Pipelines of Marker Development for Transcriptome-based Exon Capture

Part I phylogenomics

January 17, 2015

Contributors: Sonal Singhal and Ke Bi

For questions or to report bugs, please contact Ke Bi (kebi@berkeley.edu)

Reference:

[1]. Singhal S. 2013. De novo transcriptomic analyses for non-model organisms: an evaluation of methods across a multi-species data set. Molecular Ecology Resources 13:403-416.

[2]. Bi K, Linderoth T, Vanderpool D, Good JM, Nielsen R and Moritz C. 2013. Unlocking the vault: next-generation museum population genomics. Molecular Ecology 22:6018-6032.

[3]. Bi K, Vanderpool D, Singhal S, Linderoth T, Moritz C and Good JM. 2012. Transcriptome-based exon capture enables highly cost-effective comparative genomic data collection at moderate evolutionary scales. BMC Genomics 13: e403.

The pipelines are deposited in https://github.com/CGRL-QB3-UCBerkeley/MarkerDevelopmentPylogenomics

---

Scripts included in this pipeline:

**Use “chmod +x script” to make each of these perl scripts executable.

**Note: If exon identification is not possible or not desirable, users can use the entire transcripts for marker development. In this case please use “6-MarkerSelectionTRANS”. Otherwise please use “6- MarkerSelectionEXONS”.

*_1-PreCleanup_*: Reformats raw cDNA sequencing reads from Illumina HiSeq or MiSeq for _2-ScrubReads_. Specifically, in this step we will remove reads that did not pass the Illumina quality control filters and modify the sequence identifiers.

Dependencies:
FastQC: http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

Input:
Raw sequence data files are grouped and saved in folders named by their sample IDs. For instance, three libraries (CGRL_index1, CGRL_index15, CGRL_index40) are saved under “/home/ke/Desktop/SeqCap/data/rawdata/library/”. Compressed fastq sequence files are saved in each of these folders.

Fastq files use the following naming scheme:
<sample name>_<barcode sequence>_L<lane (0-padded to 3 digits)>_R<read number>_<set number (0-padded to 3 digits)>.fastq.gz

For example, in “CGRL_index15_CGACCTG_L006_R1_001.fastq.gz”:

sample name: CGRL_index15
barcode sequence: CGACCTG
lane (0-padded to 3 digits): 006
read number: 1
set number (0-padded to 3 digits): 001

```
#Make a new folder called “raw” under “~/Desktop/MarkerDevelopment/data/rawdata/”:
ke@NGS:~/Desktop/MarkerDevelopment/data/rawdata$ mkdir raw

#Copy all these compressed fastq files from each folder (CGRL_index1, CGRL_index15, CGRL_index40) to “raw”:
ke@NGS:~/Desktop/MarkerDevelopment/data/rawdata$ cp library/CGRL_index*/*.gz raw/

#Check data files in “raw”:
ke@NGS:~/Desktop/MarkerDevelopment/data/rawdata$ ls raw/*
CGRL_index15_CGACCTG_L006_R1_001.fastq.gz
CGRL_index15_CGACCTG_L006_R2_001.fastq.gz
CGRL_index1_TCGCAGG_L006_R1_001.fastq.gz
CGRL_index1_TCGCAGG_L006_R2_001.fastq.gz
CGRL_index40_TTCGCAA_L006_R1_001.fastq.gz
CGRL_index40_TTCGCAA_L006_R2_001.fastq.gz
```

Commands:

```
#cd to the working directory:
ke@NGS:~/Desktop/MarkerDevelopment/data/rawdata$ cd ..

#run  1-PreCleanup with fastq evaluation
ke@NGS:~/Desktop/MarkerDevelopment/data$ 1-PreCleanup
~/Desktop/MarkerDevelopment/data/rawdata/raw/ fastqc

~/Desktop/MarkerDevelopment/data
```

Output:

Three new folders will be created under
"~/Desktop/MarkerDevelopment/data/rawdata/raw/":
"pre-clean"
"combined"
"pre-clean/evaluation"

- Folder "pre-clean" contains reformatted raw fastq reads.

CGRL_index1_R1.fq
CGRL_index1_R2.fq
CGRL_index15_R1.fq
CGRL_index15_R2.fq
CGRL_index40_R1.fq
CGRL_index40_R2.fq

- Folder "combined" contains merged, compressed, fastq data files (not used by the following pipeline).

CGRL_index1_TCGCAGG_L006_R1.fastq.gz
CGRL_index1_TCGCAGG_L006_R2.fastq.gz
CGRL_index15_CGACCTG_L006_R1.fastq.gz
CGRL_index15_CGACCTG_L006_R2.fastq.gz
CGRL_index40_TTCGCAA_L006_R1.fastq.gz
CGRL_index40_TTCGCAA_L006_R2.fastq.gz

- Folder "evaluation" contains fastQC results for each data file.

CGRL_index1_R1.fq_fastqc/
CGRL_index1_R2.fq_fastqc/
CGRL_index15_R1.fq_fastqc/
CGRL_index15_R2.fq_fastqc/
CGRL_index40_R1.fq_fastqc/
CGRL_index40_R2.fq_fastqc/

2-ScrubReads: Clean up raw data, which includes trimming for quality, removing adapters, merging overlapping reads, removing duplicates and reads sourced from contamination

Dependencies:
cutadapt: http://code.google.com/p/cutadapt/
COPE: http://sourceforge.net/projects/coperead/
Bowtie2: http://sourceforge.net/projects/bowtie-bio/files/bowtie2/
FastQC: http://www.bioinformatics.babraham.ac.uk/projects/fastqc/
FLASh-modified: modified version of FLASh by Filipe G. Vieira.
https://github.com/MVZSEQ/Exon-capture
Trimmomatic: http://www.usadellab.org/cms/?page=trimmomatic

Input:

1. Reformatted fastq files created by [*1-PreCleanup*]:

```
#Check the raw data files:
ke@NGS:~/Desktop/MarkerDevelopment/data/rawdata/raw/pre-clean$ ls *.fq
CGRL_index1_R1.fq
CGRL_index1_R2.fq
CGRL_index15_R1.fq
CGRL_index15_R2.fq
CGRL_index40_R1.fq
CGRL_index40_R2.fq
```

2. A fasta file that contains adapter sequences:

```
#Check the format of adapter sequence file:
ke@NGS:~/Desktop/SeqCap/denovoTargetCapture/associated_files $ less -S
Adapters.fasta
>P7_index1
CAAGCAGAAGACGGCATACGAGATcctgcgaGTGACTGGAGTTCAGACGTGTGCTCTTCCGATCT
>P7_index2
CAAGCAGAAGACGGCATACGAGATtgcagagGTGACTGGAGTTCAGACGTGTGCTCTTCCGATCT
……
>P5_index1
AATGATACGGCGACCACCGAGATCTACACcctgcgaACACTCTTTCCCTACACGACGCTCTTCCGATCT
>P5_index2
AATGATACGGCGACCACCGAGATCTACACtgcagagACACTCTTTCCCTACACGACGCTCTTCCGATCT
……
```

Note: The header of each adapter sequence has to be named strictly as "**P7_index**N" or "**P5_index**N". N is the number of index. It is OK to put all adapters in this file but your libraries only use a subset of them.

3. Library info file (Tab-delimited txt file):

```
#Check the format of Library info file:
ke@NGS:~/Desktop/SeqCap/denovoTargetCapture/associated_files $ less -S libInfo.txt

library         P7      P5
```

```
CGRL_index1    1
CGRL_index15   15
CGRL_index40   40
```

Leave the “P5” column blank if you only have indexes in P7 adapters in the libraries.

4. Contaminant file:
Escherichia coli (bacteria + human + other genome resources if desired) genome in fasta format.
This file (e_coli_K12.fasta) is saved in
“~/Desktop/SeqCap/denovoTargetCapture/associated_files/ecoli/”

Commands:

#Make a new folder called “cleaned_data” in “~/Desktop/MarkerDevelopment/data/”:

```
ke@NGS:~/Desktop/MarkerDevelopment/data$ mkdir cleaned_data
```

#Run *2-ScrubReads*:

```
ke@NGS:~/Desktop/MarkerDevelopment/data$ 2-ScrubReads -f
~/Desktop/MarkerDevelopment/data/rawdata/raw/pre-clean/ -o
~/Desktop/MarkerDevelopment/data/cleaned_data/ -a
~/Desktop/SeqCap/denovoTargetCapture/associated_files/Adapters.fasta -b
~/Desktop/SeqCap/denovoTargetCapture/associated_files/libInfo.txt -t
/home/ke/Desktop/SeqCap/programs/Trimmomatic-0.32/trimmomatic-0.32.jar -c
~/Desktop/SeqCap/denovoTargetCapture/associated_files/ecoli/e_coli_K12.fasta -e
200 -m 15 -z
```

Note: I use the default values for most of the arguments. Users should adjust these parameters when processing the real datasets.

Output:

1. In “~/Desktop/MarkerDevelopment/data/cleaned_data/”, six .txt files per library are produced:
For example for library CGRL_index1, the six files are:
CGRL_index1_1_final.txt (left reads)
CGRL_index1_2_final.txt (right reads)
CGRL_index1_u_final.txt (merged or unpaired reads)
CGRL_index1.contam.out (headers of reads aligned to bacteria)
CGRL_index1.duplicates.out (headers of duplicated reads)
CGRL_index1.lowComplexity.out (headers of low complexity reads)

2. In “~/Desktop/MarkerDevelopment/data/cleaned_data/evaluation/”, you can find fastQC results for cleaned reads from each library.

3-GenerateAssemblies: Assemble RNAseq data using Trinity.

Dependencies:
Trinity http://trinityrnaseq.sourceforge.net

Input:
For each library, we will concatenate cleaned forward reads (XXX_1_final.txt) and unpaired reads (XXX_u_final.txt) and name the resulting read data file as XXX_1_final.txt.

#Make a new folder called "raw_assembly" under "~/Desktop/MarkerDevelopment/data/":

```
ke@NGS:~/Desktop/MarkerDevelopment/data$ mkdir raw_assembly
```

#Concatenate cleaned forward reads and unpaired reads and save them in "raw_assembly":

```
ke@NGS:~/Desktop/MarkerDevelopment/data$ cat
cleaned_data/CGRL_index1_1_final.txt cleaned_data/CGRL_index1_u_final.txt | sed
's/\/2$/\/1/g' > raw_assembly/CGRL_index1_1_final.txt
ke@NGS:~/Desktop/MarkerDevelopment/data$ cat
cleaned_data/CGRL_index15_1_final.txt cleaned_data/CGRL_index15_u_final.txt | sed
's/\/2$/\/1/g' > raw_assembly/CGRL_index15_1_final.txt
ke@NGS:~/Desktop/MarkerDevelopment/data$ cat
cleaned_data/CGRL_index40_1_final.txt cleaned_data/CGRL_index40_u_final.txt | sed
's/\/2$/\/1/g' > raw_assembly/CGRL_index40_1_final.txt
```

#Copy read2 of all libraries to "raw_assembly"

```
ke@NGS:~/Desktop/MarkerDevelopment/data$ cp
cleaned_data/CGRL_index*_2_final.txt raw_assembly/
```

Commands:
#Run Trinity on 4 processors.

```
 ke@NGS:~/Desktop/MarkerDevelopment/data$ 3-GenerateAssemblies trinity -a
raw_assembly/ -c 5 -e 4
```

Note: Your labtop may not be able to handle Trinity assemblies.

Output:
There are quite a few intermediate files generated in
"~/Desktop/MarkerDevelopment/data/raw_assembly/CGRL_index1/".
"~/Desktop/MarkerDevelopment/data/raw_assembly/CGRL_index15/".
"~/Desktop/MarkerDevelopment/data/raw_assembly/CGRL_index40/".

#To show final trinity assemblies that are needed for annotation:

```
ke@NGS:~/Desktop/MarkerDevelopment/data$ ls raw_assembly/CGRL_index*/*.fasta
```

```
raw_assembly/CGRL_index15/CGRL_index15.fasta
raw_assembly/CGRL_index1/CGRL_index1.fasta
raw_assembly/CGRL_index40/CGRL_index40.fasta
```

#Under “~/Desktop/MarkerDevelopment/data/” make a new folder called “annotation” and copy all files shown above to this folder:

```
ke@NGS:~/Desktop/MarkerDevelopment/data$ mkdir annotation
ke@NGS:~/Desktop/MarkerDevelopment/data$ cp
raw_assembly/CGRL_index*/*.fasta annotation/
```

#check all files in folder “annotation”

```
ke@NGS:~/Desktop/MarkerDevelopment/data$ ls annotation/*
annotation/CGRL_index15.fasta
annotation/CGRL_index40.fasta
annotation/CGRL_index1.fasta
```

##

When we did step1-3 we used a tiny fraction of the RNAseq data for the purpose of quick demonstration. To better demonstrate how to use the next script (4-AssemblyEvaluation) let’s sample some more data from each individual.

Please do the following before you start working on step 4:

```
ke@NGS:~/Desktop/MarkerDevelopment/data$ cp
~/Desktop/MarkerDevelopment/associated_data/CGRL_index*.fasta
annotation/
```

##

4-AssemblyEvaluation (Optional): Evaluate the quality of cDNA *de novo* assemblies. A few examples of the available functions are shown here.

Dependencies:
Blat: http://hgdownload.soe.ucsc.edu/downloads.html#source_downloads
Blastall:
http://blast.ncbi.nlm.nih.gov/Blast.cgi?PAGE_TYPE=BlastDocs&DOC_TYPE=Download

Input: Trinity assemblies for all libraries stored in "~/Desktop/MarkerDevelopment/data/annotation/"

```
#Display all items in "~/Desktop/MarkerDevelopment/data/annotation/"
ke@NGS:~/Desktop/MarkerDevelopment/data/annotation$ ls

CGRL_index15.fasta
CGRL_index1.fasta
CGRL_index40.fasta
```

a. 4-AssemblyEvaluation BASIC: function "BASIC" evaluates the quality of in-target assemblies by reporting basic stats: mean, median, total length, gc%, N50 etc. It also generates a distribution of contigs by binned lengths.

Commands:

```
ke@NGS:~/Desktop/MarkerDevelopment/data $ 4-AssemblyEvaluation BASIC -a annotation/
```

Output:

# In folder "~/Desktop/MarkerDevelopment/data/annotation/", you should get the following output files:

```
CGRL_index15.hist
CGRL_index1.hist
CGRL_index40.hist
basic_evaluation.out
```

Output:

1. "XXX. hist" shows distribution of contigs by binned lengths

```
#Display first few lines of the file:
ke@NGS:~/Desktop/MarkerDevelopment/data/annotation$  head CGRL_index15.hist
200:299         57
300:399         43
400:499         34
500:599         28
600:699         25
```

```
700:799     23
800:899     18
900:999     24
1000:1099   13
1100:1199   6
```

2. “basic_evaluation.out”: results of assembly evaluation
#Display first few lines of the file:

```
ke@NGS:~/Desktop/MarkerDevelopment/data/annotation$  head
basic_evaluation.out
```

b. *4-AssemblyEvaluation ANNOTATABLE*: Calculates the percentage of the assembled contigs that get a match in reference. It also calculates average percentage of matched bp and mismatches among the matched genes.

Commands:

```
ke@NGS:~/Desktop/MarkerDevelopment/data$ 4-AssemblyEvaluation
ANNOTATABLE  -a annotation/ -b 100 -c
~/Desktop/MarkerDevelopment/associated_data/Xenopus_tropicalis.JGI_4.2.cdna.all.f
a
```

Output:

#Display results in the output file “annotatable.out”:

```
ke@NGS:~/Desktop/MarkerDevelopment/data/annotation$ less annotatable.out

Assemblies    total matches(%)     matched bases(%)     avg similarity(%)
CGRL_index1    100.00  61.91  78.55
CGRL_index15    98.00  58.38  77.36
CGRL_index40    96.00  68.23  78.17
```

c. *4-AssemblyEvaluation ACCURACY*: The percentage of the correctly assembled bases estimated using the set of expressed reference transcripts

Commands:

```
ke@NGS:~/Desktop/MarkerDevelopment/data$ 4-AssemblyEvaluation ACCURACY -a
annotation/ -b 300 -c
~/Desktop/MarkerDevelopment/associated_data/Xenopus_tropicalis.JGI_4.2.pep.all.fa
```

Output:

#Display results in the output file “accuracy.out”:

```
ke@NGS:~/Desktop/MarkerDevelopment/data/annotation$ less accuracy.out

Assemblies    stop codon(%)   gaps(%)
CGRL_index1    0.000   0.000
CGRL_index15    0.692   0.000
CGRL_index40    0.348   0.000
```

d. *4-AssemblyEvaluation CONTIGUITY* : Calculates assembly contiguity (the percentage of expressed reference transcripts covered by a single, longest assembled contig) and completeness (the percentage of expressed reference transcripts covered by all matched assembled contigs)

Commands:

```
ke@NGS:~/Desktop/MarkerDevelopment/data$ 4-AssemblyEvaluation  CONTIGUITY -a annotation/ -b 300 -c ~/Desktop/MarkerDevelopment/associated_data/Xenopus_tropicalis.JGI_4.2.cdna.all.fa
```

Note: that –b in function “CONTIGUITY” refers to the number of randomly selected sequences from the reference protein database. In functions “BASIC”, “ANNOTATABLE” and “ACCURACY” –b refers to the number of randomly selected sequences in de novo assemblies

Output:

#Display results in the output file “Contiguity.out”:

```
ke@NGS:~/Desktop/MarkerDevelopment/data/annotation$ less Contiguity.out

Assemblies     complete(%)     contiguity(%)
CGRL_index1     13.46   11.47
CGRL_index15    23.36   23.36
GRL_index40    37.40   30.21
```

5-Annotation: annotate assembled contigs using a related reference protein dataset that can be found in Ensembl Genome Browser (http://www.ensembl.org/index.html)

Dependencies:
BLAST+:
http://blast.ncbi.nlm.nih.gov/Blast.cgi?PAGE_TYPE=BlastDocs&DOC_TYPE=Download
FrameDP: https://iant.toulouse.inra.fr/FrameDP/cgi-bin/framedp.cgi?__wb_cfg=/www/iant/FrameDP/cgi-bin/../cfg/FrameDP.cfg&__wb_session=WBuPAWHo&__wb_main_menu=Download&__wb_function=Download
exonerate: http://www.ebi.ac.uk/~guy/exonerate/index.html

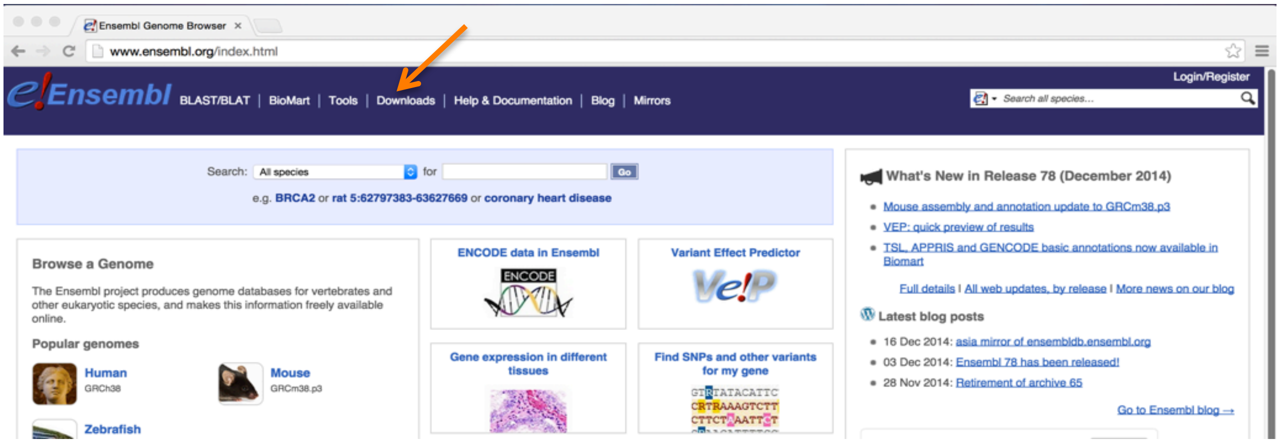
**Note: this script works only if you can find a reference database from the EGB. However, if you would like to use NCBI refseq, NR or UniProtKB/Swiss-Prot, modification of this script is needed.

** Swiss-Prot (created in 1986) is a high quality manually annotated and non-redundant protein sequence database, which brings together experimental results, computed features and scientific conclusions. UniProtKB/Swiss-Prot is now the reviewed section of the UniProt Knowledgebase.

**FrameDP: Sensitive peptide detection on noisy matured sequences. A self-training integrative pipeline for predicting CDS in transcripts which can adapt itself to different levels of sequence qualities.

Input:
1. download a reference protein dataset from the Ensembl:

Step1. Go to the Ensembl homepage http://www.ensembl.org/ and click on "Download" located at the top.


Ensembl Genome Browser
www.ensembl.org/index.html
Ensembl
BLAST/BLAT | BioMart | Tools | Downloads | Help & Documentation | Blog | Mirrors
Login/Register
Search all species...
Search: All species for Go
e.g. BRCA2 or rat 5:62797383-63627669 or coronary heart disease
Browse a Genome
Popular genomes
Human GRCh38
Mouse GRCm38.p3
Zebrafish
ENCODE data in Ensembl
Variant Effect Predictor
Gene expression in different tissues
Find SNPs and other variants for my gene
What's New in Release 78 (December 2014)
Latest blog posts
Go to Ensembl blog →

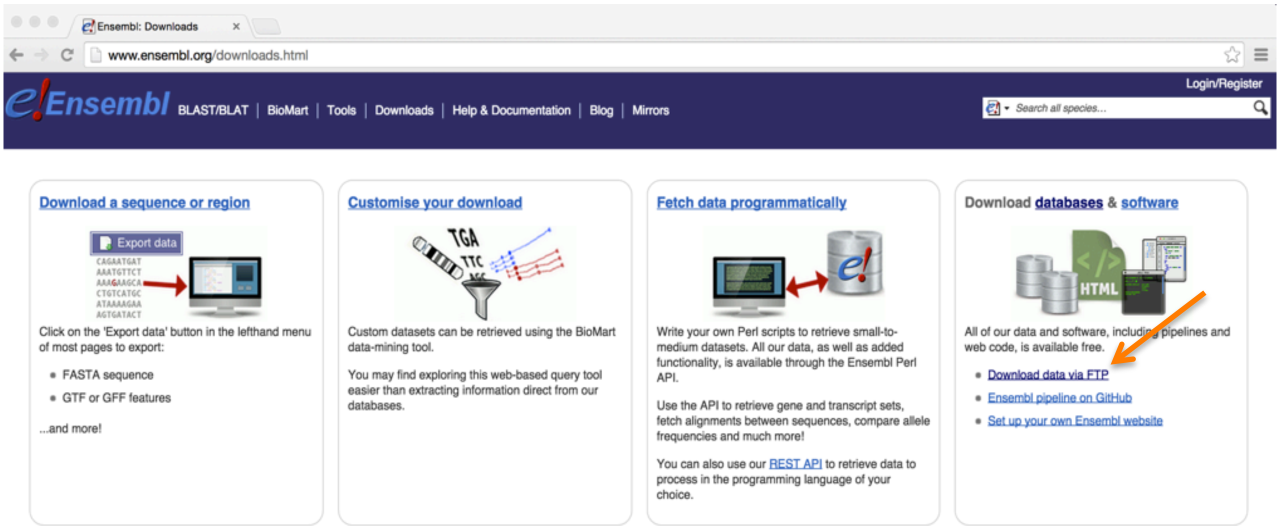
The Ensembl project produces genome databases for vertebrates and other eukaryotic species, and makes this information freely available online.

- Mouse assembly and annotation update to GRCm38.p3
- VEP: quick preview of results
- TSL, APPRIS and GENCODE basic annotations now available in Biomart

Full details | All web updates, by release | More news on our blog

- 16 Dec 2014: asia mirror of ensembldb.ensembl.org
- 03 Dec 2014: Ensembl 78 has been released!
- 28 Nov 2014: Retirement of archive 65

Step2. Click on “Download data via FTP” to the left of the download page.


Ensembl: Downloads
www.ensembl.org/downloads.html
Login/Register
Ensembl
BLAST/BLAT | BioMart | Tools | Downloads | Help & Documentation | Blog | Mirrors
Search all species...

Download a sequence or region

Click on the 'Export data' button in the lefthand menu of most pages to export:

- FASTA sequence
- GTF or GFF features

...and more!

Customise your download

Custom datasets can be retrieved using the BioMart data-mining tool.

You may find exploring this web-based query tool easier than extracting information direct from our databases.

Fetch data programmatically

Write your own Perl scripts to retrieve small-to-medium datasets. All our data, as well as added functionality, is available through the Ensembl Perl API.

Use the API to retrieve gene and transcript sets, fetch alignments between sequences, compare allele frequencies and much more!

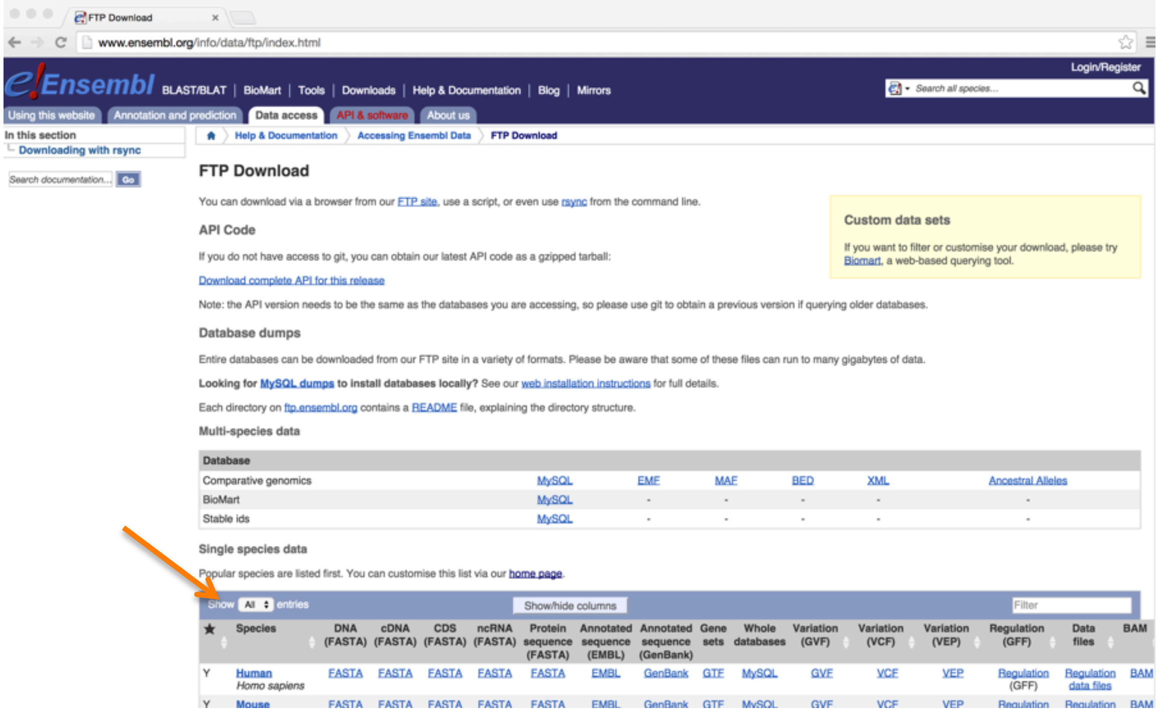
You can also use our REST API to retrieve data to process in the programming language of your choice.

Download databases & software

All of our data and software, including pipelines and web code, is available free.

- Download data via FTP
- Ensembl pipeline on GitHub
- Set up your own Ensembl website

Step3. Select “All” in the “single species data” box in the FTP download page.



FTP Download

You can download via a browser from our FTP site, use a script, or even use rsync from the command line.

API Code

If you do not have access to git, you can obtain our latest API code as a gzipped tarball:

Download complete API for this release

Note: the API version needs to be the same as the databases you are accessing, so please use git to obtain a previous version if querying older databases.

Custom data sets

If you want to filter or customise your download, please try Biomart, a web-based querying tool.

Database dumps

Entire databases can be downloaded from our FTP site in a variety of formats. Please be aware that some of these files can run to many gigabytes of data.

Looking for MySQL dumps to install databases locally? See our web installation instructions for full details.

Each directory on ftp.ensembl.org contains a README file, explaining the directory structure.

Multi-species data

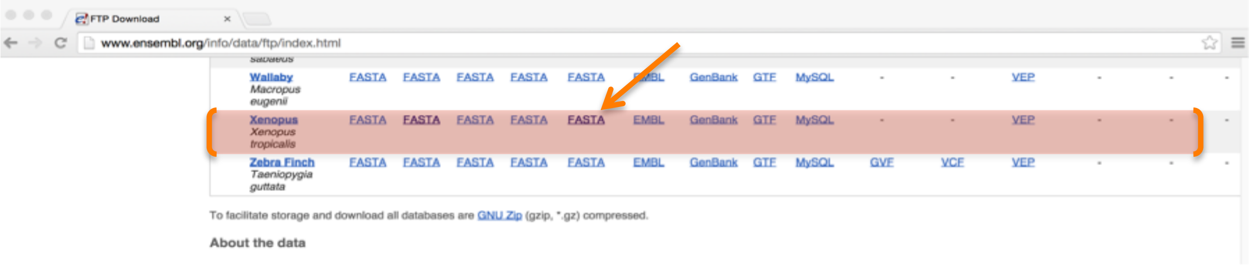
Database						
Comparative genomics	MySQL	EMF	MAF	BED	XML	Ancestral Alleles
BioMart	MySQL	-	-	-	-	-
Stable ids	MySQL	-	-	-	-	-

Single species data

Popular species are listed first. You can customise this list via our home page.

★	Species	DNA (FASTA)	cDNA (FASTA)	CDS (FASTA)	ncRNA (FASTA)	Protein sequence (FASTA)	Annotated sequence (EMBL)	Annotated sequence (GenBank)	Gene sets	Whole databases	Variation (GVF)	Variation (VCF)	Variation (VEP)	Regulation (GFF)	Data files	BAM
Y	Human *Homo sapiens*	FASTA	FASTA	FASTA	FASTA	FASTA	EMBL	GenBank	GTF	MySQL	GVF	VCF	VEP	Regulation (GFF)	Regulation data files	BAM
Y	Mouse	FASTA	FASTA	FASTA	FASTA	FASTA	EMBL	GenBank	GTF	MySQL	GVF	VCF	VEP	Regulation	Regulation	BAM

Step 4: Find and download the reference. Click on the FASTA link for Protein sequence. In this case we choose *Xenopus tropicalis* as the reference.

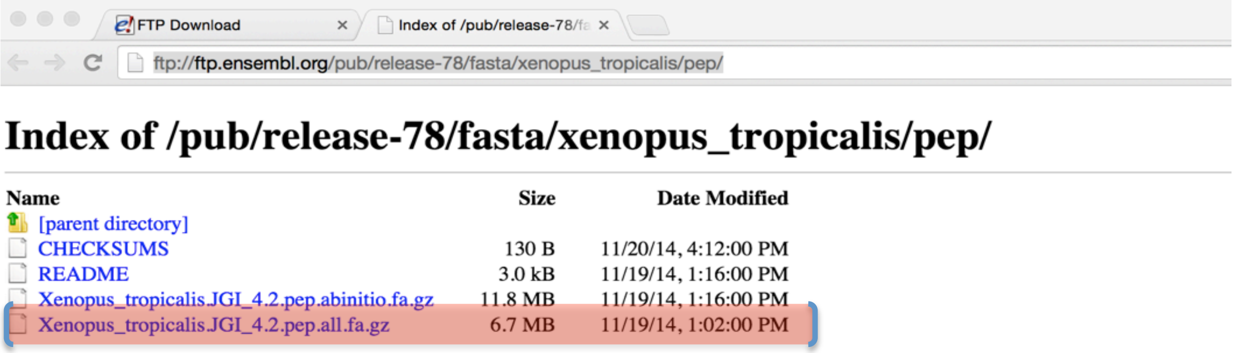


Wallaby *Macropus eugenii*	FASTA	FASTA	FASTA	FASTA	FASTA	EMBL	GenBank	GTF	MySQL	-	-	VEP	-	-	-
Xenopus *Xenopus tropicalis*	FASTA	FASTA	FASTA	FASTA	FASTA	EMBL	GenBank	GTF	MySQL	-	-	VEP	-	-	-
Zebra Finch *Taeniopygia guttata*	FASTA	FASTA	FASTA	FASTA	FASTA	EMBL	GenBank	GTF	MySQL	GVF	VCF	VEP	-	-	-

To facilitate storage and download all databases are GNU Zip (gzip, *.gz) compressed.

About the data

Step 5: From FTP server, download reference protein fasta “XXX.pep.all. fa.gz”



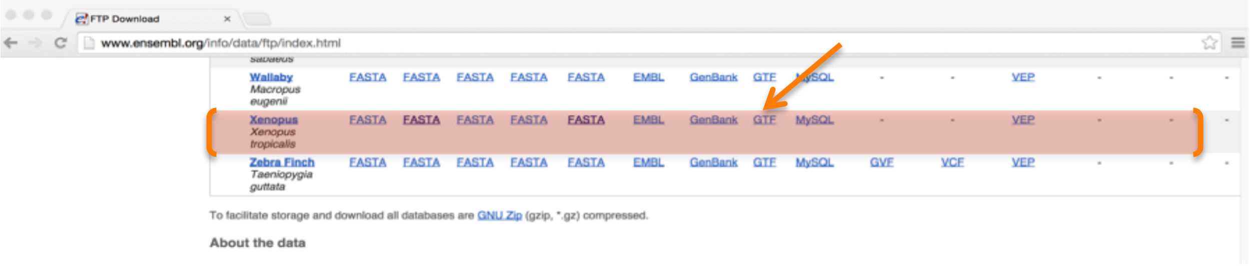
FTP Download
Index of /pub/release-78/fa
ftp://ftp.ensembl.org/pub/release-78/fasta/xenopus_tropicalis/pep/

Index of /pub/release-78/fasta/xenopus_tropicalis/pep/

Name	Size	Date Modified
[parent directory]		
CHECKSUMS	130 B	11/20/14, 4:12:00 PM
README	3.0 kB	11/19/14, 1:16:00 PM
Xenopus_tropicalis.JGI_4.2.pep.abinitio.fa.gz	11.8 MB	11/19/14, 1:16:00 PM
Xenopus_tropicalis.JGI_4.2.pep.all.fa.gz	6.7 MB	11/19/14, 1:02:00 PM

Step 6: unzip the downloaded reference fasta: *gunzip Xenopus_tropicalis.JGI_4.2.pep.all. fa.gz*

Step 7: Find and download the GTF (Gene transfer format (GTF) is a file format used to hold information about gene structure) if there is one available for the reference. In this case we can see that *Xenopus tropicalis* has a GTF so we can download it.



FTP Download
www.ensembl.org/info/data/ftp/index.html

Wallaby *Macropus eugenii*	FASTA	FASTA	FASTA	FASTA	FASTA	EMBL	GenBank	GTF	MySQL	-	-	VEP	-	-	-
Xenopus *Xenopus tropicalis*	FASTA	FASTA	FASTA	FASTA	FASTA	EMBL	GenBank	GTF	MySQL	-	-	VEP	-	-	-
Zebra Finch *Taeniopygia guttata*	FASTA	FASTA	FASTA	FASTA	FASTA	EMBL	GenBank	GTF	MySQL	GVF	VCF	VEP	-	-	-

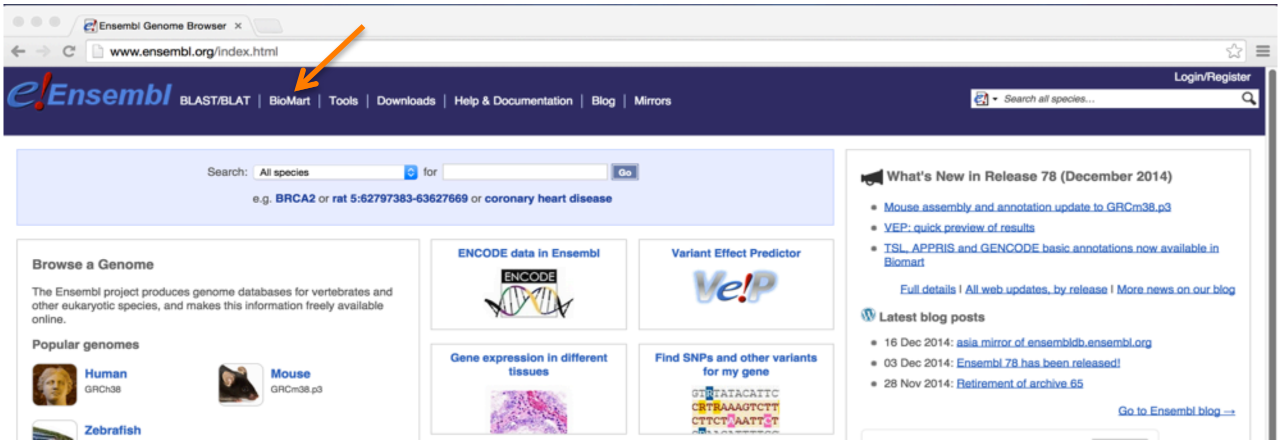
To facilitate storage and download all databases are GNU Zip (gzip, *.gz) compressed.

About the data

Step 8: unzip the downloaded GTF: *gunzip Xenopus_tropicalis.JGI_4.2.78.gtf.gz*

2. If GTF is not available then you can use Ensembl BioMart tool to obtain a gene annotation file for the reference. For the workshop I will show you how obtain this file from the BioMart tool even though we have downloaded a GTF for the reference.

Step1. Go to the Ensembl homepage http://www.ensembl.org/ and click on "BioMart" located at the top.


Ensembl Genome Browser
www.ensembl.org/index.html
Login/Register
Ensembl
BLAST/BLAT | BioMart | Tools | Downloads | Help & Documentation | Blog | Mirrors
Search all species...
Search: All species for Go
ENCODE data in Ensembl
Variant Effect Predictor
Gene expression in different tissues
Find SNPs and other variants for my gene
Popular genomes
Human GRCh38
Mouse GRCm38.p3
Zebrafish

e.g. BRCA2 or rat 5:62797383-63627669 or coronary heart disease

Browse a Genome

The Ensembl project produces genome databases for vertebrates and other eukaryotic species, and makes this information freely available online.

What's New in Release 78 (December 2014)

- Mouse assembly and annotation update to GRCm38.p3
- VEP: quick preview of results
- TSL, APPRIS and GENCODE basic annotations now available in Biomart

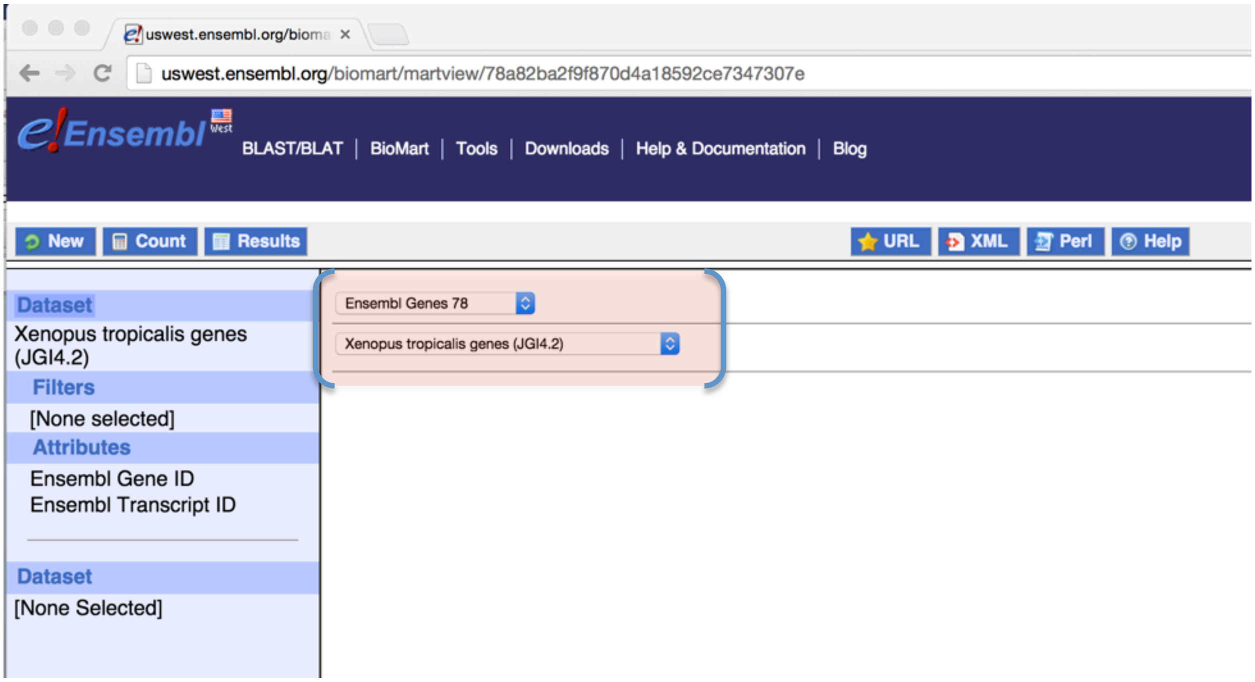
Full details | All web updates, by release | More news on our blog

Latest blog posts

- 16 Dec 2014: asia mirror of ensembldb.ensembl.org
- 03 Dec 2014: Ensembl 78 has been released!
- 28 Nov 2014: Retirement of archive 65

Go to Ensembl blog →

Step2. In the BioMart homepage, select "Ensembl Genes 78" and "Xenopus tropicalis genes (JGI4.2)".

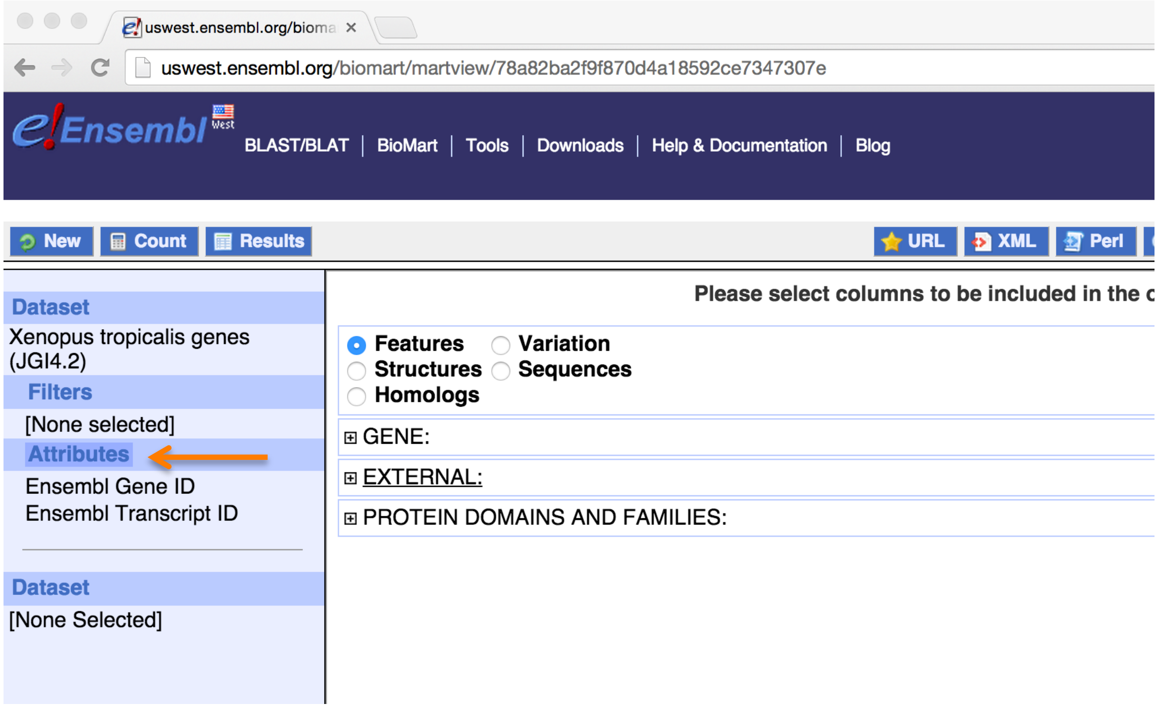

uswest.ensembl.org/bioma
uswest.ensembl.org/biomart/martview/78a82ba2f9f870d4a18592ce7347307e
Ensembl West
BLAST/BLAT | BioMart | Tools | Downloads | Help & Documentation | Blog
New
Count
Results
URL
XML
Perl
Help

Dataset
Xenopus tropicalis genes (JGI4.2)
Filters
[None selected]
Attributes
Ensembl Gene ID
Ensembl Transcript ID

Dataset
[None Selected]

Ensembl Genes 78
Xenopus tropicalis genes (JGI4.2)

Step3. Click on "Attributes" icon to the left.


uswest.ensembl.org/bioma
uswest.ensembl.org/biomart/martview/78a82ba2f9f870d4a18592ce7347307e
Ensembl West
BLAST/BLAT | BioMart | Tools | Downloads | Help & Documentation | Blog
New
Count
Results
URL
XML
Perl

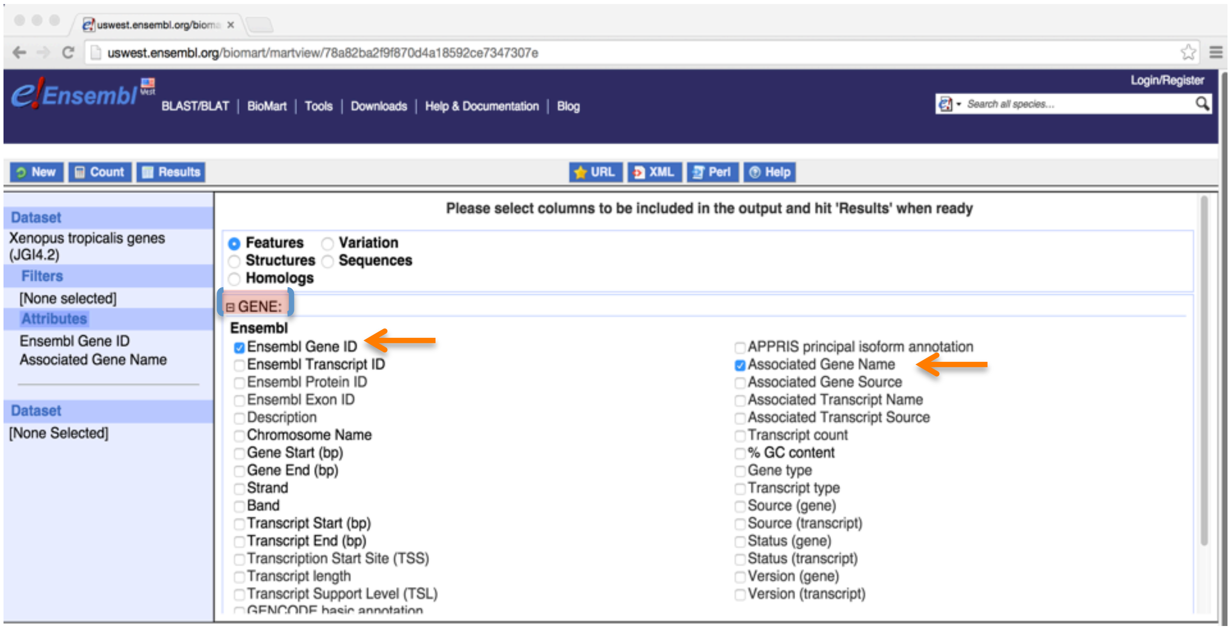
Dataset
Xenopus tropicalis genes (JGI4.2)
Filters
[None selected]
Attributes
Ensembl Gene ID
Ensembl Transcript ID
Dataset
[None Selected]

Please select columns to be included in the c

● Features ○ Variation
○ Structures ○ Sequences
○ Homologs

⊞ GENE:
⊞ EXTERNAL:
⊞ PROTEIN DOMAINS AND FAMILIES:

Step 4. Click on "GENE" to expand the manual. Check on "Ensembl Gene ID" and "Associated Gene Name".


uswest.ensembl.org/bioma
uswest.ensembl.org/biomart/martview/78a82ba2f9f870d4a18592ce7347307e
Login/Register
Ensembl West
BLAST/BLAT | BioMart | Tools | Downloads | Help & Documentation | Blog
Search all species...
New
Count
Results
URL
XML
Perl
Help

Dataset
Xenopus tropicalis genes (JGI4.2)
Filters
[None selected]
Attributes
Ensembl Gene ID
Associated Gene Name
Dataset
[None Selected]

Please select columns to be included in the output and hit 'Results' when ready

● Features ○ Variation
○ Structures ○ Sequences
○ Homologs

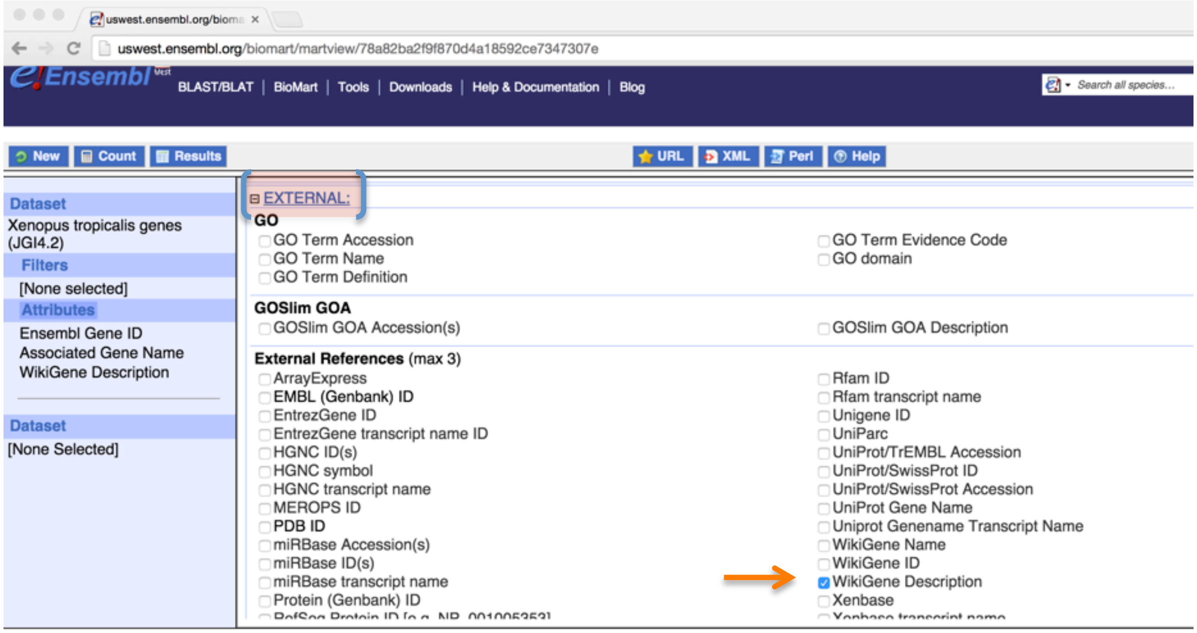
⊟ GENE:

Ensembl

☑ Ensembl Gene ID
□ Ensembl Transcript ID
□ Ensembl Protein ID
□ Ensembl Exon ID
□ Description
□ Chromosome Name
□ Gene Start (bp)
□ Gene End (bp)
□ Strand
□ Band
□ Transcript Start (bp)
□ Transcript End (bp)
□ Transcription Start Site (TSS)
□ Transcript length
□ Transcript Support Level (TSL)
□ GENCODE basic annotation

□ APPRIS principal isoform annotation
☑ Associated Gene Name
□ Associated Gene Source
□ Associated Transcript Name
□ Associated Transcript Source
□ Transcript count
□ % GC content
□ Gene type
□ Transcript type
□ Source (gene)
□ Source (transcript)
□ Status (gene)
□ Status (transcript)
□ Version (gene)
□ Version (transcript)

Step 5. Scroll down the window to find “EXTERNAL”. Click on it to expand the manual. Check on “WikiGene Description”


uswest.ensembl.org/biomart/martview/78a82ba2f9f870d4a18592ce7347307e

Ensembl | BLAST/BLAT | BioMart | Tools | Downloads | Help & Documentation | Blog | Search all species...

New | Count | Results | URL | XML | Perl | Help

Dataset
Xenopus tropicalis genes (JGI4.2)
Filters
[None selected]
Attributes
Ensembl Gene ID
Associated Gene Name
WikiGene Description

Dataset
[None Selected]

⊟ EXTERNAL:

GO

□ GO Term Accession	□ GO Term Evidence Code
□ GO Term Name	□ GO domain
□ GO Term Definition	

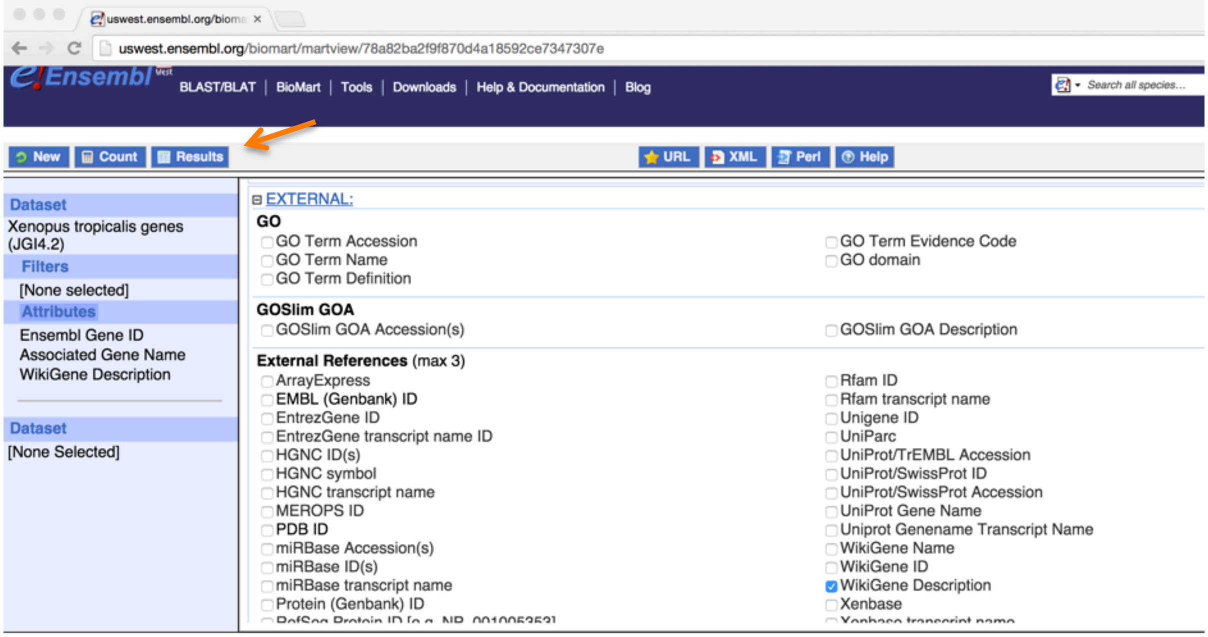
GOSlim GOA

□ GOSlim GOA Accession(s)	□ GOSlim GOA Description

External References (max 3)

□ ArrayExpress	□ Rfam ID
□ EMBL (Genbank) ID	□ Rfam transcript name
□ EntrezGene ID	□ Unigene ID
□ EntrezGene transcript name ID	□ UniParc
□ HGNC ID(s)	□ UniProt/TrEMBL Accession
□ HGNC symbol	□ UniProt/SwissProt ID
□ HGNC transcript name	□ UniProt/SwissProt Accession
□ MEROPS ID	□ UniProt Gene Name
□ PDB ID	□ Uniprot Genename Transcript Name
□ miRBase Accession(s)	□ WikiGene Name
□ miRBase ID(s)	□ WikiGene ID
□ miRBase transcript name	☑ WikiGene Description
□ Protein (Genbank) ID	□ Xenbase
□ RefSeq Protein ID [e.g. NP_001005353]	□ Xenbase transcript name

Step6. Click on “Results” icon.


uswest.ensembl.org/biomart/martview/78a82ba2f9f870d4a18592ce7347307e

Ensembl | BLAST/BLAT | BioMart | Tools | Downloads | Help & Documentation | Blog | Search all species...

New | Count | Results | URL | XML | Perl | Help

Dataset
Xenopus tropicalis genes (JGI4.2)
Filters
[None selected]
Attributes
Ensembl Gene ID
Associated Gene Name
WikiGene Description

Dataset
[None Selected]

⊟ EXTERNAL:

GO

□ GO Term Accession	□ GO Term Evidence Code
□ GO Term Name	□ GO domain
□ GO Term Definition	

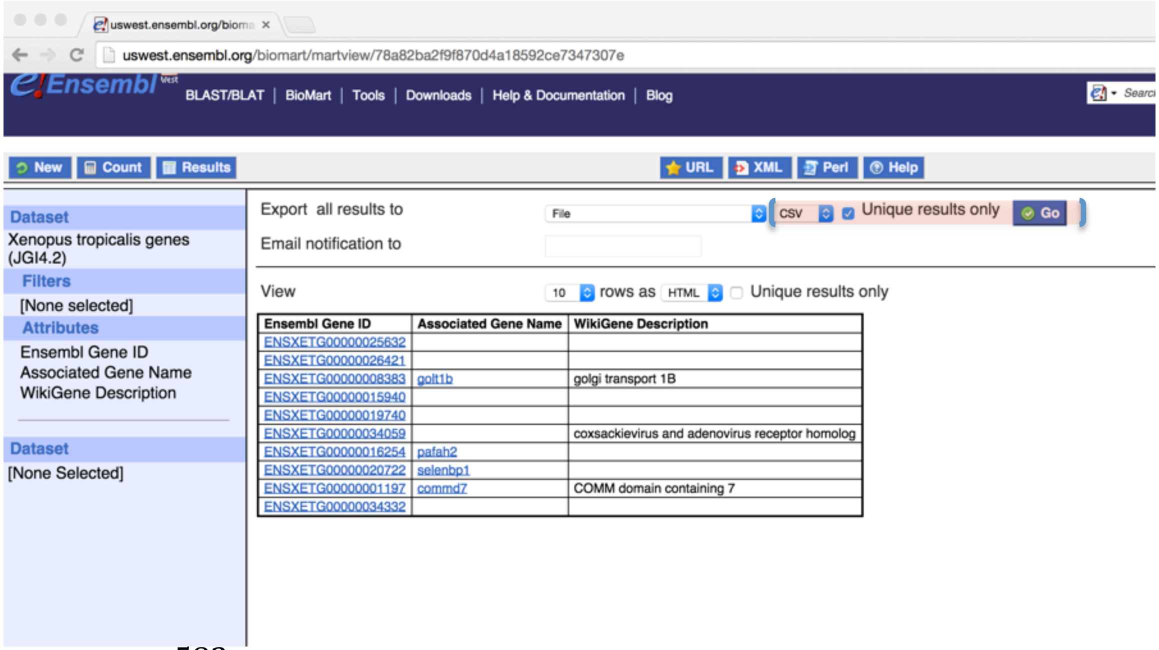
GOSlim GOA

□ GOSlim GOA Accession(s)	□ GOSlim GOA Description

External References (max 3)

□ ArrayExpress	□ Rfam ID
□ EMBL (Genbank) ID	□ Rfam transcript name
□ EntrezGene ID	□ Unigene ID
□ EntrezGene transcript name ID	□ UniParc
□ HGNC ID(s)	□ UniProt/TrEMBL Accession
□ HGNC symbol	□ UniProt/SwissProt ID
□ HGNC transcript name	□ UniProt/SwissProt Accession
□ MEROPS ID	□ UniProt Gene Name
□ PDB ID	□ Uniprot Genename Transcript Name
□ miRBase Accession(s)	□ WikiGene Name
□ miRBase ID(s)	□ WikiGene ID
□ miRBase transcript name	☑ WikiGene Description
□ Protein (Genbank) ID	□ Xenbase
□ RefSeq Protein ID [e.g. NP_001005353]	□ Xenbase transcript name

Step 7. To export the results, select “CSV” format and check on “Unique results only” box, and then click on “Go”.


uswest.ensembl.org/bioma
uswest.ensembl.org/biomart/martview/78a82ba2f9f870d4a18592ce7347307e
Ensembl West
BLAST/BLAT | BioMart | Tools | Downloads | Help & Documentation | Blog
Search
New
Count
Results
URL
XML
Perl
Help

Dataset
Xenopus tropicalis genes (JGI4.2)
Filters
[None selected]
Attributes
Ensembl Gene ID
Associated Gene Name
WikiGene Description

Dataset
[None Selected]

Export all results to File CSV ☑ Unique results only Go

Email notification to

View 10 rows as HTML □ Unique results only

Ensembl Gene ID	Associated Gene Name	WikiGene Description
ENSXETG00000025632		
ENSXETG00000026421		
ENSXETG00000008383	golt1b	golgi transport 1B
ENSXETG00000015940		
ENSXETG00000019740		
ENSXETG00000034059		coxsackievirus and adenovirus receptor homolog
ENSXETG00000016254	pafah2	
ENSXETG00000020722	selenbp1	
ENSXETG00000001197	commd7	COMM domain containing 7
ENSXETG00000034332		

Step 8. Save and rename the result to be “*Xenopus.tropicalis* _gene_name.txt”. There are three columns, separated by comma:

Ensembl Gene ID, Associated Gene Name, WikiGene Description
ENSXETG00000008383, golt1b, golgi transport 1B
ENSXETG00000034059, CARH, coxsackievirus and adenovirus receptor homolog
ENSXETG00000001197, commd7, COMM domain containing 7
……

**For this workshop, a reference protein, a GTF and the corresponding biomart gene name file are already downloaded and located in “~/Desktop/MarkerDevelopment/associated_data/”.

Input:

1. A folder that contains all trinity assemblies. These files are located in “~/Desktop/MarkerDevelopment/data/annotation/”

2. Reference protein downloaded from the ensemble: Xenopus_tropicalis.JGI_4.2.pep.all.fa.

3. Reference biomart gene annotation file: Xenopus_tropicalis_gene_name.txt

OR

4. Reference GTF file:
Xenopus_tropicalis.JGI_4.2.78.gtf

Commands:

# Run 5-Annotation without a GTF (do not execute the command during the workshop, since the runs will take quite a while to finish).

```
ke@NGS:~/Desktop/MarkerDevelopment/data$ 5-Annotation  -a
~/Desktop/MarkerDevelopment/data/annotation/ -b
~/Desktop/MarkerDevelopment/associated_data/Xenopus_tropicalis.JGI_4.2.pep.all.fa
-d ~/Desktop/SeqCap/programs/framedp-1.2.2/ -f
~/Desktop/MarkerDevelopment/associated_data/Xenopus_tropicalis_gene_name.txt -
n xenopus -e 1
```

##Copy the annotation results to "*~/Desktop/MarkerDevelopment/data*"

```
ke@NGS:~/Desktop/MarkerDevelopment/data$ scp -r
~/Desktop/MarkerDevelopment/associated_data/annotation/* annotation/
```

Output:

For each individual trinity assembly, a new folder is generated under "~/Desktop/MarkerDevelopment/data/annotation/":

```
CGRL_index1_xenopus/
CGRL_index14_xenopus /
CGRL_index40_xenopus /
```

##The annotated fasta files are named as "XXX_xenopus_annotated.fasta".

```
ke@NGS:~/Desktop/MarkerDevelopment/data/annotation$ ls
CGRL_index*/*annotated.fasta

CGRL_index15_xenopus/CGRL_index15_xenopus_annotated.fasta
CGRL_index50_xenopus/CGRL_index50_xenopus_annotated.fasta
CGRL_index1_xenopus/CGRL_index1_xenopus_annotated.fasta
```

##make a new folder "probe_design" under "~/Desktop/MarkerDevelopment/data/".

```
ke@NGS:~/Desktop/MarkerDevelopment/data$ mkdir probe_design
```

##copy all the annotated fasta files to "*probe_design*"

```
ke@NGS:~/Desktop/MarkerDevelopment/data$ cp
annotation/CGRL_index*/*annotated.fasta  probe_design/
```

## read and display the first few lines in the annotated fasta file:

```
ke@NGS:~/Desktop/MarkerDevelopment/data/probe_design$ head -4
CGRL_index15_xenopus_annotated.fasta
```

```
>contig1       gs1_ge432      ENSXETG00000014175      vwa5a   NA      5e-57
TCTCTTACATGGACCCTTCC……
>contig10      5u355_gs356_ge817_3u818 ENSXETG00000004176      mocs2
molybdenum cofactor synthesis 2 2e-82
TGTGCACAGTGTGATGTAG……
```

For contig1: "gs1" means coding region starts at position 1. "ge432" means coding region ends by position 432. No UTRs are present in this contig. "ENSXETG00000014175" is the Ensembl gene ID obtained from Xenopus reference database. "vwa5a" is the gene name. "NA" is the wiki gene description and in this case, wiki gene description is missing. "5e-57" is e-value in the BLAST search.

For contig10: "5u355" means 5UTR ends by position 355. "gs356" means coding region starts at position 356. "ge817" means coding region ends by position 817. "3u818" means 3UTR starts at position 818. "ENSXETG00000004176" is the Ensembl gene ID obtained from Xenopus reference database. "mocs2" is the gene name. "molybdenum cofactor synthesis 2" is the wiki gene description. "2e-82" is e-value in the BLAST search.

~~~~~~

Run 5-Annotation with a GTF

Commands:

```
ke@NGS:~/Desktop/MarkerDevelopment$ 5-Annotation -a
~/Desktop/MarkerDevelopment/data/annotation/ -b
~/Desktop/MarkerDevelopment/associated_data/Xenopus_tropicalis.JGI_4.2.pep.all.fa
-d ~/Desktop/SeqCap/programs/framedp-1.2.2/  -n xenopus -g
~/Desktop/MarkerDevelopment/associated_data/Xenopus_tropicalis.JGI_4.2.78.gtf -e
1
```

The output by using GTF is slightly different since the header doesn't have gene name descriptions. For example:

```
>contig1      gs1_ge432      ENSXETG00000014175      vwa5a   protein_coding      5e-
57
TCTCTTACATGGACCCTTCC……
```

"gs1" means coding region starts at position 1. "ge432" means coding region ends by position 432. No UTRs are present in this contig. "ENSXETG00000014175" is the Ensembl gene ID obtained from Xenopus reference database. "vwa5a" is the gene name. **"protein_coding" is the type of the gene**. "5e-57" is e-value in the BLAST search.
~~~~~~

**6-MarkerSelectionTRANS*: Find orthologous transcripts in transcriptomes from different species and generate input files for probe design. It can be used when exon identification is impossible and/or is not preferred.

Dependencies:
BLAST+:
http://blast.ncbi.nlm.nih.gov/Blast.cgi?PAGE_TYPE=BlastDocs&DOC_TYPE=Download
MUSCLE: http://www.drive5.com/muscle/
cd-hit-est: http://weizhongli-lab.org/cd-hit/

First of all we want to identify orthologous transcripts across transcriptomes from different species. We will run the command "6-MarkerSelectionTRANS markers" for this task:

Input:
All annotated transcripts located in "~/Desktop/MarkerDevelopment/data/probe_design"

```
##
ke@NGS:~/Desktop/MarkerDevelopment/data/probe_design$ ls
CGRL_index15_xenopus_annotated.fasta
CGRL_index40_xenopus_annotated.fasta
CGRL_index1_xenopus_annotated.fasta
```

Make a new folder "other_files" under "~/Desktop/MarkerDevelopment/data/probe_design/".
Use one of the annotated files as a "primary" annotation file. Move the rest to a folder "other_files". In the workshop we use CGRL_index1 as the "primary" annotation file.

```
ke@NGS:~/Desktop/MarkerDevelopment/data/probe_design$ mkdir other_files

ke@NGS:~/Desktop/MarkerDevelopment/data/probe_design$ mv CGRL_index15_xenopus_annotated.fasta  CGRL_index40_xenopus_annotated.fasta other_files/
```

Commands:

```
# Run 6-MarkerSelectionTRANS markers:
ke@NGS:~/Desktop/MarkerDevelopment/data$ 6-MarkerSelectionTRANS markers -f probe_design/CGRL_index1_xenopus_annotated.fasta  -d probe_design/other_files/ -a 1000
```

Output:

#Under "ke@NGS:~/Desktop/MarkerDevelopment/data/probe_design/" a new folder called "results" was created by the script.

```
ke@NGS:~/Desktop/MarkerDevelopment/data/probe_design$ cd results/
```

#Markers that passed all filters are stored in "marker_kept.txt". First take a How many markers are kept?

```
ke@NGS:~/Desktop/MarkerDevelopment/data/probe_design/results$ wc -l marker_kept.txt
1050 marker_kept.txt
```

```
ke@NGS:~/Desktop/MarkerDevelopment/data/probe_design/results$ less -S marker_kept.txt
```

Transcript_name: Ensembl Gene ID
avgDiv: Average sequence divergence (avg. %mismatches)
varianceDiv: Variance of sequence divergence
avgLength: Average length of the marker
avgGC: Average CG content of the marker
div_CGRL_index15_xenopus_annotated _vs_CGRL_index1_xenopus_annotated: sequence divergence between CGRL_index15 and CGRL_index1
div_CGRL_index15_xenopus_annotated _vs_CGRL_index40_xenopus_annotated: sequence divergence between CGRL_index15 and CGRL_index40
div_CGRL_index1_xenopus_annotated _vs_CGRL_index40_xenopus_annotated: sequence divergence between CGRL_index1 and CGRL_index40

#Select the markers that you would like to use for probe design. In this case choose the most variable 800 markers and save them in a new file "marker_final.txt"

```
ke@NGS:~/Desktop/MarkerDevelopment/data/probe_design/results$ tail -800 marker_kept.txt > marker_final.txt
```

++++++++++++++++++++++++++++++++++++
Now we will use command "6-MarkerSelectionTRANS seq" to generate input fasta files for probe design:

Input:

1. A final set of markers you would like to use for probe design - "~/Desktop/MarkerDevelopment/data/probe_design/results/marker_final.txt"

2. A folder containing all trimmed transcripts in fasta format. These files were created by "6-MarkerSelectionTRANS markers" and are named as XXX.final2 – "~/Desktop/MarkerDevelopment/data/probe_design/results/"

Commands:

```
# Run 6-MarkerSelectionTRANS seq:
ke@NGS:~/Desktop/MarkerDevelopment/data$ 6-MarkerSelectionTRANS seq -f
probe_design/results/marker_final.txt -d probe_design/results/
The target size for CGRL_index15_xenopus _annotated.final2 is 700532bp!
The target size for CGRL_index1_xenopus _annotated.final2 is 701541bp!
The target size for CGRL_index40_xenopus _annotated.final2 is 701593bp!
```

Output:

```
#A new folder "Probe_Design" was created by the script. cd to this folder:
ke@NGS:~/Desktop/MarkerDevelopment/data/probe_design/results$ cd
Probe_Design/

#Three fasta sequence files contain sequences of orthologous markers are
generated and ready for submission for probe design:
ke@NGS:~/Desktop/MarkerDevelopment/data/probe_design/results/Probe_Design$
ls *exonic_targets.txt

CGRL_index15_xenopus _annotated_exonic_targets.txt
CGRL_index1_xenopus _annotated_exonic_targets.txt
CGRL_index40 _xenopus _annotated_exonic_targets.txt
```

6-MarkerSelectionEXONS: Find orthologous exons in transcriptomes from different species and generate input files for probe design.

Dependencies:
exonerate: http://www.ebi.ac.uk/~guy/exonerate/index.html
cd-hit-est: http://weizhongli-lab.org/cd-hit/
BLAST+:
http://blast.ncbi.nlm.nih.gov/Blast.cgi?PAGE_TYPE=BlastDocs&DOC_TYPE=Download
MUSCLE: http://www.drive5.com/muscle/

We will run "6-MarkerSelectionEXONS exons" to identify orthologous exons in transcriptomes from each of the species.

**If a .gtf file is not available then we will first use a protein and genome reference to identify exons from reference species. We will then use the identified exons from the reference to identify ortholgous exons from each of the transcriptomes.

**However, if a .gtf file is available then I recommend first run "ParseGTF" to obtain exonic sequences from the reference and then run "6-MarkerSelectionEXONS exons".

+++
First of all we assume no .gtf is available so we have to identify exons using a reference protein and reference genome.

Input:
1. Under "~/Desktop/MarkerDevelopment/data/" make a new folder "probe_design_exons/":

```
ke@NGS:~/Desktop/MarkerDevelopment/data$ mkdir probe_design_exons/
```

2. copy all annotated transcripts to "~/Desktop/MarkerDevelopment/data/probe_design_exons".

```
ke@NGS:~/Desktop/MarkerDevelopment/data$ cp probe_design/CGRL_index1_xenopus_annotated.fasta probe_design/other_files/CGRL_index* probe_design_exons/

ke@NGS:~/Desktop/MarkerDevelopment/data$ cd probe_design_exons/

ke@NGS:~/Desktop/MarkerDevelopment/data/probe_design_exons$ ls
CGRL_index15 _xenopus _annotated.fasta
CGRL_index40 _xenopus _annotated.fasta
CGRL_index1 _xenopus _annotated.fasta
```

3. Repeat-masked reference genome “Xenopus_tropicalis.JGI_4.2.dna_rm.nonchromosomal.fa”

4. A reference protein reference “Xenopus_tropicalis.JGI_4.2.pep.all.fa”;

Both 3 and 4 can be downloaded through Ensembl following the instruction above. For this workshop these two files are located under “~/Desktop/MarkerDevelopment/associated_data”.

Command:

```
#Run “6-MarkerSelectionEXONS exons”
ke@NGS:~/Desktop/MarkerDevelopment/data$ 6-MarkerSelectionEXONS exons -p
'/home/ke/Desktop/MarkerDevelopment/associated_data/Xenopus_tropicalis.JGI_4.2.
pep.all.fa' -g
'/home/ke/Desktop/MarkerDevelopment/associated_data/Xenopus_tropicalis.JGI_4.2.
dna_rm.nonchromosomal.fa' -f
~/Desktop/MarkerDevelopment/data/probe_design_exons -E 1000
```

** “*6-MarkerSelectionEXONS exons”* takes very long time to run so please do not run it during the workshop. Let’s skip this step and copy the output files directly from “associated_data”:

```
ke@NGS:~/Desktop/MarkerDevelopment/data$ cp
~/Desktop/MarkerDevelopment/associated_data/probe_design_exons/*.nr
~/Desktop/MarkerDevelopment/associated_data/probe_design_exons/marker_*
probe_design_exons/
```

Output:

In “~/Desktop/MarkerDevelopment/data/probe_design_exons/” there are two output files that are relevant for the next step:

1. “marker_kept.txt”: Orthologous exonic markers identified in the three species
2. “marker_kept_one_exon_per_gene.txt” is a subset of “marker_kept.txt” ,which contains randomly selected one exon per gene.

In both 1 and 2, annotation of each column is explained below:

exon_name: Exon ID
avgDiv: Average sequence divergence (avg. %mismatches)
varianceDiv: Variance of sequence divergence
avgLength: Average length of the exons
avgGC: Average CG content of the exons
div_CGRL_index15_xenopus _annotated_vs_CGRL_index1_xenopus _annotated: sequence divergence between CGRL_index15 and CGRL_index1

div_CGRL_index15_xenopus _annotated_vs_CGRL_index40_xenopus _annotated: sequence divergence between CGRL_index15 and CGRL_index40

div_CGRL_index1_xenopus _annotated_vs_CGRL_index40_xenopus _annotated: sequence divergence between CGRL_index1 and CGRL_index40

++

Now I will demonstrate how to use *6-MarkerSelectionEXONS exons* when a gtf is available.

Command:

```
#Run “ParseGTF”:
ke@NGS:~/Desktop/MarkerDevelopment/data$ ParseGTF -f ~/Desktop/MarkerDevelopment/associated_data/Xenopus_tropicalis.JGI_4.2.78.gtf -g ~/Desktop/MarkerDevelopment/associated_data/Xenopus_tropicalis.JGI_4.2.dna_rm.toplevel.fa -o 100 -p 1
```

**do not run ParseGTF in the workshop

Output:

```
#Results are stored in “exons.unique” under
“/home/ke/Desktop/MarkerDevelopment/associated_data/results”

#cd to “/home/ke/Desktop/MarkerDevelopment/associated_data/results”
ke@NGS:~/Desktop/MarkerDevelopment/data$ cd ~/Desktop/MarkerDevelopment/associated_data/results/

#display at the results
ke@NGS:~/Desktop/MarkerDevelopment/associated_data/results$ less -S exons.unique

#copy “exons.unique” to
“~/Desktop/MarkerDevelopment/data/probe_design_exons/”
ke@NGS:~/Desktop/MarkerDevelopment/associated_data/results$ cp exons.unique ~/Desktop/MarkerDevelopment/data/probe_design_exons

#copy all annotated transcripts to
“~/Desktop/MarkerDevelopment/data/probe_design_exons/”
ke@NGS:~/Desktop/MarkerDevelopment/data$ cp probe_design/CGRL_index1_xenopus_annotated.fasta probe_design/other_files/CGRL_index* probe_design_exons/
```

Command:

#run “6-MarkerSelectionEXONS exons” (do not run it in the workshop)

```
ke@NGS:~/Desktop/MarkerDevelopment/data$ 6-MarkerSelectionEXONS exons -p '/home/ke/Desktop/MarkerDevelopment/associated_data/Xenopus_tropicalis.JGI_4.2.pep.all.fa' -g '/home/ke/Desktop/MarkerDevelopment/associated_data/Xenopus_tropicalis.JGI_4.2.dna_rm.toplevel.fa' -f ~/Desktop/MarkerDevelopment/data/probe_design_exons -E 1000
```

Output:

Same as above:

Now we will run command “6-MarkerSelectionEXONS seq” to generate input fasta files for probe design:

Input:

1. A final set of markers you would like to use for probe design. In this case we choose to use one exon per gene - “~/Desktop/MarkerDevelopment/data/probe_design/results/marker_kept_one_exon_per_gene.txt”

2. A folder containing non-redundant exonic markers in fasta format. These files were created by “6-MarkerSelectionEXONS exons” and are named as XXX _exon.fa.nr – “~/Desktop/MarkerDevelopment/data/probe_design_exons/”

Commands:

# Run *6-MarkerSelectionEXONS seq:*

```
ke@NGS:~/Desktop/MarkerDevelopment/data$ 6-MarkerSelectionEXONS seq -f probe_design_exons/marker_kept_one_exon_per_gene.txt -d probe_design_exons/
```

Output:

#A new folder “Probe_Design” was created by the script. cd to this folder:

```
ke@NGS:~/Desktop/MarkerDevelopment/data/probe_design_exons$ cd Probe_Design/
```

#Three fasta sequence files contain sequences of orthologous exonic markers are generated and ready for submission for probe design:

```
ke@NGS:~/Desktop/MarkerDevelopment/data/probe_design_exons/Probe_Design$ ls *exonic_targets.txt

CGRL_index15_xenopus _annotated_exonic_targets.txt
CGRL_index1_xenopus _annotated_exonic_targets.txt
CGRL_index40_xenopus _annotated_exonic_targets.txt
```