Pipelines for Marker Development for Transcriptome-based Exon Capture

Part II Population Genomics

February 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/MarkerDevelopmentPopGen

---

Scripts included in this pipeline:

1-PreCleanup

2-ScrubReads

3-GenerateAssemblies

4-AssemblyEvaluation

5-Annotation

6-ProcessAnnotation

7-MiningTranscripts

**Use "chmod +x script" to make each of these perl scripts executable.
**In transcriptome-based exon captures, marker development for population

genomic projects needs RNAseq from multiple tissue types from one individual sample. For this workshop, we selected cDNA from three tissues from one indivudal frog and the libraries are named as “CGRL_index1”, CGRL_index15”, and CGRL_index40”, respectively.

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 (+ 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 multi-tissue RNAseq data using Trinity.

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

Input:
For each library, 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_pop

#Concatenate cleaned forward reads and unpaired reads and save them in "raw_assembly_pop/":
*ke@NGS:~/Desktop/MarkerDevelopment/data$ cat cleaned_data/*_1_final.txt cleaned_data/*_u_final.txt | sed 's/\/2$/\/1/g' > raw_assembly_pop/combined_1_final.txt*

*ke@NGS:~/Desktop/MarkerDevelopment/data$ cat cleaned_data/*_2_final.txt > raw_assembly_pop/combined_2_final.txt*

#Concatenated files "combined_1_final.txt" and "combined_2_final.txt" are the input files for trinity assembly.

Commands:
#Run Trinity on 4 processors.
ke@NGS:~/Desktop/MarkerDevelopment/data$ 3-GenerateAssemblies trinity -a raw_assembly_pop/ -c 5 -e 4

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

Output:
#The resulting trinity assembly is named "combined.fasta" in "~/Desktop/MarkerDevelopment/data/raw_assembly_pop/combined/".

#Under "~/Desktop/MarkerDevelopment/data/", make a new folder called "annotation_pop" and "combined.fasta" shown above to this folder:

ke@NGS:~/Desktop/MarkerDevelopment/data$ mkdir annotation_pop
ke@NGS:~/Desktop/MarkerDevelopment/data$ cp raw_assembly_pop/combined/combined.fasta annotation_pop/

280

##

282

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

288 **Please do the following before you start working on step 4:**
ke@NGS:~/Desktop/MarkerDevelopment/data$ cp
290 **~/Desktop/MarkerDevelopment/associated_data/combined.fasta annotation/**

292

##

294

_____

296

298

4-AssemblyEvaluation (Optional): Evaluate the quality of RNAseq data *de novo* assemblies. A few example 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: A trinity assembly "combined.fasta" stored in "~/Desktop/MarkerDevelopment/data/annotation_pop/"

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_pop/

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

combined.hist
basic_evaluation.out

Output:
1. "combined. hist" shows distribution of contigs by binned lengths

#Display first few lines of the file:
ke@NGS:~/Desktop/MarkerDevelopment/data/annotation_pop$ head combined.hist

100:199	*2*
200:299	*865*
300:399	*619*
400:499	*483*
500:599	*426*
600:699	*376*
700:799	*350*
800:899	*318*
900:999	*284*
1000:1099	*269*

2. "basic_evaluation.out": results of assembly evaluation
#Display first few lines of the file:
ke@NGS:~/Desktop/MarkerDevelopment/data/annotation_pop$ 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_pop/ -b 100 -c ~/Desktop/MarkerDevelopment/associated_data/Xenopus_tropicalis.JGI_4.2.cdna.all.fa

Output:

#Display results in the output file "annotatable.out":

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

Assemblies	*total matches(%)*	*matched bases(%)*	*avg similarity(%)*
combined	*96.00*	*62.53*	*77.70*

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_pop/ -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_pop$ less accuracy.out

Assemblies	*stop codon(%)*	*gaps(%)*
combined	*0.355*	*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_pop/ -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_pop$ less Contiguity.out

Assemblies	*complete(%)*	*contiguity(%)*
combined	*39.86*	*31.63*

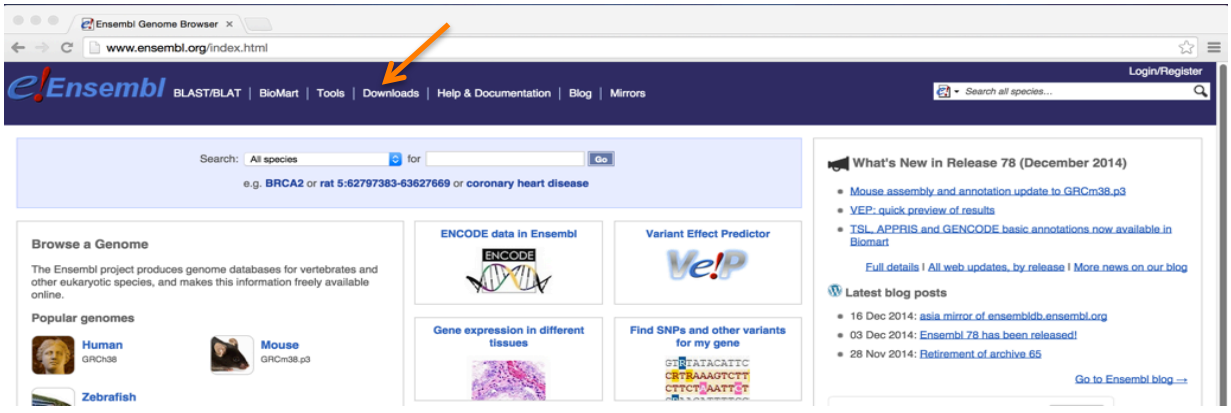
5-Annotation: annotate assembled transcripts using one or multiple reference protein datasets that can be found in Ensembl Genome Browser (http://www.ensembl.org/index.html).
Using multiple references may be helpful when there is not a closely related reference genome available.
For this workshop we use two references, Anole lizard (*Anolis carolinensis*) and clawed frog (*Xenopus tropicalis*), to annotate the assemblies (from a frog species).
The example below shows how to download protein reference and corresponding annotation files of *Xenopus tropicalis* From the Ensembl Genome Browser.

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

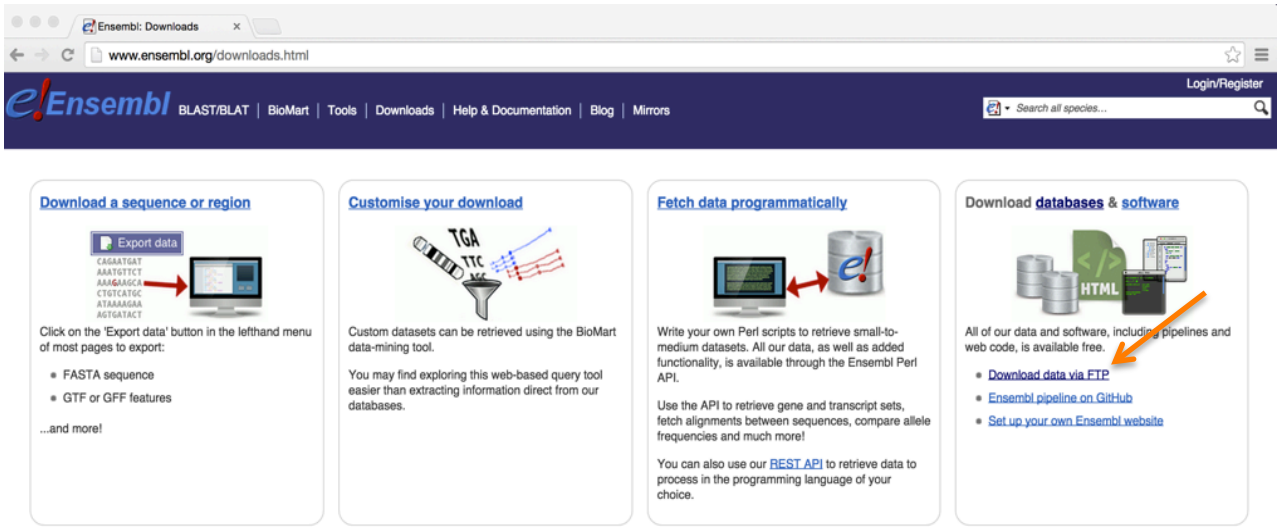
**Note: this script works only if you can find a reference database from the EGB. However, if you would like to use NCBI refseq or UniProtKB/Swiss-Prot, modification of this script is needed.
** Swiss-Prot (created in 1986) is a high quality manually annotated and non-redundant protein sequence database, 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:

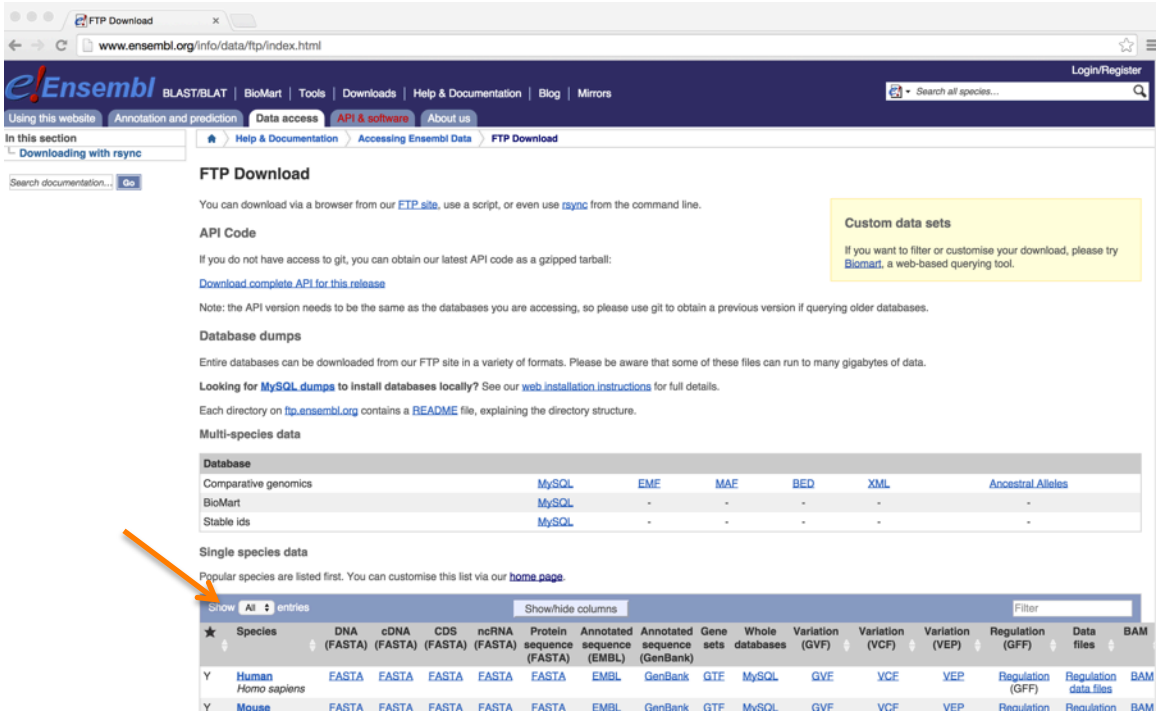
Step1. Go to the Ensembl homepage http://www.ensembl.org/ and click on "Download" located at the top.



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

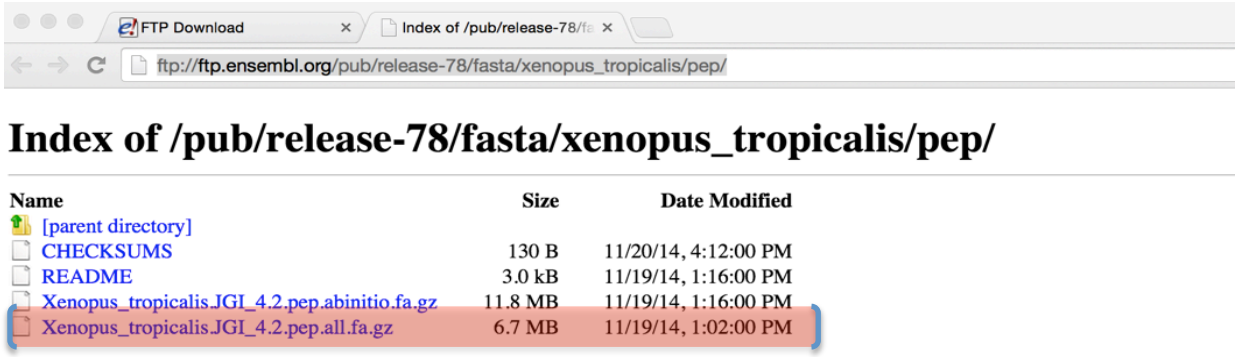


Step3. Select “All” in the “single species data” box in the FTP download page.



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.

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



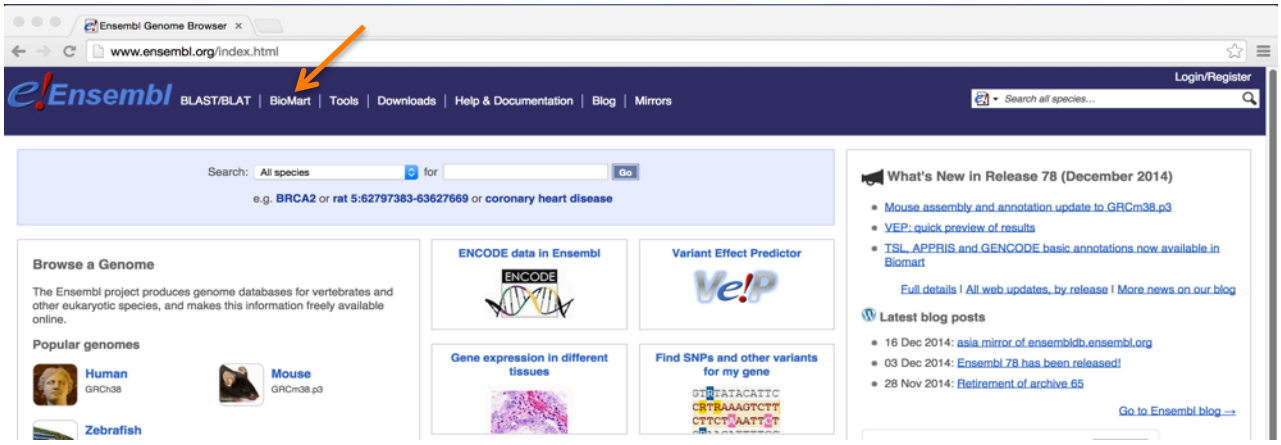
Step 6: unzip the downloaded reference fasta: *gunzip Xenopus_tropicalis.JGI_4.2.pep.all. fa.gz*

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.

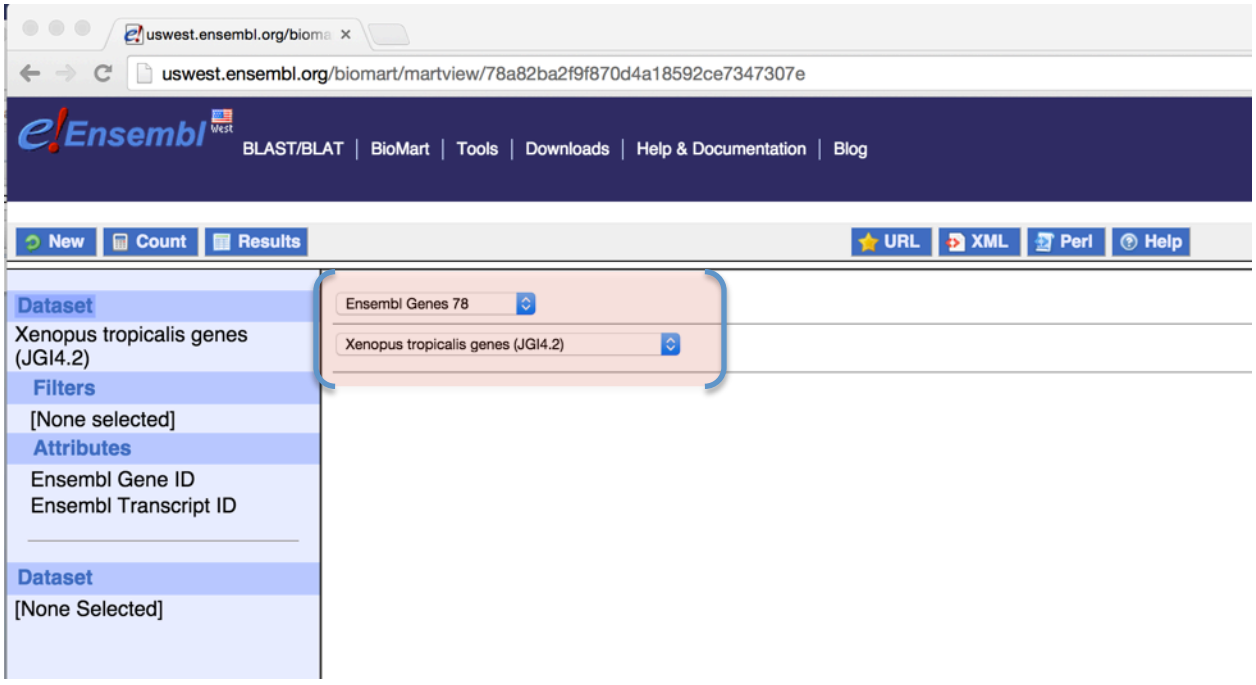
Step 8: unzip the downloaded GTF: *gunzip Anolis_carolinensis.AnoCar2.0.78.gtf.gz*

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.

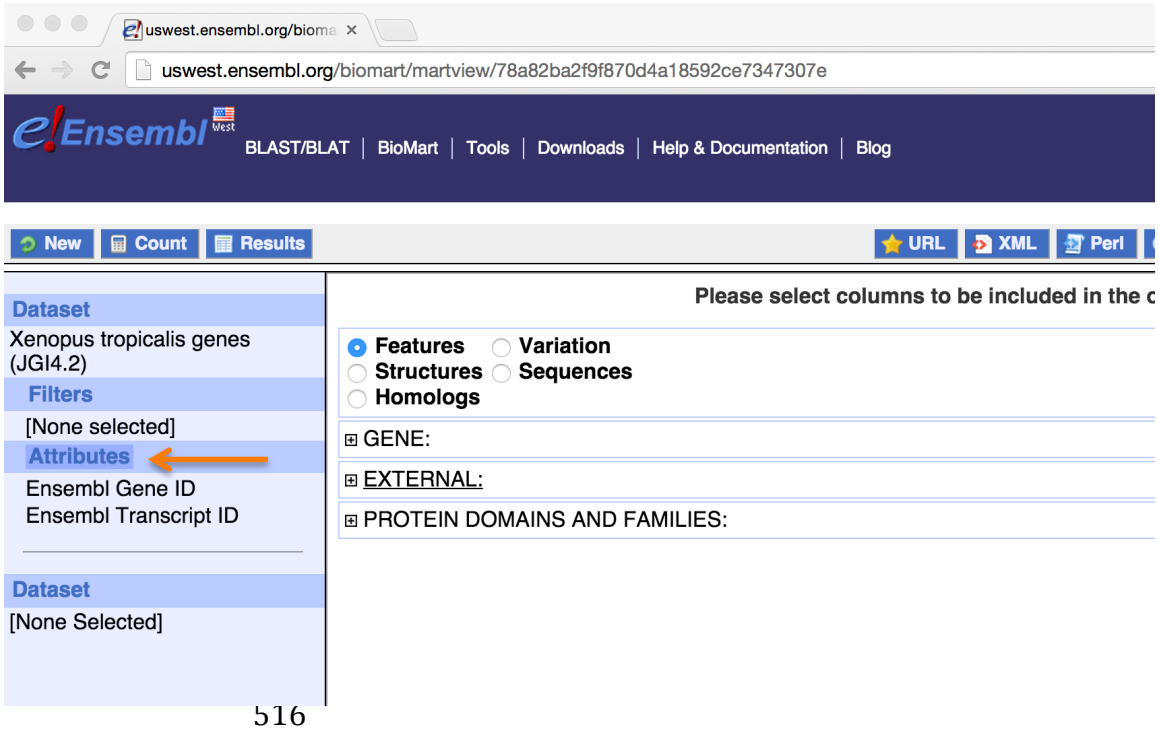
Step1. Go to the Ensembl homepage http://www.ensembl.org/ and click on "BioMart" located at the top.



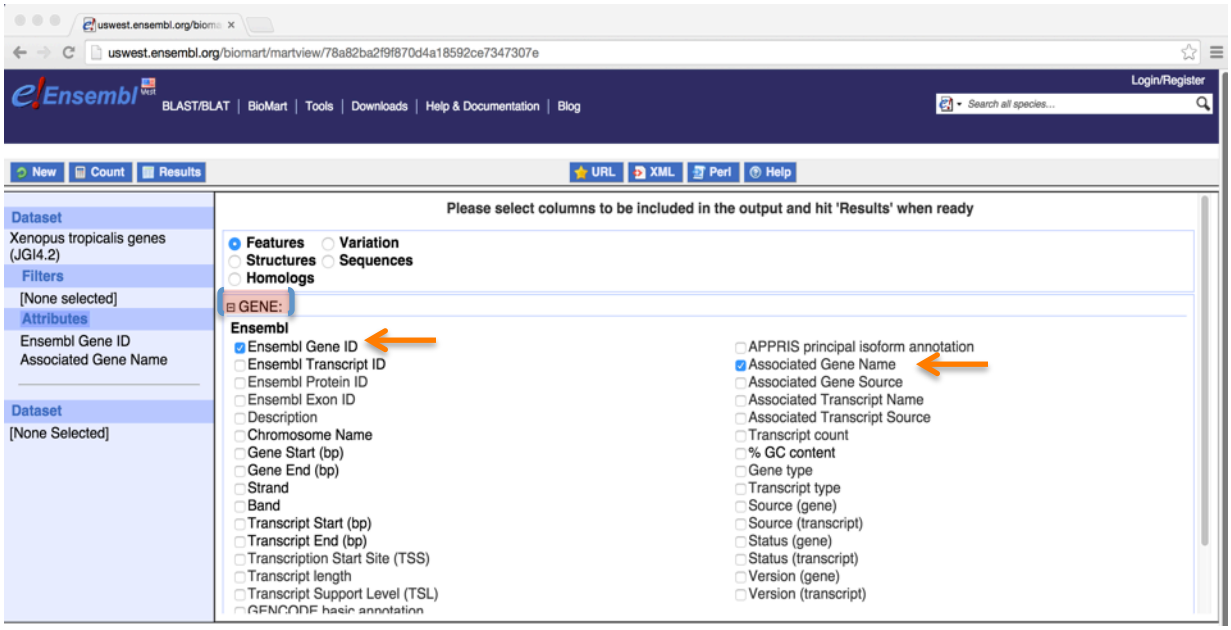
Step2. In the BioMart homepage, select "Ensembl Genes 78" and "Xenopus tropicalis genes (JGI4.2)".



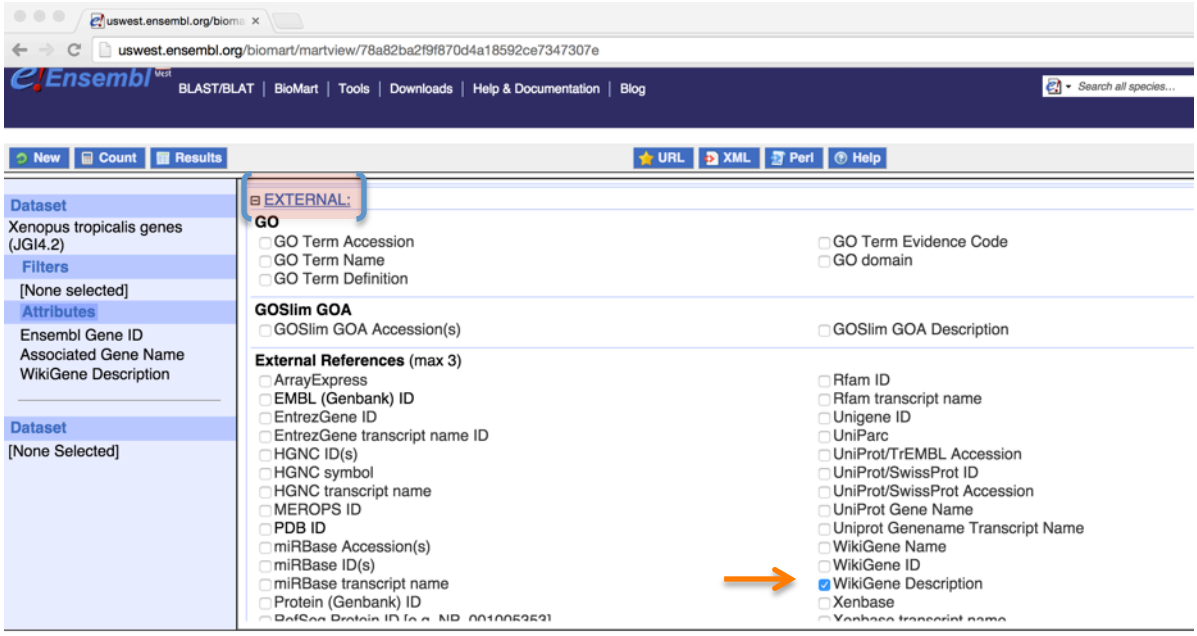
Step3. Click on “Attributes” icon to the left.



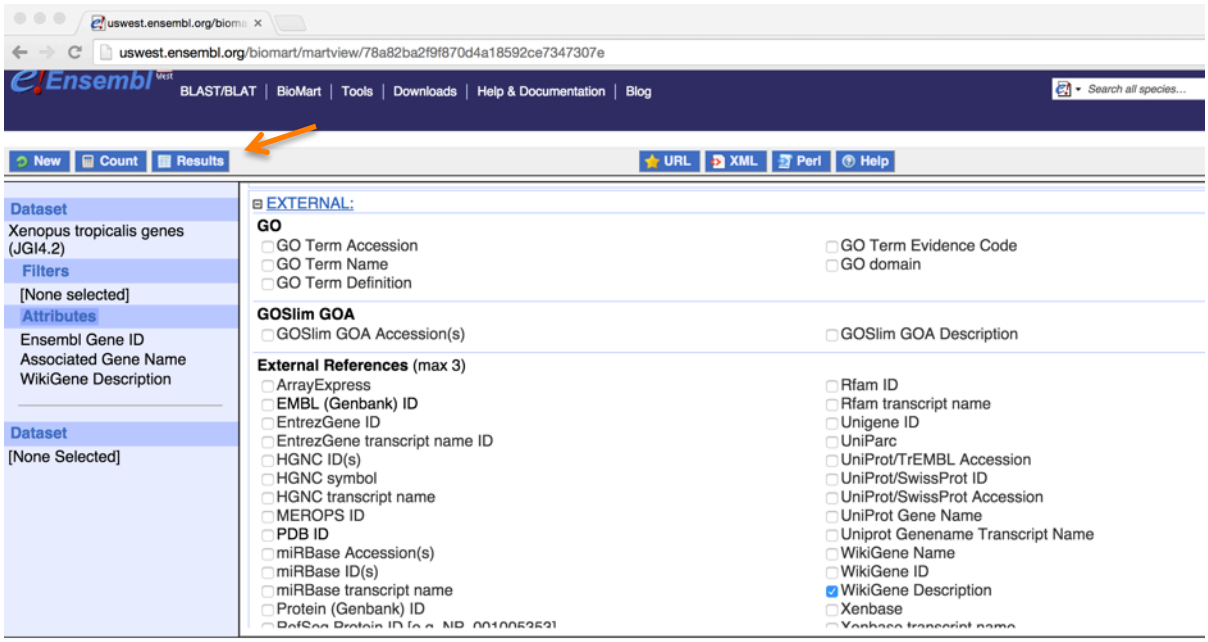
Step 4. Click on “GENE” to expand the manual. Check on “Ensembl Gene ID” and “Associated Gene Name”.



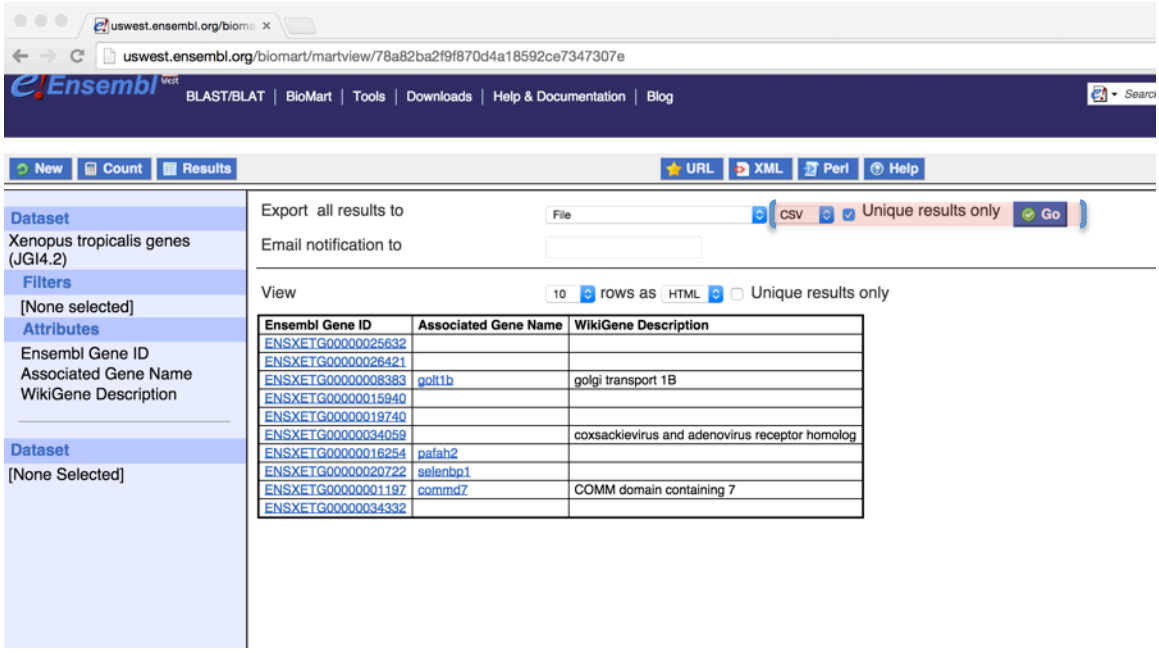
Step 5. Scroll down the window to find “EXTERNAL”. Click on it to expand the manual. Check on “WikiGene Description”



Step6. Click on “Results” icon.



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



Step 8. Save and rename the result like “*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
……

**Repeat steps demonstrated above to download *Anolis carolinensis* reference protein from the Ensembl FTP and gene annotation file from the BioMart.

**For this workshop, reference genomes and the corresponding biomart gene annotation files are already downloaded and located in “~/Desktop/MarkerDevelopment/associated_data/”.

First Annotation using *Xenopus_tropicalis:*

Input:
1. A folder that contains trinity assemblies. This file “combined.fasta” is located in “~/Desktop/MarkerDevelopment/data/annotation_pop/”

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

4. Reference gtf file:
Anolis_carolinensis.AnoCar2.0.78.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_pop/ -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

#Repeat the same steps and command above to annotate assemblies using *Anolis carolinensis* as a reference

##Copy the annotation results to "*~/Desktop/MarkerDevelopment/data*"
ke@NGS:~/Desktop/MarkerDevelopment/data$ scp -r ~/Desktop/MarkerDevelopment/associated_data/annotation_pop/* annotation_pop/

Output:
For each annotation, a new folder is generated under "~/Desktop/MarkerDevelopment/data/annotation_pop/":

combined_xenopus/
combined_anole/

"combined" is the name of the assemblies and after that is the name of the reference used for annotate the assemblies.

##The annotated fasta files are "XXX_xenopus_annotated.fasta" and "XXX_anole_annotated.fasta"

ke@NGS:~/Desktop/MarkerDevelopment/data/annotation_pop$ ls combined/*_annotated.fasta*

combined_anole/combined_anole_annotated.fasta
combined_xenopus/combined_xenopus_annotated.fasta

##make a new folder “probe_design_pop”under “~/Desktop/MarkerDevelopment/data/”.

ke@NGS:~/Desktop/MarkerDevelopment/data$ mkdir probe_design_pop

##copy all the annotated fasta files to “*probe_design_pop*”

ke@NGS:~/Desktop/MarkerDevelopment/data$ cp annotation_pop/combined/*annotated.fasta probe_design_pop/*

check annotations in one of the annotated fasta files in post_annotation:

ke@NGS:~/Desktop/MarkerDevelopment/data/probe_design_pop$ head -4 combined_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_pop/ -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……

676 "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
678 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
680 search.

682

6-ProcessAnnotation:

Four sub-functions are included: Merge, Filter, TrimORF , Exon

If transcript targets are desired then run:
Merge -> Filter -> TrimORF

If Exonic targets are desired then run:
Merge -> Filter -> Exon

Now I will demonstrate pipelines for generating transcript targets (Merge -> Filter -> TrimORF).

"6-ProcessAnnotation Merge": Merge annotations from various references. It also filters out redundant transcripts via self-blasting

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/

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

##make a folder "probe_design_pop/"
ke@NGS:~/Desktop/MarkerDevelopment/data$ mkdir probe_design_pop/

copy data files to "probe_design_pop/"
*ke@NGS:~/Desktop/MarkerDevelopment/data$ cp ~/Desktop/MarkerDevelopment/associated_data/annotation_pop/combined_*/*annotated.fasta probe_design_pop/*

ke@NGS:~/Desktop/MarkerDevelopment/data/probe_design_pop$ ls
combined_anole_annotated.fasta
combined_xenopus_annotated.fasta

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

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

ke@NGS:~/Desktop/MarkerDevelopment/data/probe_design_pop$ mv combined_anole_annotated.fasta other_files/

Commands:

Run "6-ProcessAnnotation Merge":

ke@NGS:~/Desktop/MarkerDevelopment/data$ 6-ProcessAnnotation Merge -a probe_design_pop/combined_xenopus_annotated.fasta -b probe_design_pop/other_files/ -d frog

Output:

"1-frog_annotation_filtered.final" under "~/Desktop/MarkerDevelopment/data/probe_design_pop/"

"6-ProcessAnnotation Filter": Basic filters on length, GC content, and repetitive elements

Input:

"1-frog_annotation_filtered.final" produced by "6-ProcessAnnotation Merge"

Commands:

Run "6-ProcessAnnotation Filter":

ke@NGS:~/Desktop/MarkerDevelopment/data$ 6-ProcessAnnotation Filter -f probe_design_pop/1-frog_annotation_filtered.final -a 150 -b 1000 -R frogs -g frog

Output:

" 2-frog_GC_length_repeatmasked.txt" under "~/Desktop/MarkerDevelopment/data/probe_design_pop/"

"6-ProcessAnnotation TrimORF": Trim off sequences outside the ORF

Input:

" 2-frog_GC_length_repeatmasked.txt" produced by "6-ProcessAnnotation Filter":

Commands:

Run "6-ProcessAnnotation TrimORF":

ke@NGS:~/Desktop/MarkerDevelopment/data$ 6-ProcessAnnotation TrimORF -f probe_design_pop/2-frog_GC_length_repeatmasked.txt -b ~/Desktop/MarkerDevelopment/associated_data/combined_refProt.fasta -c 100 -o frog

Output:

"3-frog_transcript_target.fasta" under "~/Desktop/MarkerDevelopment/data/probe_design_pop/"

++

Now I will demonstrate pipelines for generating exonic targets (Merge -> Filter -> Exon).

First run "6-ProcessAnnotation Merge" and "6-ProcessAnnotation Filter" following the instructions above

#Run 6-ProcessAnnotation Exon:

Input:

1. "2-frog_GC_length_repeatmasked.txt" produced by "6-ProcessAnnotation Filter":

2. Combined protein reference of "Xenopus_tropicalis.JGI_4.2.pep.all.fa" and "Anolis_carolinensis.AnoCar2.0.pep.all.fa"

ke@NGS:~/Desktop/MarkerDevelopment/associated_data$ cat Anolis_carolinensis.AnoCar2.0.pep.all.fa Xenopus_tropicalis.JGI_4.2.pep.all.fa > combined_refProt.fasta

3. Combined genomic DNA reference of "Xenopus_tropicalis.JGI_4.2.dna_rm.toplevel.fa" and "Anolis_carolinensis.AnoCar2.0.dna_rm.toplevel.fa"

ke@NGS:~/Desktop/MarkerDevelopment/associated_data$ cat Xenopus_tropicalis.JGI_4.2.dna_rm.toplevel.fa Anolis_carolinensis.AnoCar2.0.dna_rm.toplevel.fa > combined_refGenome.fasta

Commands:

Run "6-ProcessAnnotation Exon" (do not execute the command!):

ke@NGS:~/Desktop/MarkerDevelopment/data$ 6-ProcessAnnotation Exon -p ~/Desktop/MarkerDevelopment/associated_data/combined_refProt.fasta -g ~/Desktop/MarkerDevelopment/associated_data/combined_refGenome.fasta -f probe_design_pop/2-frog_GC_length_repeatmasked.txt -e 150 -E 1000 -o frog

Output:

"3-frog_exon.fa"

#Let's copy the output file from "~/Desktop/MarkerDevelopment/associated_data/" to "~/Desktop/MarkerDevelopment/data/probe_design_exons/".

ke@NGS:~/Desktop/MarkerDevelopment/data$ cp ~/Desktop/MarkerDevelopment/associated_data/3-frog_exon.fa probe_design_exons/

#check the output using "head"

>frog Contig1130_134_310 ENSXETG00000000011_exon2
GGATCATGCCAAAGTTCTTCACTACATCGGAGCTGGGGTTGCCTTCCCAACCAGTATGTT
GTTCATTTTCTTTCAGTCTATCCTGACCTACCGCATGGCACACACTTATTGGAACTGGTG
GGCTGGACACGTACGCTGTCTTCTTACGTTGTTTGGACTGGTCATTTTAGTGCTTAG
>frog Contig2807_214_363 ENSXETG00000000013_exon11
GATGCAAAAAATATAGAGGCAGAGGTAAAAGAGCTGAAGAAAGTTACAGATTTGAGCAT
TGTCCGGCTGCGGTTTACAGCATATTTGCCAGACAGCACTGGTGCCTATACTCTCCGTTT
AAAGCCAGTCATTTCTGACCCTATCCATGAC
>frog Contig3806_208_390 ENSXETG00000000060_exon2
GTCGTAGCCCTCCAGGTCTCTGAGCTTTTTGATTTCAAAAAGGTTTCCCTCCACAGCCAG
CAGGGAGATTTGGGAATCGCTGAGGATGCTCTGAGGTAACATACTGAGCTCCAGACAGT
TCTCTTCCAGACGTAAAACTTTGAGACGTGGACAATGTGAAACCTGAGCTGATATTTGG
GATAT
……

"frog" is the name of the focal species
"Contig1130_134_310": This exon is located between position 134 to 310 of Contig1130.
"ENSXETG00000000011_exon2": Ensembl gene ID and numer of the exon