Brinda Vipparthy

Bioinformatics: Tools for Genome Analysis

Summer 2024

Final Portfolio

August 12, 2024

TABLE OF CONTENTS:

Introduction	3
ORF and CDS Analysis Tools	4
File Formats in Bioinformatics	8
UCSC Genome Browser	12
IGV Analysis	16
Ensembl Tools	18

Introduction:

For this final portfolio, I wanted to focus on the various genome browser tools and databases that were implemented in solving the graded homework, quizzes, and exams in this course. The purpose of this course is to aid in the deeper understanding of genomic topics, such as genome variations, gene identification, and annotation of sequences, by providing information about tools and databases which can applied to analyze and interpret complex genomic data.

This portfolio highlights tools and databases, while providing context from the course in terms of discussion posts, graded homework, exams, and quizzes. The following topics are outlined in detail in this portfolio:

- ORF finder tools
- CDS analysis and detection tools
- Various file formats outlined in the course and their specific uses
- Various implementations and uses of UCSC Genome Browser
- IGV Analysis of different genomes
- Ensembl tools (BioMart as well as the genome browser)
- Galaxy and the various tools provided

ORF and CDS Analysis Tools:

When discussing ORFs and CDSs, it is important to differentiate between Bacterial and Eukaryotic genomes. Bacterial mRNA is polycistronic, meaning that a single gene can have multiple open reading frames, or ORFs. These ORFs are translated into separate proteins. Eukaryotic mRNA, however, is monocistronic, meaning that the mRNA contains only one coding region that corresponds to a single protein. This course discusses various tools and methods that are useful in deriving ORFs, which are the reading frames in bacterial and eukaryotic genomes, and CDSs, which are the coding regions in the gene. CDSs in eukaryotic genes contain only the exons and exclude the introns, which are spliced out.

There are a few tools that were mentioned in this course that can be useful in determining ORFs and CDSs:

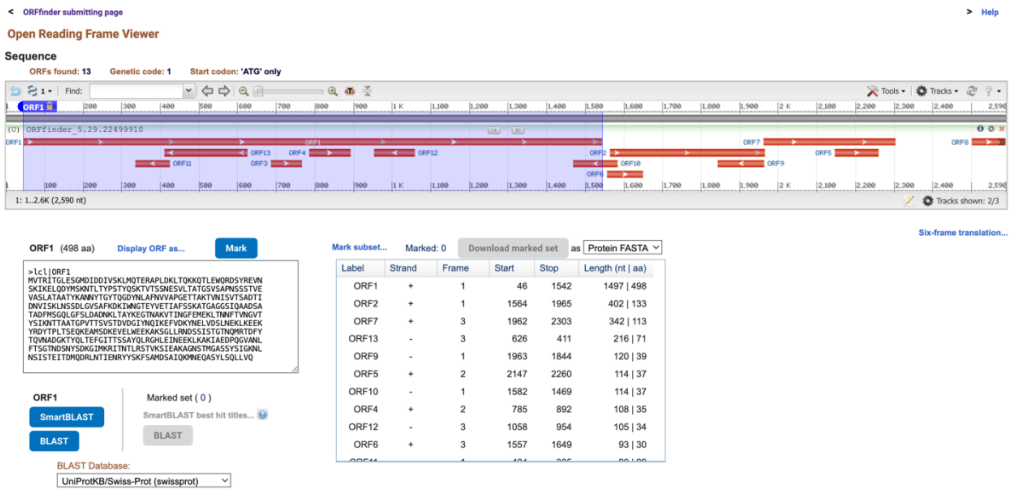
- ORF Finder
- FGENESB
- GLIMMER
- GeneMark
- EasyGene
- BPROM
- NCBI GenBank

Applications:

The below screenshot is an application of the ORF Finder Tool from Graded Homework 2. This tool was used to find the locations of the three longest ORFs in the Bacillus subtilis genome sequence.

1. Use ORF Finder to identify the locations of three coding regions (three longest ORFs) in the Bacillus subtilis genomic sequence (file:homework1.txt). (1 point)
    a. On what reading frames are each of the genes in the Bacillus DNA based on ORF Finder? (answer should be at the master pdf document)

       The three coding regions in the Bacillus subtilis genomic sequence are found in ORFs 1, 2, and 7, which are the three longest ORFs. ORF1 is located on the +1 reading frame at base pairs 46-1542, ORF2 is located on the +1 reading frame at base pairs 1564 – 1965, and ORF7 is located on the +3 reading frame at base pairs 1962 – 2303. Below is a screen shot of the output from ORF Finder. The table at the center of the page shows the reading frames in order from the largest at the top to the smallest at the bottom.



The below application is from Graded HW 2 and represents the use of the FGENESB tool to find CDSs in a bacterial S.helicoides strain.

3. Use FGENESB to identify CDSs in the partial sequence from S. helicoides strain TABS-2 (file: sheliprt.fasta). Use 'bacterial generic' as the training set. (1 point)
    a. How many CDSs are listed?

    There are 9 CDSs listed.

```
 Prediction of potential genes in microbial  genomes
 Time:   Tue Jan  1 00:00:00 2005
 Seq name: Spiroplasma helicoides strain TABS-2, partial sequence
 Length of sequence - 5500 bp
 Number of predicted genes - 9
 Number of transcription units - 6, operons - 2
     N      Tu/Op   Conserved  S             Start         End    Score
                   pairs(N/Pv)
     1     1 Op  1     .       +    CDS        635 -       991    117
     2     1 Op  2     .       +    CDS        998 -      1141    144
     3     2 Tu  1     .       -    CDS       1126 -      1365     73
     4     3 Tu  1     .       +    CDS       1334 -      1978    381
     5     4 Tu  1     .       +    CDS       2242 -      2463    231
     6     5 Op  1     .       +    CDS       2585 -      4003    998
     7     5 Op  2     .       +    CDS       4010 -      4678    423
     8     5 Op  3     .       +    CDS       4703 -      4768     72
     9     6 Tu  1     .       +    CDS       4880 -      5143    169
Predicted protein(s):
```

    b. How many mRNAs are predicted to code for those CDSs?

    There are 6 mRNAs predicted to code for these CDSs:
    i. Operon 1 contains CDS 1 and 2
    ii. Transcription unit 2 contains CDS 3
    iii. Transcription unit 3 contains CDS 4
    iv. Transcription unit 4 contains CDS 5
    v. Operon 5 contains CDS 6, 7, and 8
    vi. Transcription unit 6 constains CDS 9

The application below is an example of how GenBank can be used from the Discussion Post from M01. GenBank was used to find the CDS location of the capsular locus (cps) operon in Streptococcus pneumoniae.

(Thread) Operon Examples

Find examples of operons or genes from bacteria in GenBank. List the CDS locations and compare your results to others. In what format are polycistronic CDS regions annotated?

Brinda Vipparthy

May 23 11:17pm

I chose to focus on the capsular locus (cps) operon present in Streptococcus pneumoniae. This operon is responsible for the formation of the virulence factor of Streptococcus pneumoniae, the capsular polysaccharide. This operon contains many different genes that carry out this function, which demonstrates the polycistronic nature of bacterial organisms. For example, the wzd gene is located at 2185-2880 base pairs and is responsible for the putative regulatory protein, while the wzh gene is located at 1448-2179 base pairs and is responsible for the translation of the tyrosine protein phosphate. The information found for the wzd and wzh genes can be seen below. The format starts with the CDS location of the genes, followed by the operon that contains the gene, a note about what the gene is responsible for, codon start site, translation table, a description of the product, the protein id, and the translation.

```
CDS             2185..2880
                /gene="wzd"
                /operon="capsular locus, cps"
                /note="putative regulatory protein"
                /codon_start=1
                /transl_table=11
                /product="capsular polysaccharide biosynthesis protein
                Wzd"
                /protein_id="SBT85376.1"
                /translation="MMKEQNTIEIDVFQVLKTLWKHKLIILLVALVTGAGAFAYSTFI
                VKPEYTSTTRIYVVNRNQEGKSGLTNQDLQAGTYLVKDYREIILSQDVLEKVATNLKL
                DMPAKALTSKVQVTVPTDTRIVSISVKDKEPEEASRIANSLREVAAGKIVAVTRVSDV
                TTLEEARPATTPSSPNVRRNTLVGFLGGAVVTVITVLLIELLDTRVKRPEEVEEVLQV
                PLLGVVPDLDKMK"
```

```
CDS             1448..2179
                /gene="wzh"
                /operon="capsular locus, cps"
                /codon_start=1
                /transl_table=11
                /product="tyrosine protein phosphatase Wzh"
                /protein_id="SBT85375.1"
                /translation="MIDIHSHIVFDVDDGPKSREESKALLTEAYRQGVRTIVSTSHRR
                KGMFETPEEKIAENFLQVREIAKEVASDLVIAYGAEIYYTPDVLDKLENNRIPTLNNS
                RYALIEFSMNTPYRDIHSALNKILMLGITPVIAHIERYDVLENNEKRVRELIDMGCYT
                QINSSHVLKSKLFGEPYKFMKKRAQYFLERDLVHIIASDMHNVDGRPPHMAEAYDLVS
                QKYGEAKAQELFIDNPRKIVMDQLI"
```

Reference:
https://www.ncbi.nlm.nih.gov/nuccore/LT594599.1

File Formats in Bioinformatics:

During the course of this class, I was introduced to many different file formats that serve different purposes in the field of bioinformatics. Having a variety of file formats is particularly important in genomic data analysis because each format is designed to efficiently store, process, and analyze specific types of biological data.

BED (browser extendible data) files are useful for storing genomic regions in the form of chromosome position, start site, and end site. There are two types of BED files, 0-based and 1-based, which are necessary to note in the creation of the files. This file format is essential for tracking coordinates of exons, regulatory regions, genes, and genetic features. WIG (wiggle) file format is used for storing information at a genomic location and allows for data compression for larger data sets. This format is useful for analyzing GC percentage or probability scores. Another file format that is useful in storing probability scores is Bedgraph. This format is more useful with smaller datasets and keeps data preserved in its original format. VCF (variant calling files) are useful for finding chromosomal positions of variants and the number of reads that are aligned at each variant position. Additionally, SAM/BAM file formats are similar in nature, but BAM files are SAM files in binary format. These file formats are useful for storing aligned sequence data. FASTA/FASTQ files are useful for storing raw sequence data.

Types of file formats:

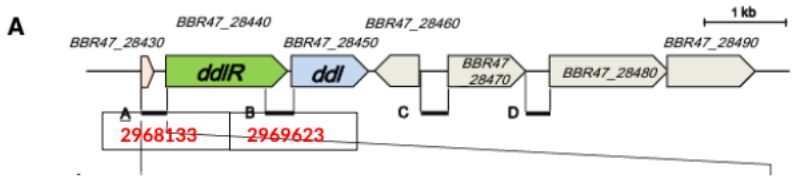
- Bed
- WIG
- Bedgraph
- SAM
- BAM
- FASTA
- FASTQ
- VCF

Application:

The application below is from Graded Homework 4 and shows the process of creating a BED file. This is a 0-based bed file, so one base pair must be subtracted from the start site. There are two bed file format lines, one for the promoter region and one for the 5'UTR.

1. **2.5 pts.** Create a BED6 file with 2 lines based on the attached paper (Takenaka_et_al-2015-FEBS_Journal.pdf). Figure 3 shows the location of transcription factor DdlR binding to the promoter region of the ddlR-ddl operon in *Brevibacillus brevis*. The chromosomal location of the ddlR CDS is **2968133..2969623**. The **zero-based** BED6 file should contain the location information of two genomic regions:

    a. 1 pts: The region bound by the DdlR transcription factor, which we will call the *promoter*. It is 170 bp in length, begins 140 nucleotides upstream from the start codon, and ends 29 nucleotides downstream from the start codon.



First, I want to label figure 3A from the pdf file with the CDS of the promoter Ddlr. This is so I can visualize the region in the figure with the given CDS. The promoter begins 140 nucleotides upstream from the start codon, which is at position 2968133.

So ,140 bp upstream of start site is the beginning of the promoter region.

- Start site = 2968133
    - 2968133 – 140 = 2967993

2967993 is the position of the start site for the promoter, but we need to compensate for the 1bp for the BED file format. 2967993 – 1 = 2967992.

- End site = 29 nucleotides downstream from start codon
    - 2968133 + 29 = 2968162

The BED file format for this would be:

chr1 2967992 2968162 promotor 0+

b. 1 pts: The 5' UTR, noting that the transcription start site, as predicted by BPROM, begins **38 nucleotides upstream from the start codon**. The 5' UTR is defined as the region from the transcription start site through the nucleotide that **immediately precedes the start codon**.

The 5'UTR, based on the description in part b, will be the promoter sequence ddlR and the nucleotide directly after ddl.

- Start site = 2968133 – 38 - 1 = 2968094
- End site = 2968133 – 1 = 2968132

The BED file format for this would be:

chr1 2968094 2968132 5UTR 0 +

The next application shows the use of Galaxy to convert from a SAM file format to a BAM file. This is from Exam 2.

Part 1 - 9 points

Part 1 uses the attached file, ERR181582a.sam. The SAM file is from a yeast sequencing run from *Saccharomyces cerevisiae* (version sacCer3) on chromosome I. The original SAM data can be found hereLinks to an external site. if you're interested (not needed for the exam).

1. (3 pts) What is a SAM file and how is a SAM file generated? Be sure to include in your answer what type of data is represented in a SAM file.

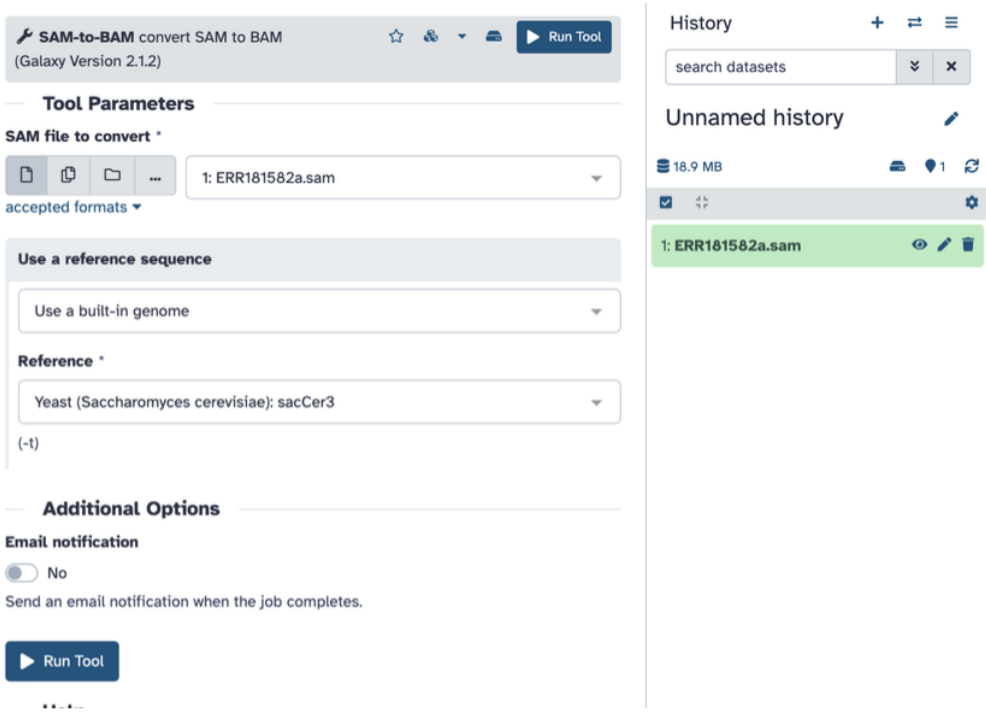
A SAM file, or a Sequence Alignment/Map file, is a file format that is intended to hold information about sequence alignment to a reference genome. This can be particularly useful for the storage of NGS data aligned to reference genomes. The SAM file type is generated by alignment programs like BWA and HISAT.

2. (3pts) Upload the SAM file to Galaxy. In Galaxy, convert the SAM file to a BAM file. Submit the BAM file.

The file name is ERR181582a_converted.bam.

3. (3 pts) List the Galaxy tool(s) you used and the parameter(s) you set to complete the previous question. Screenshot(s) or text is fine to submit.

To do the above question, the original SAM file was uploaded onto Galaxy, and the SAM-to-BAM tool was used to convert the file to a BAM file. The sacCer3 genome was used as the reference genome (seen in screenshot below).



UCSC Genome Browser:

The UCSC Genome Browser is one of the tools that was introduced in this course which is has multiple attributes contributing to the understanding, analysis, and visualization of genomic data. This tool is primarily a eukaryotic genome browser and includes data on mRNA, RefSeq gene alignments, assembly data and more. One important feature in this tool is the ability to add your own data to analyze as well as looking up genes within a variety of genomes. There are multiple tracks that allow one to view desired data. These tracks have different displays which can be implemented based on what type of information is needed.

Tracks in the UCSC Genome Browser:

- UCSC Genes
    - Based on data from RefSeq, GenBank, and UniProt
    - Allows for visualization of genomic region in the gene
- Conservation track
    - Allows for the visualization of conservation levels in various species
    - Multiple alignments of species
    - Can be useful for comparative genomics
- dbSNP 155
    - shows short genetic variants such as indels and single nucleotide variants
- ENCODE Regulation
    - Allows for the visualization of histone modifications
    - Determined by ChIP-seq assays

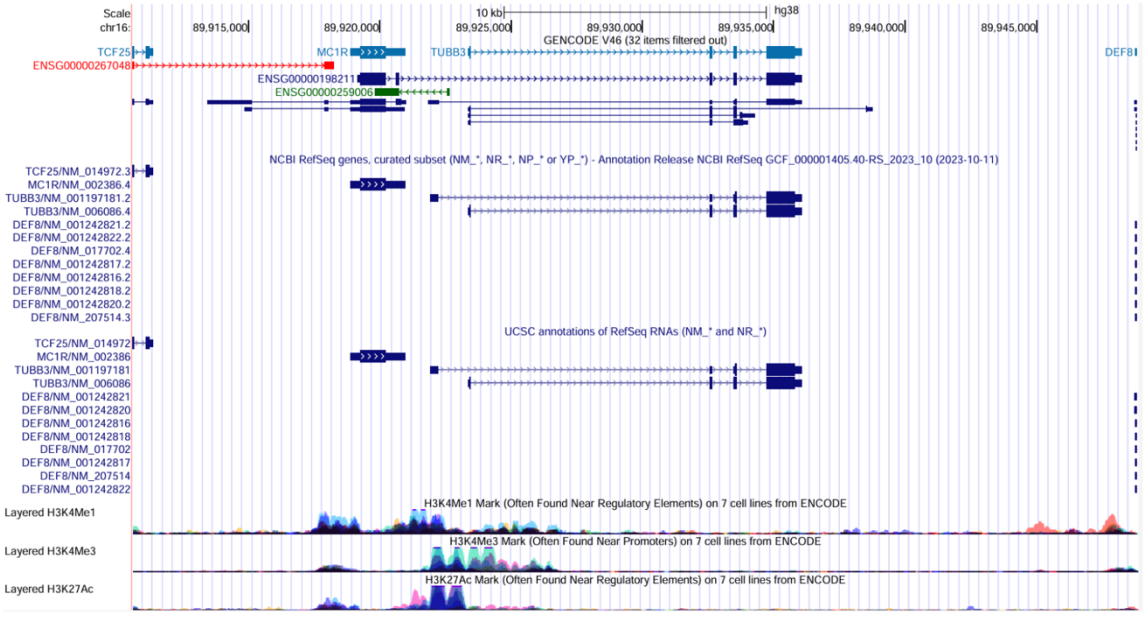
Applications:

The application below is from Exam 2. The ENCODE regulation track from UCSC Genome Browser was used to visualize histone modifications present near the transcription start site in the TUBB3 gene.

Part 2 - 8 points

1. (2 pts) Find the human TUBB3 gene using the UCSC Genome Browser (hg38). Turn on the Encode Regulation track (HINT: set display mode to "full" for these tracks) and NCBI RefSeq genes. In a few sentences, describe what you see at the TUBB3 locus in terms of the Encode Regulation tracks. Include in your answer what histone modification(s) appear(s) near the transcription start site of the TUBB3 gene. Submit a screenshot of this locus. (HINT: click View > PDF/EPS at the top of the browser page to export a PDF/EPS file.).

After turning on the ENCODE regulation track, I was only able to see information on the H3K37Ac modification, so I clicked on the track and enabled the H3K4Me3 and H3K4Me1 tracks in order to get a better understanding of which modifications are occurring. The first screenshot below is with the original ENCODE regulation track. The second screenshot includes the other two histone modifications.

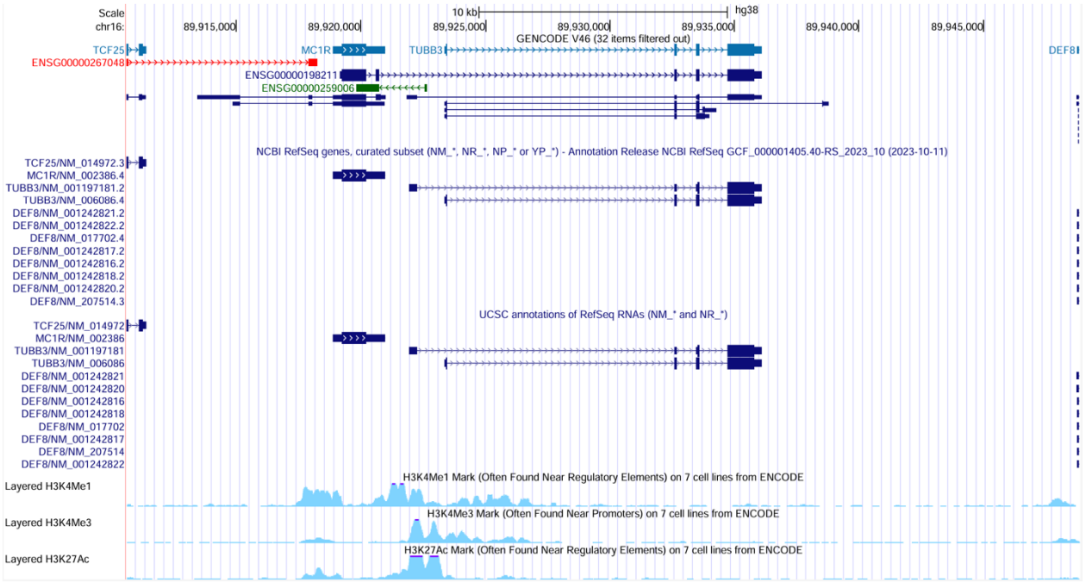


Based on the screenshot above, it can be seen that the H3K4Me3 and H3K27Ac histone modifications are present near the TSS of TUBB3.

HUVEC , or human umbilical vein endothelial cells, are used to study endothelial cell function and are retrieved from the vein of the umbilical cord (Human umbilical vein endothelial cells, Huvec Cells). Based on the Encode tracks, I believe that this gene is expressed in the HUVEC cell line. This is mainly due to the expression and presence of histone modifications H3K4Me3 and H3K27Ac in the promoter region of the TUBB3 gene. These modifications indicate an active promoter and transcription region.

References:

"Human Umbilical Vein Endothelial Cells, Huvec Cells." *Human Umbilical Vein Endothelial Cells, HUVEC Cells*, www.moleculardevices.com/applications/cell-counting/cell-counter-huvec-cells#:~:text=Human%20umbilical%20vein%20endothelial%20cells%20(HUVEC)%20are%20primary%20cells%20isolated,normal%20and%20tumor%2Dassociated%20angiogenesis. Accessed 1 Aug. 2024.



The application below is a discussion post response highlighting the importance of the CRISPR track on the UCSC Genome Browser.

(Thread) Favorite UCSC track

Pick a UCSC track that has not been mentioned and explain what it is, how the data were generated, and what is interesting about the track.

Brinda Vipparthy

Jun 24 6:32pm

Hi Joshua,

I enjoyed reading about the CRISPR target track on the UCSC genome browser. This track is extremely useful for targeting specific sites within the genome that can be cleaved for specific purposes. I found it interesting that the track shows the efficiency of cleavage at each target site. Gene therapy is a growing field of study, therefore, the CRISPR track will allow for better personalized patient care and treatment through gene editing interventions.

IGV Analysis:

IGV, also known as Interactive Genomics Viewer, is a popular database that was designed to work with large genomic datasets. In order to use this database, the IGV tool can be downloaded or used in the web browser. For this course, I preferred using the downloaded version. I found this tool extremely useful and user friendly, as it was intended for bioinformaticians as well as those who aren't trained in bioinformatics.

Applications:

The application below is from Graded Homework 6. The IGV browser was used to load datasets and visualize the EPHX1 gene. After zooming into exon4, there were multiple SNPs which were visualized. This can be seen in part 3c.

Part 3

Using IGV for hg19, load dbSNP 1.4.7 or newer (i.e. Available Datasets > Annotations > Variation and Repeats > dbSNP 1.4.7) and an exome sequencing track from the 1000 Genomes project (1000 Genomes > Alignments > GBR > exome > HG00096 exome). Go

to the EPHX1 gene and zoom in on the exon #4.

a. (0.25pts) How many SNPs overlap this exon and what are the SNP IDs?

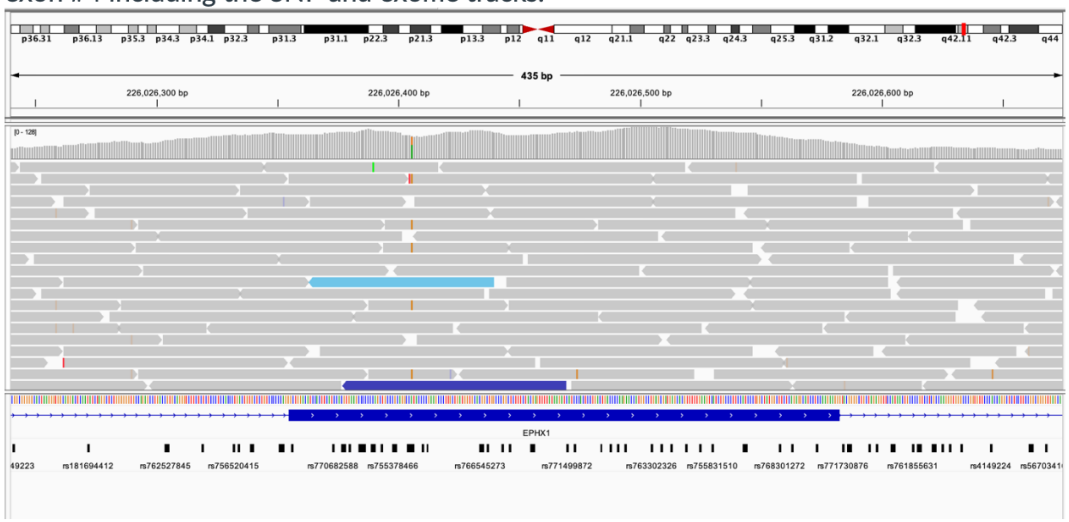
SNP IDs: rs746998584, rs770682588, rs112721617, rs745675416, rs769262345, rs775238551, rs762318357, rs763704955, rs773821078, rs761335756, rs754199890, rs755378466, rs141157588, rs752988508, rs55784606, rs2234922, rs761017131, rs373688139, rs766545273, rs769650380, rs779549215, rs147296174, rs144904318, rs199979074, rs761426131, rs771499872, rs777312225, rs139835141, rs754290186, rs765684911, rs376877493, rs763302326, rs764312984, rs151050888, rs757356785, rs371520880, rs755831510, rs779962356, rs748882501, rs768301272, rs778715423, rs140788022

There are 42 SNPs overlapping this exon region.

b. (0.25pts) At which SNP(s) in part a does this individual appear to be heterozygous? What is the sequence count for each nucleotide at this(these) position(s) (**Hint**: look at the HG00096 exome Coverage track)

The individual is heterozygous at rs2234922. The sequence count is A:57, G:34.

c. (0.25pts) Submit the image from IGV, zoomed in on but showing all of the exon #4 including the SNP and exome tracks.



Ensembl Tools:

Ensembl is a genome browser that is specific to the sequenced genomes for eukaryotes. This database offers tools for exploring gene annotations, comparative genomics, genetic variants, regulatory regions, and more. When searching for a specific gene, there are multiple tabs on the left-hand side of the page which are useful for viewing variations, different transcripts, orthologues, paralogues, and more. Ensembl is also useful for visualizing the gene location as well as the flanking genes.

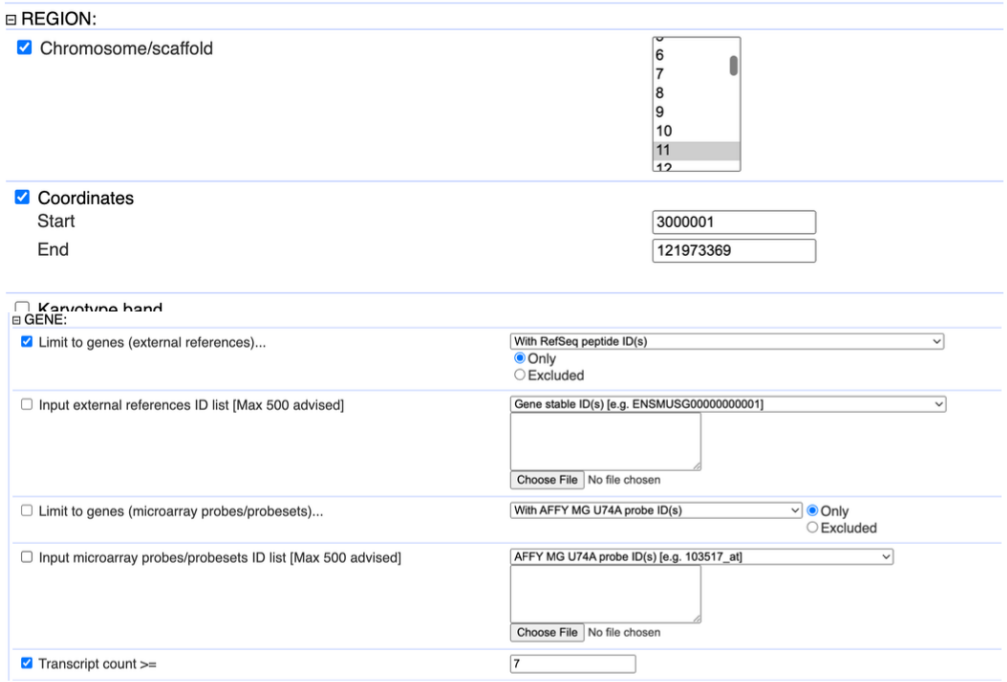
Within Ensembl, BioMart is a tool that is particularly useful for generating a table of data specific to SNPs, variants, and chromosomal positioning. BioMart is found within the Ensembl website and has filters and attributes that can be changed depending on the desired output.

Furthermore, BiomaRt is a Bioconductor package that uses R to derive data from the Ensembl database. This application is used in the command line and is useful in visualizing and analyzing high-throughput genomic data. Individuals who are proficient, or more comfortable using coding languages may prefer using R to extract data than BioMart. In my personal experience during this course, I found BioMart easier to navigate, as I do not have much experience with the R language.

Applications:

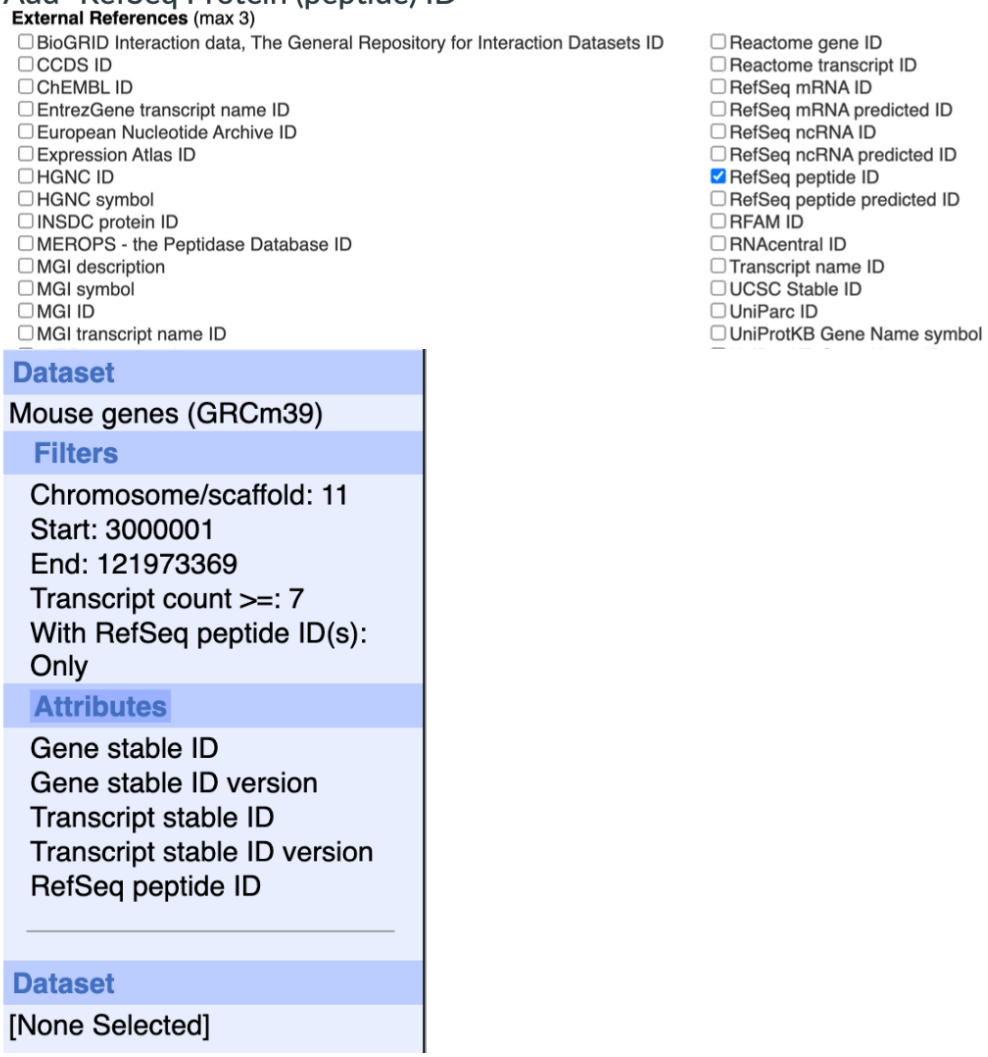
The application below is from M04 Graded homework. BioMart was used to create a dataset. The various steps in terms of filters and attributes are highlighted.

1. **2.5 pts.** Use the web-based Biomart in Ensembl to create a dataset and save it as a TSV, CSV, or XLS file. Use the following parameters to make the dataset:
    1. *Dataset:*
        1. Ensembl Genes 100 **(or the latest version)**
        2. Mouse genes (GRCm38.p6) **(or the latest version)**
    2. *Filters:*
        1. Chromosome 11
        2. Band E2 only
        3. Transcript count >=7
        4. Limit to genes with RefSeq protein (peptide) IDs only



3. *Attributes:*
    1. Default attributes

2. Add “RefSeq Protein (peptide) ID”



I decided to save the file in XLS format. The file name is problem2.xls.

The application below is from Exam 1 and depicts the use of R to extract data from Ensembl. Two tables were generated using R code.

Part 3 - 8 points

Search OMIM.org for "huntington's disease". The first five entries all have this or a similar phrase in the title. Record the five identifiers (six-digit numbers) of those five records. The corresponding biomaRt filter name for these identifiers is "mim_morbid_accession". Use biomaRt to retrieve two tables with the following attributes, limiting to the five MIM values you found:

The first five identifiers:

1. 603218
2. 604802
3. 143100
4. 606438
5. 607136

First table (2 points)
Entrez Gene ID
HGNC symbol
Ensembl Gene ID

```
  entrezgene_id hgnc_symbol ensembl_gene_id
1          3064         HTT ENSG00000197386
2          5621        PRNP ENSG00000171867
>
```

R code to get output of first table:

```
>
> library(biomaRt)
> mim_ids <- c("143100", "603218", "613004", "606269", "613005")
> ensembl <- useMart("ensembl")
>
> ensembl <- useDataset("hsapiens_gene_ensembl", mart=ensembl)
first_table <- getBM(
  attributes = c("entrezgene_id", "hgnc_symbol", "ensembl_gene_id"),
  filters = "mim_morbid_accession",
  values = mim_ids,
  mart = ensembl
)
> print(first_table)
  entrezgene_id hgnc_symbol ensembl_gene_id
1          3064         HTT ENSG00000197386
2          5621        PRNP ENSG00000171867
> |
```

Second table (2 points)
HGNC symbol
Ensembl Gene ID

Ensembl Transcript ID

```
   hgnc_symbol ensembl_gene_id ensembl_transcript_id
1          HTT ENSG00000197386       ENST00000680239
2          HTT ENSG00000197386       ENST00000680956
3          HTT ENSG00000197386       ENST00000680360
4          HTT ENSG00000197386       ENST00000681528
5          HTT ENSG00000197386       ENST00000647962
6          HTT ENSG00000197386       ENST00000649900
7          HTT ENSG00000197386       ENST00000680291
8          HTT ENSG00000197386       ENST00000355072
9          HTT ENSG00000197386       ENST00000648150
10         HTT ENSG00000197386       ENST00000506137
11         HTT ENSG00000197386       ENST00000512909
12         HTT ENSG00000197386       ENST00000510626
13         HTT ENSG00000197386       ENST00000649131
14         HTT ENSG00000197386       ENST00000509618
15         HTT ENSG00000197386       ENST00000650588
16         HTT ENSG00000197386       ENST00000650595
17         HTT ENSG00000197386       ENST00000513639
18         HTT ENSG00000197386       ENST00000513326
19         HTT ENSG00000197386       ENST00000509043
20         HTT ENSG00000197386       ENST00000509751
21         HTT ENSG00000197386       ENST00000512068
22         HTT ENSG00000197386       ENST00000513806
23         HTT ENSG00000197386       ENST00000508321
24        PRNP ENSG00000171867       ENST00000430350
25        PRNP ENSG00000171867       ENST00000379440
26        PRNP ENSG00000171867       ENST00000424424
27        PRNP ENSG00000171867       ENST00000457586
>
```

R code to get output for second table:

```
> library(biomaRt)
> mim_ids <- c("143100", "603218", "613004","606269","613005")
> ensembl <- useMart("ensembl")
> ensembl <- useDataset("hsapiens_gene_ensembl", mart=ensembl)
> second_table <- getBM(attributes =c("hgnc_symbol", "ensembl_gene_id",
"ensembl_transcript_id"), filters = "mim_morbid_accession", values =
mim_ids, mart = ensembl)
> print(second_table)
```