Table of Contents

Bioinformatics: Tool for Genome Analysis

Final Portfolio

Brian Wiley

Johns Hopkins University

MS Bioinformatics

Table of Contents

Table of Contents

Table of Contents

1. Purpose

The purpose of this portfolio is to demonstrate my research experience using bioinformatics tools, biological databases, protein bioinformatics, open source command-line tools and software programming during one of my final semesters in my Master's at Johns Hopkins.

2. Tools Used

This section briefly lists tools used. More in depth use of these tools is explained in section 4 including cases studies in which the tools were used.

2.1. General Annotations Tools

tbl2asn

Genbank

Sequin

2.2. Prokaryotic Annotation Tools

Glimmer3

ORF Finder

FGENESB

BPROM

Prodigal

2.3. Eukaryotic Annotation Tools

Augustus

Genscan

FGENESH

HMMGene

Splign

2.4. *Other Core Annotation and Workflow Tools*

samtools

bcftools

IGV

Table of Contents

2.5. Protein Bioinformatics Tools

All the tools from Protein Bioinformatics class. These are just listed to show my experience utilizing Protein Bioinformatics resources.

2.5.1. Alignment Tools

ClustalO, MAFFT, T-Coffee, Muscle, Kalign, PROBCONS

2.5.2. Profiles and Motifs

PROSITE, ProtScale Regular Expressions

2.5.3. Visualization

Chimera, Pymol

2.5.4. Secondary Structure and SASA Prediction

PredictProtein including PHDsec, PHDacc, PHDhtm by Rost Lab.

2.5.5. PDB/Structure Validation

PROCHECK, Bio3D (R), WHATCHECK

2.5.6. Comparative Modeling and Fold Recognition

FFAS, Fugue, I-Tasser, Modeller, SWISS-MODEL, Phyre2

2.6. Databases

Nuccore/Nucleotide

Pubmed

Protein

UniProt

PDBsum

Interpro

Pfam

PATRIC

Ensembl

UCSC Table Browser

dbSNP

ClinVar

2.7. Genome Browsers

NCBI Variation Viewer

Table of Contents

NCBI Genome Data Viewer

Ensemble

UCSC

2.8. File Types

Bed

GFF/GFF3

GTF

Wig/Wiggle

Sam/Bam

Genbank

Sequin

ASN.1

Fasta/Fastq

PDB file (.pdb)

VCF/BCF

2.9. Open source command-line tools

Deeptools

MACS2

bwa

bowtie2

STAR

histat2

trimmomatic

fastqc

SRA-toolkit

freebayes

VCFlib

Table of Contents

3. Software

This section includes main links to my coding portfolio on github for each of the programming languages below.

Python

De Bruijn graph and Eulerian path – Using these algorithms for mock de novo genome assembly

Modeller – Using the Modeller software package to align protein sequences and performing comparitive modelling

UCSF Chimera Driver – Using this driver to perform 3D structural alignment of protein chains in Chimera

MySQL and neo4j Python drivers

R

biomaRt – Using the biomaRt package from Ensembl

NetOGlyc2 – As gnuplot was not working for NetOGlyc2 at PROSITE, created this script to graph it

Microarray Scripts – Scripts for analyzing RNA microarrays, RNA-seq, and qRT-PCR

Perl

Module 2

Module 2 contains scripts as an intro to Perl programming using arrays, hashes and regular expressions

Module 3

Module 3 goes more in depth to using LibXML, XMLSimple, LWP::UserAgent, JSON, and DBI for programmatically interfacing with Biological database APIs an storing information in an SQLite using the DBI driver.

Module 6

Module 6 goes into programmatically using NCBI Blast as well as the API for interacting with the NCBI Pubchem and PDBe database by EMBL. This mostly uses UserAgent, LibXML and JSON packages for perl.

Table of Contents

Bash

Variant calling pipeline – Specific for finding and annotating variants in the exonic regions of BRCA1. Requires installing trimmomatic, freebayes, vcflib, bedtools, samtools and 2 arguments 1) srr.ids file holding SRR dataset ids and 2) refGene file for BRCA1 exons

---

Java

Java was used in my data structures and algorithms courses and so the high level for these code submissions are more general to all industries and not only bioinformatics.

Table of Contents

4. Case Studies

4.1. Week 1 - Gene annotation and structure

Genbank, Accessing Genbank files with Bio::DB::EUtilities in perl in database class.

In the first week of class we reviewed general annotation. A major topic was reading Genbank files. Interesting I started very early on week 3 in my Biological databases courses which had a module on E-utilities and accessing information from NCBI progromatically. I discovered a package for accessing Genbank file in perl.

I wrote a perl script in my database class, eUtils_genbank.pl to access Genbank files searching nucleotide/nuccore database and return exons of search hits or protein database and return CDSs of search hits. Usage is below:

```
Usage:

    eUtils_genbank.pl -q [query] -e [e-mail] [OPTIONS...]

Options:

    --max-return=<number>

        How many max database entries to return

    -t, --taxid=<number>

        Option filter by tax id

    -r, --refseq

        Option to filter by only RefSeq entries

    -d, --db=<String>

        Database option. Default is 'nucleotide'. Only allows for

        'nucleotide', 'nuccore', or 'protein'. I figured a way to get NG and

        NC reference number from 'gene' database but it requires parsing the

        Entrez gene file for the genomic accessions. It's a pain.
```

Table of Contents

4.2. Week 2 - Prokaryotic genes and gene prediction

Using tools in the Glimmer package, genome for Halanaerobium praevalens and contig for strain of Halanaerobium predict coding sequences for the contig. Compare this to FGENESB.

```
# obtain open reading frames from complete genome in hprev_genome.fasta

# using entropy threshold of t=1.15

long-orfs -n -t 1.15 hprev_genome.fasta Hpraevalens.longorfs

# extract sequences for training using -t for no stop codon

extract -t hprev_genome.fasta Hpraevalens.longorfs > Hpraevalens.train

# create interpolated context model from train set

build-icm -r Hpraevalens.icm < Hpraevalens.train

# run glimmer3 to predict CDS' s

# maxoverlap=50, min_gene_length=110, threshold_score >= 30

glimmer3 -o50 -g110 -t30 halan.fasta Hpraevalens.icm halan

# extract sequences from halan.predict using -t for no stop codon

extract -t halan.fasta halan.predict > halan.glimmer
```

Results:

```
>Halanaerobium sp. MDAL1, whole genome shotgun sequence
orf00001      171      350  +3    11.68
orf00003      343     1626  +1     8.96
orf00004     1629     4733  +3     6.58
orf00005     5786     4971  -3     8.13
```

From Softberry homepage run FGENESB with full genome and contig to predict CDS sequences

Selecting BACTERIAL generic as closest organism returns following results:

```
Output:
Number of transcription units - 2, operons - 1
     N      Tu/Op   Conserved  S             Start         End    Score
                    pairs(N/Pv)
     1     1 Op  1     .       +    CDS          3 -       350    298
     2     1 Op  2     .       +    CDS        343 -      1626   1063
     3     1 Op  3     .       +    CDS       1629 -      4733   1901
     4     2 Tu  1     .       -    CDS       4971 -      5786    654
```

Table of Contents

Install and use Prodigal for bacterial gene prediction to compare with Glimmer and FGENESB

Steps:

1) Download binary with:

```
wget https://github.com/hyattpd/Prodigal/releases/download/v2.6.3/prodigal.linux
```

2) Change name to prodigal and move to desired directory

3) Alternatively download source code and make with:

```
wget https://github.com/hyattpd/Prodigal/archive/v2.6.3.tar.gz
tar -xvzf v2.6.3.tar.gz
cd Prodigal-2.6.3
make install INSTALLDIR=/home/coyote/tools
```

4) Run prodigal in training mode with:

```
prodigal -i hprev_genome.fasta -t hprev_genome.train
```

5) Use training to predict CDS sequences

```
prodigal -i halan.fasta -t hprev_genome.train -a halan_protein.faa
```

Results for Prodigal are same as with FGENESB:

```
FEATURES             Location/Qualifiers
     CDS             <3..350

     CDS             343..1626

     CDS             1629..4733

     CDS             complement(4971..5786)
```

Use PATRIC to find information on bacterial genomes as well as transcriptomic experiments

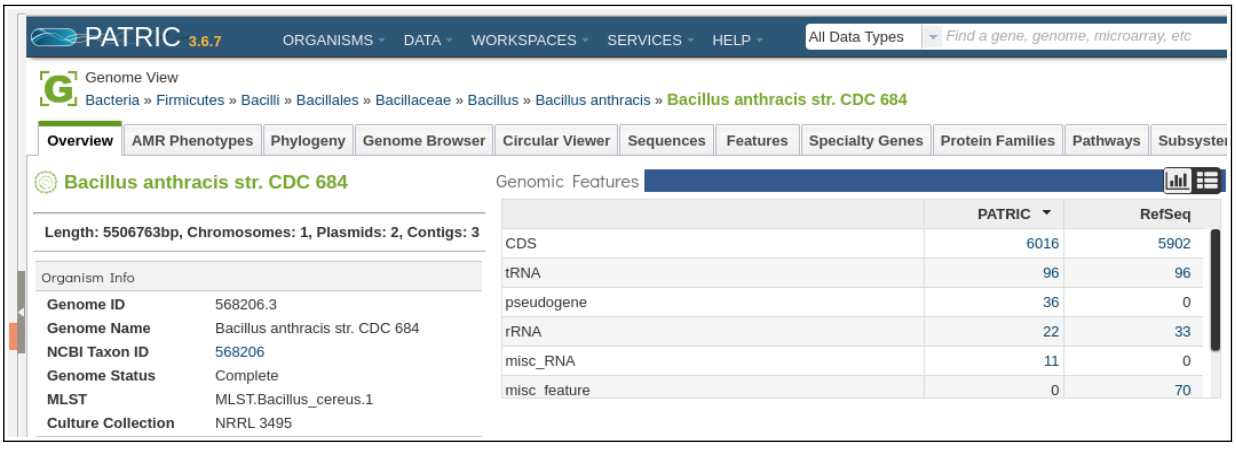


Table of Contents

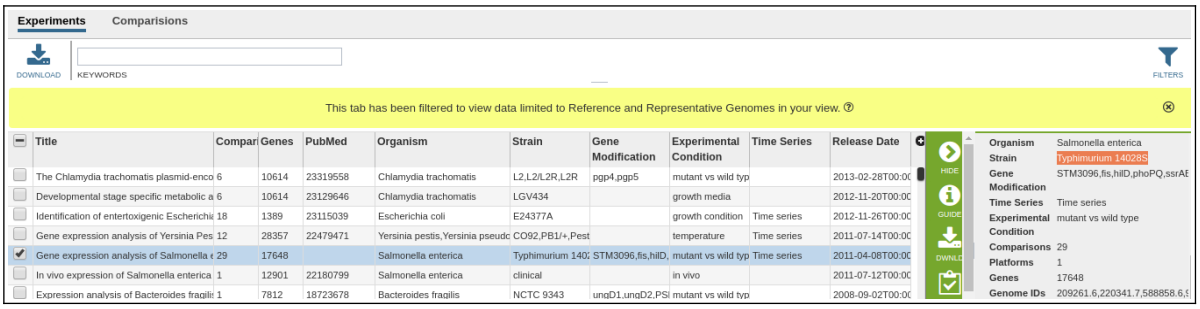


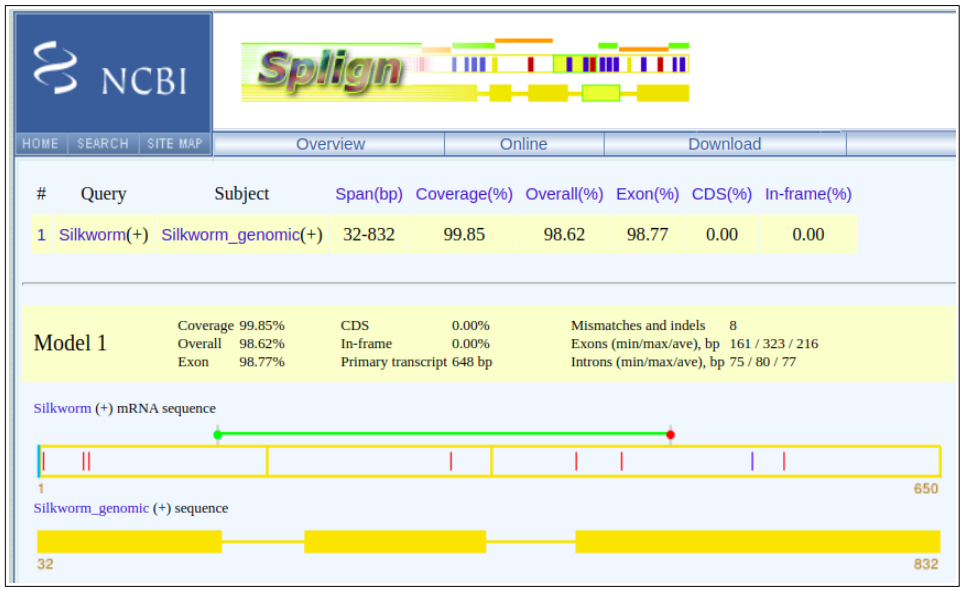
Table of Contents

4.3. Week 3 - Eukaryotic genes and gene prediction

Add tracks to Variation Viewer: Added .bed file for CpG island and H3K27ac narrowPeaks wiggle file to NCBI Variation Viewer.

Align mRNA to genomic DNA in Splign. Then annotated with tbl2asn and Sequin to generate Genbank file. Steps below:

Step 1: Splign with cDNA input from silkworm mRNA fasta file and Genomic input from silkworm genomic input file for the gene FK506-binding protein (LOC733041)



Step 2: install tbl2asn from ftp here. Using the guide by NCBI for tbl2asn to create Feature table.

Table of Contents

```
>Feature Silkworm genomic_sequence
1   878 REFERENCE
            PubMed  26679986
1   878 gene
            gene    LOC733041
            note    FK506-binding protein
            db_xref GeneID:733041
32  195 mRNA
270 430
510 832
            product silk_mRNA
159 195 CDS
270 430
510 637
            gene    LOC733041
            codon_start 1
            product FK506-binding protein
            protein_id  NP_001040498.1
32  195 exon
270 430 exon
510 832 exon
```

Step 3: Run tbl2asn with three required input files (silkworm genomic DNA, feature table, and submit file from NCBI) and update in Sequin before exporting genbank file (silkworm.gb)

```
tbl2asn -p silkworm_tbl2asn
```

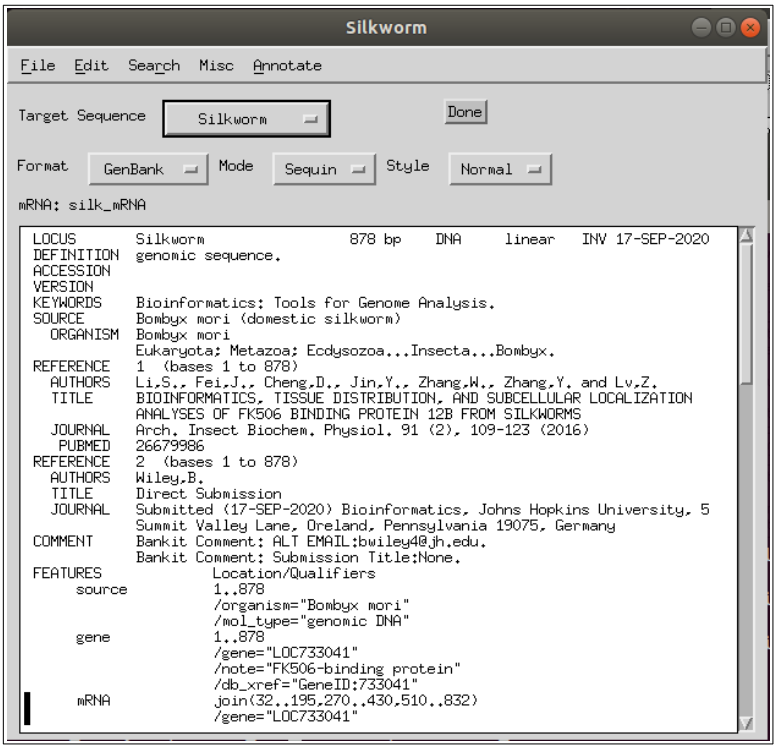


Table of Contents

Downloaded and install Augusts to run gene prediction analysis on genomic DNA from chicken from command line.

Steps:

1) Clone package from github with:

```
git clone https://github.com/Gaius-Augustus/Augustus.git
```

2) Install samtools, bcftools, and htslib to build Augustus with bam2wig

3) Run gene prediction with:

```
augustus --species=chicken chicken_gene.txt > chicken.out.gff
```

3) Obtain the proteins, coding exons, and coding sequence (mRNA) using getAnnoFasta.pl script:

```
getAnnoFasta.pl chicken.out.gff --seqfile=chicken_gene.txt
```

Exon sequences:

```
>g1.t1.cds1
atgaacttgtccaagctgaagctgtcggacatcacccagggcatccagaagctcaacagaggagtgcag
>g1.t1.cds2
gtcccttgcaatgacaccagagtggcacaagtggctttcaaggatcggaag
>g1.t1.cds3
ctgtcagagcaagagctgctgtgccaggctgccacagtgctggacaacatgaccgactgcaagaaggactacgagcccctcatcaccagtctgaagagcctgcacggcatgacg
>g1.t1.cds4
aactgcccgccgagcaccgacaatgagatctacctgcggaacttcttgccggcactggggaactacacccaggcgctgtaccggcggatctcggccacagctgcaaactga
>g2.t1.cds1
atgagctcctcactgcccaccctgctggcactgctggtgctgctggccggccctggggctgtgcccacgctgtgcttacagctctcagtgccgctgatggagagcatccggatagtgaatgacatccagggagag
>g2.t1.cds2
gtttcctgcgtcaagatgaacgtgacagatatctttgcagacaataag
>g2.t1.cds3
acaaataacaaaactgagctcttatgcaaagcctccacaattgtttgggagagccagcactgccacaagaacctgcagggtctcttcctcaacatgcgtcagctcctgaatgccagcagcacctccctcaag
>g2.t1.cds4
gcaccatgtcccacggcagcaggcaacactacttcaatggagaagttcctagcagacctacgtaccttcttccaccaactagctaaaaataagtga
```

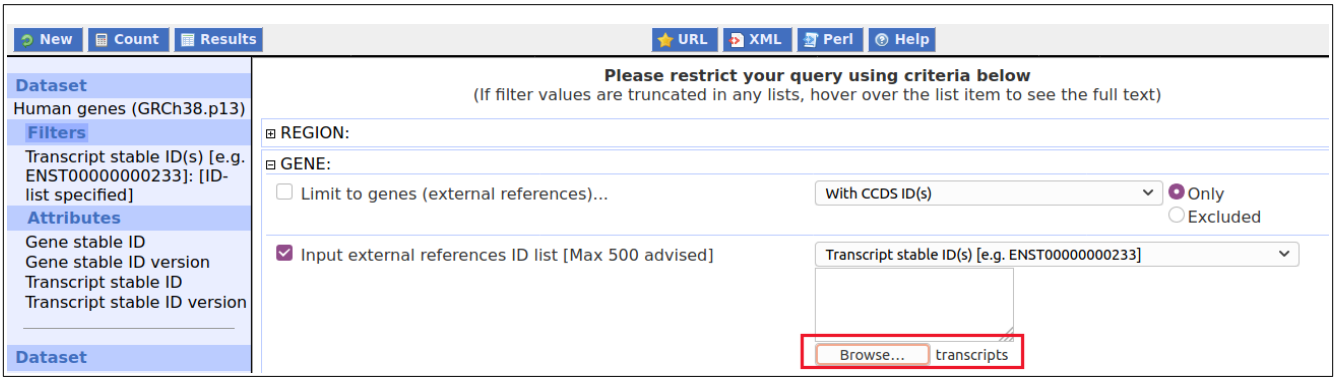
Table of Contents

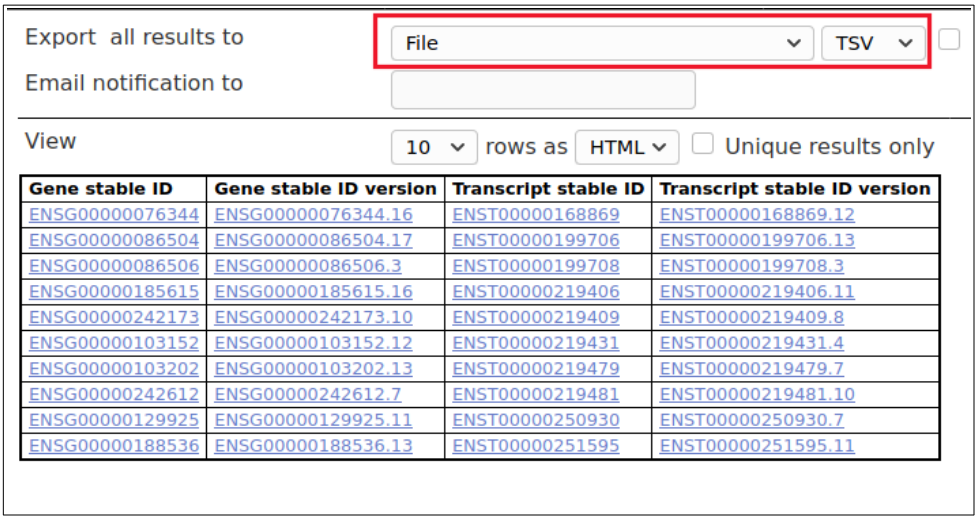
4.4. Week 4 - Ensembl and Biomart

Use BioMart interface on Ensembl website since "biomaRt" R package was working on updates.

This analysis was performed in Week 5 as when querying UCSC Table Browser when for the "Genes and Gene Predictions" group. Most tracks and tables, particularly GENCODE tracks, have a name column for "Name of gene" but list transcript IDs. The question asked how many Ensembl genes are listed in a particular region and the return from UCSC listed all the transcripts and so, after discussing with professors, to obtain the Ensembl genes we input the transcript IDs to BioMart interface and then we can export to R to count unique gene IDs in the gene ID column.

field	example	SQL type	info	description
name	ENST00000619216.1	varchar(255)	values	Name of gene
chrom	chr1	varchar(255)	values	Reference sequence chromosome or scaffold
strand	-	char(1)	values	+ or - for strand
txStart	17368	int(10) unsigned	range	Transcription start position (or end position for minus strand item)
txEnd	17436	int(10) unsigned	range	Transcription end position (or start position for minus strand item)
cdsStart	17368	int(10) unsigned	range	Coding region start (or end position if for minus strand item)
cdsEnd	17368	int(10) unsigned	range	Coding region end (or start position if for minus strand item)
exonCount	1	int(10) unsigned	range	Number of exons
exonStarts	17368,	longblob		Exon start positions (or end positions for minus strand item)
exonEnds	17436,	longblob		Exon end positions (or start positions for minus strand item)
proteinID		varchar(40)	values	UniProt display ID, UniProt accession, or RefSeq protein ID
alignID	uc031tla.1	varchar(255)	values	Unique identifier (GENCODE transcript ID for GENCODE Basic)





Gene stable ID	Gene stable ID version	Transcript stable ID	Transcript stable ID version
ENSG00000076344	ENSG00000076344.16	ENST00000168869	ENST00000168869.12
ENSG00000086504	ENSG00000086504.17	ENST00000199706	ENST00000199706.13
ENSG00000086506	ENSG00000086506.3	ENST00000199708	ENST00000199708.3
ENSG00000185615	ENSG00000185615.16	ENST00000219406	ENST00000219406.11
ENSG00000242173	ENSG00000242173.10	ENST00000219409	ENST00000219409.8
ENSG00000103152	ENSG00000103152.12	ENST00000219431	ENST00000219431.4
ENSG00000103202	ENSG00000103202.13	ENST00000219479	ENST00000219479.7
ENSG00000242612	ENSG00000242612.7	ENST00000219481	ENST00000219481.10
ENSG00000129925	ENSG00000129925.11	ENST00000250930	ENST00000250930.7
ENSG00000188536	ENSG00000188536.13	ENST00000251595	ENST00000251595.11

Table of Contents

```
# write transcript ids to table without version id
write.table(gsub("\\..*", "", bed$V4), quote = F, file="transcripts", col.names=F, row.names=F)

# read result from Ensembl BioMart
biomart <- read.table("/home/coyote/Downloads/mart_export(3).txt", sep="\t", header=T)
length(unique(biomart$Gene.stable.ID))
```

Use 'biomaRt' and 'GenomicRanges' bioconductor packages to obtain genes overlapping region

Part 3 - biomaRt

1. What Ensembl genes are in a 100,000 base pair region of chromosome 20 from 5.0 to 5.3 Mb? What chromosome band are they on, what strand, and what type of genes are they?

Hint: Attributes are: hgnc_symbol, band, strand, gene_biotype

2. Are there OMIM genes in this same region? Use "hgnc_symbol","band","strand","gene_biotype","mim_morbid_accession","mim_morbid_description" for the Attributes.

I used 'biomaRt' package in R to send request to database to answer this question. The R script, titled part3_2.R is available on github. We discussed this problem on Slack and Discussion board as a class and there was some good conversation. We did come to agreement on interpreting the question to only include ranges that had with less than 100 kb which I performed with `BiocGenerics::width()` function and as long as any base of the region overlapped the 300 kb range we would include it. For this I also used the `GenomicRanges` Bioconductor package in R and use of the `findOverlaps()` function.

Use 'biomaRt' and 'GenomicRanges' Bioconductor packages to obtain mouse genes with RefSeq peptides and write GRanges to BED file using 'rtracklayer' package.

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:

Dataset:

Ensembl Genes 93 (or the latest version)

Mouse genes (GRCm38.p6) (or the latest version)

Filters:

Chromosome 11

Band E2 only

Transcript count >=7

Table of Contents

Limit to genes with RefSeq protein (peptide) IDs only

Attributes:

Default attributes

Add “RefSeq Protein (peptide) ID”

I performed the search with ‘biomaRt’ in R, put information into GRanges object, sorted by start position and wrote to BED6 file using ‘rtracklayer’. Script is named week4_problem4_updated.R and is available on github along with the bed file week4_problem4.bed with RefSeq accession ID for the peptide entry.

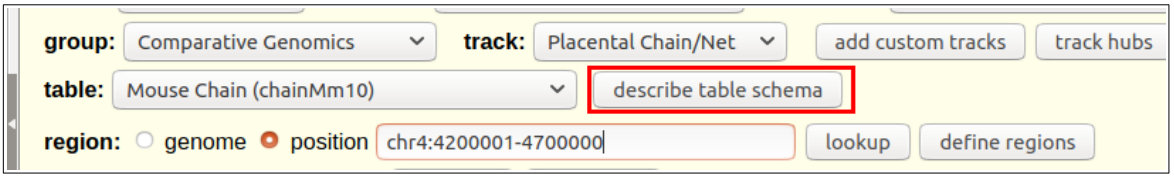
Table of Contents

4.5. Week 5 - Genome browsers and analysis platforms

Create and document a workflow for students to follow for comparative genomics analysis using Galaxy, UCSC Genome Browser, and IGV. Posted to discussion forums for students to follow.

I decided to apply all that I learned in reading and in my own experience with all three tools, UCSC, Galaxy, & IGV that includes some extra nice stuff.

An interesting function in the UCSC Table browser is the “describe table” schema. For part 4 in peer collab clicking this for the Mouse Chain table shows the fields of the data in the table and gives a description. The name of the Primary Table: chainMm10 is what shows in Galaxy.



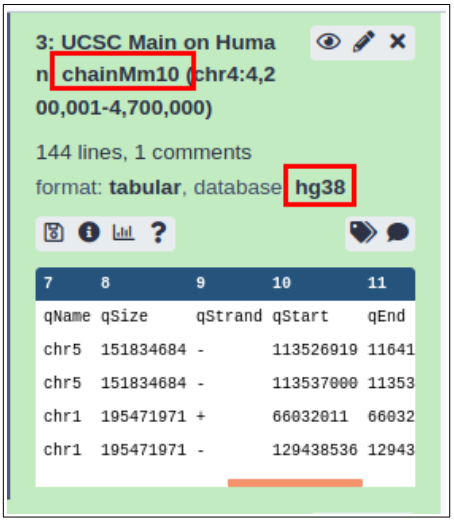
Database: hg38 **Primary Table:** chainMm10 **Row Count:** 3,678,476 **Data last updated:** 2015-04-10
Format description: Summary info about a chain of alignments

field	example	SQL type	description
bin	585 ???	smallint(5) unsigned	Indexing field to speed chromosome range queries.
score	170460	double	score of chain
tName	chr1	varchar(255)	Target sequence name
tSize	248956422	int(10) unsigned	Target sequence size
tStart	11677	int(10) unsigned	Alignment start position in target
tEnd	37616	int(10) unsigned	Alignment end position in target
qName	chr6	varchar(255)	Query sequence name
qSize	149736546	int(10) unsigned	Query sequence size
qStrand	-	char(1)	Query strand
qStart	28206106	int(10) unsigned	Alignment start position in query
qEnd	28238509	int(10) unsigned	Alignment end position in query
id	2478	int(10) unsigned	chain id

You may be asking yourself what is this “bin” field? It doesn’t seem important! Oh but wait...it is EXTREMELY important. The developers of the UCSC Table Browser, including a name you will see on about every single UCSC paper, James Kent, came up with a clever, “mystic” as they describe it, way to index the HUGE amount of data that can be queried and retrieved from extremely large tables and created a binning algorithm for faster data retrieval. You don’t need to worry about it from the user interface but if you are accessing information programatically, such as overnight batch retrieval, according to this “famous” Biostars post, Heng Li (the creator along with R. Durbin of the bwa aligner program!) strongly urges learning how to implement "binning" coordinate intervals for 1) faster retrieval and 2) to not bog down the server. If you are interested in the algorithm you can find it here based on the advice of R. Durbin as well as Python implementation here.

Table of Contents

The database used also shows up if you click the history title in Galaxy. Notice the wrong format of “tabular”.



If you are trying to perform a function or use a tool on a data but that number is “unavailable” this means that you may have used an incorrect format. For instance for the Mouse Chain, I incorrectly did not set to “Bed Format” and so you would see:

Correcting this shows:

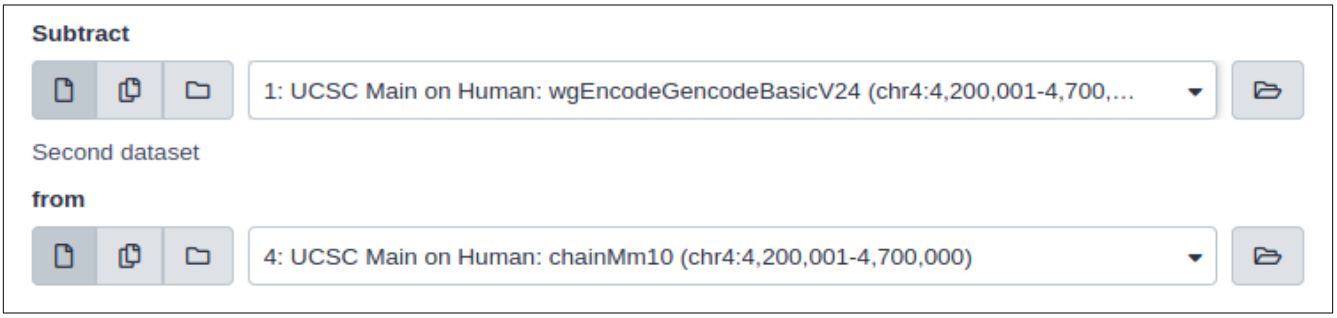


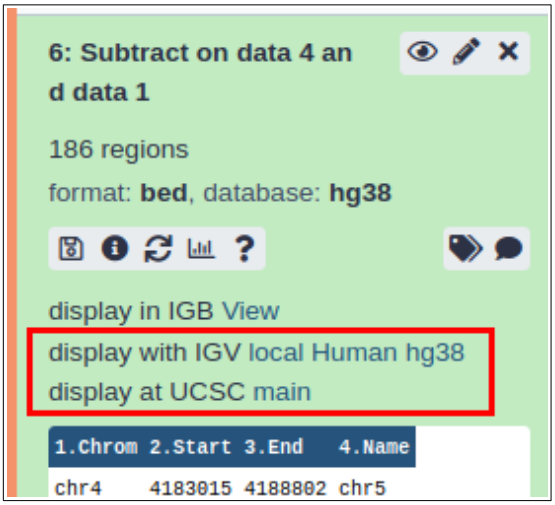
Table of Contents

You can see the Galaxy tool parameters by clicking the circled info(i) button if you forgot which parameters you chose to run the tool.

Tool Parameters

Input Parameter	Value
Subtract	• 1: UCSC Main on Human: wgEncodeGencodeBasicV24 (chr4:4,200,001-4,700,000)
from	• 4: UCSC Main on Human: chainMm10 (chr4:4,200,001-4,700,000)
Return	Intervals with no overlap
where minimal overlap is	1

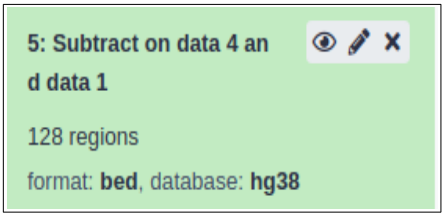
You can actually visualize results directly from Galaxy in IGV and UCSC.



I think this linking from Galaxy to IGV only works if you have IGV installed in Windows as the link asks you to open up an application and IGV is only installed as an “application” in windows. On my Linux machine I have to start IGV with their shell script that comes with the download. For me I can only save the XML file and source it into IGV.

For the “Non-overlapping pieces of intervals” return, if there is an overlapping interval then the pieces of the interval that remains after subtracting will still be returned. For the “Intervals with no overlap” return, ff there is an overlapping interval then no interval will be returned after subtracting, i.e. you will have less regions returned from “Intervals with no overlap” than “Non-overlapping pieces of intervals”.

No overlap vs. Overlapping pieces



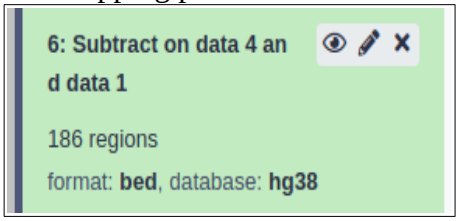
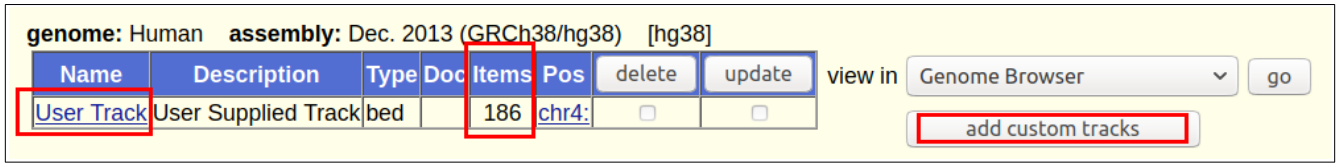
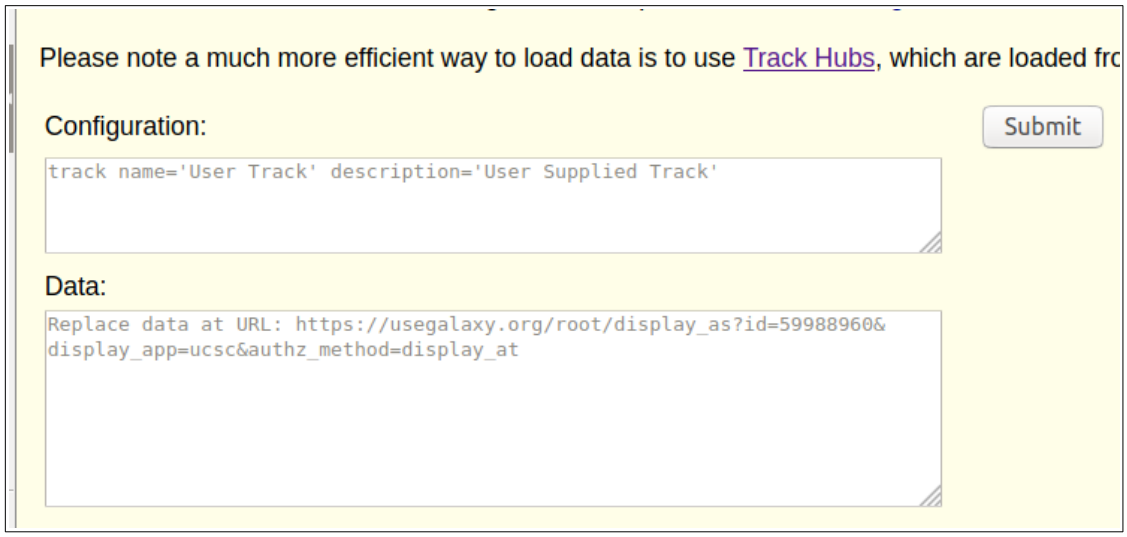


Table of Contents

If you directly click the link from Galaxy to “display at UCSC main” you can see the User Track at the top by default and clicking “manage custom tracks” shows information about the track.



Clicking the name “User Track” link you would think you can change the name but no it’s greyed out. However it does show how to label your track like we did a few weeks back.



Adding a track is not that easy I found as uploading and writing in the configuration doesn’t work. Here are the steps for adding custom tracks.

1) Format file type as bed, wig, gff3, etc.

2) Add browser line(s), e.g.

```
browser position chr4:4200001-4700000
```

3) Add track name and description, e.g.

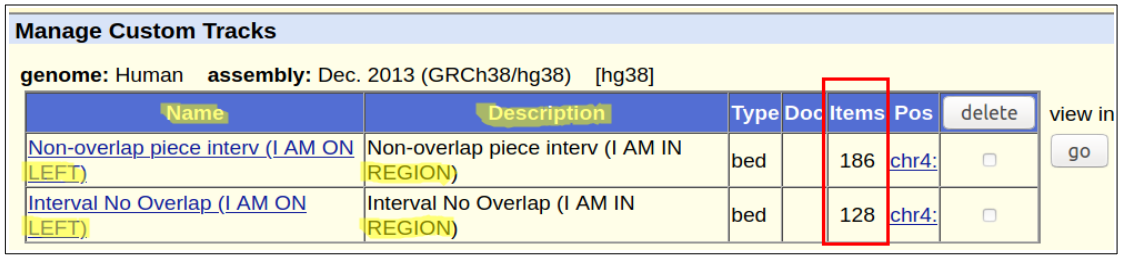
```
track name='Interval No Overlap (I AM ON LEFT)' description='Interval No Overlap (I AM IN REGION)'
```

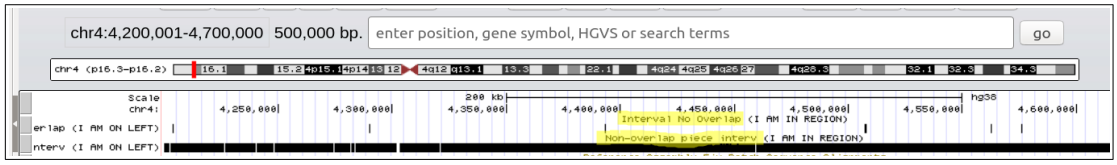
Note it is the Description that shows in the region of the browser and the Name is the track name to left.

Warning be aware of your quotes. It works like code.

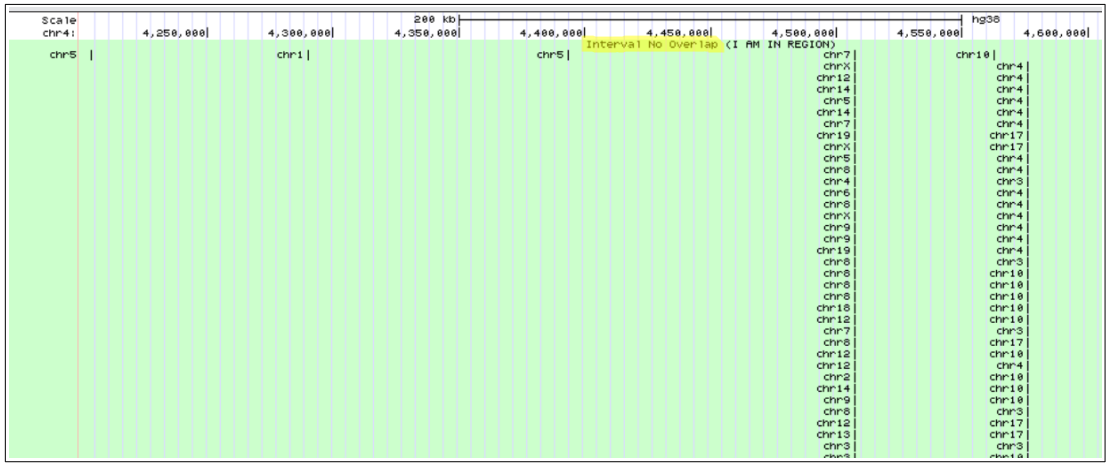
4) Don’t worry about step 4. Go back to “add custom tracks” upload and Submit

Table of Contents

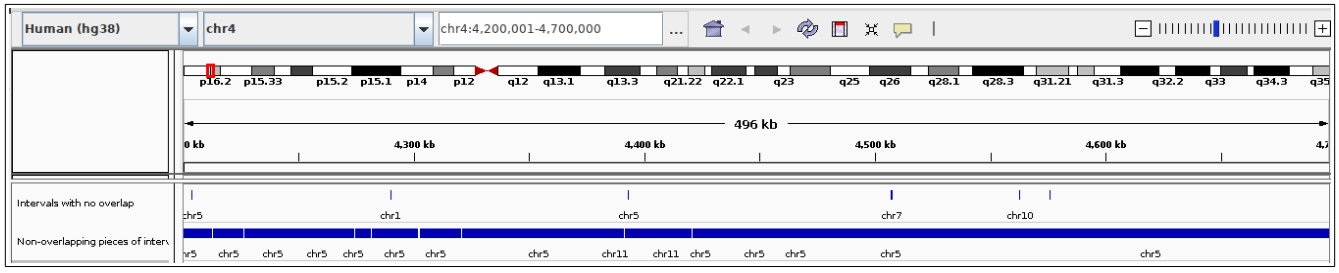




As you can see, allowing overlapping pieces (bottom track) instead of getting rid of any pieces that overlap even just by 1 base (top track) removes a lot of region in width but not necessarily the # of regions from the database that conserved with this region in Mouse.



It's much easier to upload to IGV. Below you will see in IGV with "Collapsed View" as well as "Expanded View". It is also much easier to rename the track just by right-clicking.



It's these really long pieces in red that get removed when you select the return type "Intervals with no overlap".

Table of Contents

Also another interesting fact about IGV that Isabel elluded to with respect to Broad Institute developing the program for research related to The Cancer Genome Atlas (TCGA). One of the cool labs that is part of the Bioinformatics and Integrative Genomics (BIG) at Harvard University where Heng Li is associated with, the Stegmaier Lab works with Broad and one of her lab members works for the Cancer Data Science team at the Broad. I hope to write about my interest in this Data Science team in my BIG application.

Resources:
1. Kent WJ, Sugnet CW, Furey TS, Roskin KM, Pringle TH, Zahler AM, Haussler D. The human genome browser at UCSC. Genome Res. 2002 Jun;12(6):996-1006. doi: 10.1101/gr.229102. PMID: 12045153; PMCID: PMC186604.

Reviewed and posted issue with the "Histogram of a numeric column" tool in Galaxy.

The breaks functionality that backends with the R hist() function incorrectly calculates the start break of range for bins. If we set the total range to be the number of breaks, i.e a bin width of one, then bins 1 and 2 were being combined. I open a github ticket under galaxyproject /usegalaxy-playbook #309. As the ggplot() tool in Galaxy is not very customizable, I exported to R and plotted with ggplot2. The script is also available on github at file galaxy_ggplot.R.

Table of Contents

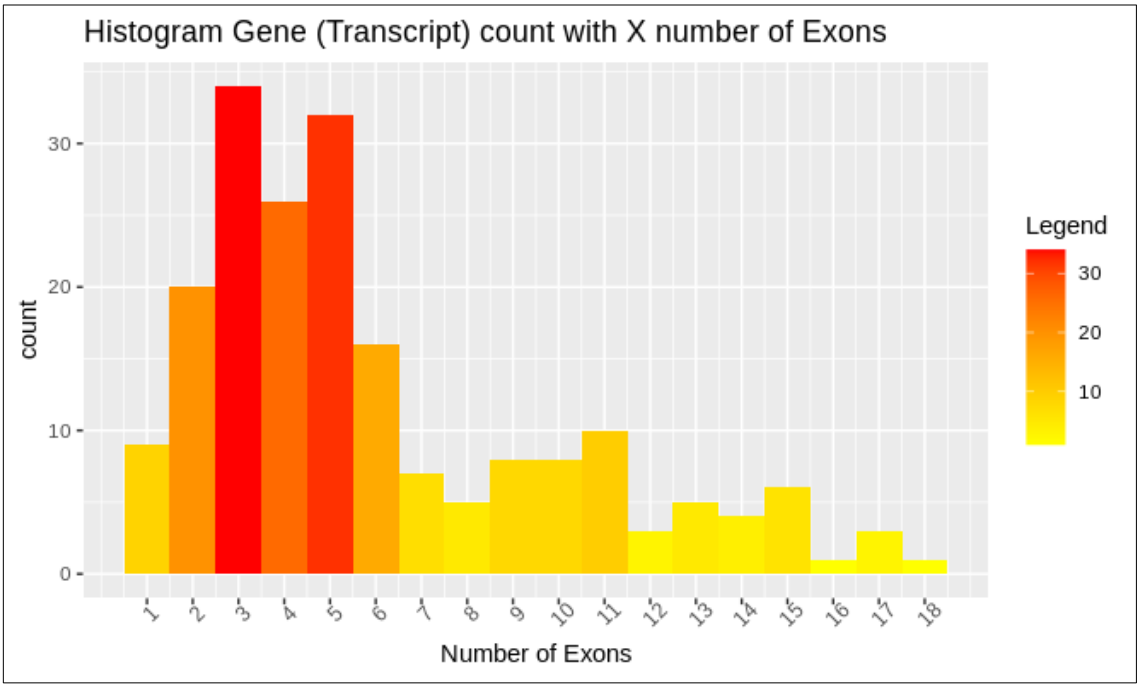


Table of Contents

4.6. Week 6 - Single nucleotide and copy number variations

Compare information from Firehose IGV server dataset to the [Broad GDAC Firehose] website to learn more annotated information.

Noticing that there were different types of prefixes for Firehose data from servers datasets in IGV, I looked up the meaning of TP and TM on the Firehose website finding they stood for Tumor Primary and Tumor Metastatic respectively. I also looked up the samples for Copy Number datasets from Firehose from 2016 and found for the samples for SNP6 CopyNum, e.g. genome_wide_snp_6-segmented_scna_minus_germline_cnv_hg19 samples, where the SCNA stands for somatic copy-number alterations, was ran with the Affymetrix Genome-Wide Human SNP Array 6.0. Looking up information on this chip from Thermo Fisher website I found that of the 1.8 million genetic markers on this chip, 946,000 of the probes detect for CNV.

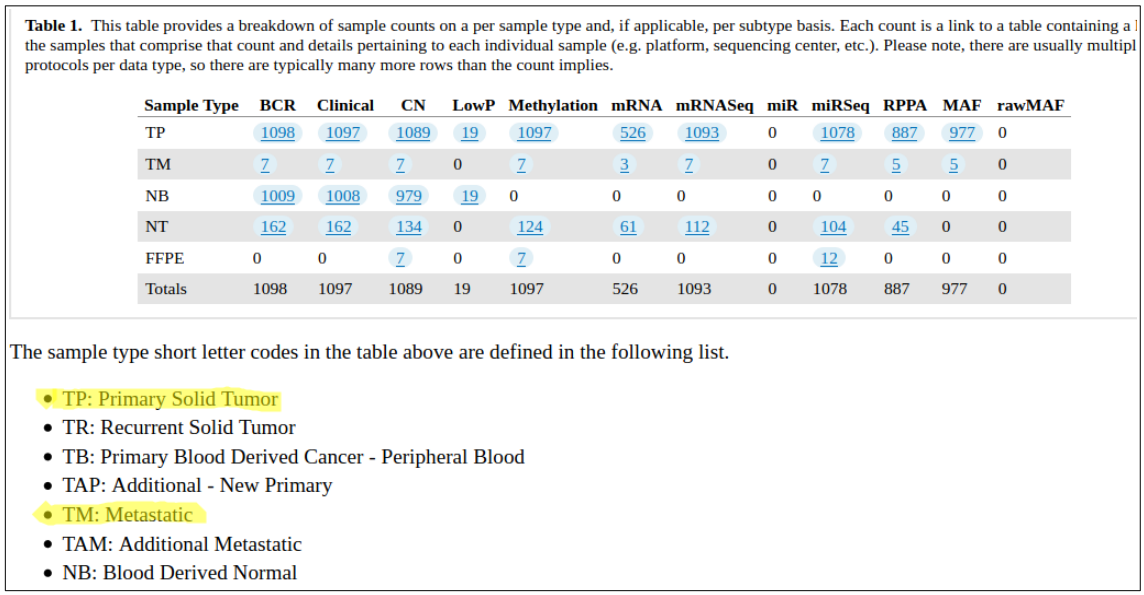
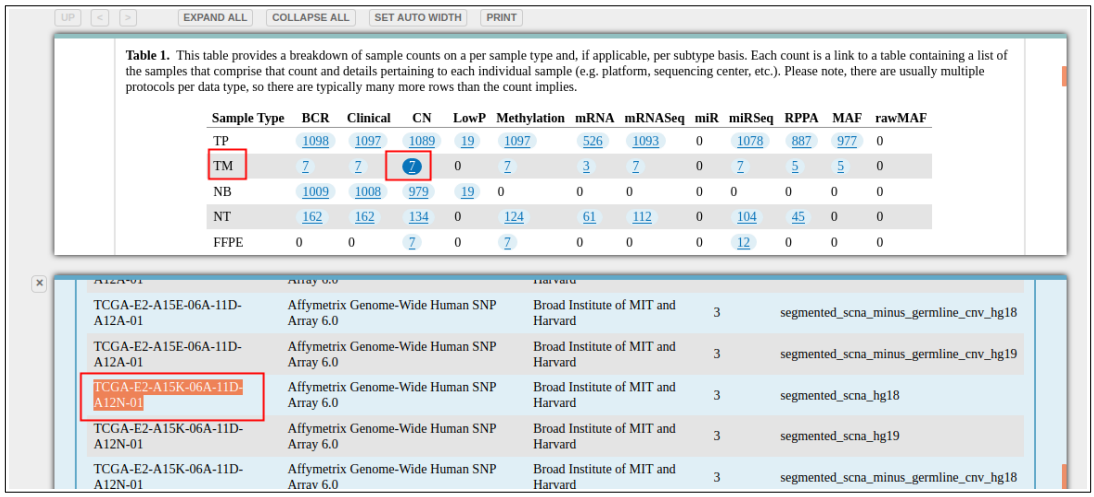


Table 1. This table provides a breakdown of sample counts on a per sample type and, if applicable, per subtype basis. Each count is a link to a table containing a l the samples that comprise that count and details pertaining to each individual sample (e.g. platform, sequencing center, etc.). Please note, there are usually multipl protocols per data type, so there are typically many more rows than the count implies.

Sample Type	BCR	Clinical	CN	LowP	Methylation	mRNA	mRNASeq	miR	miRSeq	RPPA	MAF	rawMAF
TP	1098	1097	1089	19	1097	526	1093	0	1078	887	977	0
TM	7	7	7	0	7	3	7	0	7	5	5	0
NB	1009	1008	979	19	0	0	0	0	0	0	0	0
NT	162	162	134	0	124	61	112	0	104	45	0	0
FFPE	0	0	7	0	7	0	0	0	12	0	0	0
Totals	1098	1097	1089	19	1097	526	1093	0	1078	887	977	0

The sample type short letter codes in the table above are defined in the following list.

- TP: Primary Solid Tumor
- TR: Recurrent Solid Tumor
- TB: Primary Blood Derived Cancer - Peripheral Blood
- TAP: Additional - New Primary
- TM: Metastatic
- TAM: Additional Metastatic
- NB: Blood Derived Normal



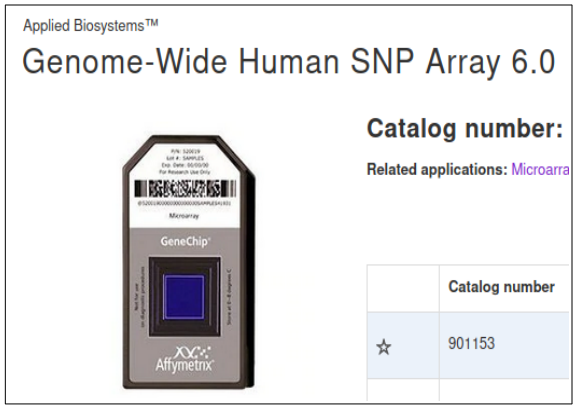
UP < > EXPAND ALL COLLAPSE ALL SET AUTO WIDTH PRINT

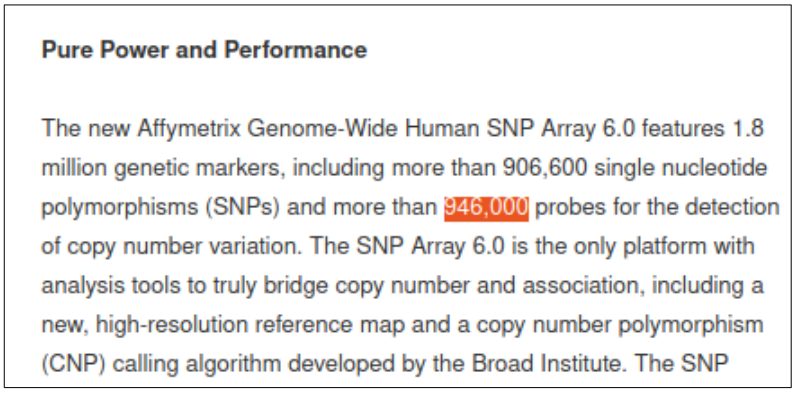
Table 1. This table provides a breakdown of sample counts on a per sample type and, if applicable, per subtype basis. Each count is a link to a table containing a list of the samples that comprise that count and details pertaining to each individual sample (e.g. platform, sequencing center, etc.). Please note, there are usually multiple protocols per data type, so there are typically many more rows than the count implies.

Sample Type	BCR	Clinical	CN	LowP	Methylation	mRNA	mRNASeq	miR	miRSeq	RPPA	MAF	rawMAF
TP	1098	1097	1089	19	1097	526	1093	0	1078	887	977	0
TM	7	7	7	0	7	3	7	0	7	5	5	0
NB	1009	1008	979	19	0	0	0	0	0	0	0	0
NT	162	162	134	0	124	61	112	0	104	45	0	0
FFPE	0	0	7	0	7	0	0	0	12	0	0	0

TCGA-E2-A15E-06A-11D-A12A-01	Affymetrix Genome-Wide Human SNP Array 6.0	Broad Institute of MIT and Harvard	3	segmented_scna_minus_germline_cnv_hg18
TCGA-E2-A15E-06A-11D-A12A-01	Affymetrix Genome-Wide Human SNP Array 6.0	Broad Institute of MIT and Harvard	3	segmented_scna_minus_germline_cnv_hg19
TCGA-E2-A15K-06A-11D-A12N-01	Affymetrix Genome-Wide Human SNP Array 6.0	Broad Institute of MIT and Harvard	3	segmented_scna_hg18
TCGA-E2-A15K-06A-11D-A12N-01	Affymetrix Genome-Wide Human SNP Array 6.0	Broad Institute of MIT and Harvard	3	segmented_scna_hg19
TCGA-E2-A15K-06A-11D-A12N-01	Affymetrix Genome-Wide Human SNP Array 6.0	Broad Institute of MIT and Harvard	3	segmented_scna_minus_germline_cnv_hg18

Table of Contents





This also shows that the software used by Thermo Fisher included with the chip is developed by Broad.

Use IGV to visualize whole-exome sequencing GBR (British in England and Scotland) sample data with the data from NCBI dbSNP 147 database.

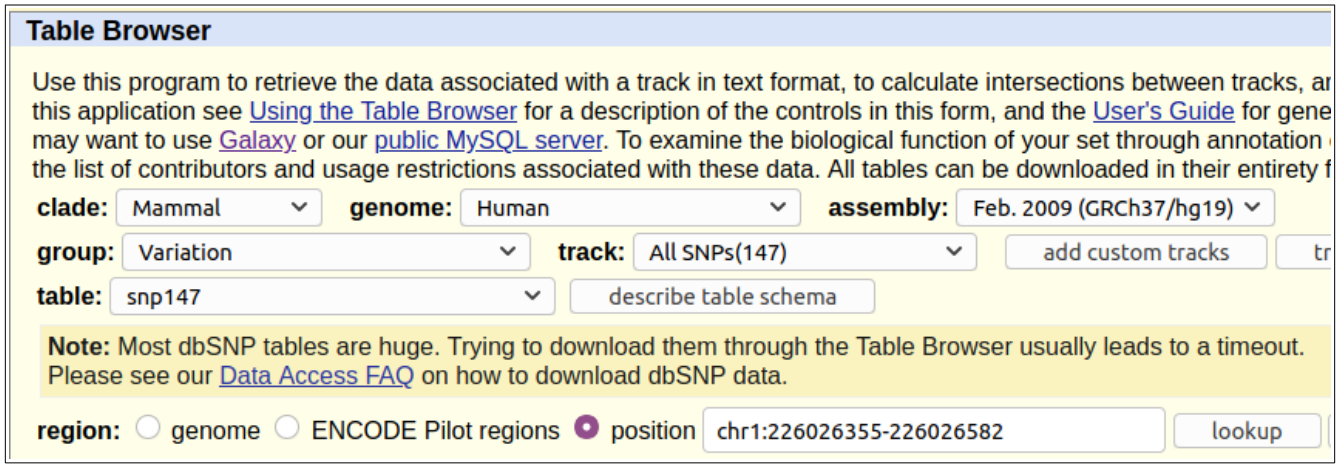
Using IGV for hg19, load dbSNP 1.4.7 or newer (i.e. Available Datasets > Annotations > Variation and Repeats > dbSNP 1.4.7) and an exome sequencing track from the 1000 Genomes project (1000 Genomes > Alignments > GBR > exome > HG00096 exome). Go to the EPHX1 gene and zoom in on the exon #4.

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

b. (0.25 pts) 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)

The sample HG00096 is a Bam (Binary alignment map) file with the results of runs from whole-exome sequencing (WES) on human B-lympocytes from a donor from the United Kingdom as part of the 1000 Genomes project. I found that there were too many SNPs to count, even on 1 exon of EPHX1, and so I used the UCSC Table Browser, track All SNPs(147) primary table snp147, with the coordinates of exon 4 of EPHX1 gene, obtained from NCBI Genome Data Viewer and confirmed with the GenBank record (see 7J section under “Be a Teacher”) to obtain an output of all the SNPs listed on this track in IGV. Exporting from the table browser was then able to read into R to print out nicely in the script galaxy_ggplot.R.

Table of Contents




```
45  ## read in export from All SNPs(147) - Simple Nucleotide Polymorphisms (dbSNP 147)
46  test <- read.table("/home/coyote/Downloads/part3_out", header=T,
47                     sep="\t", comment.char = "")
48  paste(test$name, collapse = ", ")
```

As a class through discussion on Slack we discovered there were more than just one type of variation from the dbSNP database and that in addition to SNPs, there were other classes of variation list in red box below.

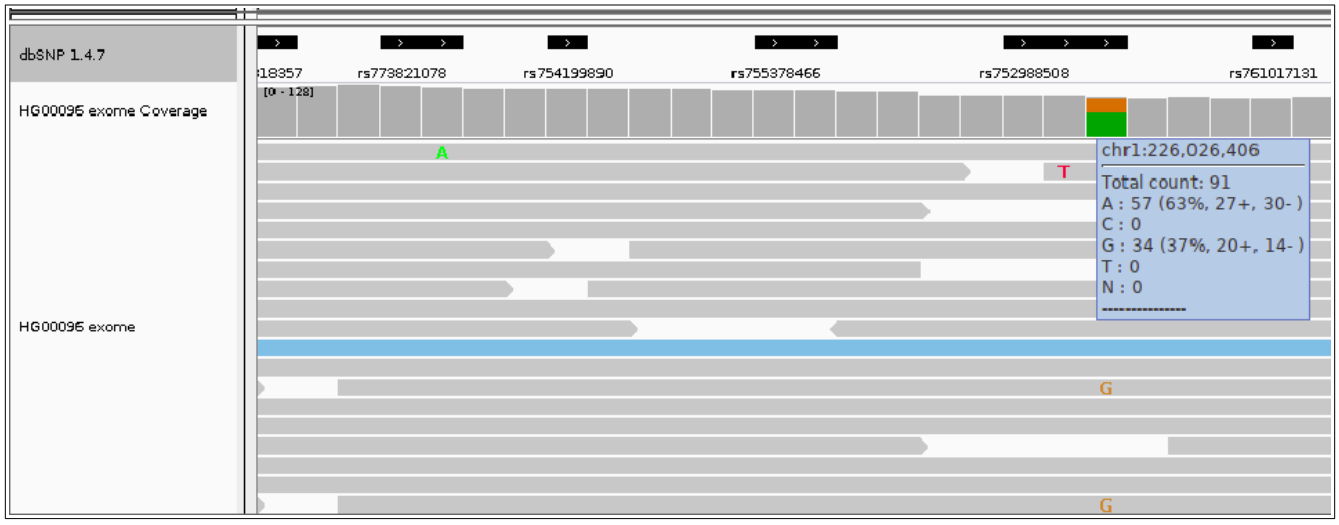
Schema for All SNPs(147) - Simple Nucleotide Polymorphisms (dbSNP 147)

Database: hg19 **Primary Table:** snp147 **Row Count:** 154,822,082 **Data last updated:** 2016-07-29
Format description: Polymorphism data from dbSNP

field	example	SQL type
bin	585	smallint(5) unsigned
chrom	chr1	varchar(31)
chromStart	10019	int(10) unsigned
chromEnd	10020	int(10) unsigned
name	rs775809821	varchar(15)
score	0	smallint(5) unsigned
strand	+	enum('+', '-')
refNCBI	A	blob
refUCSC	A	blob
observed	-/A	varchar(255)
molType	genomic	enum('unknown', 'genomic', 'cDNA')
class	deletion	enum('single', 'in-del', 'microsatellite', 'named', 'mnp', 'insertion', 'deletion')

Next using the summary track for Bam file for sample HG00096, found which specific SNP the sample was heterozygous for. Another student and I found from help page for AlignmentData for IGV at Broad that “if a nucleotide differs from the reference sequence in greater than 20% of quality weighted reads, IGV colors the bar in proportion to the read count of each base (A, C, G, T).” (Broad Institute)

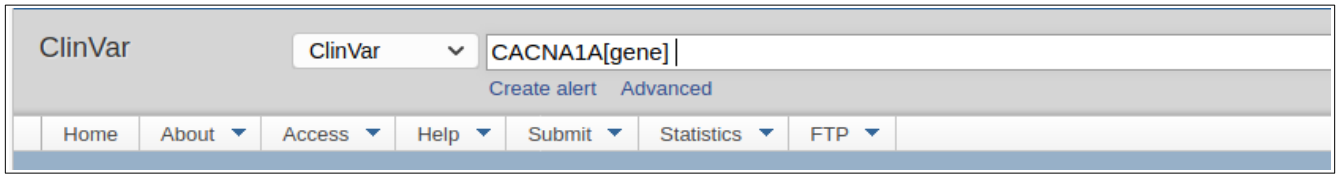
Table of Contents

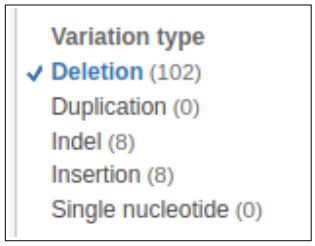


Use NCBI ClinVar and Ensembl to look up information on structural variants of a protein coding gene.

As a follow to my paper on the A712T mutation of the high-voltage Cav2.1 Voltage-dependent P/Q-type calcium channel subunit alpha-1A (Cavα1) in Protein Bioinformatics regarding to the structural implications of the variant on epileptic pathologies, the goal was to complete a meta-analysis study to quantify the amount of variants for this large (>2500 residue) protein.

First use ClinVar to determine the amount of deletion type variations and of those how many were classified as being pathogenic.





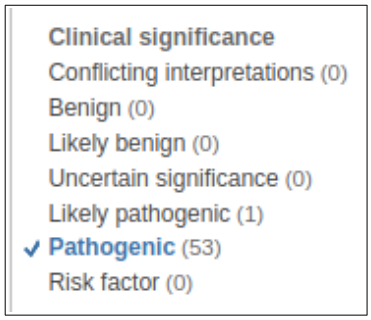
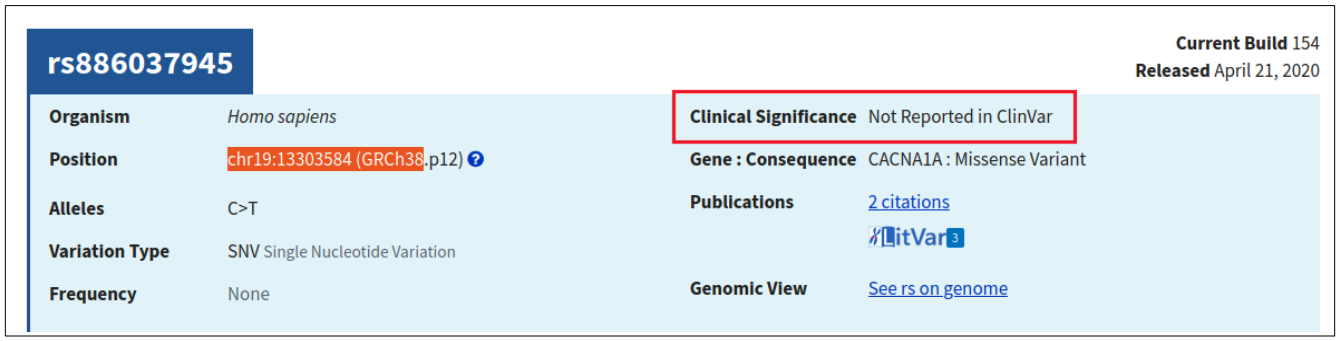


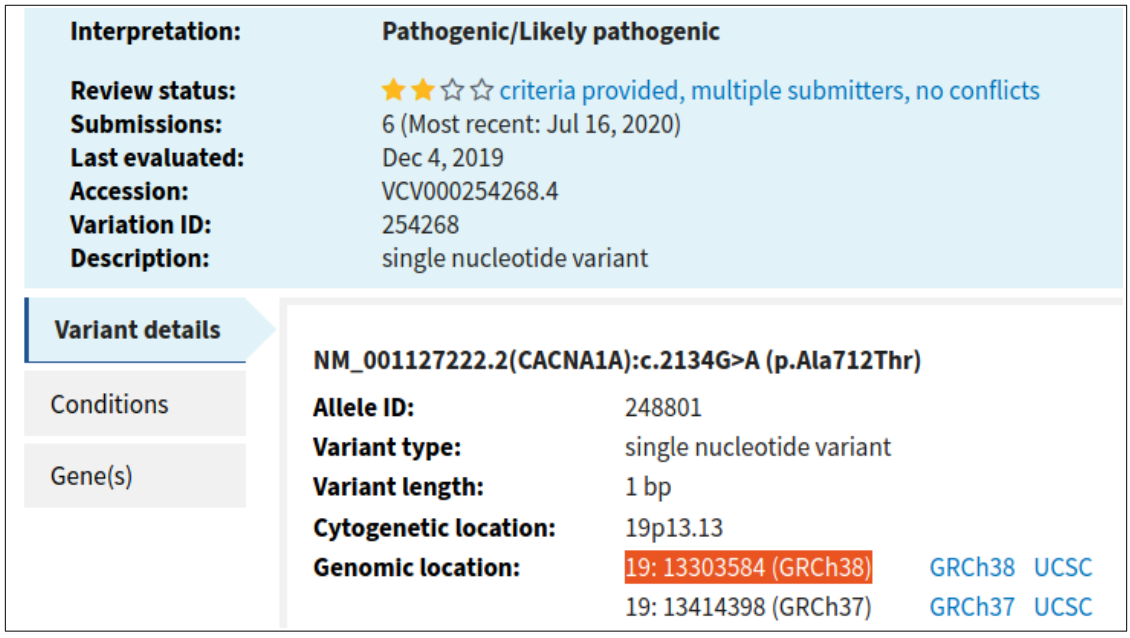
Table of Contents

As a follow to my original paper, the p.A712T variant was also listed in ClinVar and in dbSNP. These NCBI databases usually will have to links to each other, if in ClinVar, the dbSNP rs id entry will indicate this next to “Clinical Significance”, and if in dbSNP, the ClinVar entry will have the dbSNP link to the rs id next to “Links”. However neither of these entries have the valid link to eachother despite the variant being in both databases.

dbSNP:



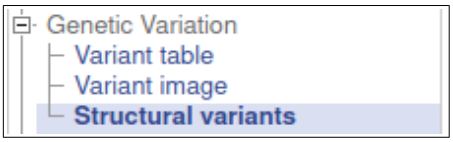
ClinVar:



Finally, use Ensembl to count the structural variants as well as how many of the structural variants are listed in the dbVar at NCBI. We can then export and/or read into R to filter for columns based on

Table of Contents

database study, e.g. dbVar, Database of Genomic Variants Archive (DGV), etc. and the variant class, e.g. CNV, insertion, etc.



Structural variants

Show 25 entries | Show/hide columns | Filter

Name	Chr:bp	Genomic size (bp)	Class	Source Study	Study description
nsv995068	19:10319474-13777860	3,458,387	CNV	DGVa:nstd37	Database of Genomic Variants Archive: Miller 2010 "Consensus statement: chromosomal microarray is a first-tier clinical diagnostic test for individuals with developmental disabilities or congenital anomalies."
nsv992659	19:13381400-13381401	-	mobile element insertion	DGVa:nstd94	Database of Genomic Variants Archive: Helman 2014 "Somatic retrotransposition in human cancer revealed by whole-genome and exome sequencing." PMID:24823667
nsv984831	19:90910-58586487	-	CNV	DGVa:nstd11	Database of Genomic Variants Archive: Walter 2009 "Acquired copy number alterations in adult acute myeloid leukemia genomes." PMID:19651600
nsv911096	19:13319510-13354187	34,678	CNV	DGVa:nstd71	Database of Genomic Variants Archive: Xu 2011 "SgD-CNV, a database for common and rare copy number variants in three Asian populations." PMID:21882294
nsv911095	19:13291258-13359502	68,245	CNV	DGVa:nstd71	Database of Genomic Variants Archive: Xu 2011 "SgD-CNV, a database for common and rare copy number variants in three Asian populations." PMID:21882294

Table of Contents

4.7. Week 7 - Genomic data file formats and manipulation tools

Use bedtools sort, subtract, intersect, and closest tools on genomic annotated bed files.

First I installed the most recent version of bedtools from github:

1. Download bedtools-2.29.2.tar.gz from github

2. Run commands to compile

```
tar -xvzf  bedtools-2.29.2.tar.gz

cd bedtools2

make
```

3. Move bedtools2 folder and add bedtools2/bin to PATH

A)

Returning to the exercise in Week 5 for analyzing intervals of conserved regions in mouse and human on 0.5 mb interval of chromosome 4, we can perform similar operations using bedtools that we performed in galaxy.

Using the subtract tools in bedtools we can subtract overlapping intervals of human exons from the conserved regions with the command:

```
bedtools subtract -a mouse.bed -b human.bed > conserved_not_in_human_exons.bed
```

And this gives us the same 186 regions in Galaxy using the parameter “Non-overlapping pieces of intervals”. If we wanted the opposite, i.e. only intervals in human exons that overlap conserved regions, we can use the intersect tool with the command:

```
bedtools intersect -a mouse.bed -b human.bed > conserved_only_in_human_exons.bed
```

If we wanted to include all parts of the interval of conserved mouse regions with this intersection, i.e the (-a) parameter mouse.bed, then we can pass the -wa parameter. This will NOT include regions of -a that have no overlap at all with -b.

```
bedtools intersect -a mouse.bed -b human.bed -wa
```

In order to mirror the subtract functionality of Galaxy with parameter “Intervals with no overlap” we use bedtools subtract and the parameter “-A Remove features with any overlap” with the command:

```
bedtools subtract -a mouse.bed -b human.bed -A > conserved_not_in_human_exons_no_overlap.bed
```

Alternatively we can obtain this same function with bedtools intersect passing the “-v Reporting the absence of any overlapping features” parameter with the command:

```
bedtools intersect -a mouse.bed -b human.bed -v > conserved_not_in_human_exons_no_overlap.bed
```

This last two calls returns the same 128 lines from my Galaxy tutorial in Week 5.

Table of Contents

B)

In another example we can use bedtools intersect to look at regions of H3K4me3 methylation and intersection with RefSeq genes.

```
bedtools intersect -a hs_chr20_H3K4me3_for_linux.bed -b hs_chr20_refseq.bed > h3k4me3_intersects_RefSeq.bed
```

As well as passing the '-v' parameter for A does not intersect with B and get regions of H3K3me3 that do not intersect exons.

C) A third example we use bedtools sort and bedtools closest to find regions of one annotation that are closest to regions in another. To summarize the closest tool we just run:

```
bedtools closest

Tool:    bedtools closest (aka closestBed)

Version: v2.29.2

Summary: For each feature in A, finds the closest

        feature (upstream or downstream) in B.
```

To find the closest exons to genome assembly gaps, we set the gap annotation file to be A and B to be the exons file. If the files are not sorted we should sort them first with bedtools sort:

```
bedtools sort -i hs_chr20_refseq.bed > hs_chr20_refseq_sorted.bed
```

And then we can run:

```
bedtools closest -a hs_chr20_gaps.bed -b hs_chr20_refseq_sorted.bed > hs_chr20_nearest_exon_gap.bed
```

We can also create a new column to report the distance with the -d parameter which measures the distance from B in the exons file to the gaps in the gaps file. If there are exons for different transcripts but for the same region then we may get more lines than in file A since these exons would be the same distance. If we just want 1 of these exons we can set the ties -t parameter to "first" and it will report the first if there are ties such as:

```
bedtools closest -t first -a hs_chr20_gaps.bed -b hs_chr20_refseq_sorted.bed > hs_chr20_nearest_unique_exon_gap.bed
```

View histone modification BAM alignment files for multiple histone modification marks in IGV

If, because we have many and/or large alignment files, we want to just view the alignment(s) for a certain chromosome of the genome or genomic interval of a particular chromosome we can use samtools package to index and subset our SAM/BAM alignment files. For instance, if we were just

interested in looking at the alignment near the PARD6B gene for the hg19 assembly and 0.5 Mb upstream and 0.5 Mb downstream, as the location for GRCh37 is NC_000020.10 (49347923..49373333) on chromosome 20, we first index the SAM/BAM file(s) with:

```
samtools index HUES64_rep1.H3K27me3.chr20.sorted.bam

samtools index HUES64_rep1.H3K4me3.chr20.sorted.bam
```

First to see the format of the 'SN' Reference sequence name, as we some plant genomes use a capital 'C' such as Chr4, while the more common type use a lowercase 'c' to be UCSC compatible, we check the header of the alignment file with:

```
samtools view -H HUES64_rep1.H3K27me3.chr20.sorted.bam

        @HD     VN:1.0 SO:coordinate

        @SQ     SN:chr20        LN:63025520

        @PG     ID:Bowtie       VN:1.0.0
```

And them subset each SAM/BAM file using samtools view and keeping header -h and in BAM format -b with:

```
samtools view -h -b HUES64_rep1.H3K27me3.chr20.sorted.bam chr20:48847923-49873333 >
PARD6B_HUES64_rep1.H3K27me3.chr20.sorted.bam
```

I have listed more about samtools is listed in 7L section under "Be a Teacher".

We can then view both alignment files within 0.5 MB of the PARD6B gene in IGV but only after we index the new subsetted alignments, as IGV required .bai index files, with:

```
samtools index PARD6B_HUES64_rep1.H3K27me3.chr20.sorted.bam
```

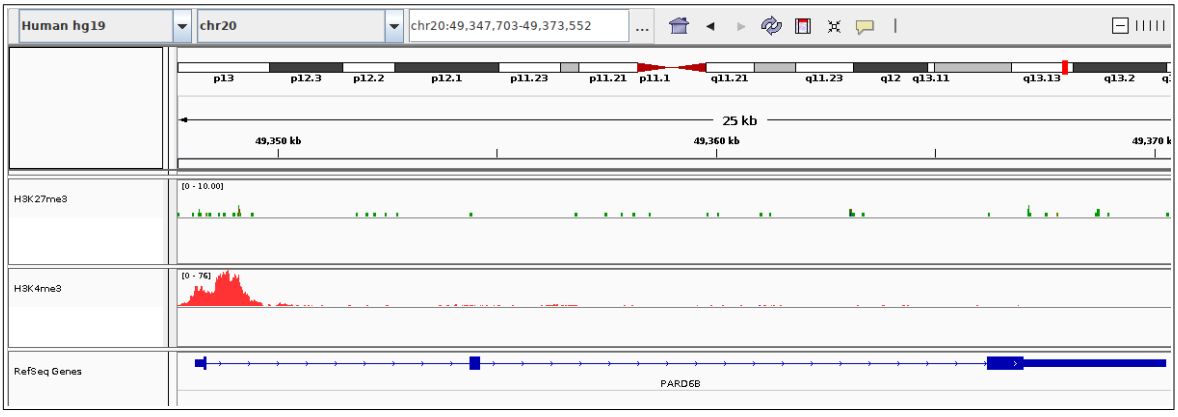
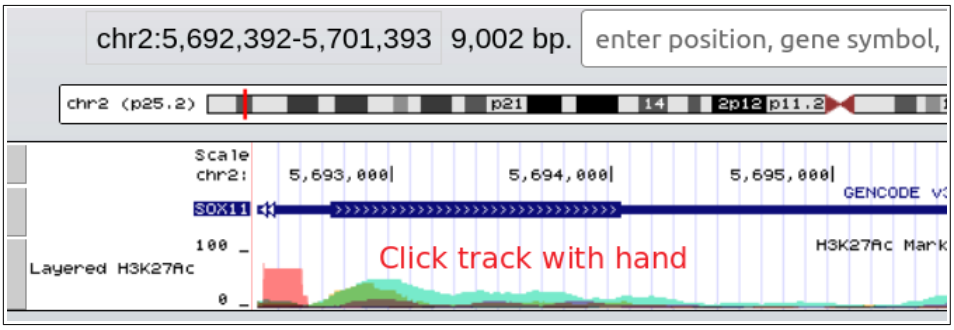


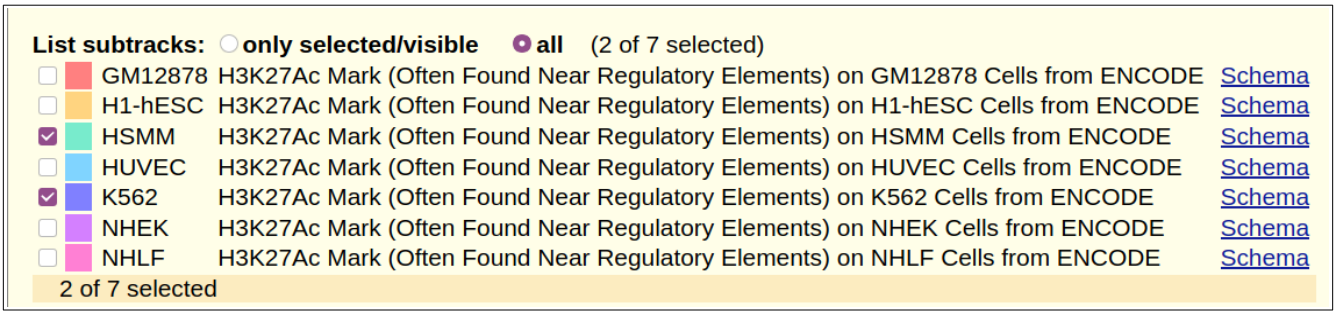
Table of Contents

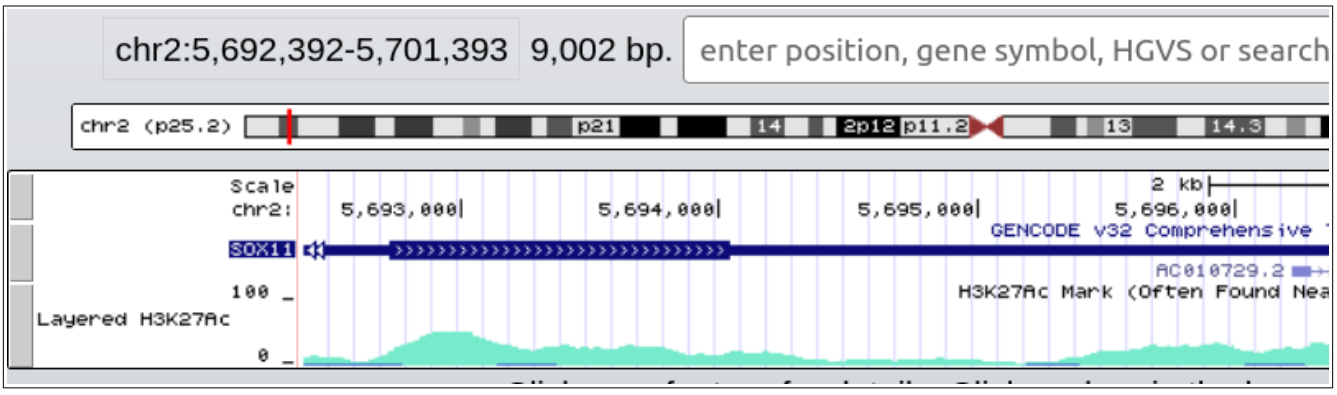
4.8. Week 8 - The ENCODE Project and model organism genomes

View SOX11 gene for the hg38 assembly in UCSC looking at the H3K27ac data for two cells lines HSSM and K562. Then add DNAse hypersensitivity



Select only these two cells lines

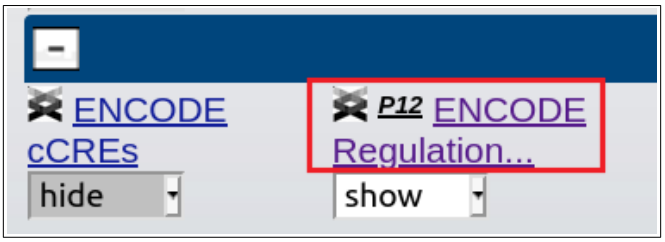


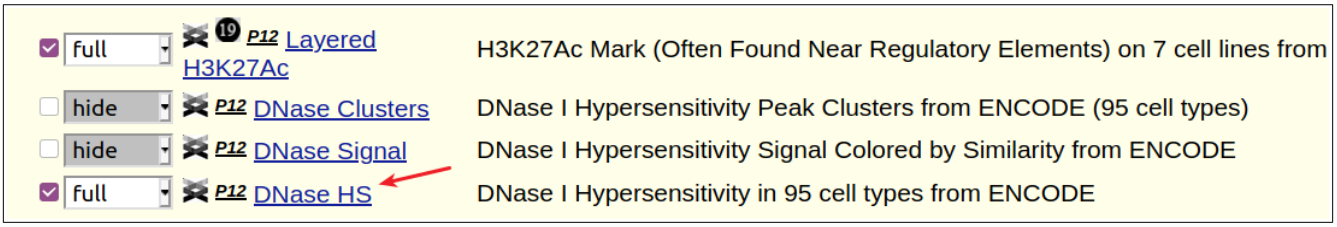


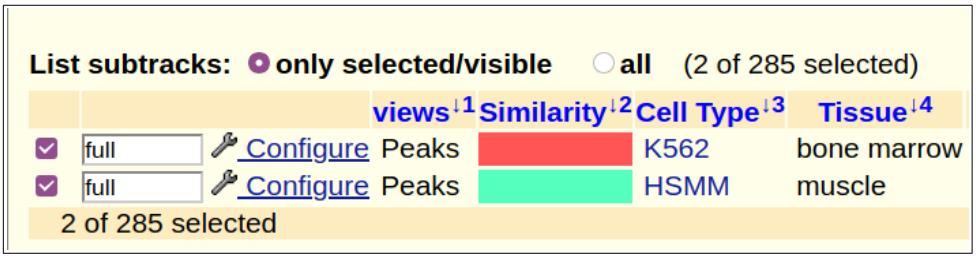
H3K27ac is high in the HSMM cell line at promoter and through the gene body while it is low for K562 cell line. The K562 cell line is a leukemia cell line from the blood of a 53 year old patient why HSMM is a skeleton muscle myoblast cell line. SOX11 is a HMG-box gene involved in embryonic development and determination of cell fate in normal cells. It would make sense that normal cell differentiation is disrupted and SOX11 is not activated with H3K27ac in a leukemia cell line why the gene would be activated on the normal differentiation from myoblast to myocyte.

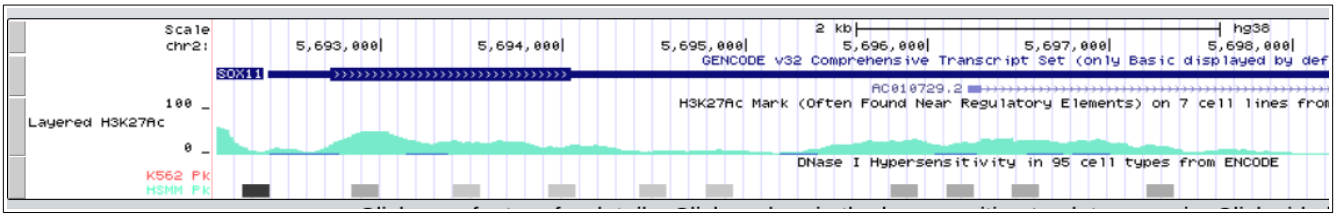
Table of Contents

Adding DNAse HS sites for only these two cell lines







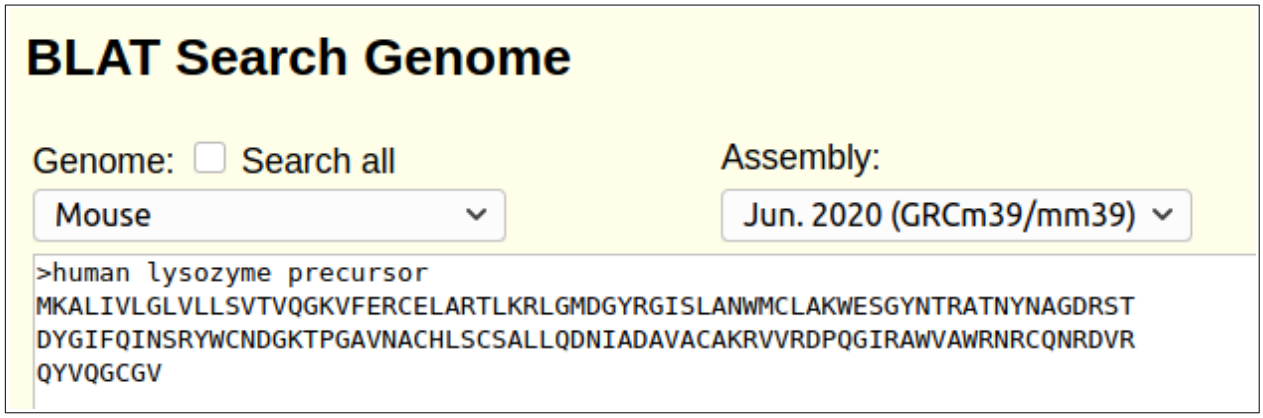


The HSMM cell line has a high scoring DNAse HS system just upstream of the promoter.

Table of Contents

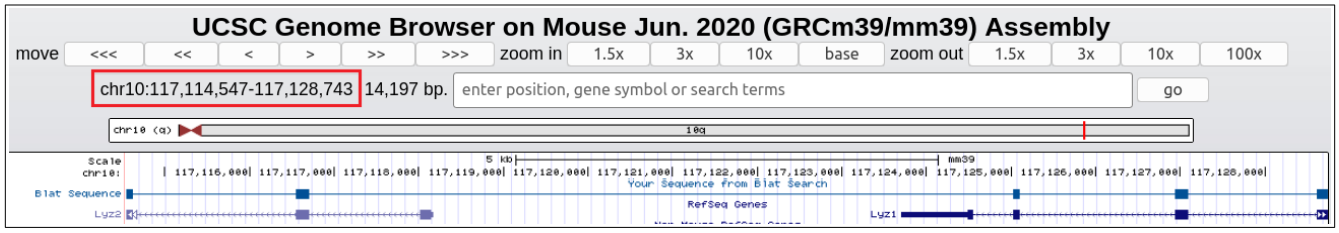
Use UCSC BLAT and Mouse Genome Informatics (MGI) to perform comparative genomics of human lysozyme with mouse genome.

1. Take the human lysozyme sequence (attached) and run BLAT against the mouse genome at UCSC. Click the browser link for the hit with the longest span. What are the chromosomal coordinates?



ACTIONS	QUERY	SCORE	START	END	QSIZE	IDENTITY	CHROM	STRAND	START	END	SPAN
browser details	human	219	1	127	148	79.6%	chr10	+-	117114547	117128743	14197
browser details	human	149	48	127	148	81.3%	chr10	+-	117125023	117127093	2071
browser details	human	60	48	99	148	69.3%	chr10	+-	117073526	117073681	156
browser details	human	45	67	85	148	89.5%	chr11	+-	103525864	103525920	57

chr10:117,114,547-117,128,743

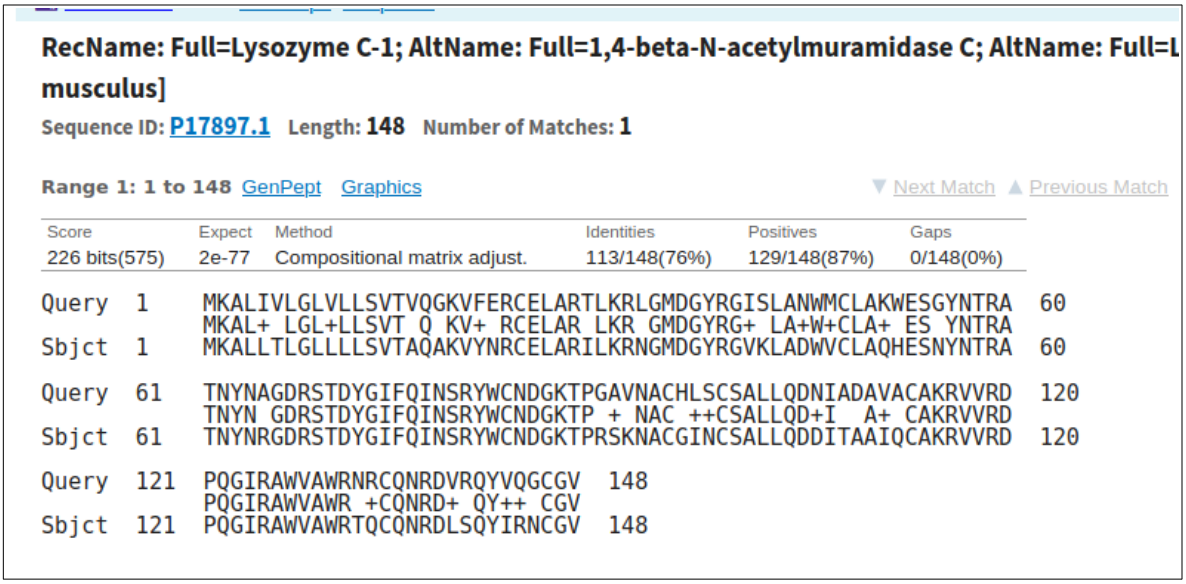


2. Go to MGI with the human sequence and run Mouse BLAST. Be sure to choose BLASTP and the UniProt Mouse database. Your best hit should be Lyz1. What is its score and P value?

P value = 2e-77 or basically 0

Score is 575

Table of Contents



RecName: Full=Lysozyme C-1; AltName: Full=1,4-beta-N-acetylmuramidase C; AltName: Full=L
musculus]

Sequence ID: P17897.1 **Length:** 148 **Number of Matches:** 1

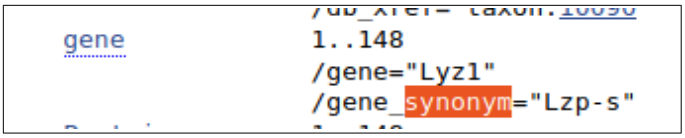
Range 1: 1 to 148 GenPept Graphics ▼ Next Match ▲ Previous Match

Score	Expect	Method	Identities	Positives	Gaps
226 bits(575)	2e-77	Compositional matrix adjust.	113/148(76%)	129/148(87%)	0/148(0%)

```
Query  1    MKALIVLGLVLLSVTVQGKVFERCELARTLKRLGMDGYRGISLANWMCLAKWESGYNTRA  60
            MKAL+ LGL+LLSVT Q KV+ RCELAR LKR GMDGYRG+ LA+W+CLA+ ES YNTRA
Sbjct  1    MKALLTLGLLLLSVTAQAKVYNRCELARILKRNGMDGYRGVKLADWVCLAQHESNYNTRA  60

Query  61   TNYNAGDRSTDYGIFQINSRYWCNDGKTPGAVNACHLSCSALLQDNIADAVACAKRVVRD  120
            TNYN GDRSTDYGIFQINSRYWCNDGKTP + NAC ++CSALLQD+I  A+ CAKRVVRD
Sbjct  61   TNYNRGDRSTDYGIFQINSRYWCNDGKTPRSKNACGINCSALLQDDITAAIQCAKRVVRD  120

Query  121  PQGIRAWVAWRNRCQNRDVRQYVQGCGV  148
            PQGIRAWVAWR +CQNRD+ QY++ CGV
Sbjct  121  PQGIRAWVAWRTQCQNRDLSQYIRNCGV  148
```

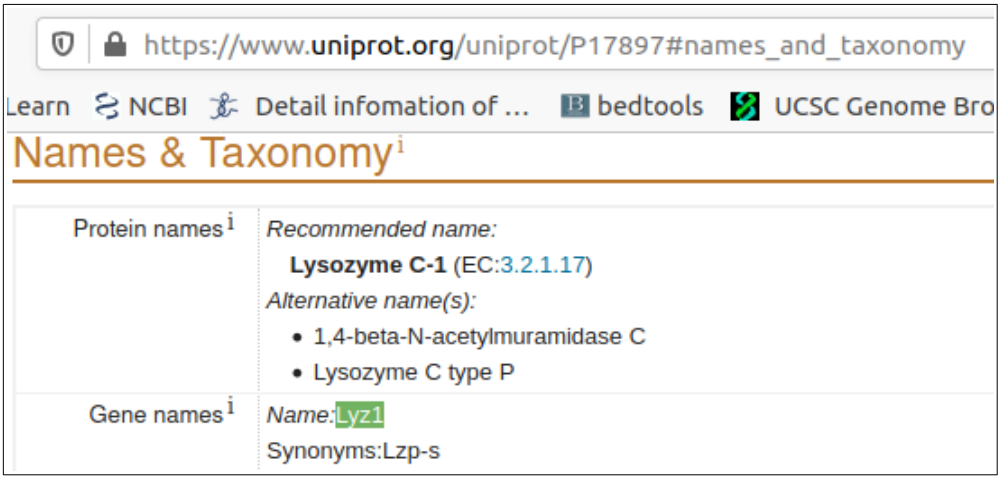
3. Click the Lyz1 link. What synonyms are listed?

Lzp-s is listed at NCBI GenPept entry



```
gene            1..148
                /gene="Lyz1"
                /gene_synonym="Lzp-s"
```

As well as in UniProt



https://www.uniprot.org/uniprot/P17897#names_and_taxonomy

Learn NCBI Detail infomation of … bedtools UCSC Genome Bro

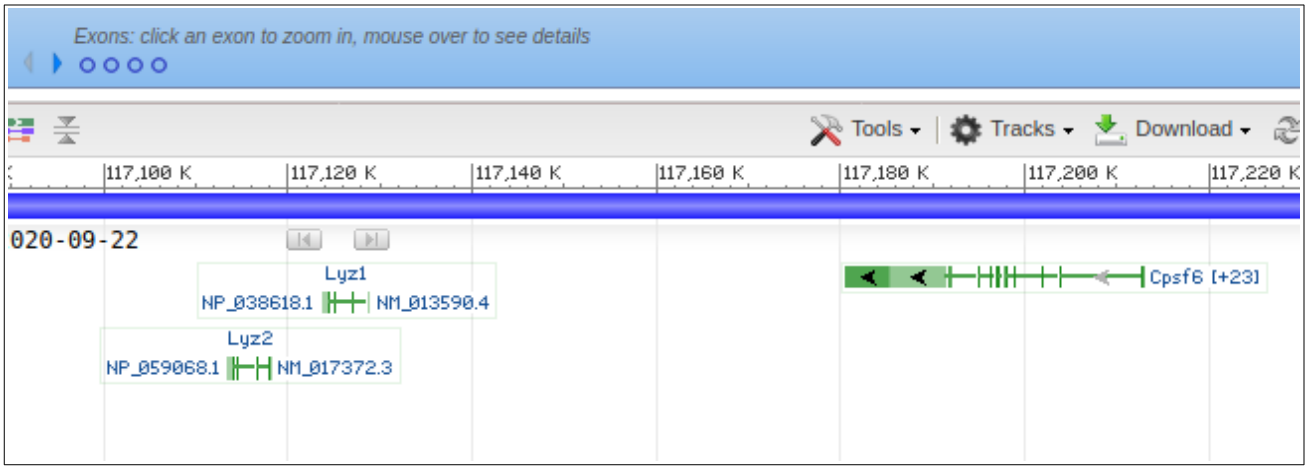
Names & Taxonomy

Protein names	*Recommended name:* **Lysozyme C-1** (EC:3.2.1.17) *Alternative name(s):* • 1,4-beta-N-acetylmuramidase C • Lysozyme C type P
Gene names	*Name:*Lyz1 Synonyms:Lzp-s

4. What are the flanking genes?

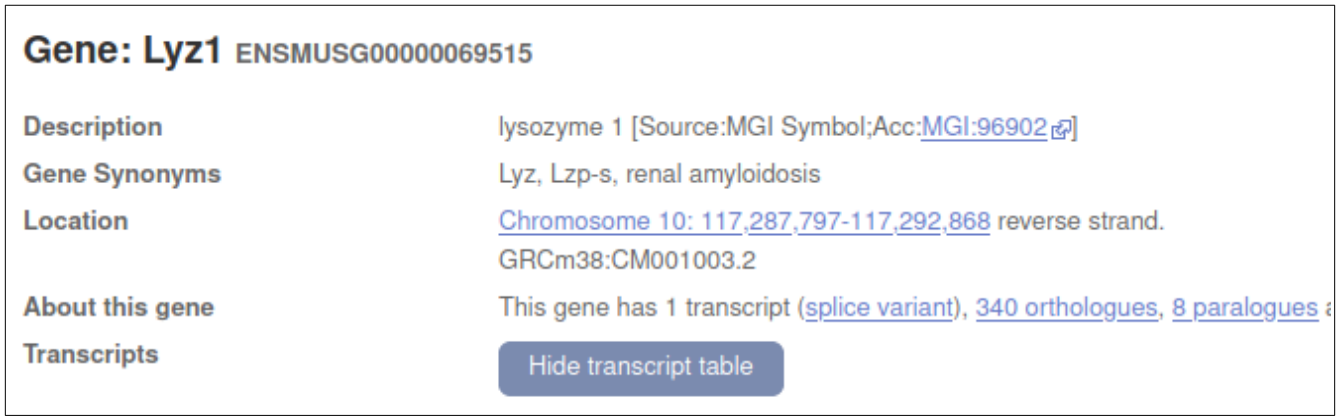
Viewing in NCBI Genome Data Viewer we see Lyz2 is just down stream on the negative strand and Cpsf6 just upstream on the negative strand

Table of Contents



5. Click the Ensembl Gene Model under Sequences. How many paralogues in mouse?

8 paralouges



6. Click the Gene Tree image. Expand Placental Mammals. List all primate species that come up.

There are no primate species that come up in Ensembl for Lyz1 despite BLAT having a hit for Human Lysozyme C with 79.6% identity.

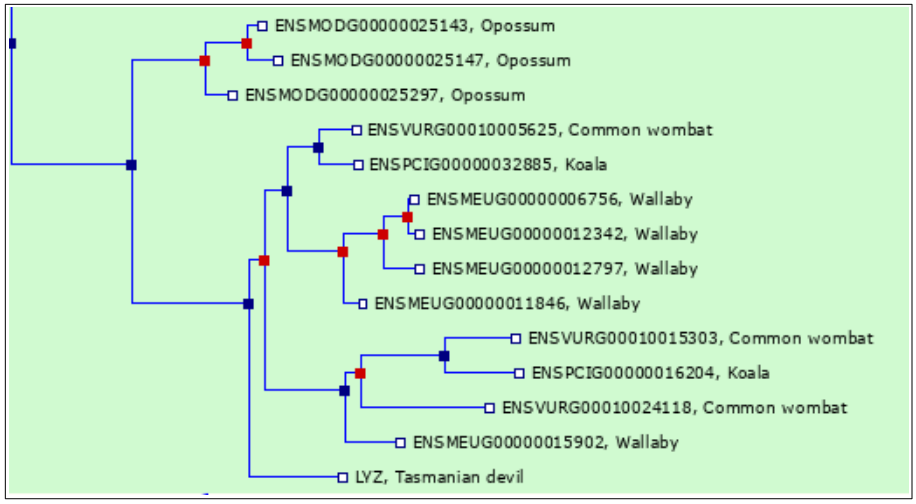


Table of Contents

Summary of orthologues of this gene Hide

Click on 'Show details' to display the orthologues for one or more groups of species. Alternatively, click on 'Configure this page' to choose a custom list of species.

Species set	Show details	With 1:1 orthologues	With 1:many orthologues	With many:many orthologues	Without orthologues
Primates (*27 species*) Humans and other primates	☑	0	0	0	27
Rodents and related species (*31 species*) Rodents, lagomorphs and tree shrews	☐	2	5	4	20
Laurasiatheria (*45 species*) Carnivores, ungulates and insectivores	☐	0	1	1	43
Placental Mammals (*108 species*) All placental mammals	☐	2	6	5	95
Sauropsida (*19 species*) Birds and Reptiles	☐	0	4	15	0
Fish (*86 species*) Ray-finned fishes	☐	0	36	38	12
All (*280 species*) All species, including invertebrates	☐	2	91	66	121

Create and document a workflow for students to follow for uploading custom yeast tracks from the Saccharomyces Genome Database (SGD) and NCBI GEO database to UCSC Genome Browser. Posted to discussion forums for students to follow.

I am going to post some screenshots since this is an all visual exercise. I was able to download a few tracks from the SGD database in wiggle format, as well as a BED and bigWig file from the two different not related studies from the NCBI GEO database.

You can download datasets while in the genome viewer of SGD but you can only download based on the chromosome you are viewing doing it this way.

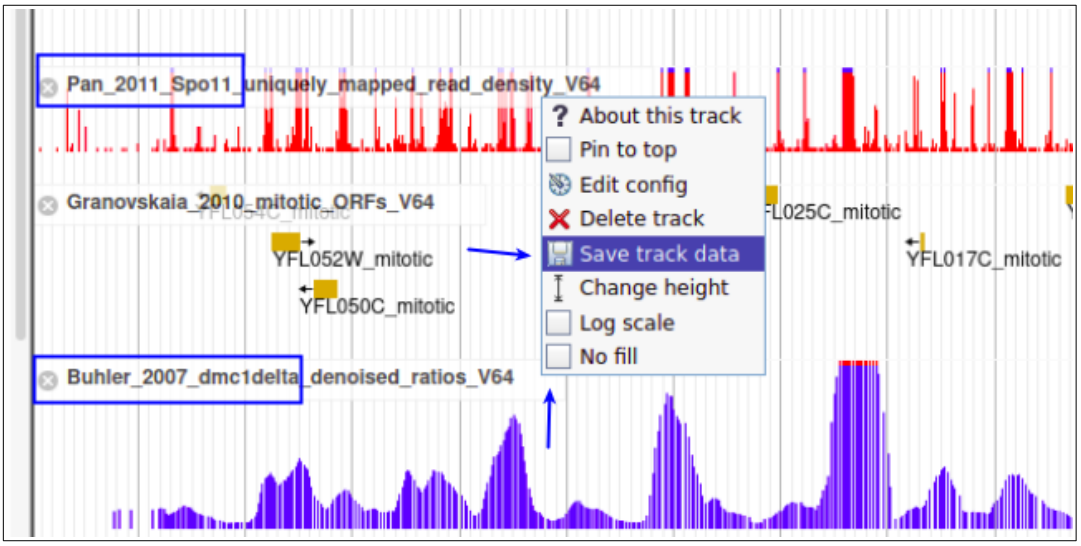
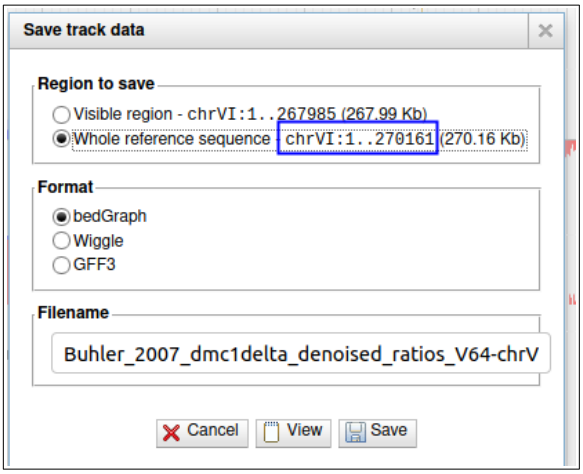
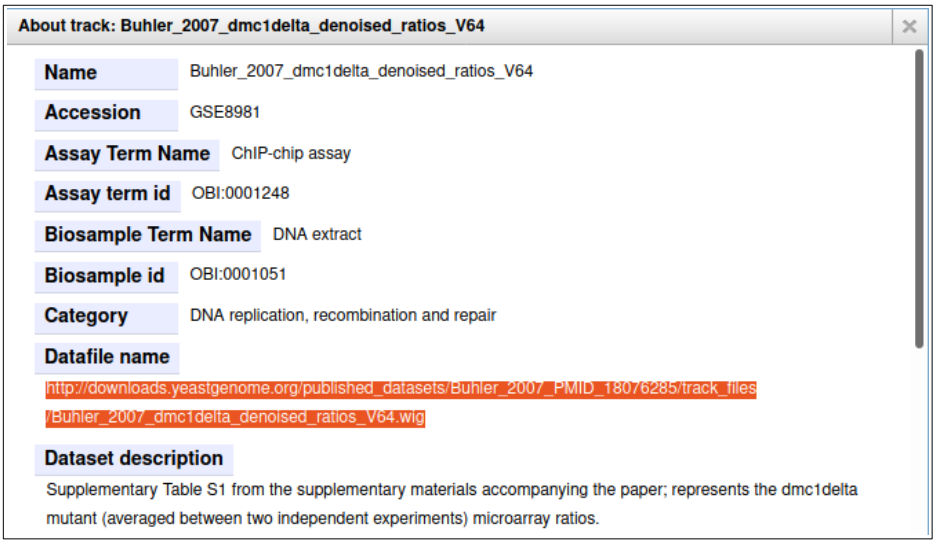


Table of Contents



Alternatively you can download from the link in the (?) question mark "About this track" to get the full dataset for entire genome.



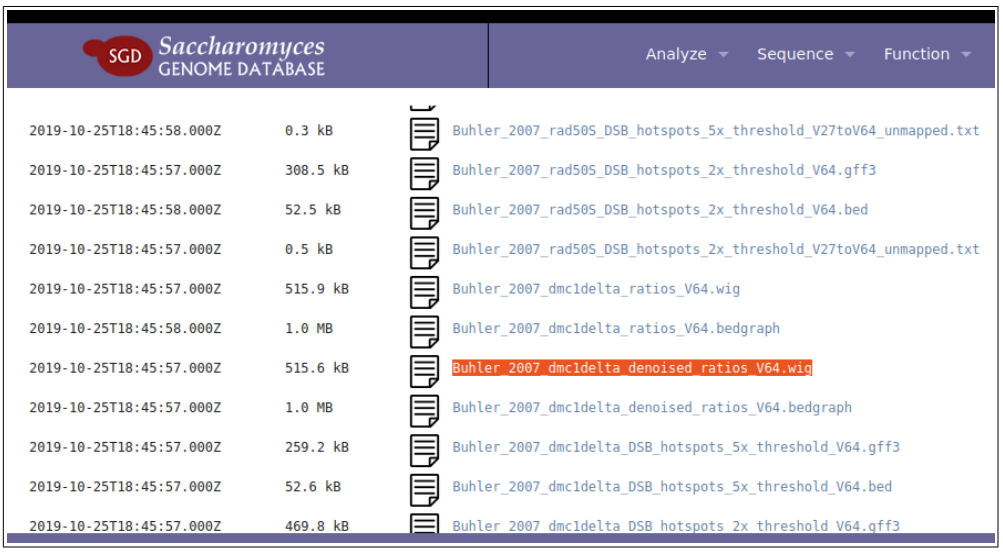
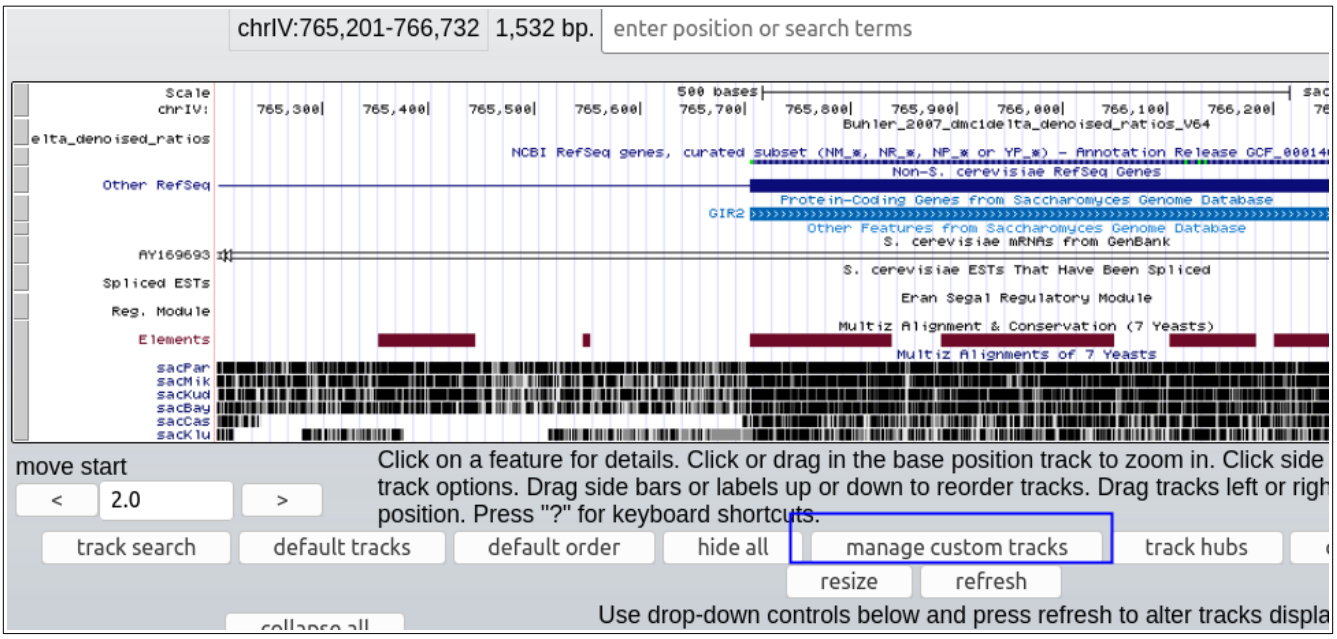
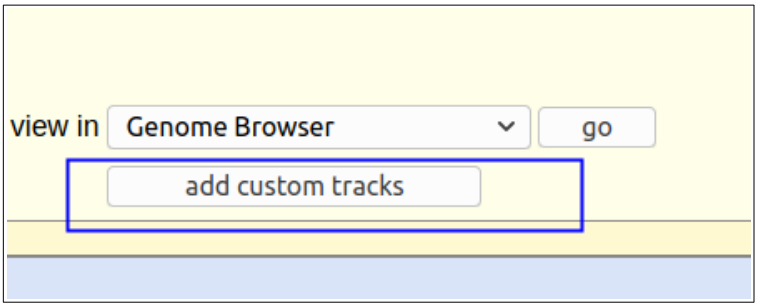


Table of Contents

To add to UCSC select "manage custom tracks"



You can upload files here. Depending on which version of UCSC yeast genome you could have issues because they change from Chr numbers (1, 5, 14) to Roman numerals (I, V, XIV).



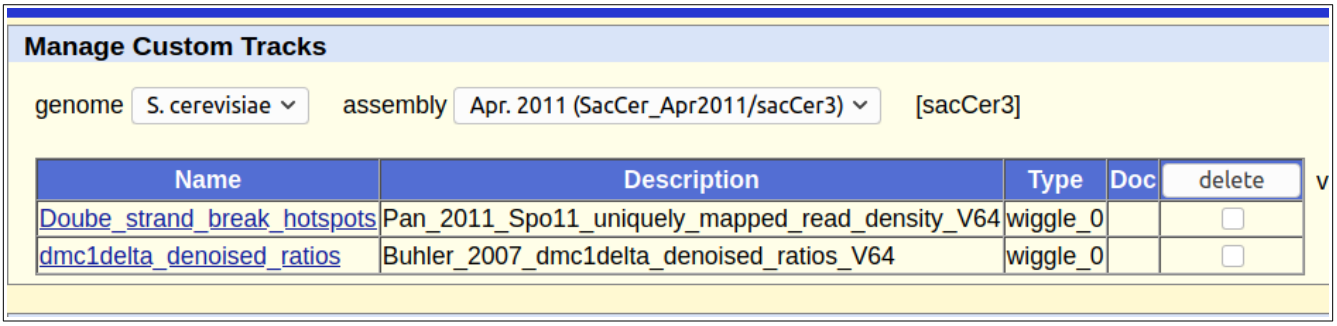
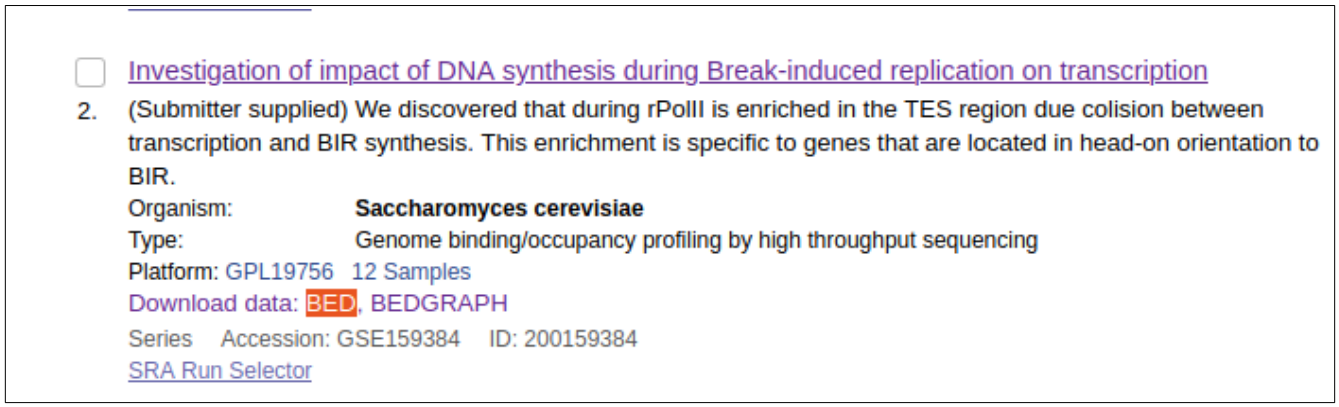


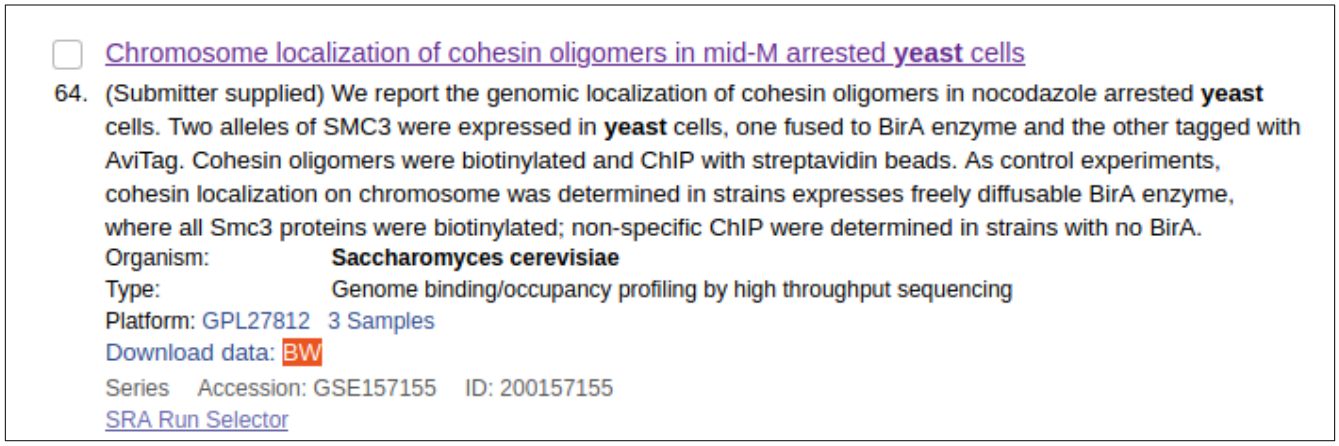
Table of Contents

Then you can get datasets from GEO as well.

BED:



bigWig:



However you cannot add binary bigWig files to UCSC this way.

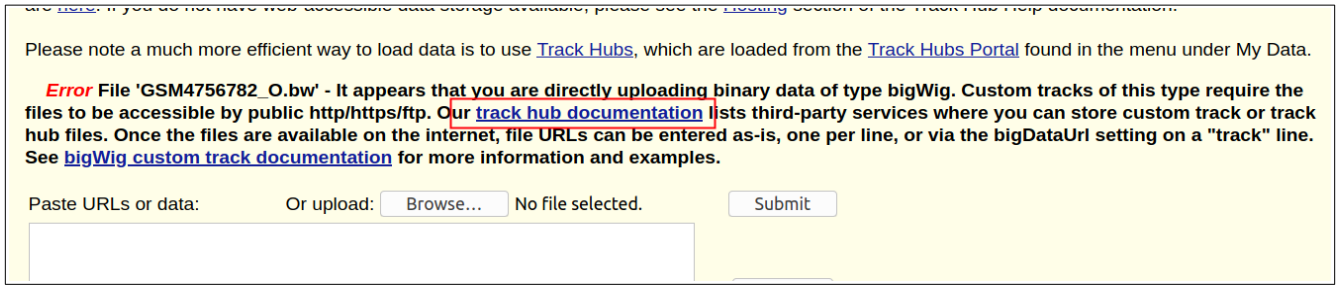
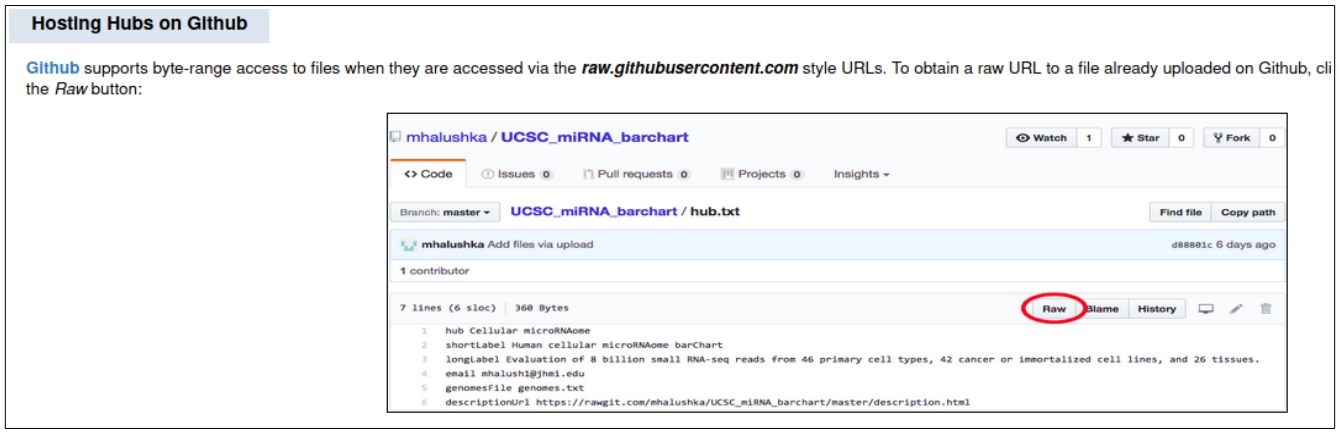


Table of Contents

However you can use github to upload files up to 100Mb in size and then copy the raw url to upload:

https://genome.ucsc.edu/goldenPath/help/hgTrackHubHelp.html#Hosting



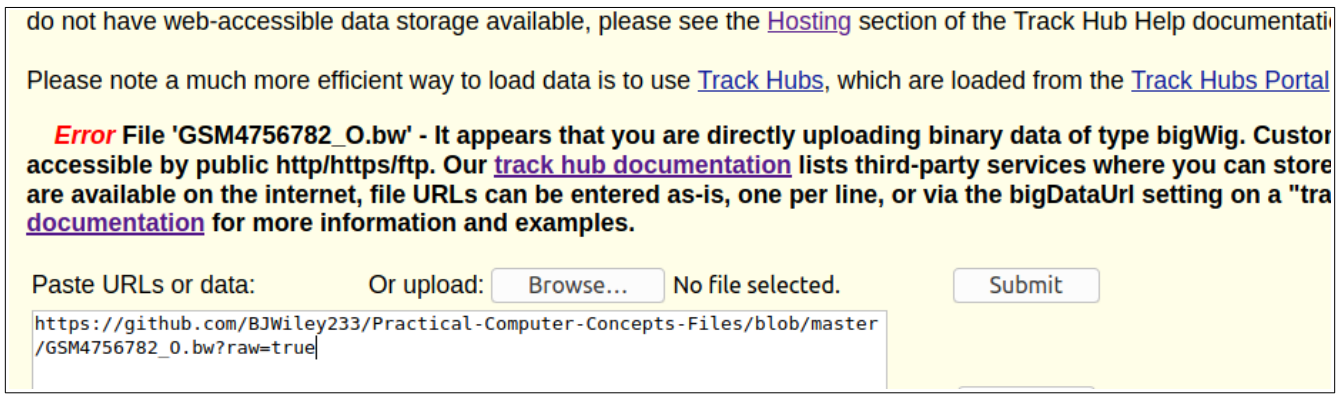


Table of Contents

In the below I have 4 custom tracks. The top 2 from SGD are wiggle format (#1 and #2), then a BW (#3) from GEO and a BED from GEO (4):

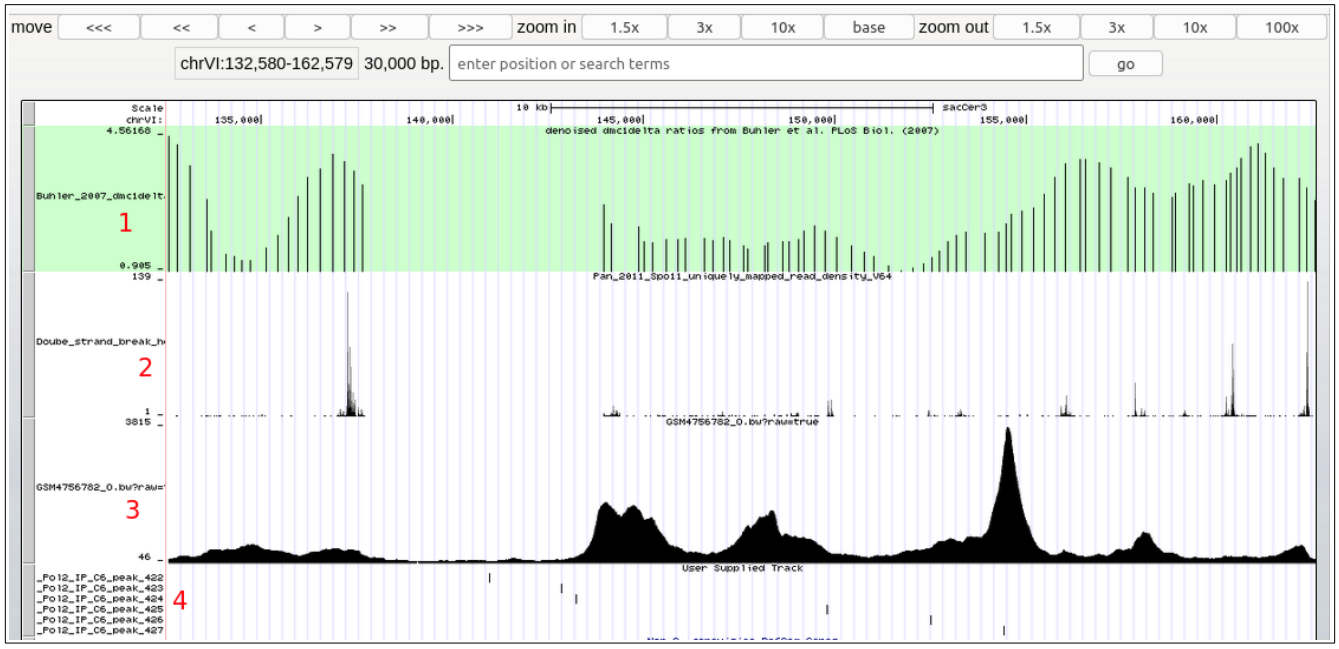


Table of Contents

4.9. Week 9 - Noncoding RNAs and ultraconserved regions

This week was an integration between the week we covered RNA databases and tools in my Biological Databases course. The project for this was to develop and create an RNA database schema for all types of non-coding RNAs including mRNA, lncRNA, tRNA, miRNA, miRNA, shRNA for the sake of the following 3 goals:

- First goal is to look for various RNA molecule types that are associated with the same or similar disease(s).
- The next is use sequence analysis algorithm (some sort of Neighborhoods analysis) to be able to search experimental sequences from Next gen sequencing to find existing similar RNA molecules or to determine which molecule your sequence is from.
- For the sequences attribute which is one of the extra attributes you can search with the lncRNA and interfering RNAs on the sequences of other RNA types including its same type (i.e. tsix and Xist) to predict additional targets then those that are listed from the source.

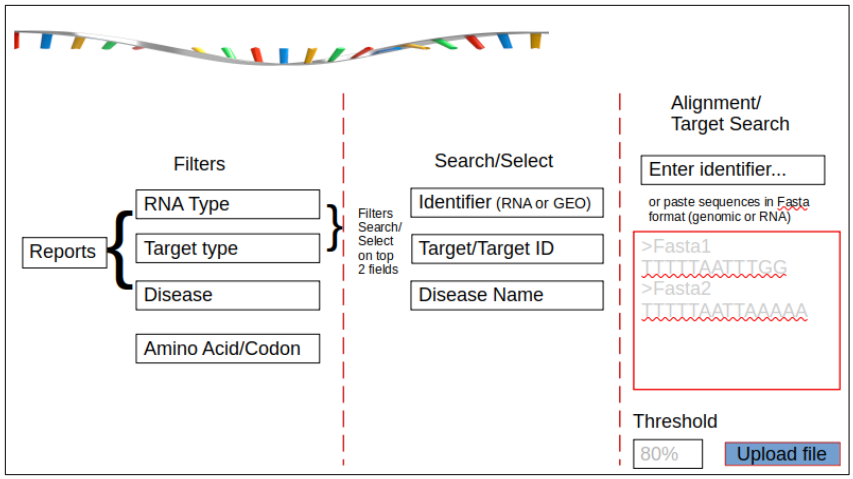
The interface would have a few search fields as well as filtering fields to update the search fields. Based on selected filters in first column of interface you can print reports or advance to second column to search based on those filters. Lastly you can search for alignments with high identity to determine if you RNA-sequence is a type of RNA molecule or is an siRNA which have closer identity to targets or search for targets if lower threshold is selected such as that for miRNA.

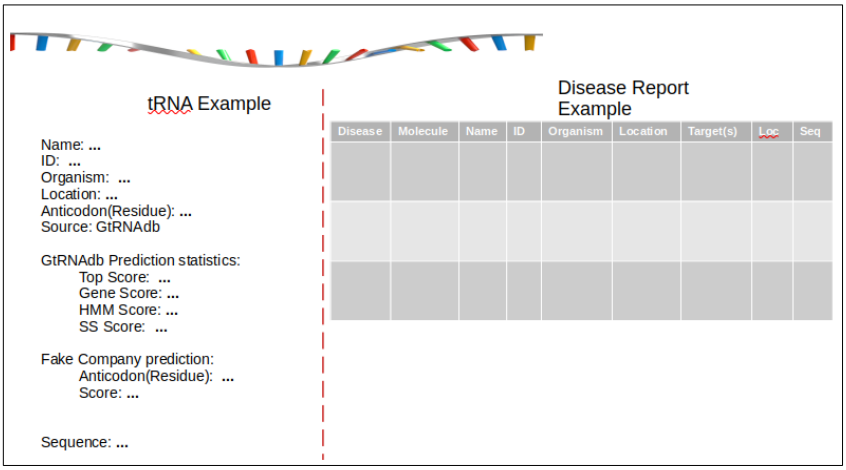
The following RNA database sources were used:

1. NCBI Gene (mRNA)
2. NCBI GEO (mRNA)
3. NONCODE (lncRNA)
4. LncTarD (lncRNA)
5. GtRNAdb (tRNA)
6. miRBase (miRNA)
7. DIANA miRGen (miRNA)
8. DIANA TarBase (miRNA)
9. MIT/ICBP siRNA Database (siRNA & shRNA)

Table of Contents

Interface:





Data Dictionary:

Data element name	Description	Source	Source field	Datatype	Example(s)
rnaMoleculeName	RNA molecule name from database	1. NCBI Gene GenBank file 2. NONCODE 3. GtRNAdb 4. miRBase 5. DIANA miRGen 6. MIT/ICBP siRNA Database	1. product 2. KeyName 3. GtRNAdb Gene Symbol 4. ID 5. miRNA name 6. N/A	String/ VARCHAR	1. BRCA1 DNA repair associated, transcript variant 1 2. lnc-BRCA1-1-5_dup1 3. tRNA-Arg-CCT-2-1 4. hsa-mir-4698 5. hsa-mir-192 6. NONE
identifier	The id for the lncRNA from database	1. NCBI Gene GenBank file 2. NONCODE 3. GtRNAdb 4. miRBase 5. DIANA miRGen 6. MIT/ICBP	1. transcript_id 2. NONCODE TRANSCRIPT ID / NONCODE ID 3. tRNAscan-SE ID	String/ VARCHAR	1. NM_007294.4 2. NONHSAT053833.2 3. chr17.trna19 4. MI0017331 5. MI0000234 6. 1107

Table of Contents

		siRNA Database	4. Accession 5. DIANA miRNAGen 6. siRNA ID#		
rnaType	1 of 5 types of RNA	1. NCBI Gene GenBank file 2. NONCODE 3. GtRNAdb 4. miRBase 5. DIANA miRGen 6. MIT/ICBP siRNA Database	1. FEATURES 2. NONCODE 3. GtRNAdb 4. N/A 5. N/A 6. Probe type	String enum(mRNA, lncRNA, tRNA, miRNA, siRNA, shRNA)	1. mRNA 2. lncRNA 3. tRNA 4. miRNA 5. miRNA 6. shRNA
targets	Target gene(s), protein (transcription factors), or RNA	1. LncTarD 2. miRBase 3. DIANA miRGen 4. MIT/ICBP siRNA Database	1. Target 2. Predicted targets 3. TF name 4. Target gene	MAP for LncTarD & miRBase or ARRAY	1. {miR-214-3p: Epithelial ovarian cancer, WNT1: Colon cancer} 2. {TargetMiner: [NM_001122679, NM_000024, NM_000024], TargetScanVert: [RP11-379H8.1, UBE2W, LAGE3], miRDB: [BRWD3, MED13L] 3. [GATA3, Jun, Fos] 4. [AKT1]
antiCodon	The anticodon with the associated amino acid	GtRNAdb	Predicted tRNA Isotype / Anticodon	String/ VARCHAR	CCT (Arg)
evidenceGtRNAdb	The scores from the tRNAscan-SE for the prediction evidence in GtRNAdb	GtRNAdb		MAP of Top Scoring, Gene Score, Mature tRNA Score, HMM Score, Secondary Structure Score	 Arg (99.8 bits) 71.1 71.1 52.80 18.30
evidenceFakeCo	The evidence from Fake Company, Inc.	Evidence from user Community	Anticodon and score	MAP of Anticodon (Residue), Score	CCT (Arg) 85
geoID		NCBI GEO		String/ VARCHAR	GSE123068
geoCondition	This will be the condition being tested that is stated in the design of the dataset.	1. NCBI GEO Dataset *2. Internal GEO Sample within the dataset* *3. Internal GEO Sample within the dataset*	1. Overall design, *2. Characteristics* *3. Treatment protocol*	String/ LONGTEXT and/or Array of Sample Objects	1. 3’ RNA-Seq (QuantSeq) profiling of sorted cDCs populations from WNT1 overexpressing and control (Empty) lung tumors. *2. cell type: Ovarian Clear Cell Carcinoma cell line* *3. No treatment*
genomicSequence	The genomic sequence. Some have both genomic and	1. NCBI Gene GenBank file 2. NONCODE 3. GtRNAdb	1. ORIGIN with join() 2. Sequence 3. Genomic	String/ LONGTEXT	DNA sequence. Can be long for some and short for other short type RNAs.

Table of Contents

	RNA with uracil	4. miRBase 5. DIANA miRGen 6. MIT/ICBP siRNA Database	Sequence 4. Sequence 5. DIANA miRNAGen 6. Sequence		
location	Location in genome including chromosome and positions (concatenated if multiple fields, with hyphen if all 3 fields)	1. NCBI Gene GenBank file 2. NONCODE 3. GtRNAdb 4. miRBase 5. DIANA miRGen	1. chromosome, REGION 2. Chromosome, Start Site, End Site 3. Locus 4. Genome context 5. TSS Coordinates	String/ VARCHAR	1. 17:complement(43044295..43125364) 2. chr17:43221167-43303331 3. chr17:75034431-75034503 (-) 4. chr12: 47187812-47187891 [+] 5. chr11:64660708-64660709 [-]
associatedDiseases	Diseases associated with RNA	1. NONCODE 2. LncTarD 2. DIANA miRGen 3. DIANA TarBase	1. Disease name 2. DiseaseName 3. Cluster diseases 4. Related Diseases	ARRAY	1. AIDS,dermatomyositis 2. Hepatocellular carcinoma 3. Adenocarcinoma, Colorectal Neoplasms, Hematologic Neoplasms 4. Adenocarcinoma, Colorectal Neoplasms, Hematologic Neoplasms, Uterine Cervical Neoplasms
Source, sourceEvidence	Which database or source the database came from	The database that is queried		String/ VARCHAR, LONGTEXT	Database name from each database and all their evidence from that entry that was obtained.

Schema :

This is an Object Oriented database that can be implemented in Mongo DB or Hadoop Ozone.

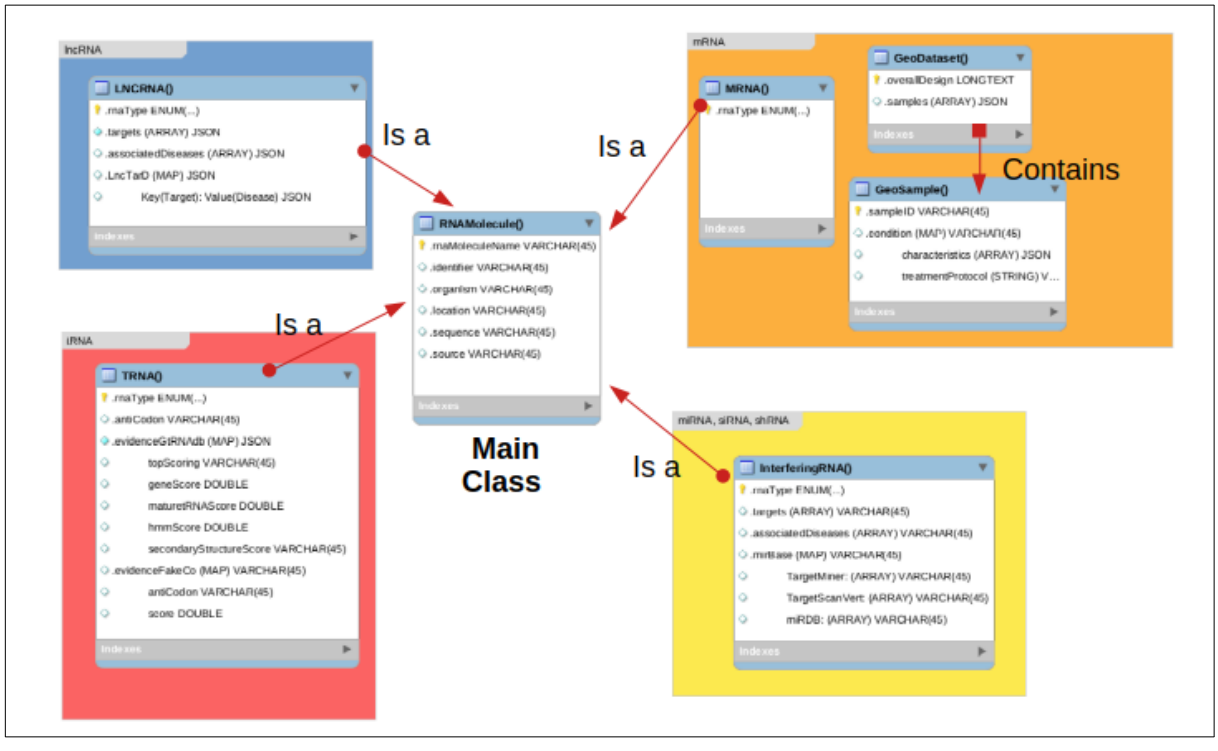


Table of Contents

4.10. Week 10 - Next Generation Sequencing and Analysis

Perform NGS analysis using command-line tools. Although these analysis were performed from the command-line, each one of them can also be performed in Galaxy. The example sample is one from the SNP Project section 5 from the Sequence Read Archive (SRA). An entire workflow to run a list of SRR IDs is available on github.com here.

NCBI SRA-toolkit to download SRR sample files with raw Illumina paired-end reads.

srr.ids file:
SRR10023023
SRR10022985
SRR10022871
SRR10022930

Command:

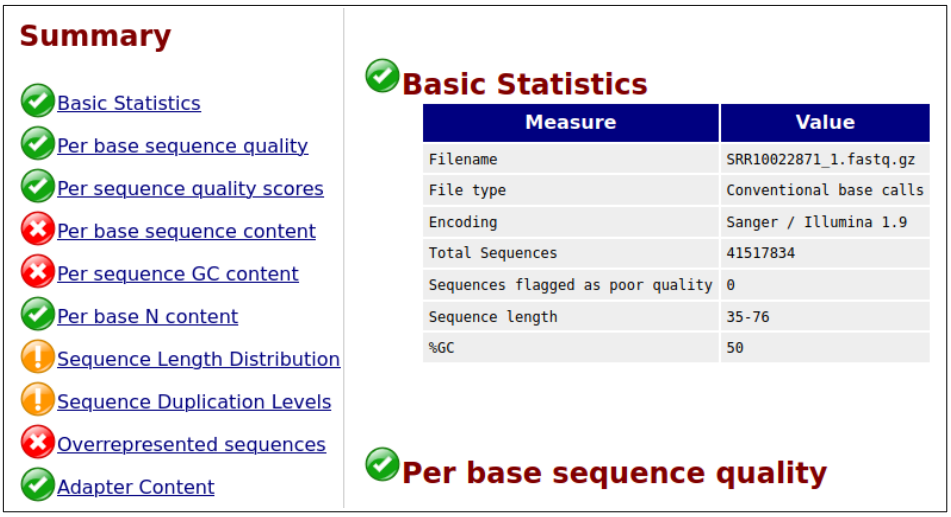
```
parallel -a srr.ids fastq-dump --split-files --origfmt --gzip;
```

fastqc

Command:

```
parallel -a srr.ids fastq-dump --split-files --origfmt --gzip;
```

Output:



Summary

- Basic Statistics
- Per base sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content

Basic Statistics

Measure	Value
Filename	SRR10022871_1.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	41517834
Sequences flagged as poor quality	0
Sequence length	35-76
%GC	50

Per base sequence quality

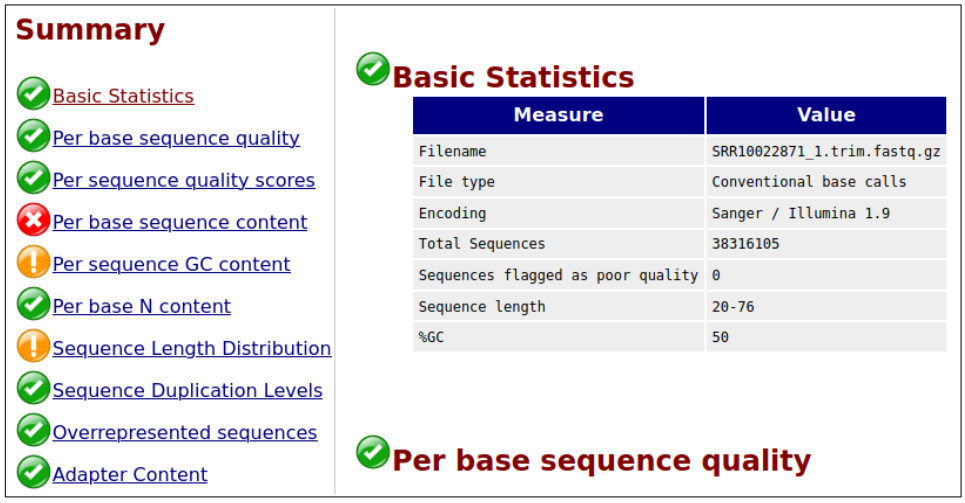
Table of Contents

trimmomatic

Command:

```
java -jar /path/to/trimmomatic-0.39.jar PE SRR10022871_1.fastq.gz SRR10022871_2.fastq.gz \
        SRR10022871_1.trim.fastq.gz SRR10022871_1un.trim.fastq.gz \
        SRR10022871_2.trim.fastq.gz SRR10022871_2un.trim.fastq.gz \
        SLIDINGWINDOW:4:20 \
        MINLEN:20;
```

After running fastqc again:



bwa

bwa requires running alignment first for each sample in paired-end to output the forward and reverse sequence alignment index (sai) files before running alignment. It also requires creating the index for the reference.

Commands:

```
bwa index hg19.fa -p hg19
```

```
bwa aln hg19 SRR10022871_1.trim.fastq.gz > SRR10022871_1.trim.sai
```

```
bwa aln hg19 SRR10022871_2.trim.fastq.gz > SRR10022871_2.trim.sai
```

```
bwa sampe hg19 SRR10022871_1.trim.sai SRR10022871_2.trim.sai SRR10022871_1.trim.fastq.gz
SRR10022871_2.trim.fastq.gz > SRR10022871_paired.hg19.trim.sam
```

Table of Contents

bowtie2

bowtie2 has indexes that can be downloaded but indexes can also be created before aligning.

Commands:

```
bowtie2-build hg19.fa hg19;
```

```
bowtie2 -x hg19 -1 SRR10022871_1.trim.fastq.gz -2 SRR10022871_2.trim.fastq.gz --threads 32 --fast -S SRR10022871_paired.hg19.trim.sam;
```

samtools sort

Sorting the alignment file as well as creating a bam index .bai file is required to run variant caller freebayes.

Commands:

```
samtools sort SRR10022871_paired.hg19.trim.sam -o SRR10022871_paired.hg19.trim.sorted.bam -O bam;
```

```
samtools index SRR10022871_paired.hg19.trim.sorted.bam;
```

freebayes

Option to run freebayes on a specific region is used.

Command:

```
freebayes -f hg19.fa SRR10022871_paired.hg19.trim.sorted.bam -r chr17:0-81195210 > SRR10022871_paired.hg19.trim.vcf;
```

VCFlib

vcffilter

Filtering for allele frequency = 0.5 since we have two samples in the paired-end

Command:

```
vcffilter -f "DP > 10 & AF = 0.5" SRR10022871_paired.hg19.trim.vcf > SRR10022871_paired.hg19.trim.filtered.vcf
```

Table of Contents

vcfannotate

Using the geneRef table for NCBI RefSeq as indicated by GATK, we can annotate our vcf file for variants that overlap certain regions.

Command:

```
vcfannotate -b refGene_BRCA1_exons_hg19_transcript1.bed \
-k REFGENE \
SRR10022871_paired.hg19.trim.filtered.vcf >
SRR10022871_paired.hg19.trim.filtered.annot.vcf
```

We can then get the lines with:

```
grep "^[^#]" SRR10022871_paired.hg19.trim.filtered.annot.vcf | grep "REFGENE"
```

Table of Contents

4.11. Week 11 – ChIP-seq

Use deepTools commands multiBamSummary, plotPCA, plotCorrelation, and plotFingerprint to analyze NGS ChIP-seq runs for TAL1 on two different mouse cell lines, G1E cells and megakaryocytes from fetal liver.

1. Install deepTools from github.

    Commands:

    ```
    git clone https://github.com/deeptools/deepTools

    cd deepTools

    python setup.py install
    ```

2. After performing alignment of Single-end reads from fastq files sequencing from mouse made available by the Mouse ENCODE Consortium with bwa aligner in Galaxy, download BAM files for input and TAL1 antibody for chromatin immunoprecipitation for both cell lines.

3. Run multiBamSummary

    Commands:

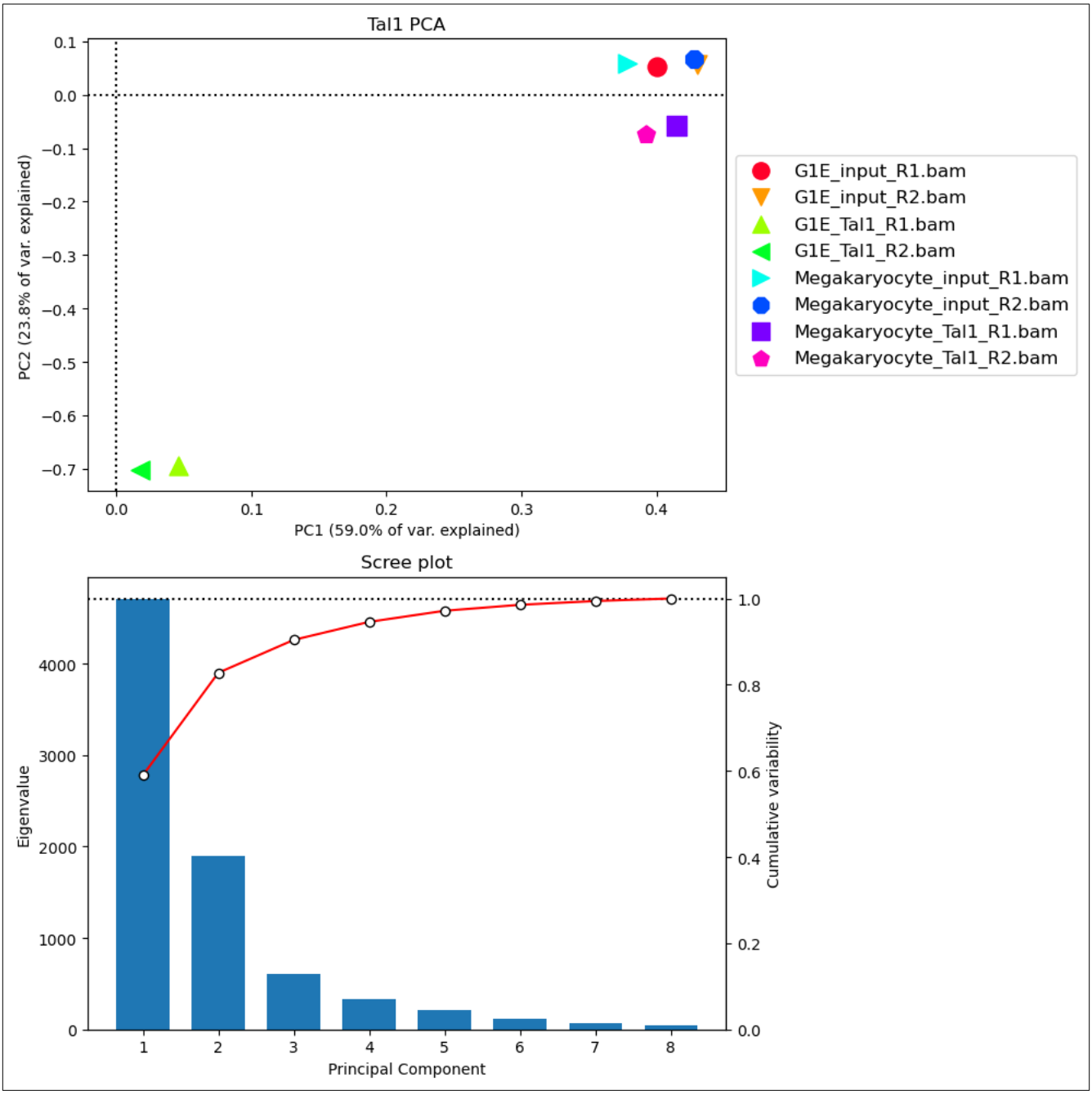
    ```
    ls *.bam | parallel samtools index '{}'

    multiBamSummary bins --bamfiles *.bam --binSize 1000 -out readCounts.npz
    ```

4. Run plotPCA on .npz output from multiBamSummary

    Command:

    ```
    plotPCA -in readCounts.npz -T "Tal1 PCA" -o Tal1_pca.png
    ```

Table of Contents



This shows that the G1E alignment files for Tal1 immunoprecipitation is separate from the G1E input control and other samples.

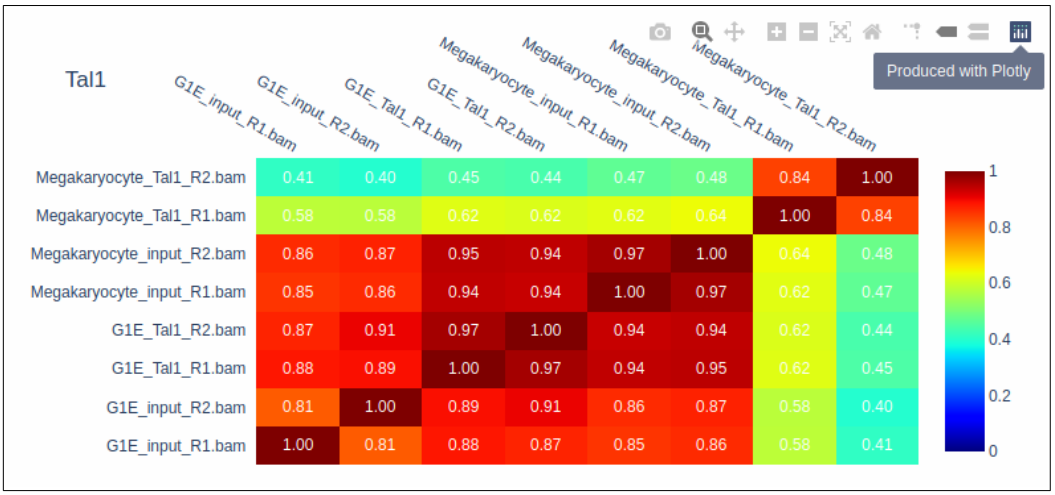
Table of Contents

5. Run plotCorrelation on all alignment files to determine correlation patterns across samples. Note that plotting with ploty required updating the source code in two places of the correlation.py file. I have submitted enhancement requests but manually updated my source code.

https://github.com/deeptools/deepTools/issues/1013 (enhancements)

Command:

```
plotCorrelation --corData readCounts.npz --corMethod pearson --whatToPlot heatmap –
plotTitle "Tal1" --removeOutliers --skipZeros --plotNumbers --plotFileFormat plotly -o
Tal1.html
```



Running with .png format gives us the dendrogram as well. This matches the PCA plot above showing the G1E Tal1 antibody precipitate is classified in its own group.

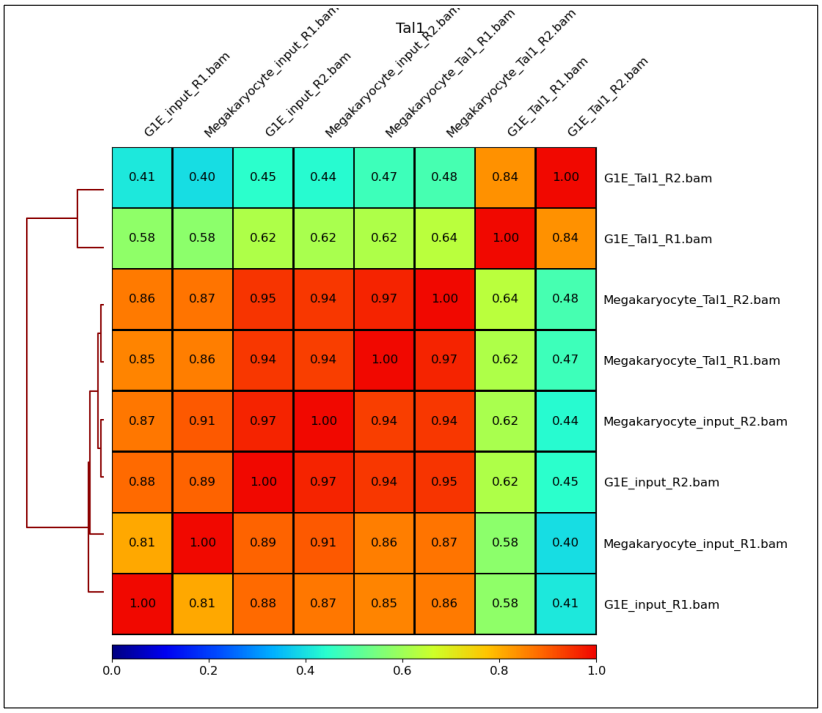
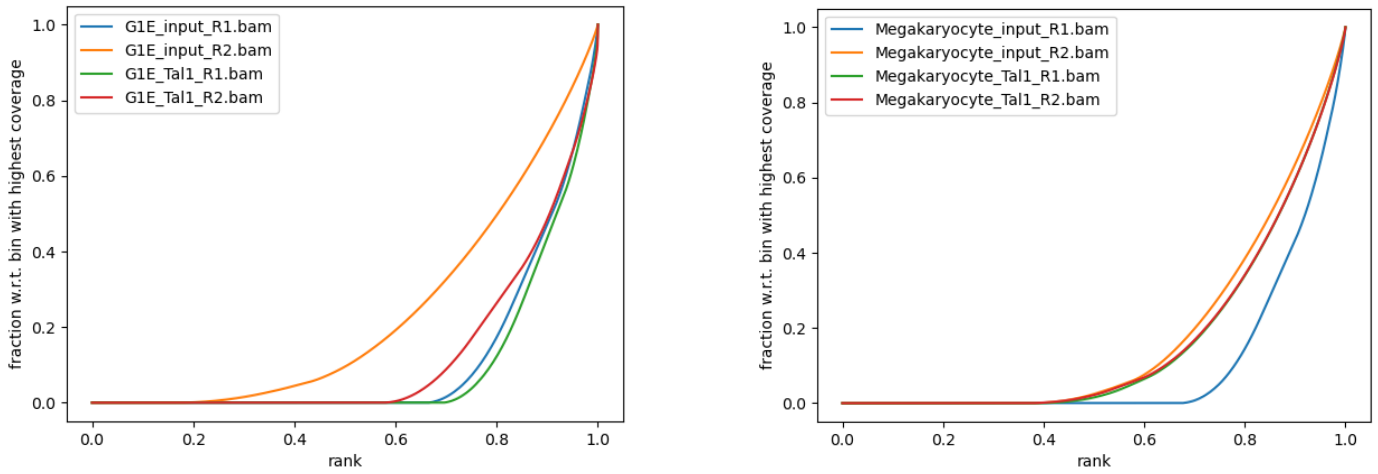


Table of Contents

6. Run plotFingerPrint on alignment files for input control and TAL1 for each cell type separately

Commands:

```
plotFingerprint -b G1E*.bam -plot G1E_fp.png --binSize 100 –skipZeros

plotFingerprint -b Megakaryocyte*.bam -plot Megakaryocyte_fp.png --binSize 100 --skipZeros
```



For G1E cells run 1 (blue and green) showed poor enrichment of Tal1 compared to control while for megakaryocytes run 2 (orange and red) showed poor enrichment of Tal1 compared to control.

Use MACS2 callpeak on each of the cell lines to visualize peaks in the experiment minus the background input control with the bedGraph files in IGV for determining Tal1 binding sites.

MACS2 is a much larger python package and also runs on cython so it is better to create its own conda environment and as well as installing cython with:

```
conda create -n macs_env

conda activate macs_env

git clone https://github.com/taoliu/MACS.git

cd MACS

python setup.py install

conda install cython
```

Table of Contents

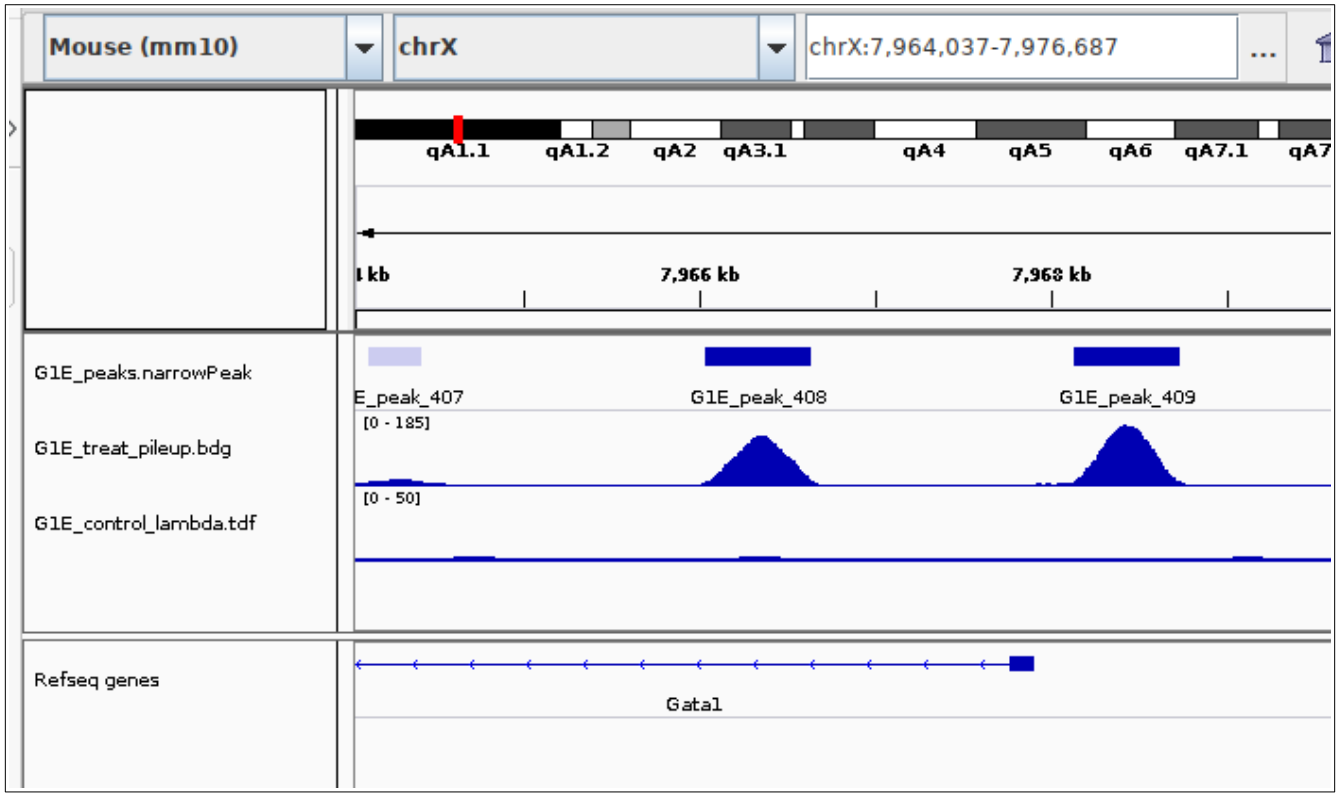
First run MACS2 callpeak on G1E cells with pooling:

```
macs2 callpeak -t G1E_Tal1*.bam -c G1E_input*.bam -f BAM --gsize 1.87e9 --bdg --name G1E
```

Convert large control file to .tdf format for viewing in IGV using IGVtools.

```
igvtools toTDF G1E_control_lambda.bdg G1E_control_lambda.tdf mm10.chrom.sizes
```

Viewing in IGV we can then see the narrow peaks for Tal1 binding. Interesting one of the highest scores/peaks is at the beginning of the GATA1 gene, that is Tal1 transcription factor regulates the transcription of the gene in which it complexes with. Many transcription factors like GATA1 bind and regulate their own gene transcription. According to Tsai et al., GATA1 is a positive regulator of its own promoter[1] and so it makes sense that since TAL1 complexes with GATA1 we would see binding where GATA1 binds at the GATA1 promoter.

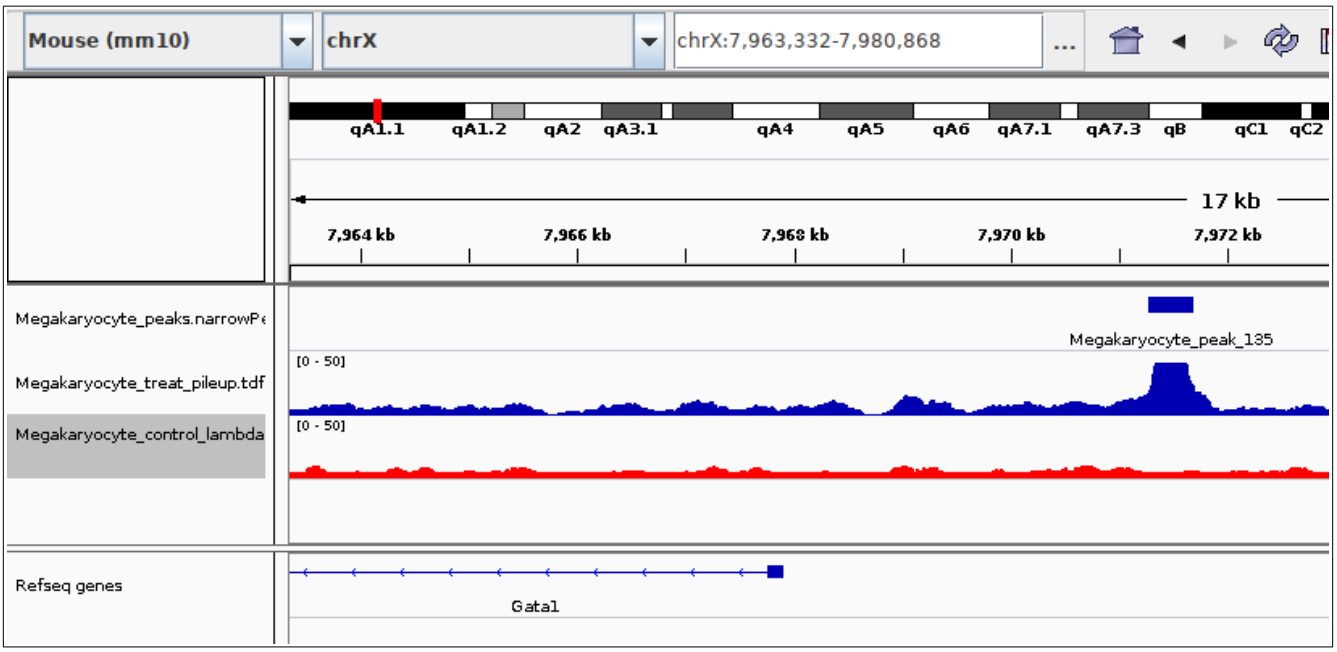


Run MACS2 callpeak on megakaryocytes cells with pooling:

```
macs2 callpeak -t Megakaryocyte_Tal1*.bam -c Megakaryocyte_input*.bam -f BAM --gsize 1.87e9
--bdg --name Megakaryocyte
```

Table of Contents

There is also peak enrichment at the GATA1 gene and RUNX1, albeit a little more upstream of the promoter for GATA1 then in G1E cells.



Using bedtools find intersecting regions of narrow peaks for GAL1 between G1E cells and megakaryocytes. Then find regions that exists only in G1E and again for only in megakaryocytes.

```
bedtools intersect -a G1E_peaks.narrowPeak -b Megakaryocyte_peaks.narrowPeak > narrow_peaks_G1E_megak.bed
```

```
wc -l narrow_peaks_G1E_megak.bed
```

```
grep  “chr19” narrow_peaks_G1E_megak.bed | wc -l
```

We see 50 overlapping sites for TAl1 binding on both cells types, most of them being on chromosome 19 (30 or 60%) including upstream GATA1 promoter.

For G1E only:

```
bedtools intersect -a G1E_peaks.narrowPeak -b Megakaryocyte_peaks.narrowPeak -v > narrow_peaks_G1E_only.bed
```

For megakaryocytes only:

```
bedtools intersect -b G1E_peaks.narrowPeak -a Megakaryocyte_peaks.narrowPeak -v > narrow_peaks_megak_only.bed
```

Table of Contents

Finally using deepTools bamCompare, computeMatrix, and plotHeatmap plot signals between the two samples.

bamCompare Commands:

```
bamCompare -b1 G1E_Tal1_R1.bam -b2 G1E_input_R1.bam -o G1E_R1.bw

bamCompare -b1 G1E_Tal1_R2.bam -b2 G1E_input_R2.bam -o G1E_R2.bw

bamCompare -b1 Megakaryocyte_Tal1_R1.bam -b2 Megakaryocyte_input_R1.bam -o
Megakaryocyte_R1.bw

bamCompare -b1 Megakaryocyte_Tal1_R2.bam -b2 Megakaryocyte_input_R2.bam -o
Megakaryocyte_R2.bw
```

computeMatrix Commands:

```
computeMatrix reference-point -R G1E_peaks.narrowPeak -S G1E_R1.bw G1E_R2.bw –
referencePoint center --beforeRegionStartLength 5000 --afterRegionStartLength 5000 –
missingDataAsZero --skipZeros -out G1E_matrix

computeMatrix reference-point -R Megakaryocyte_peaks.narrowPeak -S Megakaryocyte_R1.bw
Megakaryocyte_R2.bw --referencePoint center --beforeRegionStartLength 5000 --
afterRegionStartLength 5000 --missingDataAsZero --skipZeros -out Megakaryocyte_matrix

computeMatrix reference-point -R G1E_peaks.narrowPeak -S G1E_R1.bw G1E_R2.bw --
referencePoint center --beforeRegionStartLength 5000 --afterRegionStartLength 5000 --
missingDataAsZero --skipZeros -out G1E_matrix

computeMatrix reference-point -R Megakaryocyte_peaks.narrowPeak -S Megakaryocyte_R1.bw
Megakaryocyte_R2.bw --referencePoint center --beforeRegionStartLength 5000 --
afterRegionStartLength 5000 --missingDataAsZero –skipZeros -out Megakaryocyte_matrix
```

plotHeatmap Commands:

```
plotHeatmap --matrixFile G1E_matrix --samplesLabel "G1E R1" "G1E R2" -out
G1E_plotHeatmap.png --heatmapHeight 10

plotHeatmap --matrixFile Megakaryocyte_matrix --samplesLabel "Megakaryocyte R1"
"Megakaryocyte R2" -out Megakaryocyte_plotHeatmap.png --heatmapHeight 10
```

Table of Contents

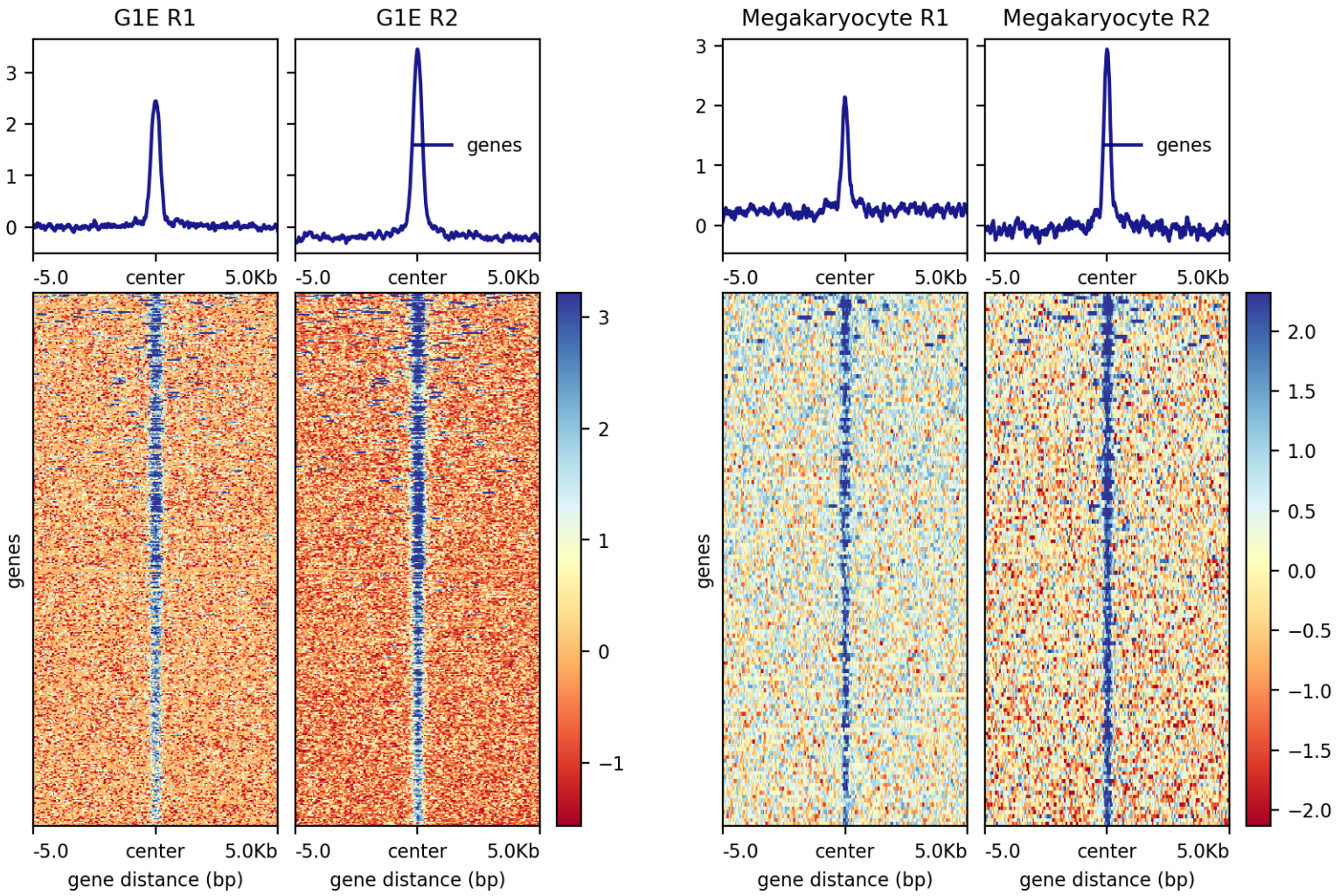
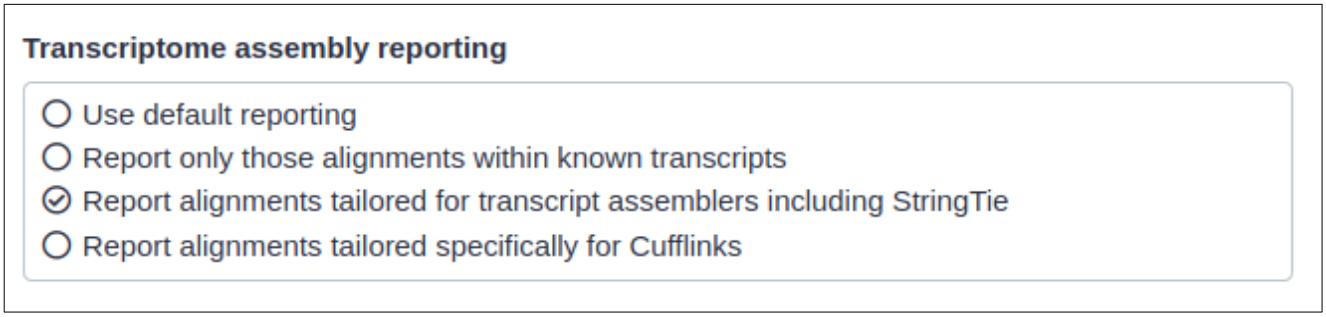


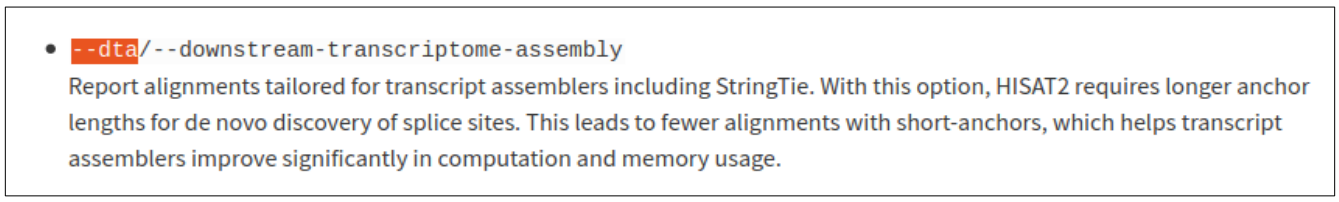
Table of Contents

4.12. Week 12 - RNA-seq: Galaxy

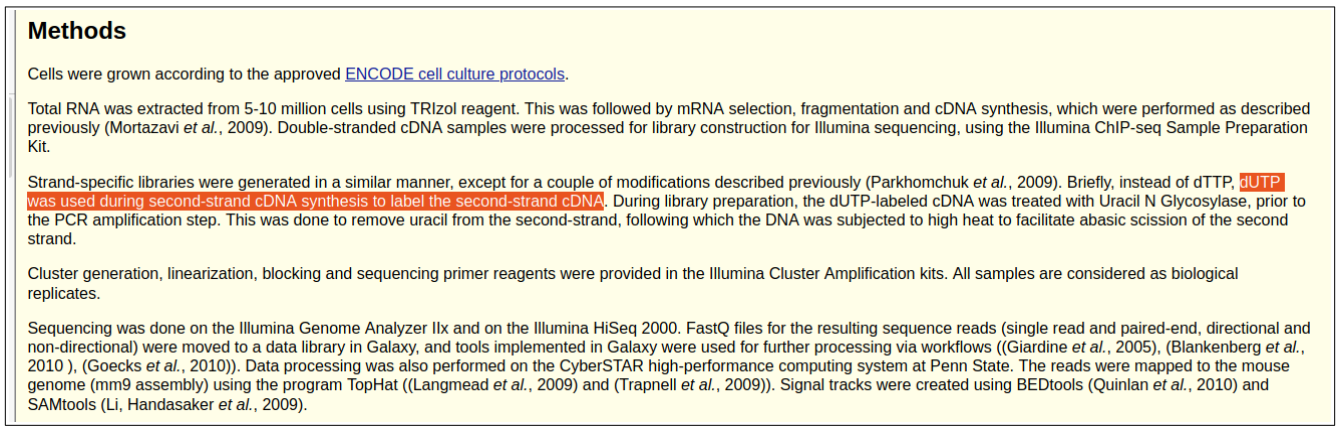
Using the RNA-seq files for the same cell lines from the same study in the previous ChIP-seq section from Wu et al., utilize Galaxy workflow for transcriptome assembly and transcript counts to analyze differential gene expression. Tools used in order of RNA-seq workflows include fastqc, trimmomatic, HISAT2, StringTie2, GffCompare, subread's featureCounts, and DESeq2.

Aligning with HISAT2 followed by StringTie workflow requires choosing output for downstream analysis by StringTie. This is equivalent to passing the "- - dta" parameter for the command-line version.





The RNA strandness parameters given in the Galaxy tutorial are incorrect. Looking at the library methods protocol for this study in UCSC Genome Browser we see the dUTP synthesis method was used. According to Biostar's moderator Devon Ryan at this post, this method used for paired-end sequencing is reverse stranded.



Methods

Cells were grown according to the approved ENCODE cell culture protocols.

Total RNA was extracted from 5-10 million cells using TRIzol reagent. This was followed by mRNA selection, fragmentation and cDNA synthesis, which were performed as described previously (Mortazavi *et al.*, 2009). Double-stranded cDNA samples were processed for library construction for Illumina sequencing, using the Illumina ChIP-seq Sample Preparation Kit.

Strand-specific libraries were generated in a similar manner, except for a couple of modifications described previously (Parkhomchuk *et al.*, 2009). Briefly, instead of dTTP, dUTP was used during second-strand cDNA synthesis to label the second-strand cDNA. During library preparation, the dUTP-labeled cDNA was treated with Uracil N Glycosylase, prior to the PCR amplification step. This was done to remove uracil from the second-strand, following which the DNA was subjected to high heat to facilitate abasic scission of the second strand.

Cluster generation, linearization, blocking and sequencing primer reagents were provided in the Illumina Cluster Amplification kits. All samples are considered as biological replicates.

Sequencing was done on the Illumina Genome Analyzer IIx and on the Illumina HiSeq 2000. FastQ files for the resulting sequence reads (single read and paired-end, directional and non-directional) were moved to a data library in Galaxy, and tools implemented in Galaxy were used for further processing via workflows ((Giardine *et al.*, 2005), (Blankenberg *et al.*, 2010), (Goecks *et al.*, 2010)). Data processing was also performed on the CyberSTAR high-performance computing system at Penn State. The reads were mapped to the mouse genome (mm9 assembly) using the program TopHat ((Langmead *et al.*, 2009) and (Trapnell *et al.*, 2009)). Signal tracks were created using BEDtools (Quinlan *et al.*, 2010) and SAMtools (Li, Handasaker *et al.*, 2009).

Table of Contents

10 Which setting to use depends on how the sequencing libraries were prepared, though likely one of `no` or `reverse` are correct for anything sequenced in the past few years. `no` is appropriate for datasets that are not strand-specific. `reverse` is correct of libraries made with a dUTP-based method (this is the most common method). `yes` is appropriate for older datasets where read #1 indicates the strand of he original fragment sequenced. When in doubt, you can run all 3 on one sample. If the counts from `yes` and `reverse` are roughly equal for most genes then the dataset is unstranded (i.e., `no` is the correct setting). If either `yes` or `reverse` produces much higher counts than the other then the appropriate setting is the one giving the higher counts. This will pretty much always be `reverse` these days. You can also just ask the people who did the library prep. what kit they used or whether it was dUTP-based.

We can confirm this by running RseQC infer_experiment.py and we see that the reads correspond to number 2.

```
This is PairEnd Data
Fraction of reads failed to determine: 0.0025
Fraction of reads explained by "1++,1--,2+-,2-+": 0.0059
Fraction of reads explained by "1+-,1-+,2++,2--": 0.9915
```

Meaning that library type is “fr-firststrand” which is important for Subread’s featureCount if you are passing the paired-end parameter with the strandSpecific argument (see below).

2. 1+-,1-+,2++,2–

- read1 mapped to ‘+’ strand indicates parental gene on ‘-‘ strand
- read1 mapped to ‘-‘ strand indicates parental gene on ‘+’ strand
- read2 mapped to ‘+’ strand indicates parental gene on ‘+’ strand
- read2 mapped to ‘-‘ strand indicates parental gene on ‘-‘ strand

Read 1 on the opposite strand
+-,-+ (SE) 1+-,1-+,2++,2-- (PE)
library-type fr-firststrand
--rna-strandedness R (SE), RF (PE)
stranded --reverse)

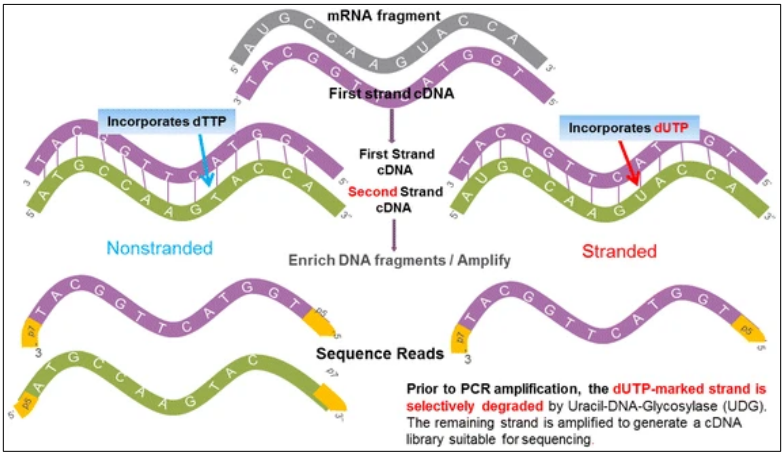
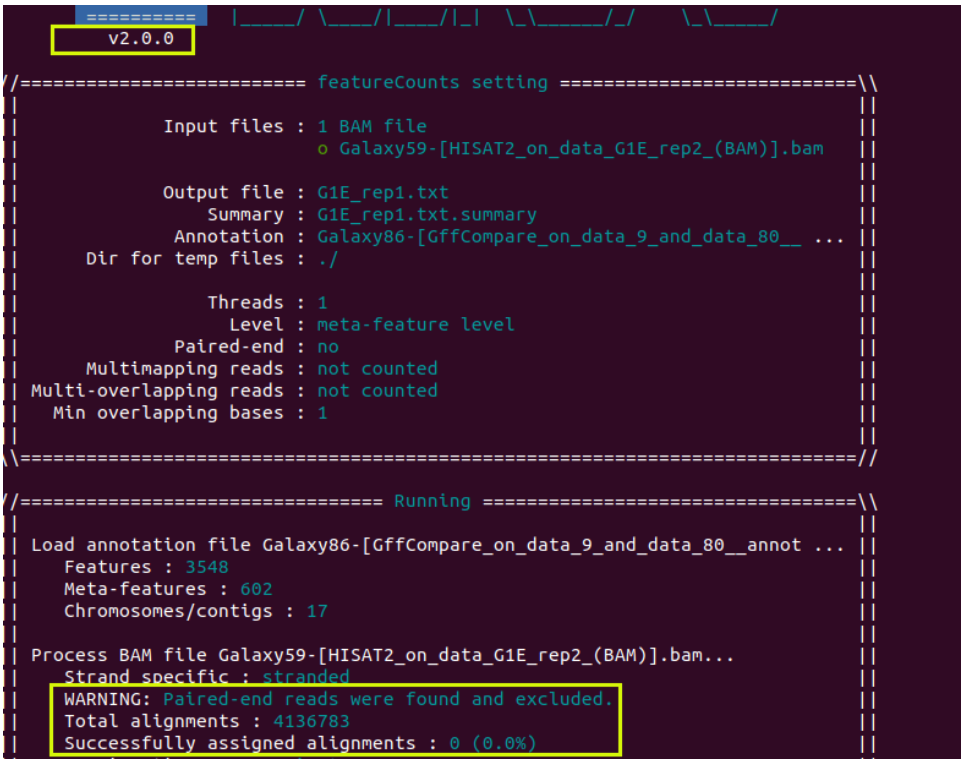


Table of Contents

Discovered that with new version of subread > 2.0.0 that for featureCounts, if data is paired then the "-p" parameter must be passed for command line and "isPairedEnd = TRUE" for Rsubread function featureCounts() if you specify the strand and this must be the correct strandness of the prepared library and alignment steps. If you incorrectly performed the alignment with HISAT2/TopHat and/or assembly with StringTie you will loose alignment assignment in the counting step due to ambiguous alignments.

Without passing "-p" to featureCounts 2.0.0 and if data is aligned as paired-end using aligner such as hisat2 we get 0 assigned alignments.



Likewise for Rsubread:

Forgetting to indicating "isPairedEnd" will get you no alignments assigned if sequencing library is paired-end.

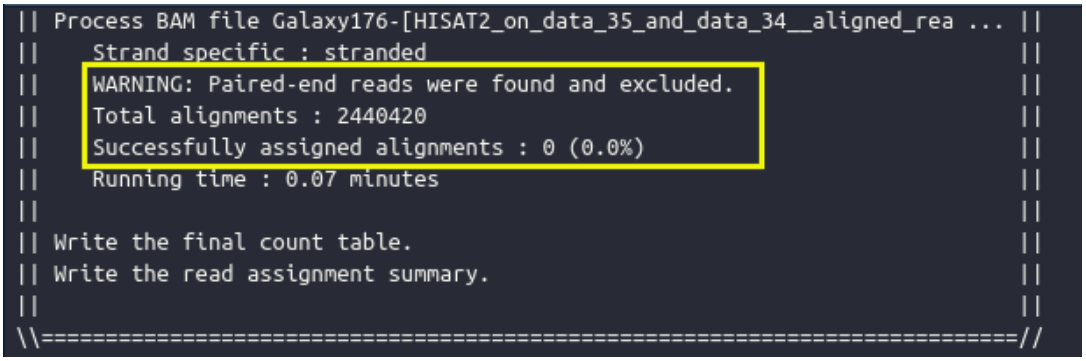
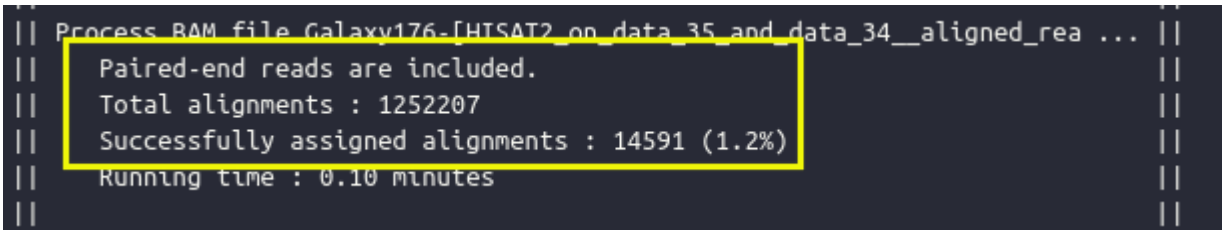
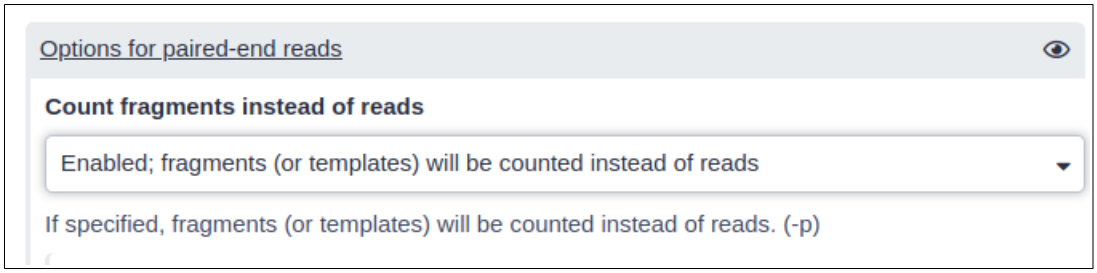


Table of Contents

While indicating unstranded will get little alignments assigned.



Instead need to make sure to select '-s' parameter in command-line, "strandSpecific" for R versions and in addition to pair-end in Galaxy select whether Stranded is "forward" or 1 for command-line and R if library and alignments were fr-secondstrand or in our case since library is fr-firststrand we select "reverse" or 2 for command-line and R.



In Galaxy the version 1.6.4 there is no "check" to exclude pair-end reads if you didn't specify pair-end as well counting forward and reverse mappings, i.e choosing "0" or "unstranded". With newer versions of Subread this won't fly.

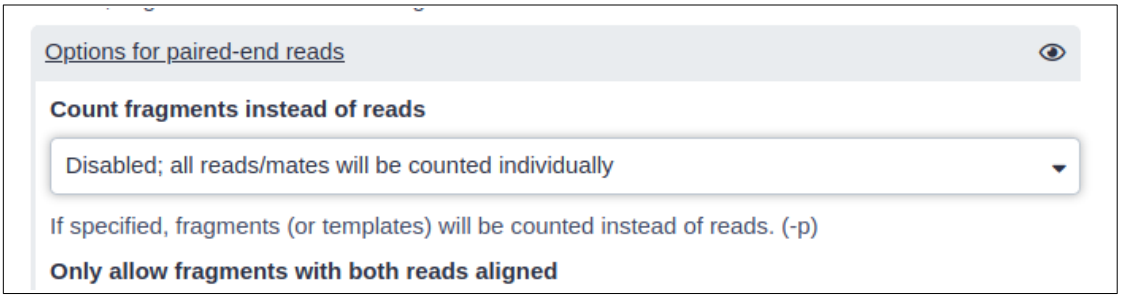
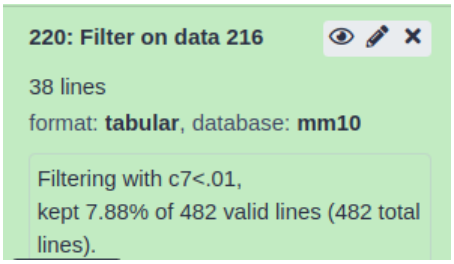
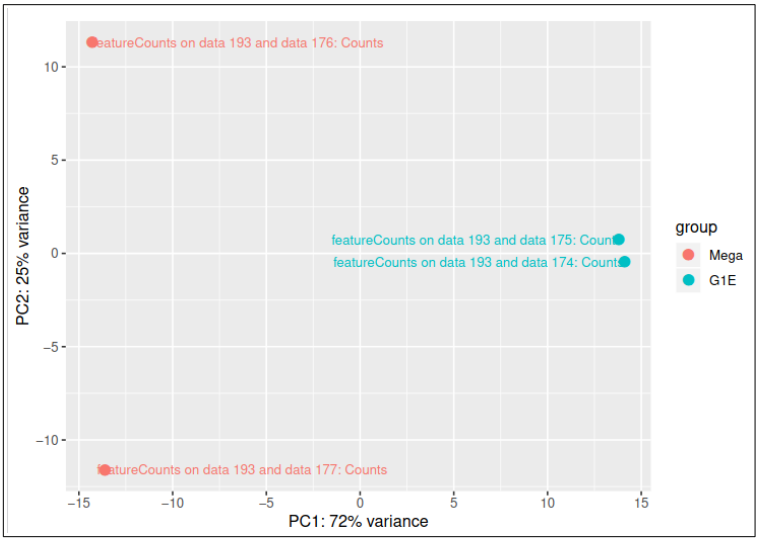


Table of Contents

Running DESeq2 showed that the replicates for G1E cells and Megakaryocytes were grouped respectively and there were 38 transcripts that had statistically signficant p-values ajusted for multiple comparisons. There was also around an even amount of transcripts that were up and down-regulated between the 2 cell types.





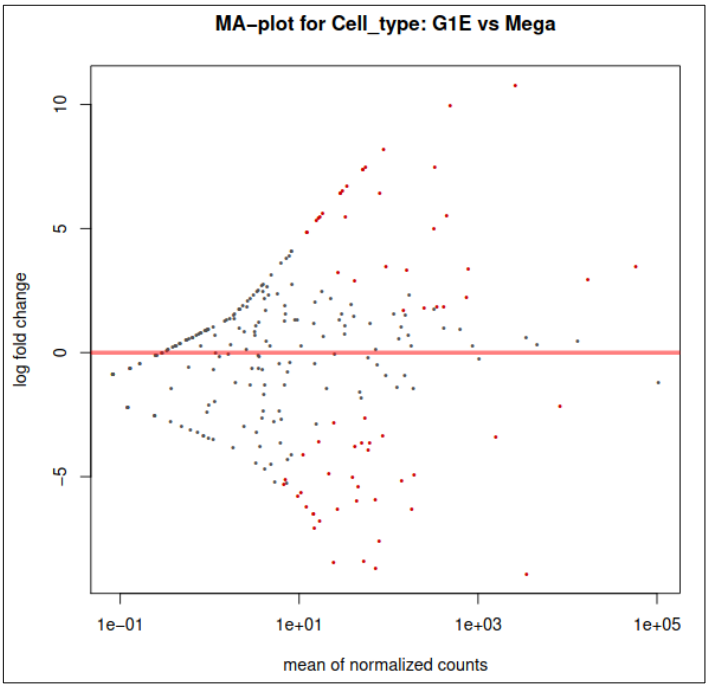


Table of Contents

Viewing the first up-regulated gene with highest confidence we see that homoebox protein Hoxb13 is expressed extremely in these G1E cells which makes sense and these cells are embryonic stem cells that proliferate as immature erythroblasts unless GATA1 expression is restored.

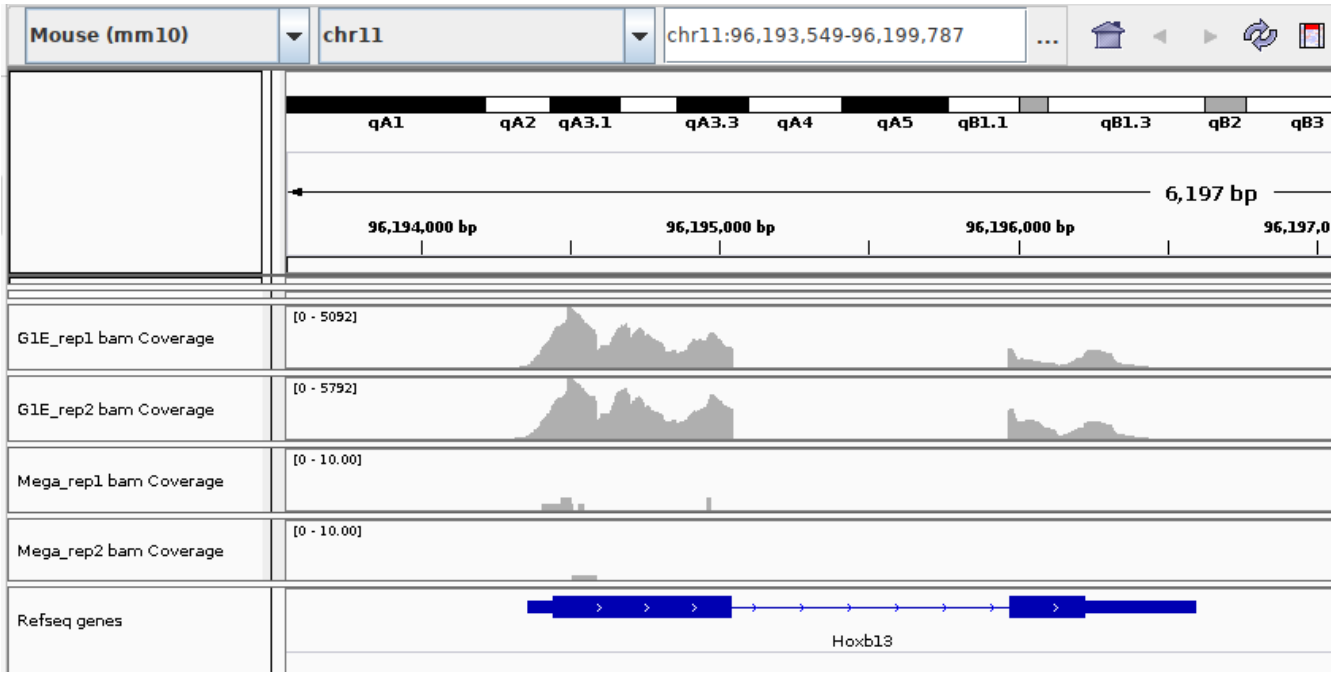


Table of Contents

4.13. Week 13 - RNA-seq analysis using command-line and R

Followed the hisat2, StringTie, and Ballgown workflow from online tutorial.

Extracting splice sites, exons and building genomic index:

```
extract_exons.py ../chrX_data/genes/chrX.gtf > chrX.exon
extract_splice_sites.py ../chrX_data/genes/chrX.gtf > chrX.ss
```

Write the samples names to file with awk to run loops:

```
awk -F, '{gsub(^"/,"")} NR>1 {print $1}' geuvadis_phenodata.csv > samples.txt
```

Read in array of sample names with loop and align to genome:

```
readarray -t mySamples < chrX_data/samples.txt
time
for id in "${mySamples[@]}"; do
	hisat2 -p 8 --dta -x chrX_data/indexes/chrX_tran \
		-1 chrX_data/samples/${id}_chrX_1.fastq.gz \
		-2 chrX_data/samples/${id}_chrX_2.fastq.gz \
		-S map/${id}_chrX.sam;
done;
```

--dta = downstream-transcriptome-assembly

-x = indexes

Sort samples using loop:

```
for id in "${mySamples[@]}";
	do samtools sort -o map/${id}_chrX.bam map/${id}_chrX.bam;
done;
```

Assemble transcripts for each sample:

```
for id in "${mySamples[@]}"; do
	stringtie map/${id}_chrX.bam -l ${id} \
	-p 8 -G chrX_data/genes/chrX.gtf \
	-o assembly/${id}_chrX.gtf;
done;
```

Table of Contents

-l is for label prefix for output transcripts. Stringtie will assembly referenced transcripts from GTF as well as novel transcripts by the assembler. These are prefix labels for the novel transcripts.

-G is the the reference GTF or GFF which is being used because we have little de novo transcript data from just the X chromosome.

Prepend folder to file names for assembly GTFs:

```
sed -i.bak 's/^/assembly\//' chrX_data/mergelist.txt
```

Merge the transcripts:

```
stringtie --merge -p 8 -G chrX_data/genes/chrX.gtf -o stringtie_merged.gtf
chrX_data/mergelist.txt
```

In the merge mode, StringTie takes as input a list of GTF/GFF files, the list in mergelist.txt

Count number of transcripts:

```
grep -v '^##' stringtie_merged.gtf | awk '$3=="transcript"' | wc -l

4598
```

We get more than in tutorial.

Compare GTFs with GffCompare:

```
gffcompare -r  chrX_data/genes/chrX.gtf -o merged  stringtie_merged.gtf
```

-r = An optional “reference” annotation GFF file.

Now prepare files for R “Ballgown” library:

```
for id in "${mySamples[@]}"; do
        mkdir ballgown/${id};
        stringtie -e -B -p 8 -G stringtie_merged.gtf \
                -o ballgown/${id}/${id}_chrX.gtf map/${id}_chrX.bam
done;
```

-e Limits the processing of read alignments to only estimate and output the assembled transcripts matching the reference transcripts given with the -G option (requires -G, recommended for -B/-b)

-B This switch enables the output of *Ballgown* input table files (*.ctab) containing coverage data for the reference transcripts given with the -G option.

After creating a ballgown object in R we can see some of the expression counts as FPKM (fragments per kilobase of exon model per million reads mapped).

```
bg.chrX <- ballgown(dataDir = "ballgown", samplePattern = "ERR",
                    pData = pheno_data)
head(gexpr(bg.chrX), 3)
```

Just as will microarrays we filter out genes/transcripts with low variance as equal expressed genes (EEGs) should not differ much among treatment groups (Hackstadt and Hess, 2009).

```
bg.chrX.filt <- subset(bg.chrX, "rowVars(texpr(bg.chrX)) > 1")
```

Get statistics on FPKM differential expression significance and fold-change:

```
results.transcripts <- stattest(bg.chrX.filt,
                                feature = "transcript",
                                covariate = "sex",
                                getFC = T, meas = "FPKM")
```

Create dataframe with gene names and filter for significance. We see that with males Xist and Tsix are negatively differentially expressed which makes sense as they do not require X chromosome inactivation.

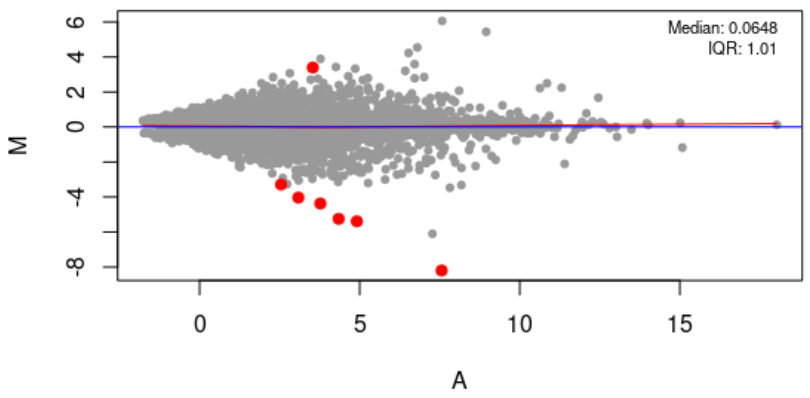
```
results.transcripts.2 <- data.frame(geneNames = geneNames(bg.chrX.filt),
                                    geneIDs = geneIDs(bg.chrX.filt),
                                    results.transcripts)
results.transcripts.2 %>% filter(qval < .05)
```

	geneNames	geneIDs	feature	id	fc	pval	qval
1107	KDM6A	MSTRG.269	transcript	1107	0.04821989	1.192030e-05	6.436964e-03
2308	TSIX	MSTRG.519	transcript	2308	0.10240589	2.677989e-06	1.735337e-03
2309	.	MSTRG.520	transcript	2309	0.02383869	1.139096e-06	9.226678e-04
2310	XIST	MSTRG.520	transcript	2310	0.00340101	4.943179e-11	1.601590e-07
2311	.	MSTRG.520	transcript	2311	0.02630457	6.121697e-08	9.917149e-05
2312	.	MSTRG.520	transcript	2312	0.06098885	4.297798e-07	4.641622e-04
2342	.	MSTRG.529	transcript	2342	10.51988169	1.038361e-04	4.806128e-02

Table of Contents

Create MA plot with affy package:

```
idx <- which(results.transcripts.2$qval < .05)
library(affy)
ma.plot(log2(results.transcripts.2$mean), log2(results.transcripts.2$fc),
        col="#999999", pch=19, cex=0.7)
points(log2(results.transcripts.2$mean)[idx], log2(results.transcripts.2$fc)[idx],
       col="#FF0000", pch=19)
```



And volcano plot:

```
plot(log2(results.transcripts.2$fc), -log10(results.transcripts.2$pval),
     col="#999999", pch=19, cex=0.7)
points(log2(results.transcripts.2$fc)[idx], -log10(results.transcripts.2$pval)[idx],
       col="#FF0000", pch=19)
```

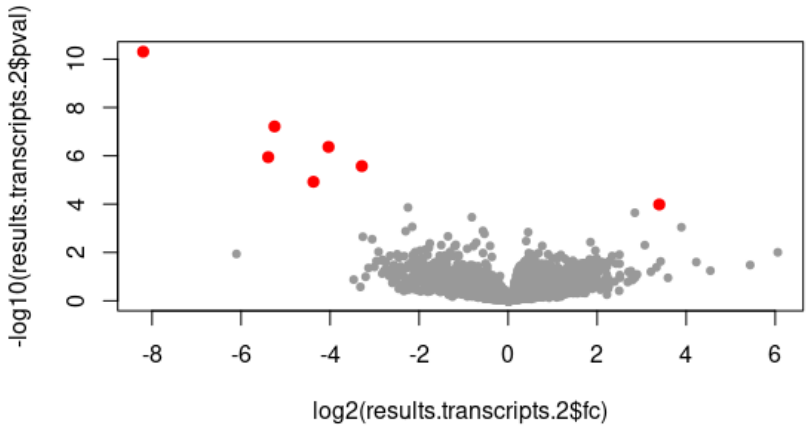


Table of Contents

5. SNP Project

SNP Choice

The HGVS for the SNP variant that we chose, rs1799966, for the hg38 assembly is NC_000017.11:g.43071077T>A[1]. The gene is the BRCA1 DNA repair associated gene and is located on the minus strand with a cytogenetic location of 17q21.31[2]. The primary isoform 1 transcript variant is 7,088 nt long and the most observed mutation, occurring at mRNA position 4950 in exon 15 of isoform 1,[3] is mutated from adenine to thymine responsible for the missense mutation from wild-type serine [AGT] to cysteine [TGT] at location 1613 of the 1863 residue length protein[4]. rs1799966 is associated with the age of diagnosis of hereditary breast cancer[5]. There have also been reports that this SNP affects chemotherapy response and survival of non-small cell lung cancer[6]. The inspiration for choosing this SNP comes from the paper “Next Generation Sequencing Reveals High Prevalence of BRCA1 and BRCA2 Variants of Unknown Significance in Early-Onset Breast Cancer in African American Women” (PMID 28439188)[7]. Another notable paper that has more extensive statistical analysis on rs1799966 is “Association between BRCA1 polymorphisms rs799917 and rs1799966 and breast cancer risk: a meta-analysis” (PMID 30832521)[8].

Links to Resources for above.

We utilized a variety of tools including but not limited to SIFT, Poly-Phen2, NCBI Genome Data Viewer, UCSC Genome Browser, Ensembl variation, dbSNP, ClinVar, UCSC Variant Annotation Integrator, Galaxy, VCFlib, bowtie2, freebayes, and samtools to perform SNP analysis on rs1799966 and within the BRCA1 gene.

Abstract

The BRCA1 gene codes for large E3 ubiquitin ligase protein of 1863 residues in length and weighing approximately 208 kDa. In addition to tagging proteins with Lys-6 polyubiquitination for degradation BRCA1 also is a vital component of cellular complexes responsible for repair in DNA mutation and DNA damage and in doing so is a tumour suppressor gene. Much research has been performed and collected on genetic variants in both BRCA1 and BRCA2 genes and it has come to fruition that despite some genetic variants in these genes being the cause for increased risk in breast, cervical and colon cancers, most variants are not pathological. We choose to look at a small sample from the breast cancer study of benign breast tumors (BBB) at Northwestern University, available on NCBI GEO at project PRJNA561179, specifically to run genetic variant analysis on whole exome sequencing (WES) to determine if a specific variant, rs1799966 c.4837A>T, would be the cause for increased risk for transition of BBB to breast cancer. Our null hypothesis based on previous medical research for our

variant indicating benign pathology is that there will be no difference between the selected control of BBBs that did not form breast cancer with those that develop breast cancer at least 1 year after biopsy with respect to having the c.4837A>T variant. Based on our findings, there showed no statistical significance to be able to reject our null hypothesis therefore agreeing with prior research findings.

Link to Final Paper, Supplemental, and Code.

Table of Contents

6. “Be a Teacher. Please, please, please be a teacher... Even if you’re not a teacher, be a teacher” -Tim Minchin

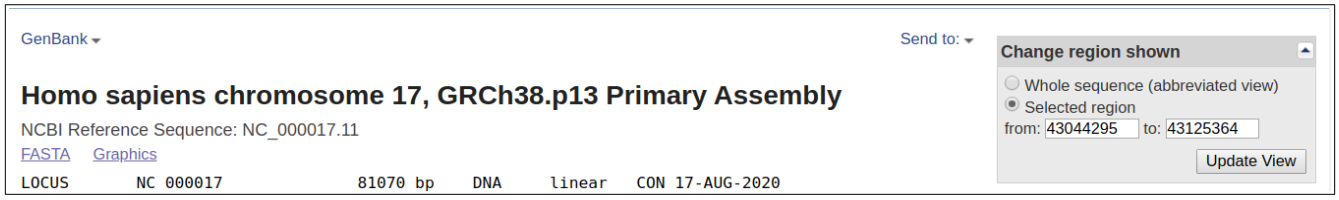
This section, maybe the longest one of them all, is inspired by the commencement speech by Tim Minchin to give a list of experiences where I could act as teacher to help my fellow students. This includes discussions, threads on slacks, and e-mails. I also include links to 3 of my PowerPoint presentations in which I felt a lot of passion during presenting those to my fellow classmates. Each entry or teaching experience is proceed by a letter starting with A) and going on to AA), AB)...BC) and so forth. Each entry briefly describes what the experiences entails.

A) During the course project, my teammate asked about why he/she was not able to find the join(..) statements for the CDS in a GenBank entry with the prefix NM. Here is my response:

Sometimes NM entries don't have the join. I saw this happen often or it will list the exons all as separate features so you can extract the exons however Biopython can do this with their parsing module with joins so separate features is not necessary. Mostly it's the "Gene" Genbank record that has the joins. So this mRNA entry has it here:
go to the

```
db_xref="GeneID:672"
```

in NM_007294.4. Then go to Genomic regions, transcripts, and products and select the GenBank link. It defaults to the region in genome:(edited)



Then scroll down to CDSs (there are 5 of them) and confirm its the correct transcript variant in note:

```
CDS             join(1269..1348,9586..9639,18832..18909,20409..20497,
                21104..21243,25485..25590,28076..28121,29443..29519,
                30505..33930,34333..34421,42790..42961,48751..48877,
                50844..51034,54127..54437,57670..57757,61414..61491,
                61992..62032,68230..68313,74248..74302,76171..76244,
                77662..77722,79563..79687)
                /gene="BRCA1"
                /gene_synonym="BRCAI; BRCC1; BROVCA1; FANCS; IRIS; PNCA4;
                PPP1R53; PSCP; RNF53"
                /note="isoform 1 is encoded by transcript variant 1;
                Derived by automated computational analysis using gene
                prediction method: BestRefSeq."
                /codon_start=1
                /product="breast cancer type 1 susceptibility protein
                isoform 1"
                /protein_id="NP_009225.1"
                /db_xref="CCDS:CCDS11453.1"
                /db_xref="Ensembl:ENSP00000350283.3"
                /db_xref="GeneID:672"
```

Table of Contents

B) In discussions for connecting to a remote server at the University, our professors asked what are ways to download information. After realizing BlackBoard would require UN and PW here is my response:

Hey All,
FYI for peer collab unit 2. Under Part Four if you want to upload files to the server from blackboard, you need to provide wget with username and password to access downloads from blackboard. So for example after you make directories for "unit2" and subdirectory for "fastas" (optional) you can get the fastas for S. malt by the code below with user name being your email without @jh.edu and your password:

```
wget --user bwiley4 --password <your password> 'https://blackboard.jhu.edu/bbcswebdav/pid-8336676-dt-content-rid-91961117_2/xid-91961117_2' -O smaltprt.fasta
wget --user bwiley4 --password <your password> 'https://blackboard.jhu.edu/bbcswebdav/pid-8336676-dt-content-rid-91961111_2/xid-91961111_2' -O smaltprt.fasta
```

Then you can check if it worked with this and you should see same results:

```
wc -l smaltprt.fasta
# 93 smaltprt.fasta
wc -l smalt.fasta
# 69320 smalt.fasta
```

C) I talk about here while reviewing different bacterial gene prediction programs my experience seeing different start codons being supplied.

So interestingly for part 1 of peer collab, I was seeing start codons in FGENESB of 'TTG' for 2247, 3653 start and stop (+) and also a start codon of 'GTG' 8859, 9728 start and stop (+) and I found this paper that indicates that FGENESB will use TTG as a start codon. The *brpA* gene was predicted to have different start codons using FGENESB depending on the settings used; the alternative start codon TTG (leucine) was predicted using “generic bacterial”And this page indicates that GTG along with ATG are the most common start promoters. So I did not know that programs and research has shown that start promoters can be other than ATG so it is important to check the docs for the program.

D) After hearing a lot about Augustus and downloading and trying out myself here I indicate my excitement about the tool in relation to functionality of that to another tool.

As I have read so much about Augustus I decided to download it and they have a cool script (getAnnoFasta.pl) to extract the sequences from the output which if you see on their website is a gff format output. This script is like that of glimmer "**extract**" protocol. Here is 1 of the ouput files with the exons extracted.

Table of Contents

```
>g1.t1.cds1
atgaacttgtccaagctgaagctgtcggacatcacccagggcatccagaagctcaacagaggagtgcag
>g1.t1.cds2
gtcccttgcaatgacaccagagtggcacaagtggctttcaaggatcggaag
>g1.t1.cds3
ctgtcagagcaagagctgctgtgccaggctgccacagtgctggacaacatgaccgactgcaagaaggactacgagcccctcatcaccagtctgaagagcctgcacggcatgacg
>g1.t1.cds4
aactgcccgccgagcaccgacaatgagatctacctgcggaacttcttgccggcactggggaactacacccaggcgctgtaccggcggatctcggccacagctgcaaactga
>g2.t1.cds1
atgagctcctcactgcccaccctgctggcactgctggtgctgctggccggccctggggctgtgcccacgctgtgcttacagctctcagtgccgctgatggagagcatccggatagtgaatgacatccagggagag
>g2.t1.cds2
gtttcctgcgtcaagatgaacgtgacagatatctttgcagacaataag
>g2.t1.cds3
acaaataacaaaactgagctcttatgcaaagcctcccacaattgtttgggagagccagcactgccacaagaacctgcagggtctcttcctcaacatgcgtcagctcctgaatgccagcagcacctccctcaag
>g2.t1.cds4
gcaccatgtcccacggcagcaggcaacactacttcaatggagaagttcctagcagacctacgtaccttcttccaccaactagctaaaaataagtga
```

E) At the start of the week on Biomart, I am excited to off my help as I had a lot of experience and self teaching after learning and researching Bioconductor “biomarRt” package. I received a lot of questions which was exciting.

Hi All! Welcome to week 4. I was super excited to see Ensembl and Biomart as the topic. I have been working with the "biomaRt" library in R for the past year and profusely with my database class last week. I would consider myself extremely experienced in the R package if you need help you have questions about your output.

Question 1:
Hi Brian, I have been trying the initial steps to install and get used to the biomaRt package by following the user guide provided. I selected a dataset of homosapiens and then started to build biomaRt query. So when I put in "listFilters()" it shows me error. Can you help me with the same? I will attach a picture here. Am I missing a step?

My response:
useMart() returns a "mart" object and listFilters() takes a "mart" object so you pass your mart to listFilters(). Such as listFilters(ensembl). you can see which type of parameters a function takes with ?listFilters

Question 2:
@Brian Wiley Thanks! I keep running into proxy problems even if I use Sys.setenv("http_proxy" = "http://my.proxy.org:9999"). It keeps saying Ensembl site unresponsive, trying useast mirror
Error in curl::curl_fetch_memory(url, handle = handle) :
Could not resolve proxy: my.proxy.org

My response:
you should be able to just load library and use a mart. What happens if you go to a new R script and run:

```
library(biomaRt)
human.mart <- useMart(biomart = "ENSEMBL_MART_ENSEMBL",
                      dataset = "hsapiens_gene_ensembl",
                      host = "useast.ensembl.org")
```

interestingly in 4 years of R I have never used Sys.setenv()

Question 3:

Table of Contents

Here another question Brian.... some of the filter functions are obvious but for the less obvious is there a quick way to find out about their function? How do we find a filter for a specific task? Hopefully it makes sense.…

My response:
I use grep...ALOT when looking for filters (this uses regular expressions if you want to learn more you can google regex in R). There two columns in return from listFilters(mart). They are name and description. You can search both columns, for instance:

```
filters <- listFilters(mouse.mart)
filters[grep("[Rr]ef[Ss]eq", filters$name), ]
filters[grep("[Rr]ef[Ss]eq", filters$description), ]
```

This will search the description column and return both columns (i.e. the rows since subsetting is first index [something,]) where there is either "refseq" or "RefSeq" in the name or description. You can also do it for attributes.

```
attrs <- listAttributes(mouse.mart)
attrs[grep("[Bb]and", attrs$name), ]
```

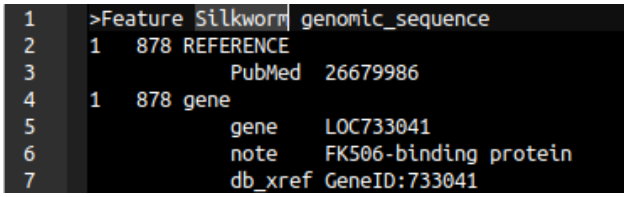
Question 4:
Hi! I keep getting this error message when I try to run getBM
> getBM(attributes=c("ensembl_transcript_id", "hgnc_symbol"), filters="entrezgene_id", values="6928", mart=ensembl)
Error: failed to load HTTP resource
Does anyone know how to fix this issue? @Brian Wiley? Thank you in advance!!

My response:
how did you call useMart()
I think there could be issues if you use the general ensembl host because of location which is why I do "host=useast.ensembl.org"
also looks like server might be down https://www.ensembl.org/

F) After email NCBI support to ask about GenBank submissions, I advised students about tbl2asn for input files to sequin. Below are the response exchanges.

Student 1:
One thing that confused me for a while was that the command doesn't include a flag for specifying the feature table name. It has to have the same name as the fasta file and be in the same folder (and end with ".tbl"). I used the simple command given in this description: https://www.ncbi.nlm.nih.gov/genbank/tbl2asn2/ which was just "tbl2asn -t template.sbt -i x.fsa" (where "x" is whatever your .fsa and .tbl files are called)

My response:
yea so its this that needs to match (the second word after "Feature" in the feature table with the first word of sequence in fasta file



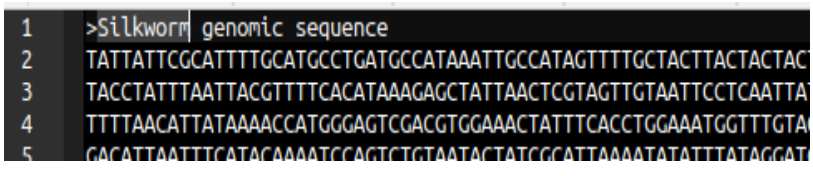


Table of Contents

example:
```
>Feature Sc_16
```
with
```
>Sc_16 [organism=Saccharomyces cerevisiae]
ttt
```

The Sc_16 matches

Student 2:
@Brian Wiley I guess I will try tbl2asn because I can't quite get Sequin to work right and nobody else seems to have input on it. If you wouldn't mind detailing the process for tbl2asn, that would be awesome. Did you tbl2asn for peer-collaborating practice problem #1 for Unit 4?

My response:
sorry just seeing this now while working on portfolio, I used the tutorials at these links below:
https://www.ncbi.nlm.nih.gov/genbank/tbl2asn2/
more on feature table format:
https://www.ncbi.nlm.nih.gov/genbank/tbl2asn2/#tbl
https://www.ncbi.nlm.nih.gov/Sequin/table.html

G) Here is a discussion I posted about prefixes for accession ids in NCBI

Thanks for the example. The graphics format shows both transcript variant 1 and 2 and their CDS respectively. It also shows RNA alignments. One is from mRNA in this GenBank entry for each variant NM_000162.5 and NM_033507.3. There are other alignments from other studies. The 'M' in NM is an mRNA entry. 'NG' prefix like in your example is incomplete genomic region, 'NC' complete genomic region, etc. See prefixes here. The NM_001354800.1 alignment is from this entry and the NM_033508.3 is from this entry. Many NG and NC entries are going to have other NM entries in the graphics under alignments I assume.

[EDIT] Just found another great resource for prefixes from Sequin.

H) This discussion talks about RNA-seq software I have used, a few annotation file formats, as well as an alignment software.

Thanks for the post. I have not come across seeing WIG file with "auxillary" probability scores or GC content. I guess you may see this with really large step sizes. The only experience I have with WIG files is in this tutorial for RSEM which is a tool for RNA-seq analysis. It comes with a script to convert BAM to WIG called 'rsem-bam2wig'. I would also like to add two more format files that we will see coming up that are closer to the representation of BED files. They are GTF and GFF3. Like BED which has many optional columns they have many columns which are required, each has 9 columns. I have posted links to the most formal and extensive descriptions I have seen for these files.
GTF: https://mblab.wustl.edu/GTF22.html
GFF3: https://learn.gencore.bio.nyu.edu/ngs-file-formats/gff3-format/
The Hisat2 tutorial for building index can also use GTF files to create exons and splice sites to incorporate in building index. Here is link: http://daehwankimlab.github.io/hisat2/howto/. It gets the GTF for human Hg38 assembly here http://ftp.ensembl.org/pub/release-84/gtf/homo_sapiens/

Table of Contents

I) Instead of writing a post on reading for one of the modules, I decided to post an applied summary of the material that combines the use of Galaxy, UCSC Genome Browser and IGV.
This is posted under Week 5 which was the most appropriate location for it.

J) For a particular exon of a gene we were to list all the SNPs listed in the dbSNP 147 database for the hg19 assembly. A fellow student was getting way more SNPs then myself and another student.

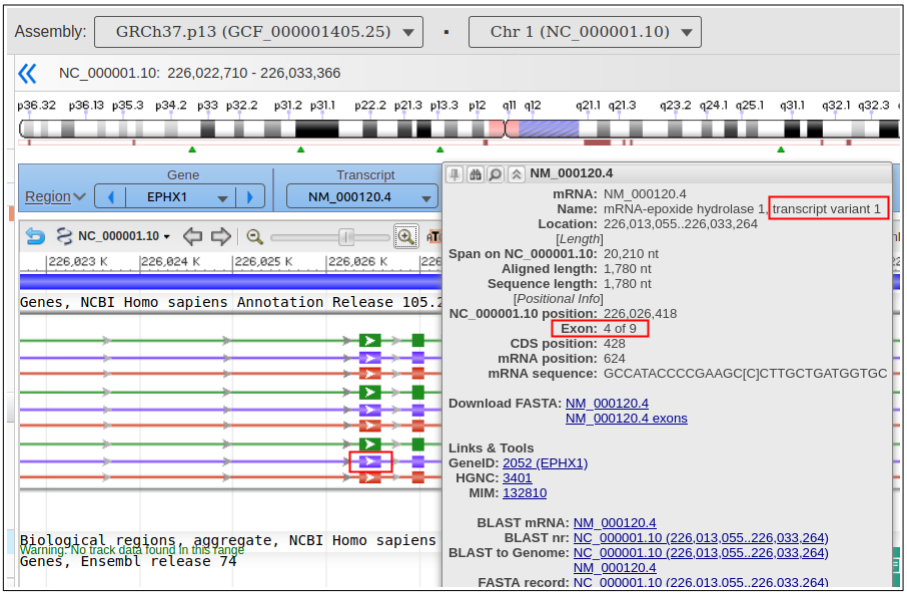
Student 1:
For part 3, writing all the SNPs that are in the exon, does anyone have a faster way than just writing them all down manually? There are a lot of SNPs and it's going slowly, I keep losing my place and it's very tedious. I tried exporting features to a BED file, but when I do that it just has the ranges, with no names. I also tried going to Ensembl and exporting the SNPs from dbSNP in the 4th exon of the gene to an excel file. It seems like this basically matches up but I wish there was some way to verify...

My initial response to student 1:
I used UCSC "All SNPs(147)" on the region for transcript variant 1 exon #4 coordinates for the hg19 asssembly. Exported to table, read into R and printed with paste(data$variant_ids, sep=",")

Student 2:
Hey, how do you specify the exon? I just put in the location and got 204 results in the table browser

My response to student 2:
I looked at coordinates for exon #4 for the main transcript variant 1. If you zoom in close enough in NCBI Genome Viewer you can hover to see exons. Then you can either zoom in more (to the base level) to get the coordinates

Table of Contents



Alternatively you can go to the GenBank record: NC_000001.10 (226,013,055..226,033,264) > Select full view > Look for transcript 1 > go to the 4th mRNA join and add those positions to first position of gene

```
mRNA            /db_xref="MIM:132810"
                join(<1..191,3372..3559,6426..6606,13301..13528,
                13864..13993,14476..14684,17013..17121,19145..19270,
                19793..20210)
                /gene="EPHX1"
                /gene_synonym="EPHX; EPOX; HYL1; MEH"
                /product="epoxide hydrolase 1, microsomal (xenobiotic),
                transcript variant 1"
                /note="Derived by automated computational analysis using
                gene prediction method: BestRefSeq."
                /transcript_id="NM_000120.3"
                /db_xref="GeneID:2052"
                /db_xref="HGNC:HGNC:3401"
                /db_xref="MIM:132810"
```

Student 2:
My bad, I misspoke. I got 225997776-226033264 from IGV but when I put that region into the table browser, I get way more than 44 SNPs

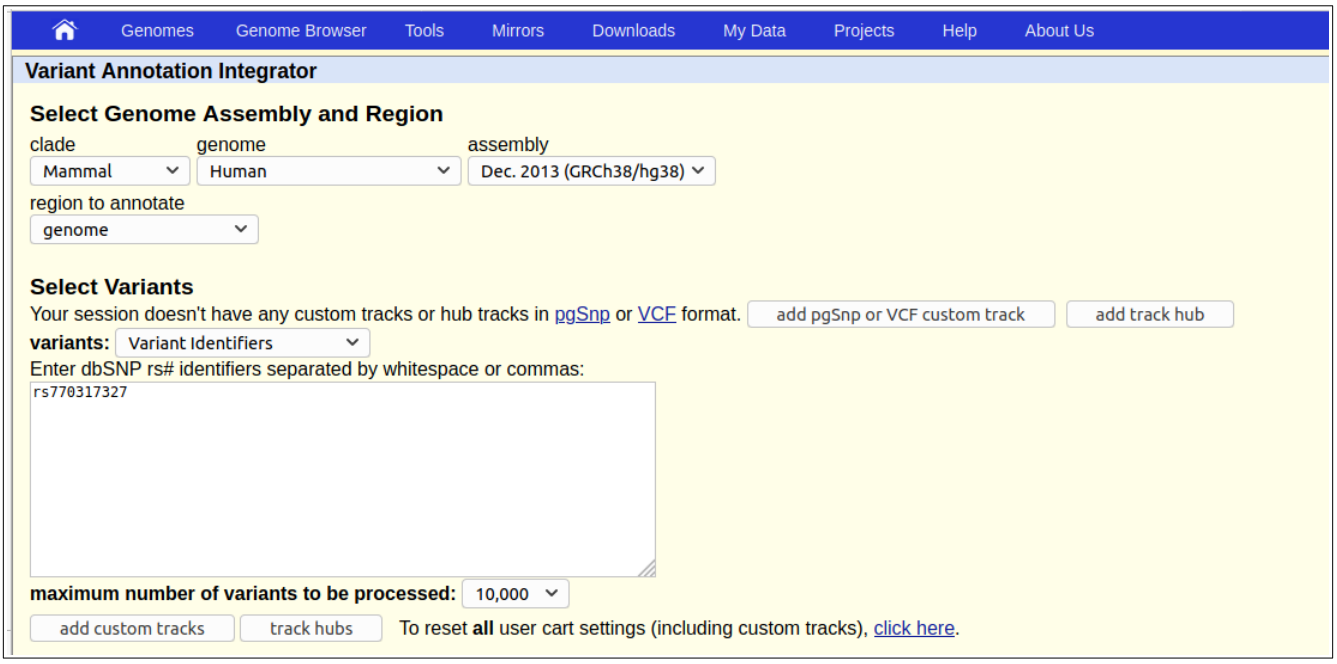
My response:
I see you hovered over the Gene in IGV. Try looking a little more closely and also see how many NM_** there are? I usually check the shortest NM first to confirm its transcript variant 1

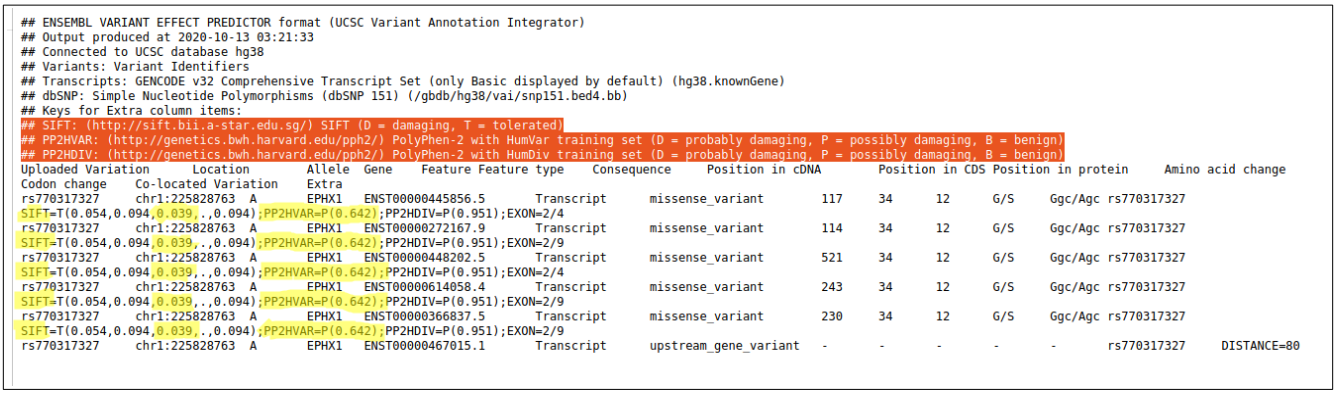
Student 2:
oh wow, the collapsed view messed me up. I didn't realize the transcripts were all on top of each other and the first set of coordinates in IGV are for the whole gene. I got it now, Thank you!

Table of Contents

K) Slack post about Variant Annotation Integrator tool at UCSC

Also check out this COOL tool from UCSC. It's under "Tools > Variant Annotation Integrator". You can search by variant ID such as rs770317327 for question 2b and you will see the predictions for SIFT, PolyPhen2 using HumVar training set and HumDiv training set. We used this for our Part 4 of project. Can you tell which training set Ensembl uses?





```
## ENSEMBL VARIANT EFFECT PREDICTOR format (UCSC Variant Annotation Integrator)
## Output produced at 2020-10-13 03:21:33
## Connected to UCSC database hg38
## Variants: Variant Identifiers
## Transcripts: GENCODE v32 Comprehensive Transcript Set (only Basic displayed by default) (hg38.knownGene)
## dbSNP: Simple Nucleotide Polymorphisms (dbSNP 151) (/gbdb/hg38/vai/snp151.bed4.bb)
## Keys for Extra column items:
## SIFT: (http://sift.bii.a-star.edu.sg/) SIFT (D = damaging, T = tolerated)
## PP2HVAR: (http://genetics.bwh.harvard.edu/pph2/) PolyPhen-2 with HumVar training set (D = probably damaging, P = possibly damaging, B = benign)
## PP2HDIV: (http://genetics.bwh.harvard.edu/pph2/) PolyPhen-2 with HumDiv training set (D = probably damaging, P = possibly damaging, B = benign)
Uploaded Variation      Location        Allele  Gene    Feature Feature type    Consequence     Position in cDNA        Position in CDS Position in protein     Amino acid change
Codon change    Co-located Variation    Extra
rs770317327     chr1:225828763  A       EPHX1   ENST00000445856.5       Transcript      missense_variant        117     34      12      G/S     Ggc/Agc rs770317327
SIFT=T(0.054,0.094,0.039,.,0.094);PP2HVAR=P(0.642);PP2HDIV=P(0.951);EXON=2/4
rs770317327     chr1:225828763  A       EPHX1   ENST00000272167.9       Transcript      missense_variant        114     34      12      G/S     Ggc/Agc rs770317327
SIFT=T(0.054,0.094,0.039,.,0.094);PP2HVAR=P(0.642);PP2HDIV=P(0.951);EXON=2/9
rs770317327     chr1:225828763  A       EPHX1   ENST00000448202.5       Transcript      missense_variant        521     34      12      G/S     Ggc/Agc rs770317327
SIFT=T(0.054,0.094,0.039,.,0.094);PP2HVAR=P(0.642);PP2HDIV=P(0.951);EXON=2/4
rs770317327     chr1:225828763  A       EPHX1   ENST00000614058.4       Transcript      missense_variant        243     34      12      G/S     Ggc/Agc rs770317327
SIFT=T(0.054,0.094,0.039,.,0.094);PP2HVAR=P(0.642);PP2HDIV=P(0.951);EXON=2/9
rs770317327     chr1:225828763  A       EPHX1   ENST00000366837.5       Transcript      missense_variant        230     34      12      G/S     Ggc/Agc rs770317327
SIFT=T(0.054,0.094,0.039,.,0.094);PP2HVAR=P(0.642);PP2HDIV=P(0.951);EXON=2/9
rs770317327     chr1:225828763  A       EPHX1   ENST00000467015.1       Transcript      upstream_gene_variant   -       -       -       -       -       rs770317327     DISTANCE=80
```

L) Discussion posts about some tools in the samtools and bcftools packages of tools

Post 1:
I found 3 nice tidbits of ways to get information on a SAM/BAM file.

1. If you are interested in getting the unique RNAMEs, i.e. the unique chromosomes of the reference genome, you can use the 'awk' command in addition to samtool tools view. To get unique on a column using awk I found this post:

Table of Contents

https://www.commandlinefu.com/commands/view/10840/display-unique-values-of-a-column

And so you would pipe the 'samtools view' command to only see lines without the header with something like:

```
samtools view plant.sam | awk '{ a[$3]++ } END { for (b in a) { print b } }'
```

2. How do I know my SAM/BAM file is sorted?

You may want to check if you file is sorted (this is required by most variant callers), before you sort it because this can use a lot of memory. After samtools sort is called to sort a SAM/BAM file, it will include a new line (maybe overwrite existing line) in the comments at the header. So you can call samtools view with the -H parameter to see the header.

```
samtools view -H plant.sam
```

If the file is unsorted you will see a line for example from this post

@HD VN:1.0 SO:unsorted

If it is sorted like this plant.sam file it will show

@HD VN:1.3 SO:coordinate

meaning it was sorted by coordinate.

3. Some times you have a SAM/BAM file with more reference chromosomes that you wish to analyze, i.e. you have a SAM/BAM alignment file for the entire human genome what you only want chromosome 22.

See this post: https://carleshf87.wordpress.com/2013/10/28/extracting-reads-for-a-single-chromosome-from-bamsam-file-with-samtools/

First you need an index file. If you don't have one you can create one with the command with samtools index. However you need to convert bam to sam first and index only works on a binary compressed files.

```
samtools view -b plant.sam > plant.bam
```

```
samtools index plant.bam
```

gives you the file "plant.bam.bai". Now to extract just the 1 chromosome of *A. thal* you can call command (-h will include header so you keep this in the subset):

```
samtools view -h plant.bam Chr1 > plant.Chr1.sam
```

Table of Contents

Then checking the new file to confirm we just call view again with the first 10 lines:

```
samtools view plant.Chr1.sam | head -n10
```

To really confirm your new sam file is a subset you can use the word count command with lines parameter to count lines:

```
samtools view plant.Chr1.sam | wc -l
1185

samtools view plant.sam | wc -l
104154
```

If you only want a region of the 1st chromosome, we can use a familiar region format we have see with biomaRt, for instance just the first 1 mega base pair of chromosome 1 with:

```
samtools view -h plant.bam Chr1:1-1000000 > plant.Chr1_1mb.sam
```

Then we can see that we cut out only about 125 lines showing that about 90% of the reads from the first chromosome are in the the 1st mega base pairs:

```
samtools view plant.Chr1_1mb.sam | wc -l
1062
```

Post 2:
FYI I just found out that you would have to pass the binary -b parameter in the subsetting view command so that you can create the index .bai file required for viewing in IGV.

Post 3:

Thanks for this post Anonomous. There is a sister package to samtools, called bcftools in which some of those tools are used in the variant calling process. These two package in addition to htslib are all written by same people. The two most popular tools of bcftools for variant calling are:

bcftools mpileup

bcftools call

You usually use mpileup before call. You used to use samtools mpileup and now its under the bcftools package. See comment from bcftools website:

The **mpileup** command was transferred to bcftools in order to avoid errors resulting from use of incompatible versions of samtools and bcftools when using in the mpileup+bcftools call pipeline.

Below is example of protocol:

Table of Contents

Examples:

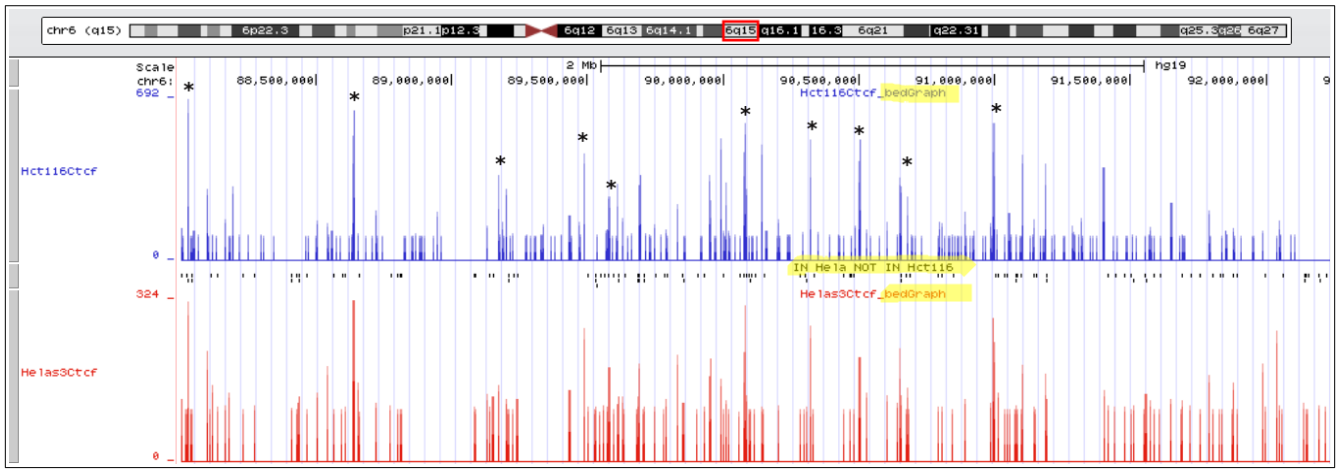
Call SNPs and short INDELs, then mark low quality sites and sites with the read depth exceeding a limit. (The read depth should be adjusted to about twice the average read depth as higher read depths usually indicate problematic regions which are often enriched for artefacts.) One may consider to add **-C50** to mpileup if mapping quality is overestimated for reads containing excessive mismatches. Applying this option usually helps for BWA-backtrack alignments, but may not other aligners.

```
bcftools mpileup -Ou -f ref.fa aln.bam | \
bcftools call -Ou -mv | \
bcftools filter -s LowQual -e '%QUAL<20 || DP>100' > var.flt.vcf
```

M) This is a Slack post I added after performing some analysis on intersecting regions between annotation files of 2 different cell lines. I wanted to let students know that sometimes you need to question the results you get and you may have to perform additional work to make more appropriate interpretations.

For part 2 of peer collab, I created a custom Python script to convert discontinuous bed files that have scores into a continuous bedGraph files so that you can view a bar graph in UCSC with the score column. Since although you may see from question 4 of part 2 there are many regions in Hela cell line not in HCT116 (there were 148 of the 211 regions in Hela cell not in HCT116 cell lines for 6q15) and think "yes CTCF binding is completely different in the cell lines because ~75% of regions of Hela are not in HCT116" but if you look at the spikes from the bedGraphs of each they really coincide. You can see all the regions in the middle track in Hela not in HCT116 and there are a lot but most of the peaks (noted with *) are very close on both tracks.

I have attached Python script if you are interested.



BedGraph Track Format

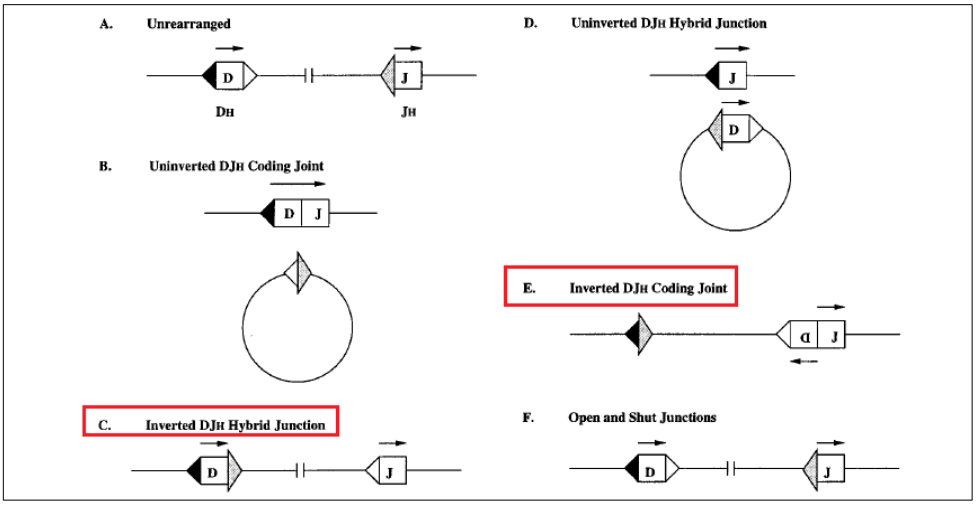
The bedGraph format allows display of continuous-valued data in track format. This display type is useful for probability scores and transcriptome data. This track type is similar to the wiggle (WIG) format, but unlike the wiggle format, data exported in the bedGraph format are preserved in their original state. This can be seen on export using the table browser. For more details on data compression in wiggle tracks see the notes section of the wiggle **track description page**. If you have a very large data set and you would like to keep it on your own server, you should use the **bigWig** data format. In fact, an attempt to load a bedGraph custom track over 50,000,000 lines will result in an error message, but can be addressed by turning the bedGraph into a bigWig (see **Example 3**). Note that bedGraph files cannot easily be converted to wiggle files; converting bedGraph to bigWig and using `bigWigToWig` will return the original bedGraph file.

Table of Contents

The simple script bedTobedGraph.py is only for sorted bed file with no overlapping regions where column 5 is a score column.

N) Here I add a post to discussion in which in many scenarios we associate chromosomal inversions with mutation and disease while in the immune this system this is occurring frequently and is necessary for diverse immunity.

Chromosomal inversions are not always considered to be associated with disease or mutation. In fact they provide a critical (crucial?) source of variation in the immune system during maturation of B and T cells in which BCR and TCR genes undergo V(D)J recombination. In molecular biology we learn of three types of recombination, more formally conservative site specific recombination (CSSR): 1) deletion, 2) duplication (insertion) and 3) inversion (Watson et al., 2013). A study (a LONG time ago) of VDJ recombination of the Heavy chain IgH of the BCR, probably the most studied gene in Cancer from a recombination perspective, in mouse indicated that "There are approximately 20,000 inverted DJH coding joints and inverted DJH hybrid junctions per day 16 fetal liver" (Sollbach et al., 1995). Below shows a summary described by authors research indicating two possible DJH recombination events 1) uninverted (coding joint) DJH coding points and 2) inverted DJH coding points.



1. Watson, J. D., Baker, T. A., Bell, S. P., Gann, A., Levine, M., & Losick, R. M. (2013). Molecular biology of the gene. 7th Edition.

2. Sollbach, A. E., & Wu, G. E. (1995). Inversions produced during V(D)J rearrangement at IgH, the immunoglobulin heavy-chain locus. Molecular and cellular biology, 15(2), 671–681. https://doi.org/10.1128/mcb.15.2.671

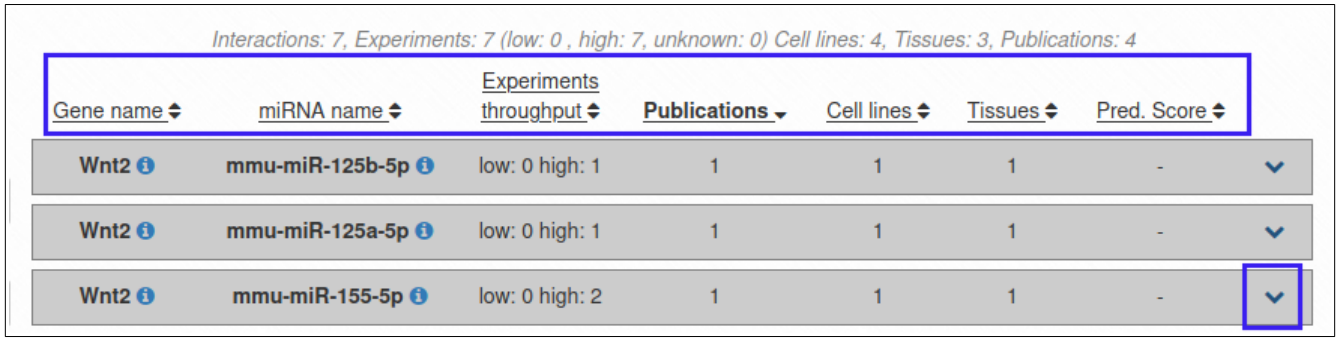
Table of Contents

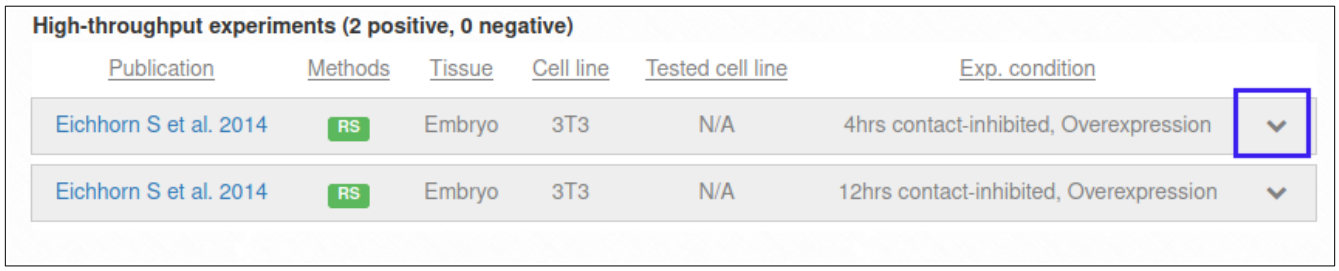
O) Discussion post for using DIANA TarBase, miRGen, and lncBase tools

We used the sister database to DIANA TarBase v.8 in my database course a few weeks back, called DIANA miRGen. I searched the other tools on the DIANA Tools packages and this is where I found TarBase. There is also another tool called lncBase for miRNA and lncRNA connections. Although it indicates in the weekly reading that Tarbase does "miRNA-mRNA targeting" it does not do any prediction with mRNA. DIANA tools are strictly focused on miRNAs and their interactions. TarBase allows searching for interactions between **miRNA and genes.** I learned something interesting in miRNA nomenclature just now in searching for the gene Wnt2. The prefix in miRNA names such as mmu- stands for mouse miRNA and hsa- stand for human miRNAs.

In all three DIANA tools below you can only search terms that are in the database. It has auto complete populate functionality when you type so you can select as you type and shows first 5 hits while typing (may be slow to populate).

Below are screen shots from TarBase views. This is good summary information then publication information such as tissue, cell line, experiment condition, method such as RNA-seq, result, validation, and source.





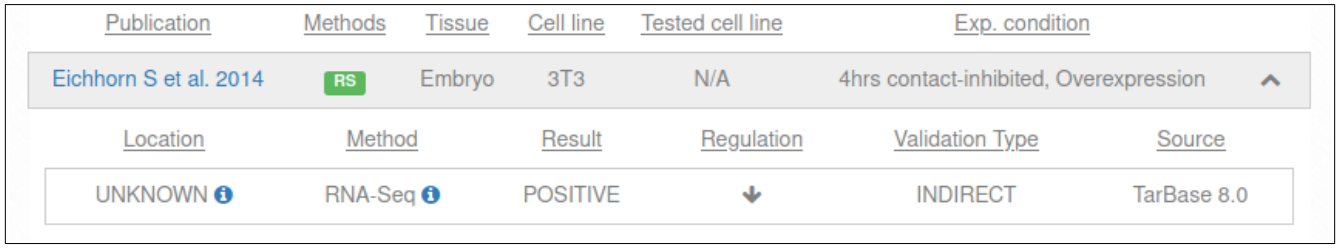
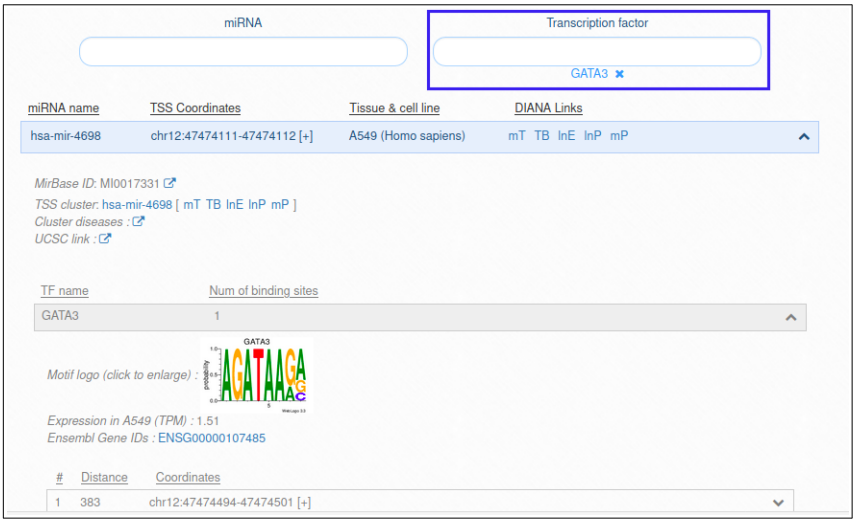
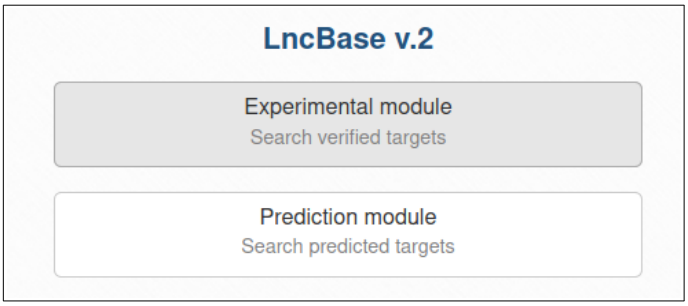


Table of Contents

miRGen allows searching for interactions between **miRNA and transcription factors**.



And as indicated above lncBase allows searching for interactions between **miRNA and lncRNA**. This database has two search methods, an Experimental module and a Prediction module. It helps to obtain lncRNA identifiers from Ensembl to post into here. For instance I obtained HOTAIR identifier ENSG00000228630 on Ensembl and searched this. There are also links to click to other DIANA databases based on your results.



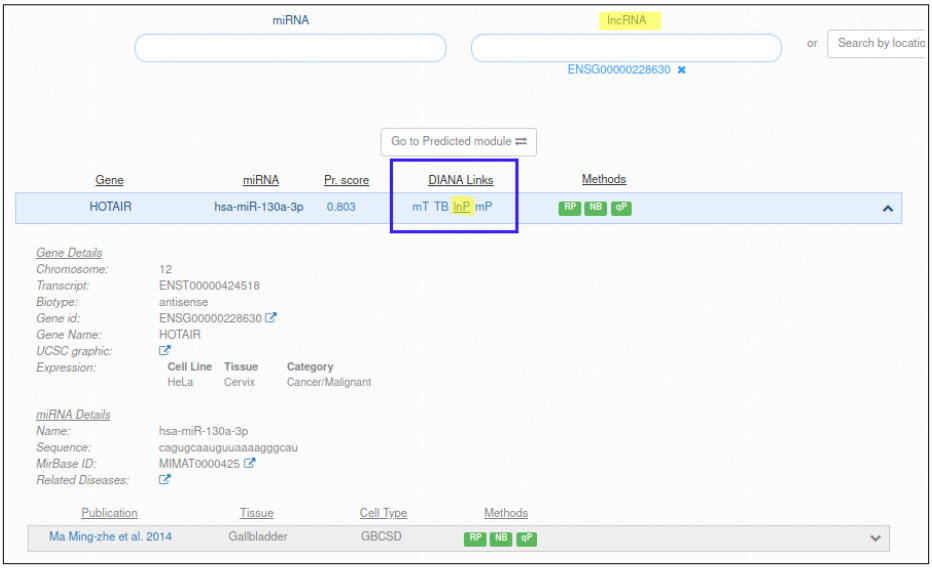
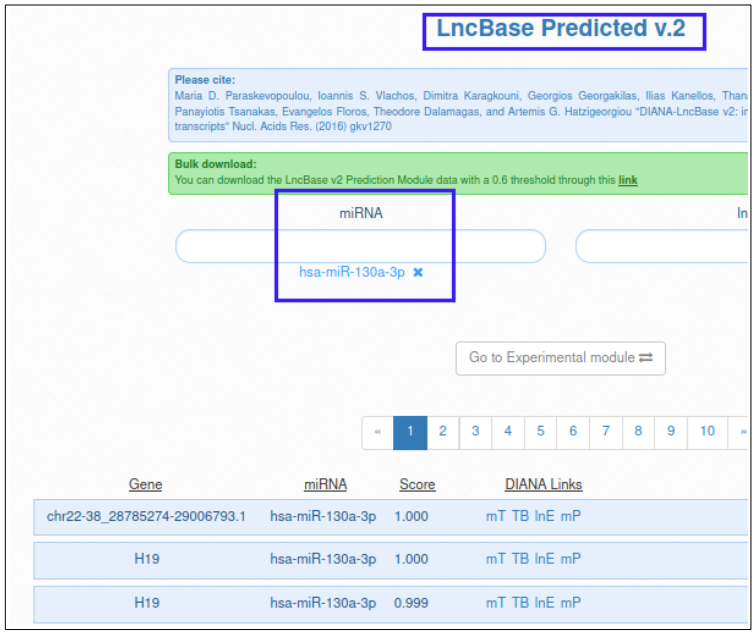


Table of Contents



P) This post show my preparation for Master's in Bioinformatics and my ability to obtain materials for self-teaching.

I did a little of mock de novo sequence assembly on the UCSD Coursera site using de Bruijn graphs followed by the Eulerian cycle algorithm which is algorithm in graph theory last year. Used together they algorithms are the meat and potatoes of de novo assembly by graphs.

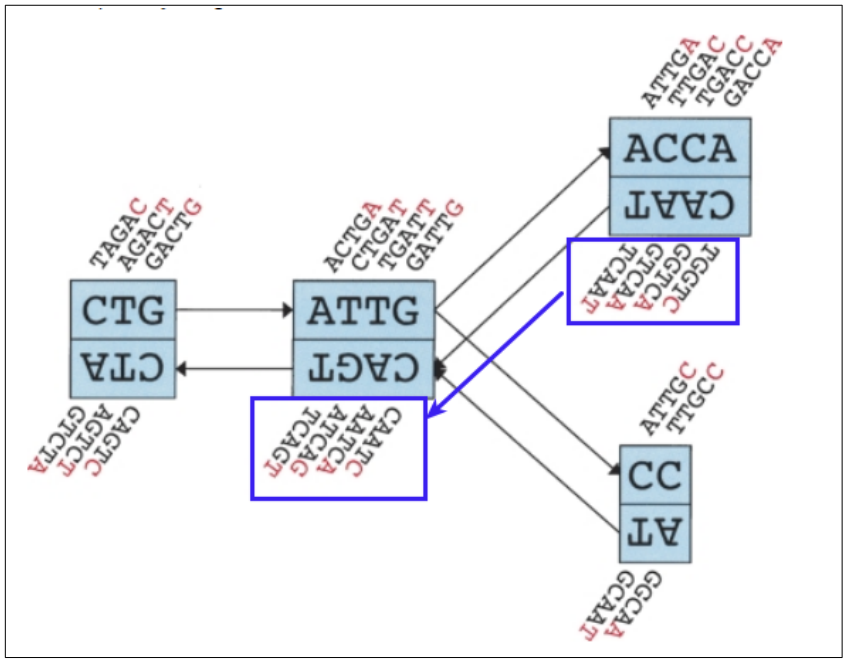


Table of Contents

You can see the de Bruijn patterns created in the image above velvet. For example we will do the blue boxes I highlighted. I tried to do bold underline bold underline to show it:
TGGTC -> **GGTC**A -> G**TCAA** -> **TCAA**T -> C**AATC** -> **AATC**A -> A**TCAG** -> **TCAG**T

When you walk the graph you should end up with the sequence below starting with:

TGGTC for the beginning and then each last letter under last node: AATCAGT gives

TGGTCAATCAGT

An example problem for de Bruijn graphs is here if you want to try it:
https://stepik.org/lesson/200/step/8?unit=8243

Q) This post portrays my ability to find material and evidence to aid in supporting my understanding of experiments.

On a reference note, trying to understand a little background on TAL1 and the design study before I start diving into the data as this is the field I will trying to go into with blood cancers. I see there are two different names for TAL1, one dealing with erythroid differentiation, seems like the background part of the study, and the other name is associated with the lymphoblast lineage. UniProt indicates the the translocation with the T-cell receptor alpha is reason for role in ALL but I am assuming this study is not looking at that at all and focusing on the non lympoblastic type leukemias. My question is for the fastqsanger files. There is G1E_input and G1E_Tal1, but in the study they indicate they use G1E cells with GATA1 knockout (GATA1 and TAL1 form heterocomplexes) and then G1E-ER4 cells which are subline that are induced by etradiol to restore GATA1 (similar to mechanism of induction by ER hormone below). So are the G1E_input fastqsanger for the G1E cells and the G1E_Tal1 fastqsanger for the G1E-ER4 cell line?

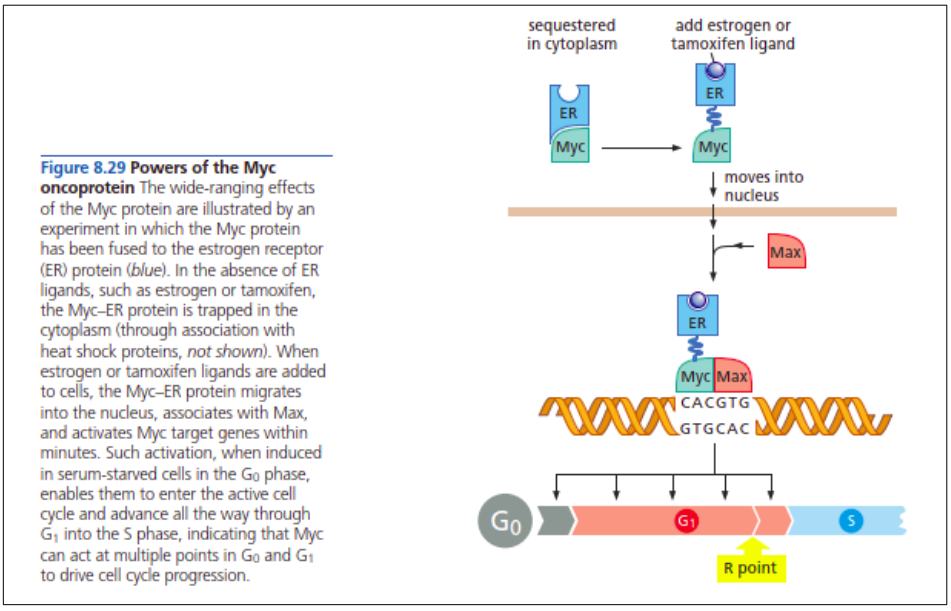


Figure 8.29 Powers of the Myc oncoprotein The wide-ranging effects of the Myc protein are illustrated by an experiment in which the Myc protein has been fused to the estrogen receptor (ER) protein (*blue*). In the absence of ER ligands, such as estrogen or tamoxifen, the Myc–ER protein is trapped in the cytoplasm (through association with heat shock proteins, *not shown*). When estrogen or tamoxifen ligands are added to cells, the Myc–ER protein migrates into the nucleus, associates with Max, and activates Myc target genes within minutes. Such activation, when induced in serum-starved cells in the G_0 phase, enables them to enter the active cell cycle and advance all the way through G_1 into the S phase, indicating that Myc can act at multiple points in G_0 and G_1 to drive cell cycle progression.

Table of Contents

From the paper it indicates "TAL1 ChIP-seq data were determined in our laboratory in **G1E cells**, G1E-ER4 + E2 (ER4)cells, Ter119+erythroblasts (Ebl) from fetal liver (Wu et al. 2011), and cultured **megakaryocytes from fetal liver** (Pimkin et al. 2014)." So the tutorial sounds like it is only using the 2 bolded cell lines above.

Now that I look at the samples on SRA looks like the input files are background without antibody (antibody says Input) and TAL1 are for the anti-TAL1. I am assuming this is just for one of the G1E cells lines and not both the G1E and G1E-ER4?

R) Two fellow students were having trouble understanding the functionality of VCFannotate tool in the VCFlib tools package.

Student 1:
I got 28 variants also using HISAT2, but then also 28 after doing the intersection with RefSeq, so I'm not sure if I did something wrong. I downloaded the chr22:0-51304566 region from UCSC Table Browser, imported the bed file into Galaxy and performed the VCFannotate using the bed file and the VCF file and still get 28?? Anybody else?

I'm still interested in knowing what I did wrong. Did you see Brian's comment about the order of the files? Perhaps I entered them backwards in the VCFsnnotate tool?

My response:
vcfannotate is a "quasi" intersect. It's really a "LEFT Join" in the SQL world if the VCF file is first, meaning you get all your first (left) entries even if there is no hit in your second file (right) entries but if hit then add to INFO column.

Student 1's response:
Thanks Brian! Again, good info, perhaps I entered my query files in reverse order...?

My response:
Sorry I meant to give the SQL Left join as an example. By default in vcfannotate the VCF file is always "First" and the annotations always to the VCF file from the input file or bed file "Second". Here is the info from vcfannotate:

```
(base) ~$ vcfannotate
usage: vcfannotate [options] [<vcf file>]

options:
    -b, --bed   use annotations provided by this BED file
    -k, --key   use this INFO field key for the annotations
    -d, --default  use this INFO field key for records without annotations

Intersect the records in the VCF file with targets provided in a BED file.
Intersections are done on the reference sequences in the VCF file.
If no VCF filename is specified on the command line (last argument) the VCF
read from stdin.
```

Table of Contents

Student 2:
I've been having a bit of an issue with the samples in Galaxy- running VCFannotate on the exons bed file and then my filtered VCF gives me a file that has the same number of lines as the filtered VCF file. I'm about to try command line bedtools intersect to see if that remedies it.

My response:
It's supposed to give you the same number of lines. It doesn't do just a bedtools intersect. It only gives a specific note in the last column of the VCF file if there is an intersect, otherwise that line is not modified. You count the number of lines by using the command line with regular expressions or you can also do intersect to count but VCFannotate is annotating the entire VCF file, its not "intersecting" two bed files.

If you wanted to can pass any key you'd like to VCFannotate, I did "--key REFGENE" and then counted with grep `"^[^#]" vcf_annotated_file.vcf | grep "REFGENE" | wc -l` to get the counts.

```
usage: vcfannotate [options] [<vcf file>]

options:
    -b, --bed   use annotations provided by this BED file
    -k, --key   use this INFO field key for the annotations
    -d, --default  use this INFO field key for records without annotations
```

S) Here I discuss my project for Epigenetics and Leukemic Stem Cells (aka Cancer Stem Cells CSCs) on a forum discussion for Epigenetics.

In my Epigenetic course we gave a presentation for our final project on "Cancer Stem Cells" also known as "Leukemic Stem Cells" where we identified Epigenetic regulation in the onset of hematopoeitic cancers. The influence for our project was Dr. Andrew Feinberg from Hopkins as well as Dr. Stephen Baylin from Hopkins (Papers below). The interesting finding from these researchers was titled the "Epigenetic progenitor model of Cancer" which is different from the typical viewed "clonal genetic model" where cancer is thought to derived from one ancestral cell with tumor suppressor mutation or oncogene progression. Instead in the Epigenitic progenitor model, irratic or abnormal epigenetic programming goes awry, either by means of tumor progenitor genes, chronic inflammation, aging, etc. leading to polyclonal expansion of undifferentiated stem cells.

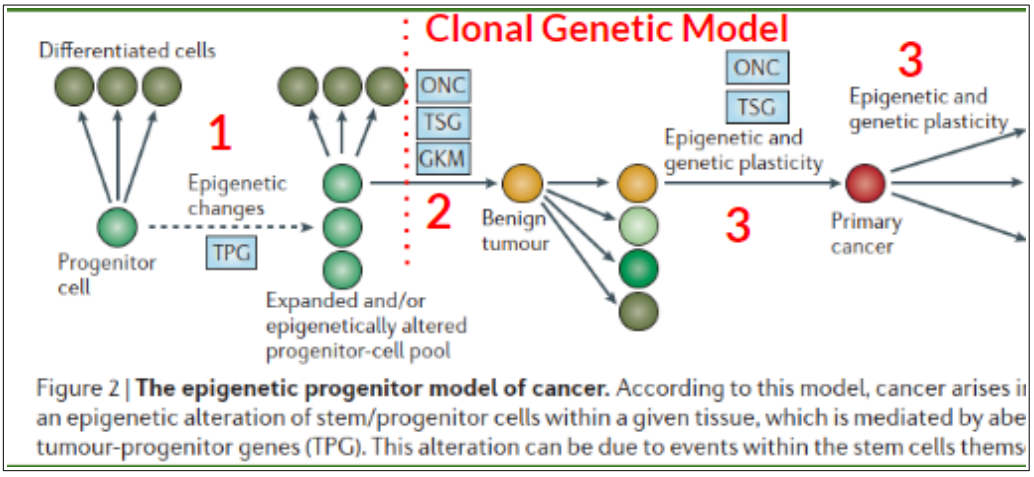


Figure 2 | **The epigenetic progenitor model of cancer.** According to this model, cancer arises i an epigenetic alteration of stem/progenitor cells within a given tissue, which is mediated by abe tumour-progenitor genes (TPG). This alteration can be due to events within the stem cells thems

Table of Contents

Figure from Feinberg.

Attractive therapies, called differentiation therapies include targets in epigenetics. For example H3K79 methylation by DOT1L activates self-renewal homeobox genes HOXA9 and MEIS1. They genes are responsible for maintenance of the proper pool level of self-renewing HSCs in the bone marrow. When DOT1L is upregulated this leads to increased marks in H3K79me which positively regulates transcription of HOXA9 and MEIS1 leading to increased level of undifferentiated HSCs in multiple leukmias such as MLL-fusion type leukemias, AML and ALL. Pinonmetostat is a DOT1L Histone Lysine Methyltransferase inhibitor that is on trial. By inhibiting DOTL1 methytransferase activity on H3K79 this leads to downstream differentiation of these leukemic stem cells to eradicate the cancer.

An interesting lab and Dana Farber in Boston works with differentiation therapy specificly with HDACi (Histone deacetylase inhibitors) in neuroblastomas.

http://stegmaierlab.dfci.harvard.edu/research.html

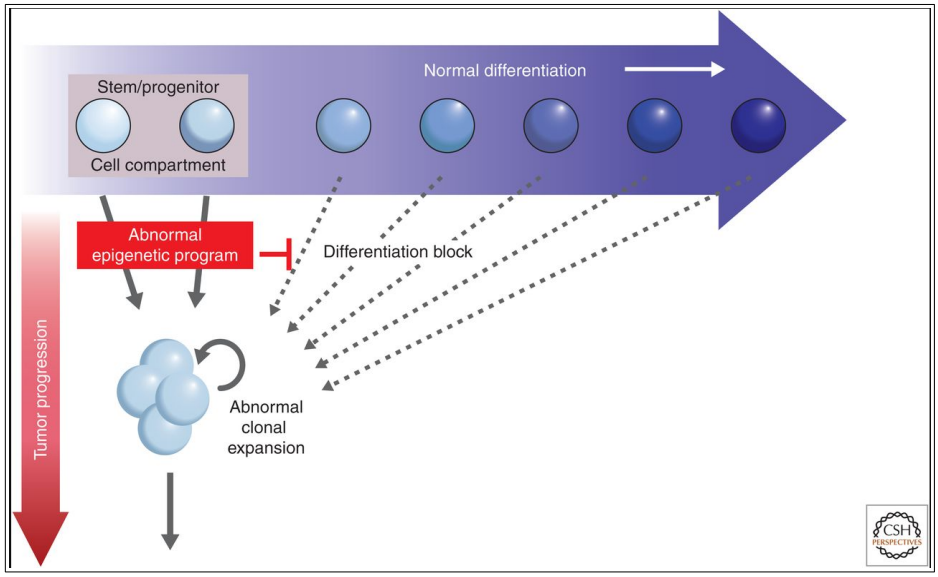


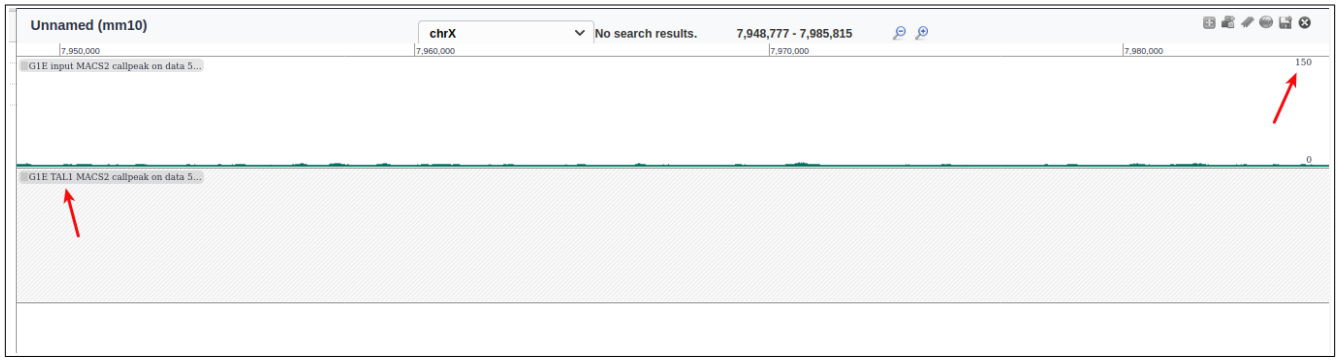
Figure from Baylin and Jones.

T) Here I look into a bug/issue on Galaxy for adding additional tracks to Trackster to an initially visualized track.

Hey All, as a reference there has been a bug in Galaxy for some time now when using the "Add tracks" function in Trackster to an initially added track. You can see the first track added in the new visualization but any track added there after will not show, basically making it unusable to visualize control vs treatment because they need to be on the same scale and not their individual scales in separate visualizations. Therefore you need to use either IGV or UCSC genome browsers or NCBI Genome Data Viewer, or other if you have found others.

Table of Contents

See response from Jenna J, main Galaxy maintainer, in first post:
https://help.galaxyproject.org/t/trackster-not-loading-dataset-with-server-indexed-database-assigned/2308/3
https://github.com/galaxyproject/usegalaxy-playbook/issues/276



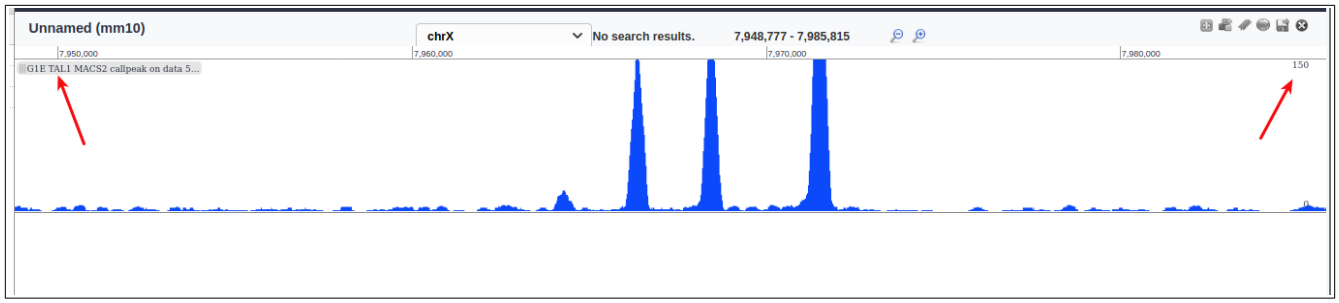


Table of Contents

7. Future Opportunity & Outlook

Examples job fields:

Bioinformatics

Genomics

Statistical Genetics

Computer Aided Drug Design (CADD)

Computational Protein Engineering

Biological Data Science

Examples jobs applied to:

Software Engineering Intern

Bioinformatics Engineer

R&D Bioinformatics Analyst

Intern, Computational Biology

Data Scientist Intern

Data Analyst Bioinformatics

Protein Design Data Scientist Intern

CADD & Molecular Analytics Intern

R&D Data Science Graduate Program

Data Sciences & AI Graduate Programme

Bioinformatics Co-Op

Computational & Structural Chemistry Co-Op

Early Career Biological Data Liaison

Co-op, Computational Sciences Bioinformatics

Computational Biologist - Summer Intern - Single Cell Analysis

Cheminformatics Intern

...

Short 5 year plan Option 1:

Apply for 2 year graduate programs and internships in the fields above. My goals during these 2 years are to use computational tools and algorithms to model small molecules and protein drugs as therapies for blood cancers. I hope to narrow down a specific type of hematological cancer throughout this time. I completed a project on Leukemia stems cells in my Epigenetics course on the role of differentiation in ALL and AML which was really exciting however I am also interested in immunotherapies targeting receptors in aggressive B-cell lymphomas and B-cell acute lymphoblastic leukemia.

After 2 years apply to roles in Level 2 and 3 Bioinformatician and Computed Aided-Drug Design to define molecular targets in pediatric leukemia.

Table of Contents

Short 5 year plan Option 2:

Applying to 5 PhD programs and 1 Master's programs below in Bioinformatics and Medicinal Chemistry to continue my education in Cancer Bioinformatics and Computed Aided-Drug Design for targets in pediatric leukemia.

University of California San Francisco – PhD Bioinformatics

European Bioinformatics Institute (EMBL-EBI) – PhD Bioinformatics, Cortes-ciriano group

Harvard University – BIG PhD Bioinformatics

University of Michigan – Medicinal Chemistry (Bioinformatics Tab)

University of Illinois at Chicago – Pharmaceutical Science (Chemistry in Drug Discovery Concentration)

Stevens Institute of Technology - Computational and Medicinal Chemistry