GENOME ANALYSIS TOOLS

A handbook

ASHWINI VASANTH
JOHNS HOPKINS UNIVERSITY -AAP
Summer 2021
AS.410.635.81.SU21 Bioinformatics: Tools for Genome Analysis
Dr. Sijung Yun, Dr. Sajung Yun

Contents

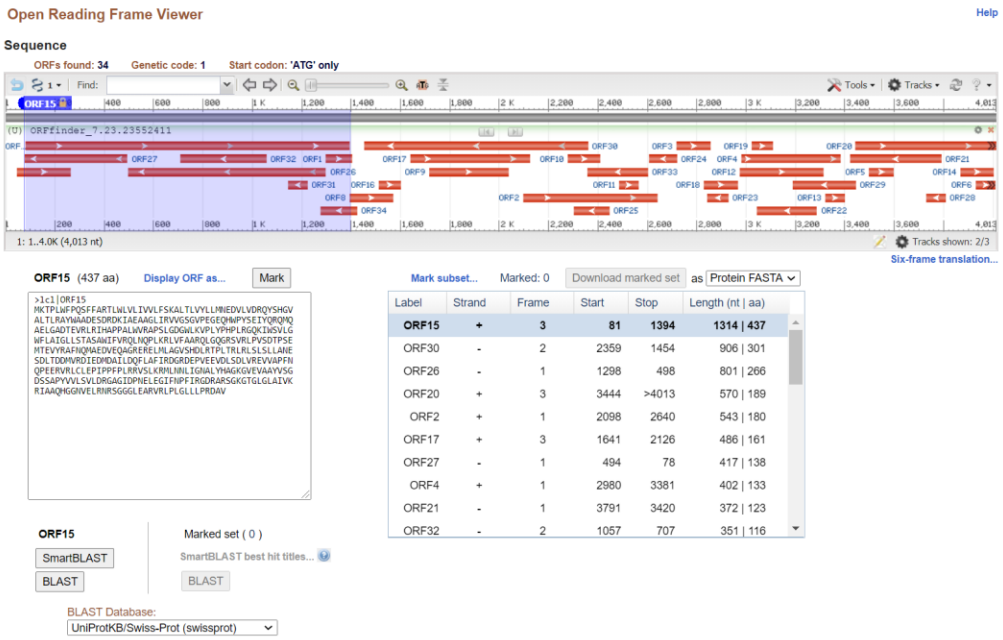
Introduction

The course “Bioinformatics: Tools for Genome Analysis” introduced me to several tools used by Bioinformaticians worldwide. These included prokaryotic and eukaryotic gene prediction tools – some command line and others with web interface, genome browsers, Galaxy, IGV, R programming language, NGS data analysis etc. This document lists the key tools studied in this course along with a summary of its usage and results and intends to be a ready reckoner for anyone new to the field of Bioinformatics. Also included in the document is a list of specialized databases that host species specific information. The document is linked to the websites associated with the tools and genome browsers.

Gene Prediction Tools

ORF Finder

The **ORF Finder** looks for Open Reading Frames (ORFs) in the DNA sequence that is entered in the interface. It lists the ORFs, and the protein sequence translated from each ORF. It is a good way to look for protein coding sequences in a newly sequenced DNA.

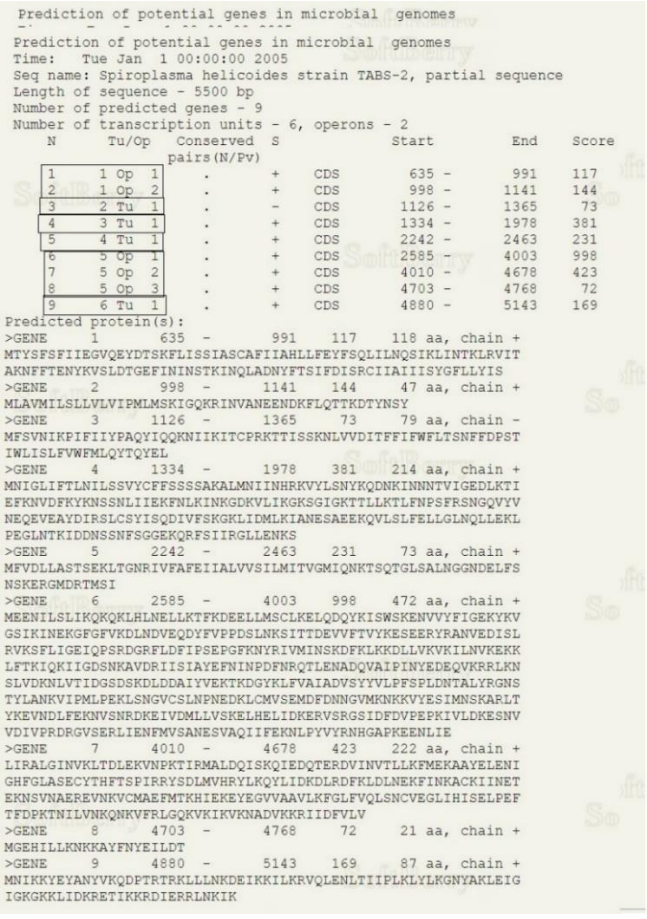


A DNA sequence in FASTA format or an accession ID can be entered in the interface. After selecting the parameters and submitting the form, a webpage like the one above is displayed that shows the total number of ORFs predicted with the start and stop position of each, and the corresponding protein sequence encoded.

FEGENESB

FGENESB is a bacterial operon and gene finding program. The prediction algorithm is based on Markov chain models. It is a very accurate ab initio prediction program available for bacterial genes.

USAGE EXAMPLE: Identify coding sequences (CDS) in the partial sequence of a bacterial strain such as S. helicoides strain TABS-2 using 'bacterial generic' as the training set. In the example below, nine CDSs are predicted by FGENESB. There are six predicted mRNAs of which two are operons. The mRNAs predicted are enclosed in black rectangles in the screenshot below.



```
 Prediction of potential genes in microbial genomes

 Prediction of potential genes in microbial genomes
 Time:   Tue Jan  1 00:00:00 2005
 Seq name: Spiroplasma helicoides strain TABS-2, partial sequence
 Length of sequence - 5500 bp
 Number of predicted genes - 9
 Number of transcription units - 6, operons - 2
     N      Tu/Op   Conserved  S             Start         End    Score
                    pairs(N/Pv)
    [1      1 Op  1]     .       +    CDS        635 -       991    117
    [2      1 Op  2]     .       +    CDS        998 -      1141    144
    [3      2 Tu  1]     .       -    CDS       1126 -      1365     73
    [4      3 Tu  1]     .       +    CDS       1334 -      1978    381
    [5      4 Tu  1]     .       +    CDS       2242 -      2463    231
    [6      5 Op  1]     .       +    CDS       2585 -      4003    998
    [7      5 Op  2]     .       +    CDS       4010 -      4678    423
    [8      5 Op  3]     .       +    CDS       4703 -      4768     72
    [9      6 Tu  1]     .       +    CDS       4880 -      5143    169
Predicted protein(s):
>GENE     1       635  -       991    117    118 aa, chain +
MTYSFSFIIEGVQEYDTSKFLISSIASCAFIIAHLLFEYFSQLILNQSIKLINTKLRVIT
AKNFFTENYKVSLDTGEFININSTKINQLADNYFTSIFDISRCIIAIIISYGFLLYIS
>GENE     2       998  -      1141    144     47 aa, chain +
MLAVMILSLLVLVIPMLMSKIGQKRINVANEENDKFLQTTKDTYNSY
>GENE     3      1126  -      1365     73     79 aa, chain -
MFSVNIKPIFIIYPAQYIQQKNIIKITCPRKTTISSKNLVVDITFFIFWFLTSNFFDPST
IWLISLFVWFMLQYTQYEL
>GENE     4      1334  -      1978    381    214 aa, chain +
MNIGLIFTLNILSSVYCFFSSSSAKALMNIINHRKVYLSNYKQDNKINNNTVIGEDLKTI
EFKNVDFKYKNSSNLIIEKFNLKINKGDKVLIKGKSGIGKTTLLKTLFNPSFRSNGQVYV
NEQEVEAYDIRSLCSYISQDIVFSKGKLIDMLKIANESAEEKQVLSLFELLGLNQLLEKL
PEGLNTKIDDNSSNFSGGEKQRFSIIRGLLENKS
>GENE     5      2242  -      2463    231     73 aa, chain +
MFVDLLASTSEKLTGNRIVFAFEIIALVVSILMITVGMIQNKTSQTGLSALNGGNDELFS
NSKERGMDRTMSI
>GENE     6      2585  -      4003    998    472 aa, chain +
MEENILSLIKQKQKLHLNELLKTFKDEELLMSCLKELQDQYKISWSKENVVYFIGEKYKV
GSIKINEKGFGFVKDLNDVEQDYFVPPDSLNKSITTDEVVFTVYKESEERYRANVEDISL
RVKSFLIGEIQPSRDGRFLDFIPSEPGFKNYRIVMINSKDFKLKKDLLVKVKILNVKEKK
LFTKIQKIIGDSNKAVDRIISIAYEFNINPDFNRQTLENADQVAIPINYEDEQVKRRLKN
SLVDKNLVTIDGSDSKDLDDAIYVEKTKDGYKLFVAIADVSYYVLPFSPLDNTALYRGNS
TYLANKVIPMLPEKLSNGVCSLNPNEDKLCMVSEMDFDNNGVMKNKKVYESIMNSKARLT
YKEVNDLFEKNVSNRDKEIVDMLLVSKELHELIDKERVSRGSIDFDVPEPKIVLDKESNV
VDIVPRDRGVSERLIENFMVSANESVAQIIFEKNLPYVYRNHGAPKEENLIE
>GENE     7      4010  -      4678    423    222 aa, chain +
LIRALGINVKLTDLEKVNPKTIRMALDQISKQIEDQTERDVINVTLLKFMEKAAYELENI
GHFGLASECYTHFTSPIRRYSDLMVHRYLKQYLIDKDLRDFKLDLNEKFINKACKIINET
EKNSVNAEREVNKVCMAEFMTKHIEKEYEGVVAAVLKFGLFVQLSNCVEGLIHISELPEF
TFDPKTNILVNKQNKVFRLGQKVKIKVKNADVKKRIIDFVLV
>GENE     8      4703  -      4768     72     21 aa, chain +
MGEHILLKNKKAYFNYEILDT
>GENE     9      4880  -      5143    169     87 aa, chain +
MNIKKYEYANYVKQDPTRTRKLLLNKDEIKKILKRVQLENLTIIPLKLYLKGNYAKLEIG
IGKGKKLIDKRETIKKRDIERRLNKIK
```

BPROM

BPROM is a bacterial sigma 70 promoter prediction program. The interface takes a FASTA sequence as input and generates an output like the one below. The number of promoters predicted along with the locations of the -10 and -35 signals are shown. The algorithm predicts the transcription start site (TSS) of bacterial genes that are regulated by the sigma 70 promoters.

```
>Lactococcus lactis subsp. lactis ptsHI operon, complete sequence
 Length of sequence-      2592
 Threshold for promoters -  0.20
 Number of predicted promoters -      7
 Promoter Pos:    225 LDF-  8.79
 -10 box at pos.    210 TGGTACAAT Score    78
 -35 box at pos.    190 TTGCAA    Score    55
 Promoter Pos:   2543 LDF-  5.41
 -10 box at pos.   2528 AATTAATAT Score    53
 -35 box at pos.   2505 TTGATA    Score    58
 Promoter Pos:   1005 LDF-  3.54
 -10 box at pos.    990 TGTTAAATT Score    66
 -35 box at pos.    973 TTGGCT    Score    33
 Promoter Pos:   1860 LDF-  3.46
 -10 box at pos.   1845 AGGTATCAT Score    71
 -35 box at pos.   1826 TTGCAG    Score    49
 Promoter Pos:   1392 LDF-  2.99
 -10 box at pos.   1377 TGCTAATAT Score    67
 -35 box at pos.   1352 CTGACG    Score    25
 Promoter Pos:    561 LDF-  2.12
 -10 box at pos.    546 CAGAATAAT Score    40
 -35 box at pos.    527 ATGACT    Score    31
 Promoter Pos:   2216 LDF-  0.70
 -10 box at pos.   2201 TGGAAGAAT Score    41
 -35 box at pos.   2176 ATGAAA    Score    30

 Oligonucleotides from known TF binding sites:

 For promoter at    225:
       purR:  TTTCGTTT at position     200 Score -   6
       purR:  ATTTCAAG at position     217 Score -   9
        fnr:  TCAAGAGT at position     220 Score -  13
       nagC:  ATATTTTA at position     233 Score -   7
       nagC:  ATTTTAGA at position     235 Score -   6
 For promoter at   2543:
      rpoS17:  AGAGGGAG at position    2483 Score -  10
        fis:  CTCATTTT at position    2499 Score -   9
       argR:  AATTAATA at position    2528 Score -  11
 For promoter at   1005:
        crp:  TTAAATTG at position     992 Score -  10
 No such sites for promoter at   1860
 For promoter at   1392:
     rpoD19:  CACCTAAA at position    1391 Score -   6
 For promoter at    561:
       argR:  ATAATCAT at position     550 Score -   9
 No such sites for promoter at   2216
```

Sample output:

BPROM output:

First line - name of your sequence;
Second and Third lines - LDF threshold and the length of presented sequence
4th line - The number of predicted promoters
Next lines - positions of predicted promoters, and their scores with 'weights' of two conserved promoter boxes. Promoter position assign to the first nucleotide of the transcript (Transcription Start Site position).
After that we present elements of Transcriptional factor binding sites for each predicted promoter (if they found).

For example:

```
)prom  Sat Jan 18 21:11:25 EST 2003
-Region of E.coli genome between protein_id="AAC76687.1" and protein_id="AAC7668
Length of sequence-       420
Threshold for promoters -  0.20
Number of predicted promoters -      1
Promoter Pos:    145 LDF-  6.02
-10 box at pos.    130 ctttatgat Score    66
-35 box at pos.    109 tttaat    Score    36

Oligonucleotides from known TF binding sites:

For promoter at    145:
       fis:  TCTTTAAT at position     107 Score -   6
    rpoD17:  TTATGATA at position     132 Score -   7
     lexA:  ATAAATAA at position     137 Score -  14
    rpoD17:  ATAATAAT at position     141 Score -   8
```

GLIMMER

GLIMMER (Gene Locator and Interpolated Markov ModelER) is a UNIX-based program for finding the genes in microbial DNA (bacteria, archaea, and virus). It makes use of interpolated Markov models (IMMs) to recognize coding regions and to differentiate them from non-coding regions. The software version and the download link can be found here.

USAGE EXAMPLE: Given a complete bacterial genome in FASTA format, GLIMMER 3 can be used to find long ORFs in the genome sequence. In bacteria, most long ORFs correspond to actual genes. Hence, the long ORFs can be used to build a training set to analyze the rest of the genome. If working with an unknown DNA sequence, a related genome can be used to build training set. The .predict file produced as the output lists the ORFs with the start and end coordinates and the strand.

Use the command line version of Glimmer to analyze CDSs in a partial sequence from Spiroplasma helicoides strain TABS-2, whose genome was submitted to GenBank on August 23, 2016 (file: sheliprt.fasta). The training set will be the full genome of S. helicoides strain TABS-2 (file: sheli.fasta).

UNIX commands:

$ long-orfs -n -t 1.15 sheli.fasta sheli.longorfs

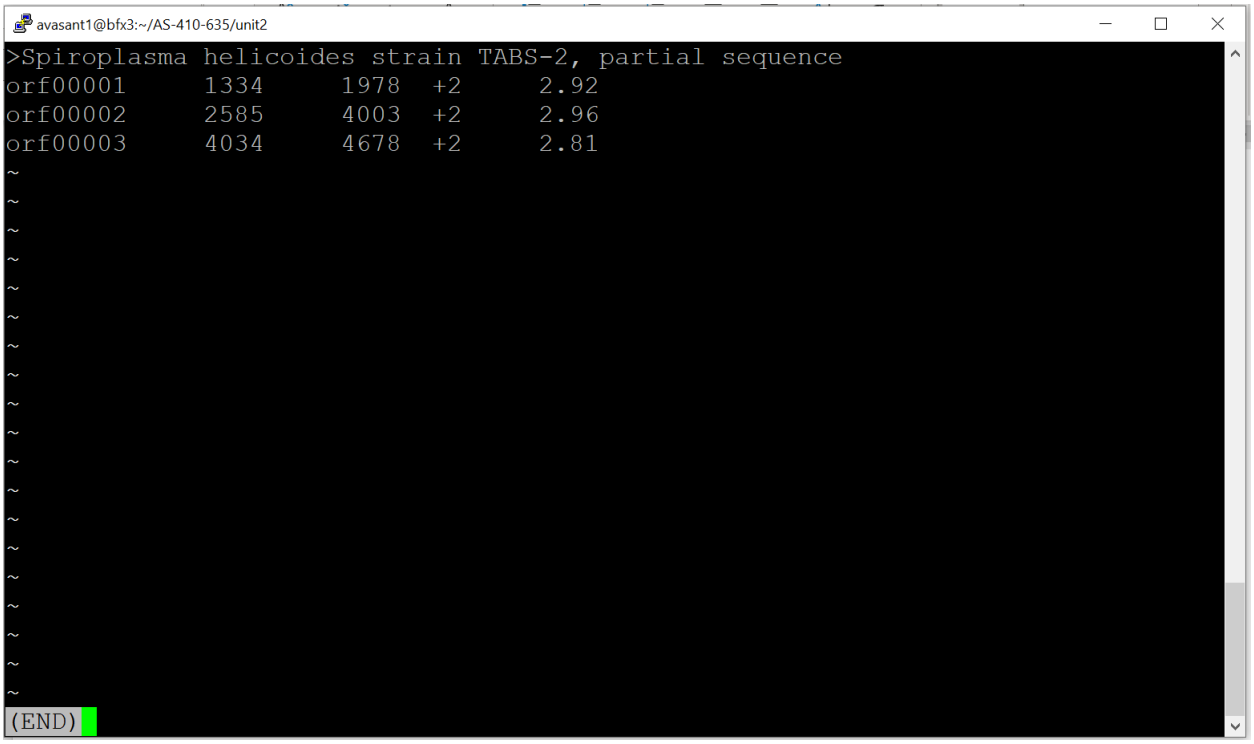
$ extract -t sheli.fasta sheli.longorfs > sheli.train

$ build-icm sheli.icm < sheli.train

$ glimmer3 -o50 -g110 -t30 sheliprt.fasta sheli.icm sheliprt

$ extract -t sheliprt.fasta sheliprt.predict > sheliprt.glimmer

Output format:



Command usage information:

long-orfs [options] <sequence-file> <output-file> - reads the DNA sequence in the <sequence-file> and outputs the coordinates of the long, non-overlapping ORFs in it.

- -n: do not include header information in the output.
- -t: only genes with the entropy score less the number specified will be considered.

extract [options] <sequence-file> <coords> - read the FASTA-format <sequence-file> and extract from it the subsequences specified by the coordinates. The default value for <coords> is the name of the file containing lines of the form <tag> <start> <stop> [<frame>]..

- -t: omit the last three characters of each output string

build-icm [options] output_file < input-file – read sequences from the standard input and output to the output_file the interpolated context model (icm) built from them.

glimmer3 [options] <sequence-file> <icm-file> <tag> - read DNA sequences in the <sequence-file> and predict genes in them using the <icm-file>. Output goes to the file <tag>.detail and predictions go to <tag>.predict.

- -o <n>: sets the maximum overlap length to <n>. Overlaps that are this short or shorter are ignored.
- -g <n>: set the minimum gene length to <n>.

GENEMARK

GeneMark.hmm prokaryotic version is a prokaryotic gene prediction tool. The sequence in FASTA format is provided as input to the program. The sets of pre-computed species are available for the user to select as a parameter to the program. The program output provides the list of predicted genes along with the strand, start and end coordinates, and the gene length.

GeneMark.hmm prokaryotic

Prokaryotic GeneMark.hmm version 1
Alexander Lukashin and Mark Borodovsky
GeneMark.hmm: new solutions for gene finding.
Nucleic Acids Research (1998) 26, pp 1107-1115

Prokaryotic GeneMark.hmm version 2
John Besemer, Alexandre Lomsadze and Mark Borodovsky
GeneMarkS: a self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions.
Nucleic Acids Research (2001) 29, pp 2607-2618

This webpage provides access to gene prediction program GeneMark.hmm prokaryotic (version 3.25) and to the sets of pre-computed species specific algorithm parameters (model parameters). The list of currently supported species is available here . These parameter sets were derived by application of the GeneMarkS that carried out unsupervised training on each genome.

Browse GeneMark.hmm prokaryotic manual

Input sequence and Select species

Enter sequence (FASTA or multi FASTA format)

```
>Stenotrophomonas maltophilia strain SM45 whole genome shotgun
sequence
ACCAAGTCGAAGCGTTCGGCCAACTCCTTGCGCTTGCCCGCGAGCTCGTCCATGCCTTCGAGCAGCTGCT
TCTGCTCCTCGGTGTACTTGGCAAGCTCCTGCCGCTGCGCATACAGCGAGTTCCAGATATTGGCGAATTG
CTGAAGGCTGTTCGGGCCACTCATCTGGCTAAGCATGTCCCTCGTCTTCTGAGAGATCTCACCCATGCTC
```

or, upload file: Choose File No file chosen

Select species

Stenotrophomonas_maltophilia_K279a

Action Start GeneMark.hmm Reset

Options

Output format for gene prediction	Output options	Optional: results by E-mail
◉ LST ○ GFF	☐ Protein sequence ☐ Gene nucleotide sequence Coding potential graph (not for multi FASTA) ☐ PDF ☐ PostScript	E-mail Subject GeneMark.hmm prokaryotic ☐ Compress files

Advanced options

☐ Switch off gene start related motif(s)

Contact Us | Home

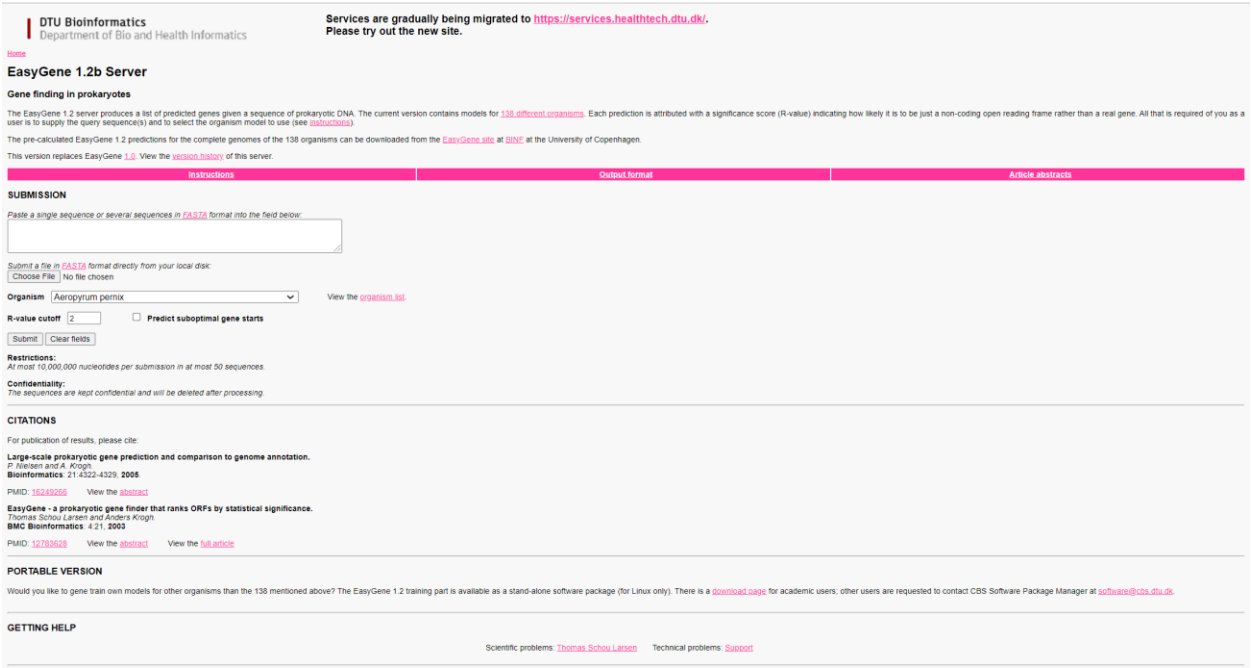
Output file:

```
GeneMark.hmm PROKARYOTIC (Version 3.26)
Date: Sun Jul 25 14:55:43 2021
Sequence file name: seq.fna
Model file name: /home/genemark/parameters/prokaryotic/Stenotrophomonas_maltophilia_K279a/GeneMark_hmm_combined.mod
RBS: true
Model information: Stenotrophomonas_maltophilia_K279a

FASTA definition line: Stenotrophomonas maltophilia strain SM45 whole genome shotgun sequence
Predicted genes
   Gene     Strand     LeftEnd     RightEnd       Gene      Class
     #                                          Length
     1        -           <3         3407         3405        2
     2        -         3400         4050          651        1
     3        -         4121         4921          801        1
     4        -         4935         5417          483        1
     5        -         5410         5745          336        1
     6        -         5742         6233          492        2
     7        -         6230         6364          135        1
```

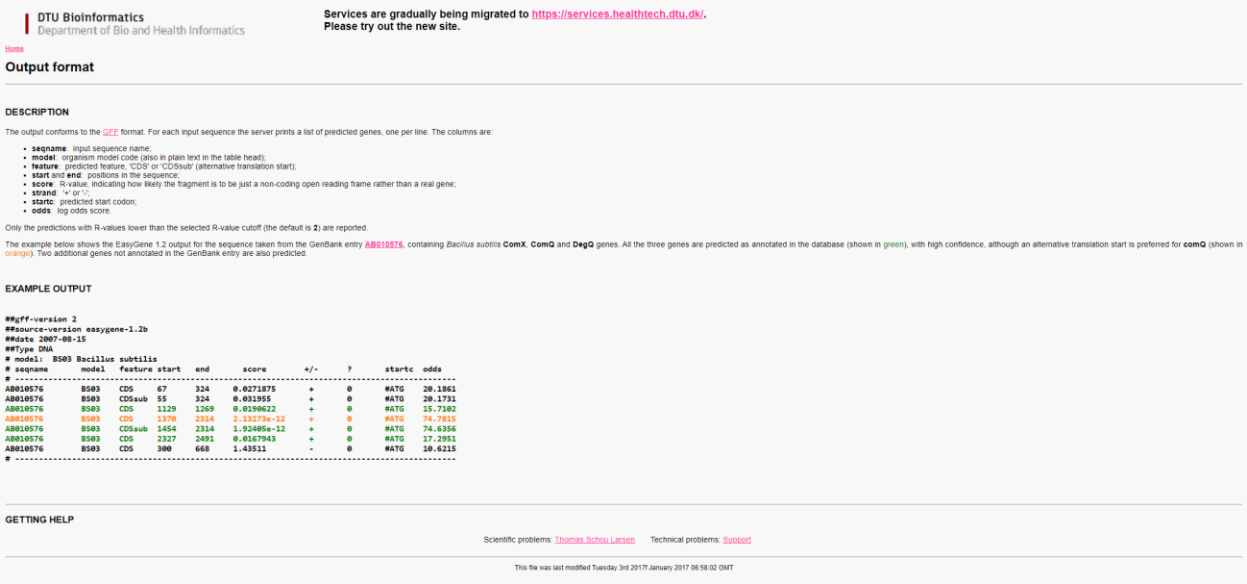
EasyGene 1.2b

EasyGene 1.2b server is a gene prediction tool for prokaryotic DNA. Each prediction comes with a significance score that indicates its likelihood of being a real gene.

Web-interface:



Understanding the output:



DTU Bioinformatics
Department of Bio and Health Informatics

Services are gradually being migrated to https://services.healthtech.dtu.dk/.
Please try out the new site.

Home

Output format

DESCRIPTION

The output conforms to the GFF format. For each input sequence the server prints a list of predicted genes, one per line. The columns are:

- **seqname**: input sequence name;
- **model**: organism model code (also in plain text in the table head);
- **feature**: 'predicted feature, 'CDS' or 'CDSsub' (alternative translation start);
- **start** and **end**: positions in the sequence;
- **score**: R-value, indicating how likely the fragment is to be just a non-coding open reading frame rather than a real gene;
- **strand**: '+' or '-';
- **startc**: predicted start codon;
- **odds**: log odds score.

Only the predictions with R-values lower than the selected R-value cutoff (the default is **2**) are reported.

The example below shows the EasyGene 1.2 output for the sequence taken from the GenBank entry AB010576, containing *Bacillus subtilis* **ComX**, **ComQ** and **DegQ** genes. All the three genes are predicted as annotated in the database (shown in green), with high confidence, although an alternative translation start is preferred for **comQ** (shown in orange). Two additional genes not annotated in the GenBank entry are also predicted.

EXAMPLE OUTPUT

```
##gff-version 2
##source-version easygene-1.2b
##date 2007-08-15
##Type DNA
# model:  BS03 Bacillus subtilis
# seqname       model   feature start   end       score       +/-     ?       startc  odds
# ------------------------------------------------------------------------------------------
AB010576        BS03    CDS     67      324     0.0271875       +       0       #ATG    20.1861
AB010576        BS03    CDSsub  55      324     0.031955        +       0       #ATG    20.1731
AB010576        BS03    CDS     1129    1269    0.0190622       +       0       #ATG    15.7182
AB010576        BS03    CDS     1370    2314    2.13273e-12     +       0       #ATG    74.7815
AB010576        BS03    CDSsub  1454    2314    1.92405e-12     +       0       #ATG    74.6356
AB010576        BS03    CDS     2327    2491    0.0167943       +       0       #ATG    17.2951
AB010576        BS03    CDS     300     668     1.43511         -       0       #ATG    10.6215
# ------------------------------------------------------------------------------------------
```

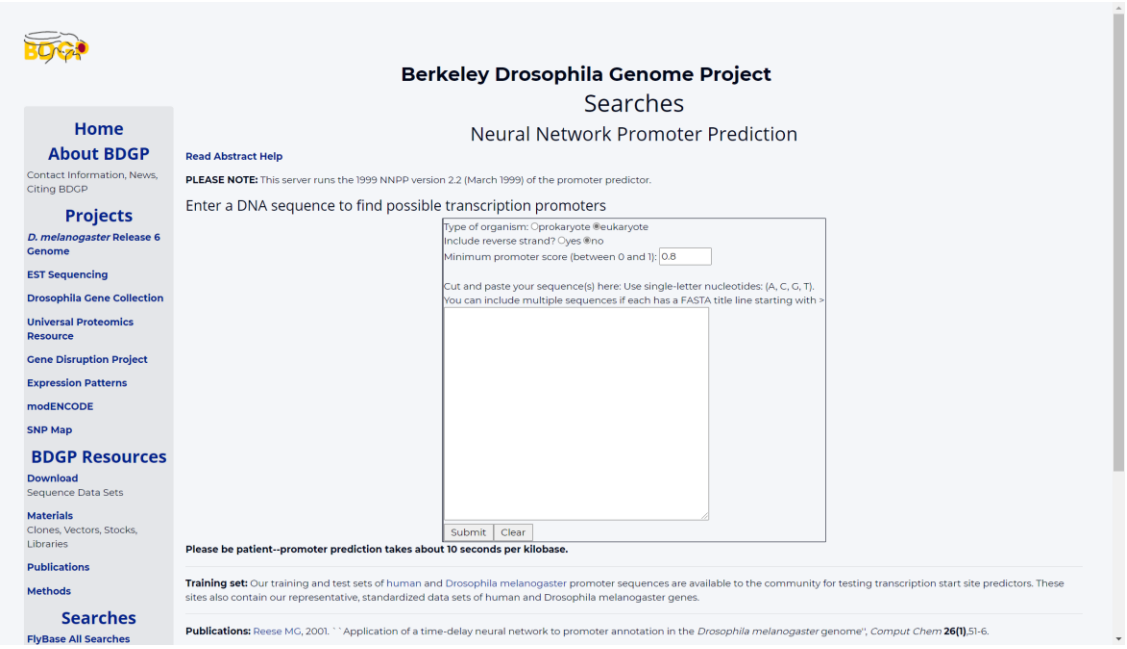
GETTING HELP

Scientific problems: Thomas Schou Larsen Technical problems: Support

This file was last modified Tuesday 3rd 2017f January 2017 06:58:02 GMT

Neural Network Promoter Prediction

It is found in the Berkeley Drosophila Genome Project page. This **program** predicts possible transcription promoters for both prokaryotic and eukaryotic DNA sequence. The output shows the start and end positions of the predicted promoters and the score associated with each prediction.



Berkeley Drosophila Genome Project
Searches
Neural Network Promoter Prediction

Home
About BDGP
Contact Information, News, Citing BDGP
Projects
D. melanogaster Release 6 Genome
EST Sequencing
Drosophila Gene Collection
Universal Proteomics Resource
Gene Disruption Project
Expression Patterns
modENCODE
SNP Map
BDGP Resources
Download
Sequence Data Sets
Materials
Clones, Vectors, Stocks, Libraries
Publications
Methods
Searches
FlyBase All Searches

Read Abstract Help

PLEASE NOTE: This server runs the 1999 NNPP version 2.2 (March 1999) of the promoter predictor.

Enter a DNA sequence to find possible transcription promoters

Type of organism: Oprokaryote ◉eukaryote
Include reverse strand? Oyes ◉no
Minimum promoter score (between 0 and 1): 0.8

Cut and paste your sequence(s) here: Use single-letter nucleotides: (A, C, G, T).
You can include multiple sequences if each has a FASTA title line starting with >

Submit | Clear

Please be patient--promoter prediction takes about 10 seconds per kilobase.

Training set: Our training and test sets of human and Drosophila melanogaster promoter sequences are available to the community for testing transcription start site predictors. These sites also contain our representative, standardized data sets of human and Drosophila melanogaster genes.

Publications: Reese MG, 2001. " Application of a time-delay neural network to promoter annotation in the *Drosophila melanogaster* genome", *Comput Chem* **26(1)**,51-6.

Output:

Promoter predictions for 1 prokaryotic sequence with score cutoff 0.80 (transcription start shown in larger font):

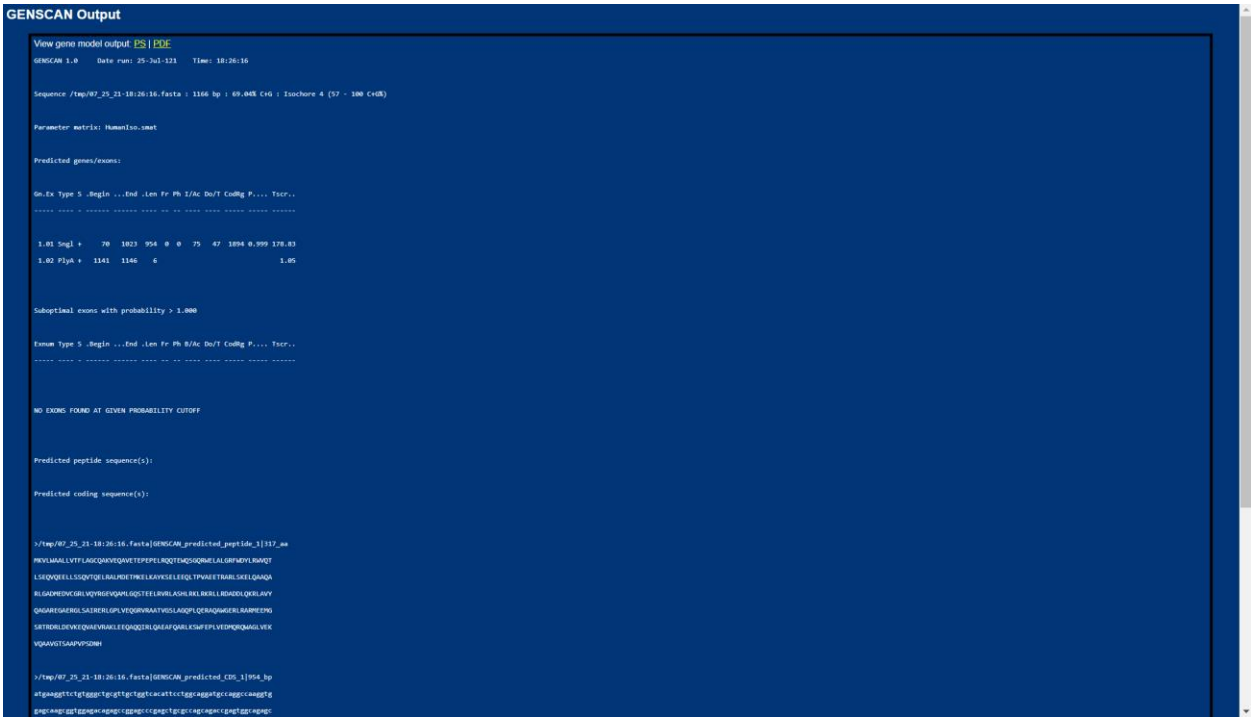
Promoter predictions for Spiroplasma :

Start	End	Score	Promoter Sequence
24	69	0.96	ATTTGTTTTTTTTATTCATAAATAGTTATTATTGTATGTC**T**ATCAACTTC
51	96	0.98	ATTATTGTATGTCTATCAACTTCTGGTTCTAAAATTAAAT**C**ATTGTAATT
92	137	0.96	ATTGTAATTTGAATAAAGAAGATAAACTTTAAAATGTTTT**T**TCTTAAGAG
126	171	0.95	TGTTTTTTCTTAAGAGTTCTATACAGATTTAAATTACTTC**T**GTAGTCATG
148	193	0.99	ACAGATTTAAATTACTTCTGTAGTCATGATTTATTAATAA**A**ATCTTTTTA
175	220	0.96	GATTTATTAATAAAATCTTTTTACTTCTATCTAATACTTT**G**TAATACTTG
188	233	0.92	AATCTTTTTACTTCTATCTAATACTTTGTAATACTTGTAA**T**TAAATGTAA
220	265	0.97	ACTTGTAATTAAATGTAAGGATAGGCAGATTAATTGTATT**T**TCTATGTGA
255	300	0.94	GTATTTTCTATGTGATTGTCAGCATAAGCTACAACATTTC**T**AAGATCAGC
266	311	0.96	GTGATTGTCAGCATAAGCTACAACATTTCTAAGATCAGCA**A**CTTGTCATT
309	354	0.91	TGTCATTTTTCACTTTTAAGTATTTTTAAAAAATTTTTAG**C**TGGTTTTGT
332	377	0.83	TTTTAAAAAATTTTTAGCTGGTTTTGTTCTATATTTTTTC**T**TAAATGAAT
380	425	1.00	ATTTGTTTTGAAAATTTTAGCCATAAATTTAAAAACATAA**T**CTAGTCATT
391	436	0.96	AAATTTTAGCCATAAATTTAAAAACATAATCTAGTCATTG**C**ATATAATCA
404	449	0.99	AAATTTAAAAACATAATCTAGTCATTGCATATAATCACTT**C**CTTAAACGC
426	471	0.87	CATTGCATATAATCACTTCCTTAAACGCATCTAATTGATA**T**ATAATTATT
456	501	0.95	CTAATTGATATATAATTATTTTAATTATATATAACTTATT**T**ATATATTTA
469	514	1.00	AATTATTTTAATTATATATAACTTATTTATATATTTAAAA**T**ATAAAAATA
488	533	0.96	AACTTATTTATATATTTAAAATATAAAAATAATATATCAC**A**TATTATTTT

GENSCAN

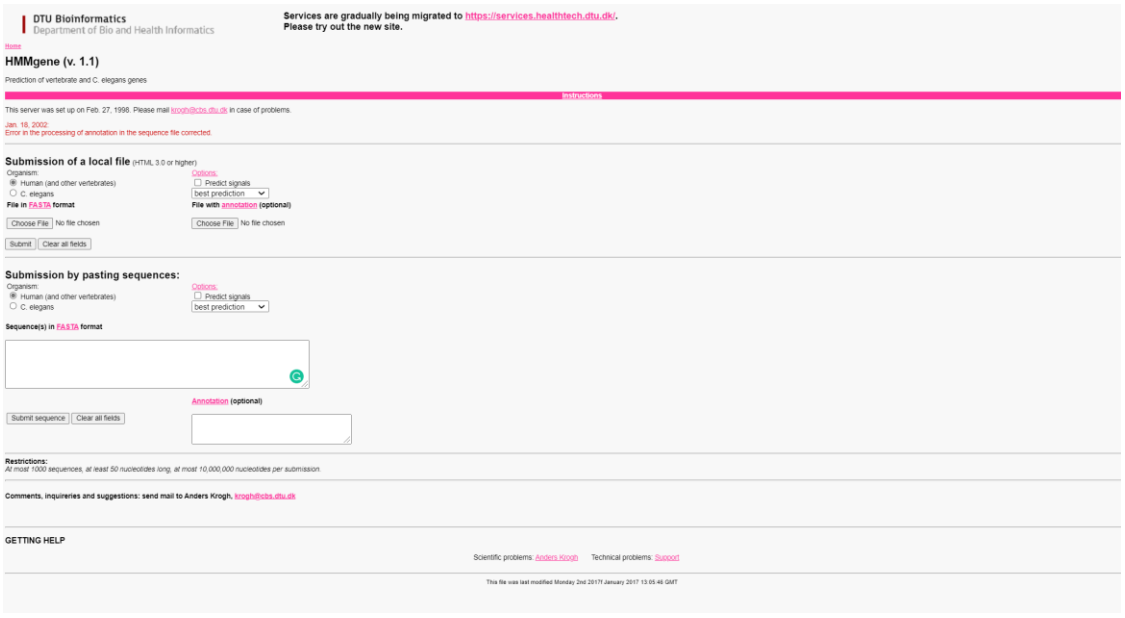
[GENSCAN](#) is one of the first successful eukaryotic gene prediction program. It predicts the locations and the exon-intron structures of genes in sequences up to 1 million base pairs in length. The output of the program includes the genes predicted with the start and stop coordinates, poly(A) tail location, and the strand information. It also shows the sequence of the predicted peptide.

Output:

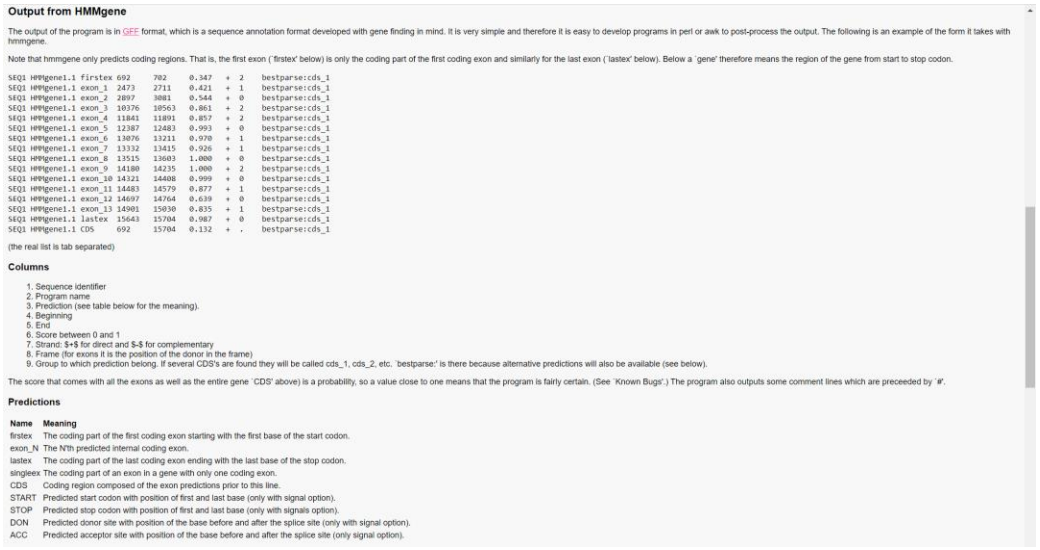


HMMGene

HMMGene is a gene prediction tool for vertebrates and C.elegans. The web-interface provides the user the option to choose the organism, enter the input sequence, and select predict signals that predicts the splice sites and probabilities associated with start/stop codons predicted.



Sample output with explanation of output format:



Output from HMMgene

The output of the program is in GFF format, which is a sequence annotation format developed with gene finding in mind. It is very simple and therefore it is easy to develop programs in perl or awk to post-process the output. The following is an example of the form it takes with hmmgene.

Note that hmmgene only predicts coding regions. That is, the first exon (`firstex' below) is only the coding part of the first coding exon and similarly for the last exon (`lastex' below). Below a `gene' therefore means the region of the gene from start to stop codon.

```
SEQ1 HMMgene1.1 firstex 692     702     0.347   +  2    bestparse:cds_1
SEQ1 HMMgene1.1 exon_1  2473    2711    0.421   +  1    bestparse:cds_1
SEQ1 HMMgene1.1 exon_2  2897    3081    0.544   +  0    bestparse:cds_1
SEQ1 HMMgene1.1 exon_3  10376   10563   0.861   +  2    bestparse:cds_1
SEQ1 HMMgene1.1 exon_4  11841   11891   0.857   +  2    bestparse:cds_1
SEQ1 HMMgene1.1 exon_5  12387   12483   0.993   +  0    bestparse:cds_1
SEQ1 HMMgene1.1 exon_6  13076   13211   0.970   +  1    bestparse:cds_1
SEQ1 HMMgene1.1 exon_7  13332   13415   0.926   +  1    bestparse:cds_1
SEQ1 HMMgene1.1 exon_8  13515   13603   1.000   +  0    bestparse:cds_1
SEQ1 HMMgene1.1 exon_9  14180   14235   1.000   +  2    bestparse:cds_1
SEQ1 HMMgene1.1 exon_10 14321   14408   0.999   +  0    bestparse:cds_1
SEQ1 HMMgene1.1 exon_11 14483   14579   0.877   +  1    bestparse:cds_1
SEQ1 HMMgene1.1 exon_12 14697   14764   0.639   +  0    bestparse:cds_1
SEQ1 HMMgene1.1 exon_13 14901   15030   0.835   +  1    bestparse:cds_1
SEQ1 HMMgene1.1 lastex  15643   15704   0.987   +  0    bestparse:cds_1
SEQ1 HMMgene1.1 CDS     692     15704   0.132   +  .    bestparse:cds_1
```

(the real list is tab separated)

Columns

1. Sequence identifier
2. Program name
3. Prediction (see table below for the meaning).
4. Beginning
5. End
6. Score between 0 and 1
7. Strand: $+$ for direct and $-$ for complementary
8. Frame (for exons it is the position of the donor in the frame)
9. Group to which prediction belong. If several CDS's are found they will be called cds_1, cds_2, etc. `bestparse:' is there because alternative predictions will also be available (see below).

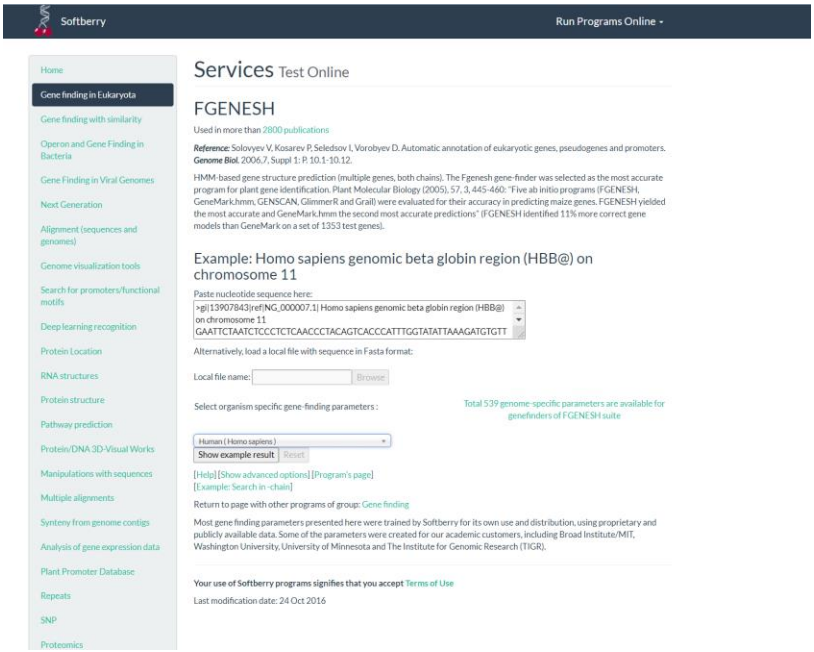
The score that comes with all the exons as well as the entire gene `CDS' above) is a probability, so a value close to one means that the program is fairly certain. (See `Known Bugs'.) The program also outputs some comment lines which are preceeded by `#'.

Predictions

Name	Meaning
firstex	The coding part of the first coding exon starting with the first base of the start codon.
exon_N	The N'th predicted internal coding exon.
lastex	The coding part of the last coding exon ending with the last base of the stop codon.
singleex	The coding part of an exon in a gene with only one coding exon.
CDS	Coding region composed of the exon predictions prior to this line.
START	Predicted start codon with position of first and last base (only with signal option).
STOP	Predicted stop codon with position of first and last base (only with signals option).
DON	Predicted donor site with position of the base before and after the splice site (only with signal option).
ACC	Predicted acceptor site with position of the base before and after the splice site (only signal option).

FGENESH

FGENESH is an HMM-based gene structure prediction program that can predict multiple genes considering both strands. The DNA sequence in FASTA format is input to the program along with the organism information. The output is quite detailed with the number of predicted genes, exons, strand information. For each predicted gene, the positions of the TSS, CDSs, and the poly(A) sites are provided. The predicted protein sequence for each of the genes can also be seen.



Sample output:

Example: Homo sapiens genomic beta globin region (HBB@) on chromosome 11

Show picture of predicted genes in PDF file

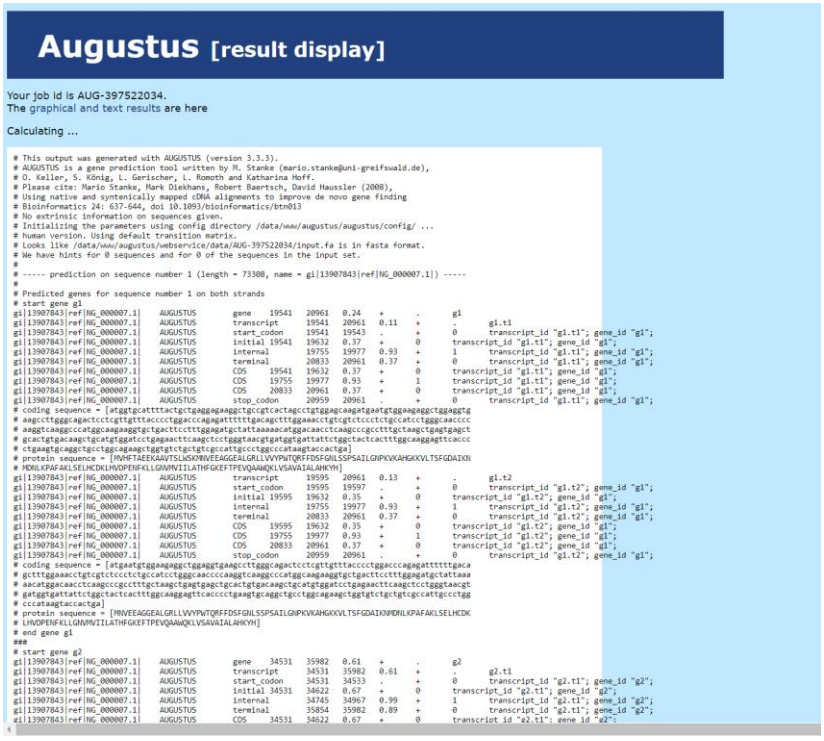
```
 FGENESH 2.6 Prediction of potential genes in Homo_sapiens genomic DNA
 Time     :   Sun Jul 25 18:43:27 2021
 Seq name: gi|13907843|ref|NG_000007.1| Homo sapiens genomic beta globin region (HBB@) on
 Length of sequence: 73308
 Number of predicted genes 10: in +chain 10, in -chain 0.
 Number of predicted exons 21: in +chain 21, in -chain 0.
 Positions of predicted genes and exons: Variant   1 from   1, Score:180.171887
   G Str   Feature   Start        End    Score          ORF          Len

   1 +       TSS      19456               -7.09
   1 +     1 CDSf     19541 -     19632   16.13     19541 -     19630     90
   1 +     2 CDSi     19755 -     19977   13.37     19756 -     19977    222
   1 +     3 CDSl     20833 -     20961    3.34     20833 -     20961    129
   1 +       PolA     21055               1.12

   2 +       TSS      34446               -7.09
   2 +     1 CDSf     34531 -     34622   13.42     34531 -     34620     90
   2 +     2 CDSi     34745 -     34967   21.52     34746 -     34967    222
   2 +     3 CDSl     35854 -     35982    2.92     35854 -     35982    129
   2 +       PolA     36043               1.12

   3 +       TSS      39382               -7.09
   3 +     1 CDSf     39467 -     39558   13.42     39467 -     39556     90
   3 +     2 CDSi     39681 -     39903   21.52     39682 -     39903    222
   3 +     3 CDSl     40770 -     40898    3.66     40770 -     40898    129
   3 +       PolA     40959               1.12

   4 +       TSS      44415               -8.69
   4 +     1 CDSf     45995 -     46151   16.58     45995 -     46150    156
   4 +     2 CDSl     46997 -     47100   -1.94     46999 -     47100    102
   4 +       PolA     47243               1.12

   5 +       TSS      54707               -4.39
   5 +     1 CDSf     54790 -     54881   13.44     54790 -     54879     90
   5 +     2 CDSi     55010 -     55232   17.01     55011 -     55232    222
   5 +     3 CDSl     56425 -     56535    2.53     56425 -     56535    111
   5 +       PolA     56931               1.12

   6 +       TSS      62104               -6.59
   6 +     1 CDSf     62187 -     62278   12.99     62187 -     62276     90
   6 +     2 CDSi     62409 -     62631   20.06     62410 -     62631    222
   6 +     3 CDSl     63482 -     63610    9.54     63482 -     63610    129
   6 +       PolA     63718               1.12
```

AUGUSTUS

AUGUSTUS is a gene prediction program for eukaryotic sequences. The program takes as input the sequence in FASTA format along with the organism information. For every predicted gene, the program lists the transcript coordinates, start/stop positions, CDSs, coding sequence, and the predicted protein sequence.


Augustus [job submission]
Paste your sequence(s) here help
>gi|13907843|ref|NG_000007.1| Homo sapiens genomic beta globin region (HBB@) on chromosome 11
GAATTCTAATCTCCCTCTCAACCCTACAGTCACCCATTTGGTATATTAAAGATGTGTTGTCTACTGTCTAGTATCCCTCA
AGTAGTGTCAGGAATTAGTCATTTAAATAGTCTGCAAGCCAGGAGTGGTGGCTCATGTCTGTAATTCCAGCACTGGAGAG
GTAGAAGTGGGAGGACTGCTTGAGCTCAAGAGTTTGATATTATCCTGGACAACATAGCAAGACCTCGTCTCTACTTAAAA
AAAAAAAATTAGCCAGGCATGTGATGTACACCTGTAGTCCCAGCTACTCAGGAGGCCGAAATGGGAGGATCCCTTGAGC
TCAGGAGGTCAAGGCTGCAGTGAGACATGATCTTGCCACTGCACTCCAGCCTGGACAGCAGAGTGAAACCTTGCCTCACG
AAACAGAATACAAAAACAAACAAACAAAAAACTGCTCCGCAATGCGCTTCCTTGATGCTCTACCACATAGGTCTGGGTAC
or upload a file in (multiple) FASTA format
Choose File No file chosen
or fill in an example.
Organism: Homo sapiens
Report genes on: both strands forward strand only reverse strand only
Alternative transcripts: none few middle many
Reset all input Run AUGUSTUS
expert options
Upload cDNA (ESTs, mRNAs) sequences. Non-commercial users only. help
or upload a cDNA sequence file (FASTA).
Choose File No file chosen
or fill in an example.
Paste here constraints that anchor the prediction help

Output:


Augustus [result display]
Your job id is AUG-397522034.
The graphical and text results are here
Calculating ...
This output was generated with AUGUSTUS (version 3.3.3).
AUGUSTUS is a gene prediction tool written by M. Stanke (mario.stanke@uni-greifswald.de),
O. Keller, S. König, L. Gerischer, L. Romoth and Katharina Hoff.
Please cite: Mario Stanke, Mark Diekhans, Robert Baertsch, David Haussler (2008),
Using native and syntenically mapped cDNA alignments to improve de novo gene finding
Bioinformatics 24: 637-644, doi 10.1093/bioinformatics/btn013
No extrinsic information on sequences given.
Initializing the parameters using config directory /data/www/augustus/augustus/config/ ...
human version. Using default transition matrix.
Looks like /data/www/augustus/webservice/data/AUG-397522034/input.fa is in fasta format.
We have hints for 0 sequences and for 0 of the sequences in the input set.
#
----- prediction on sequence number 1 (length = 73308, name = gi|13907843|ref|NG_000007.1|) -----
#
Predicted genes for sequence number 1 on both strands
start gene g1
gi|13907843|ref|NG_000007.1| AUGUSTUS gene 19541 20961 0.24 + . g1
gi|13907843|ref|NG_000007.1| AUGUSTUS transcript 19541 20961 0.11 + . g1.t1
gi|13907843|ref|NG_000007.1| AUGUSTUS start_codon 19541 19543 . + 0 transcript_id "g1.t1"; gene_id "g1";
gi|13907843|ref|NG_000007.1| AUGUSTUS initial 19541 19632 0.37 + 0 transcript_id "g1.t1"; gene_id "g1";
gi|13907843|ref|NG_000007.1| AUGUSTUS internal 19755 19977 0.93 + 1 transcript_id "g1.t1"; gene_id "g1";
gi|13907843|ref|NG_000007.1| AUGUSTUS terminal 20833 20961 0.37 + 0 transcript_id "g1.t1"; gene_id "g1";
gi|13907843|ref|NG_000007.1| AUGUSTUS CDS 19541 19632 0.37 + 0 transcript_id "g1.t1"; gene_id "g1";
gi|13907843|ref|NG_000007.1| AUGUSTUS CDS 19755 19977 0.93 + 1 transcript_id "g1.t1"; gene_id "g1";
gi|13907843|ref|NG_000007.1| AUGUSTUS CDS 20833 20961 0.37 + 0 transcript_id "g1.t1"; gene_id "g1";
gi|13907843|ref|NG_000007.1| AUGUSTUS stop_codon 20959 20961 . + 0 transcript_id "g1.t1"; gene_id "g1";
coding sequence = [atggtgcattttactgctgaggagaaggctgccgtcactagcctgtggagcaagatgaatgtggaagagctgaggtg
aagcctttgggcagactcctcgttgtttaccccctggacccagagatttttttgacagctttggaaacctgtcgtctcccctctgccatcctgggcaacccc
aaggtcaaggcccatggcaagaaggtgctgacttccttttggagatgctattaaaaacatggacaacctcaagcccgcctttgctaagctgagtgagct
gcactgtgacaagctgcatgtggatcctgaggaacttcaagctcctgggtaacgtgatggtgattatctgactactttggcaaggagttcaccc
ctgaagtgcaggctgcctggcagaagctggtgtctgctgtcgccattgccctggcccataagtaccactga]
protein sequence = [MVHFTAEEKAAVTSLWSKMNVEEAGGEALGRLLVVYPWTQRFFDSFGNLSSPSAILGNPKVKAHGKKVLTSFGDAIKN
HDNLKPAFAKLSELHCDKLHVDPENFKLLGNMVIILATHFGKEFTPEVQAAWQKLVSAVAIALAHKYH]
gi|13907843|ref|NG_000007.1| AUGUSTUS transcript 19595 20961 0.13 + . g1.t2
gi|13907843|ref|NG_000007.1| AUGUSTUS start_codon 19595 19597 . + 0 transcript_id "g1.t2"; gene_id "g1";
gi|13907843|ref|NG_000007.1| AUGUSTUS initial 19595 19632 0.35 + 0 transcript_id "g1.t2"; gene_id "g1";
gi|13907843|ref|NG_000007.1| AUGUSTUS internal 19755 19977 0.93 + 1 transcript_id "g1.t2"; gene_id "g1";
gi|13907843|ref|NG_000007.1| AUGUSTUS terminal 20833 20961 0.37 + 0 transcript_id "g1.t2"; gene_id "g1";
gi|13907843|ref|NG_000007.1| AUGUSTUS CDS 19595 19632 0.35 + 0 transcript_id "g1.t2"; gene_id "g1";
gi|13907843|ref|NG_000007.1| AUGUSTUS CDS 19755 19977 0.93 + 1 transcript_id "g1.t2"; gene_id "g1";
gi|13907843|ref|NG_000007.1| AUGUSTUS CDS 20833 20961 0.37 + 0 transcript_id "g1.t2"; gene_id "g1";
gi|13907843|ref|NG_000007.1| AUGUSTUS stop_codon 20959 20961 . + 0 transcript_id "g1.t2"; gene_id "g1";
coding sequence = [atgaatgtggaagaggctggaggtgaagccttgggcagactcctcgttgtttaccccctggacccagagatttttttgaca
gctttggaaacctgtcgtctcccctctgccatcctgggcaaccccaaggtcaaggcccatggcaagaaggtcgtcaacttccttttggagatgctattaaa
aacatggacaacctcaagcccgcctttgctaagctgagtgagctgcactgtgacaagctgcatggtggatcctgagaacttcaagctcctgggtaacgt
gatggtgattatctggctactcactttggcaaggagttcaccctgaagtgcaggctgcctggcagaagctggtgtctgctgtcgccattgccctgg
cccataagtaccactga]
protein sequence = [MNVEEAGGEALGRLLVVYPWTQRFFDSFGNLSSPSAILGNPKVKAHGKKVLTSFGDAIKHMDNLKPAFAKLSELHCDK
LHVDPENFKLLGNMVIILATHFGKEFTPEVQAAWQKLVSAVAIALAHKYH]
end gene g1
###
start gene g2
gi|13907843|ref|NG_000007.1| AUGUSTUS gene 34531 35982 0.61 + . g2
gi|13907843|ref|NG_000007.1| AUGUSTUS transcript 34531 35982 0.61 + . g2.t1
gi|13907843|ref|NG_000007.1| AUGUSTUS start_codon 34531 34533 . + 0 transcript_id "g2.t1"; gene_id "g2";
gi|13907843|ref|NG_000007.1| AUGUSTUS initial 34531 34622 0.67 + 0 transcript_id "g2.t1"; gene_id "g2";
gi|13907843|ref|NG_000007.1| AUGUSTUS internal 34745 34967 0.99 + 1 transcript_id "g2.t1"; gene_id "g2";
gi|13907843|ref|NG_000007.1| AUGUSTUS terminal 35854 35982 0.89 + 0 transcript_id "g2.t1"; gene_id "g2";
gi|13907843|ref|NG_000007.1| AUGUSTUS CDS 34531 34622 0.67 + 0 transcript_id "g2.t1"; gene_id "g2";

Splign

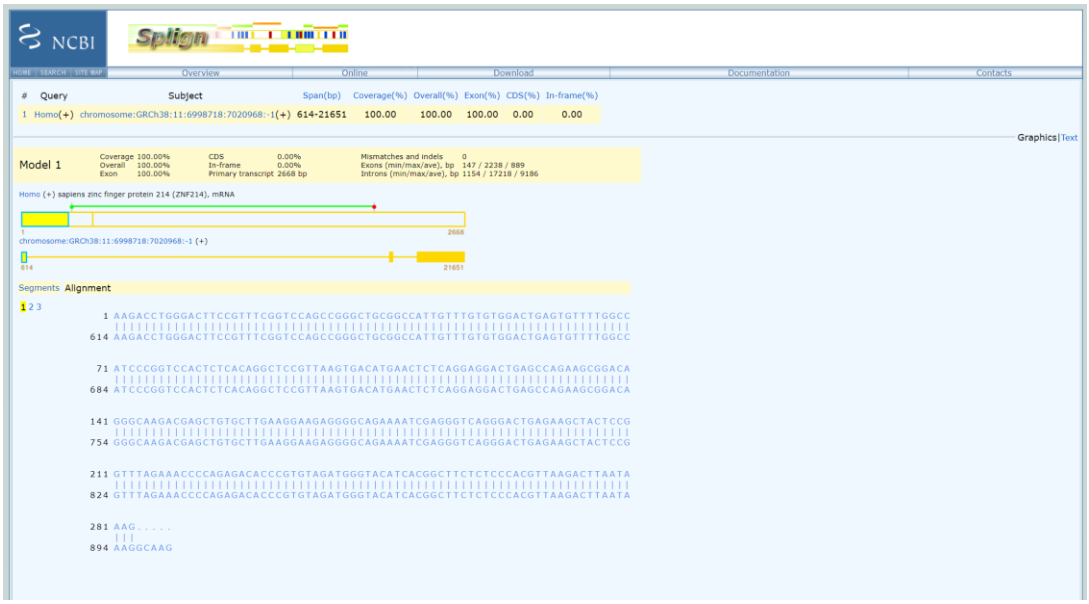
Splign, hosted by NCBI aligns cDNA to genomic DNA and identifies splice-junction locations, frameshifts, alternate gene models where possible. Splign also supports cross-species alignments. The output of the program is the gene model for the input cDNA sequence.

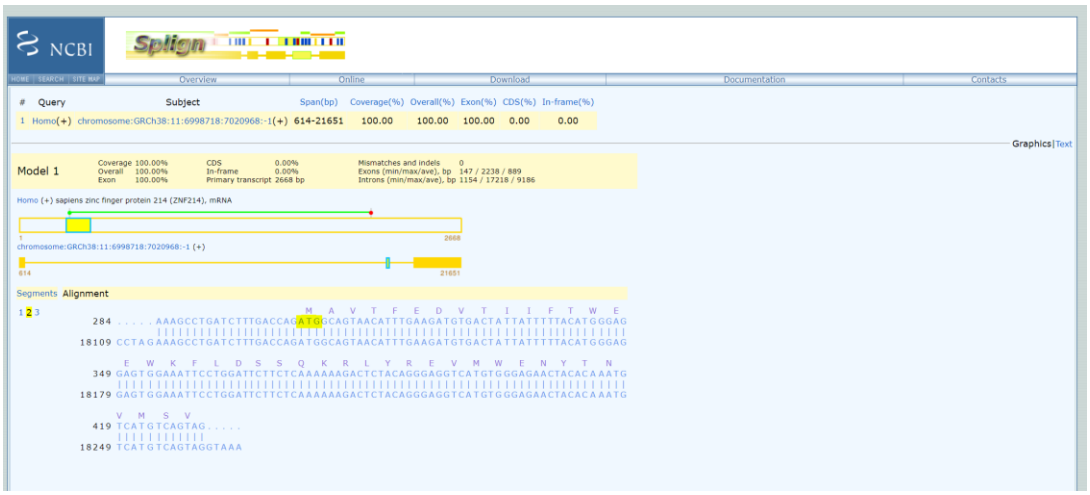
Example: Given the cDNA and the genomic sequences, Splign can be used to find the mRNA and CDS coordinates in the genomic DNA.

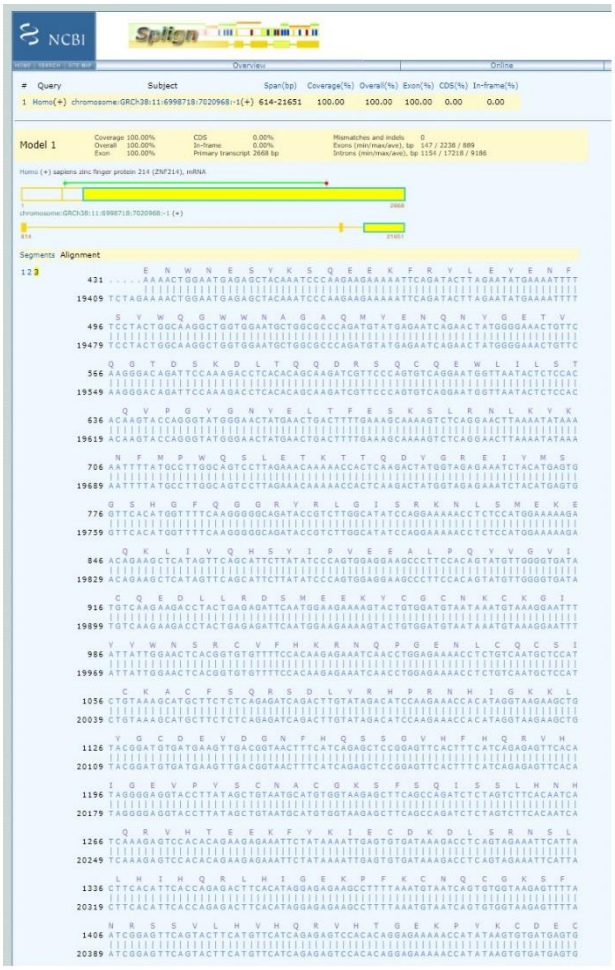
mRNA locations: 614-896, 18114- 18260, 19414-21651 (3 exons, with exon 1 being the 5'UTR)

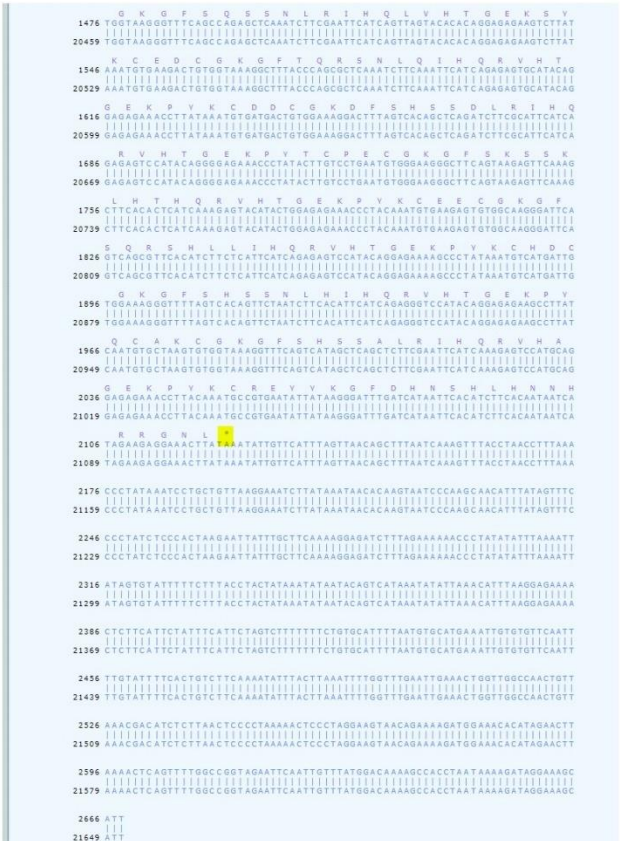
CDS locations: 303-430, 431-2124.

Three exons can be seen in the output below, with the green line indicating the coding sequence (CDS starts in Exon2 and continues into exon 3). Exon1 is the 5'UTR.

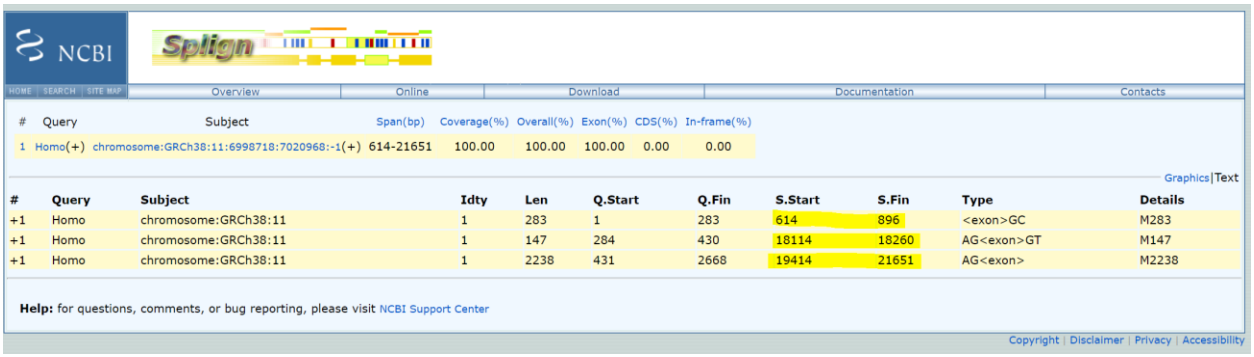



NCBI
Splign
Overview
Online
Download
Documentation
Contacts
Query Subject Span(bp) Coverage(%) Overall(%) Exon(%) CDS(%) In-frame(%)
1 Homo(+) chromosome:GRCh38:11:6998718:7020968:-1(+) 614-21651 100.00 100.00 100.00 0.00 0.00
Graphics|Text
Model 1
Coverage 100.00%
Overall 100.00%
Exon 100.00%
CDS 0.00%
In-frame 0.00%
Primary transcript 2668 bp
Mismatches and indels 0
Exons (min/max/ave), bp 147 / 2238 / 889
Introns (min/max/ave), bp 1154 / 17218 / 9186
Homo (+) sapiens zinc finger protein 214 (ZNF214), mRNA
chromosome:GRCh38:11:6998718:7020968:-1 (+)
Segments Alignment


NCBI
Splign
Overview
Online
Query Subject Span(bp) Coverage(%) Overall(%) Exon(%) CDS(%) In-frame(%)
1 Homo(+) chromosome:GRCh38:11:6998718:7020968:-1(+) 614-21651 100.00 100.00 100.00 0.00 0.00
Model 1
Homo (+) sapiens zinc finger protein 214 (ZNF214), mRNA
chromosome:GRCh38:11:6998718:7020968:-1 (+)
Segments Alignment



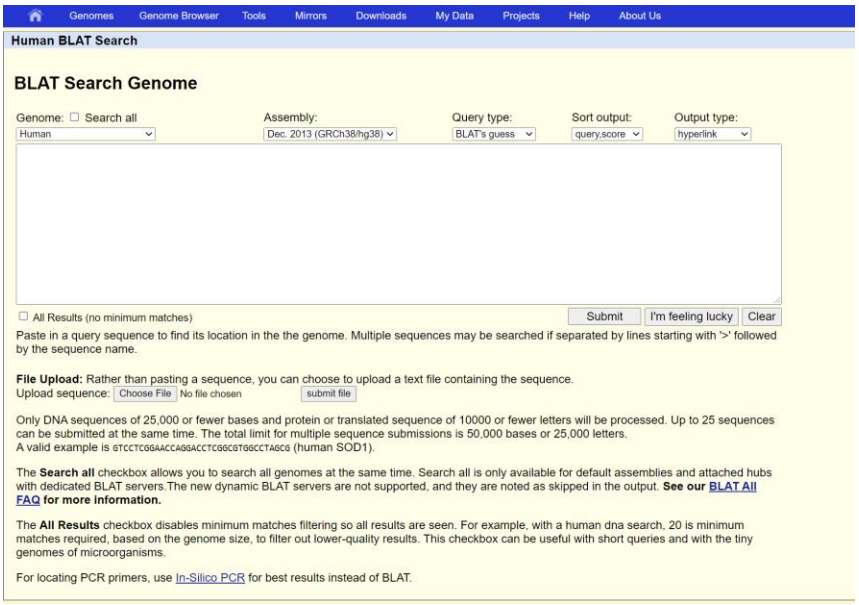
Text view of the output:



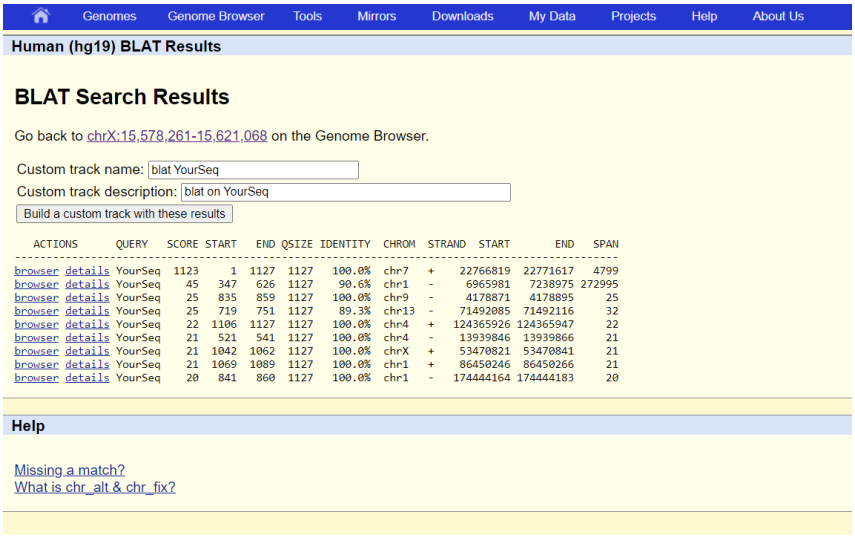
When an EST-sequence is available instead of a full-length cDNA, a genomic BLAST can be performed to align the EST to the genomic sequence in order to determine the location of the gene. Using this method, the full structure of the gene is not available.

BLAT

BLAT (BLAST-like alignment tool) at UCSC is similar to NCBI's Splign. It aligns sequences to an entire genome. But unlike Splign, BLAT can align the query sequence to anywhere in the genome and it is not necessary to specify a region of genomic DNA for alignment purposes. It does not however have a graphical interface. The user can select the organism and assembly for the input sequence. The output provides the user with links to open the elements in the UCSC genome browser and provides the score, percent identity, start and end coordinates, chromosome, strand, and span information.


Genomes
Genome Browser
Tools
Mirrors
Downloads
My Data
Projects
Help
About Us
Human BLAT Search
BLAT Search Genome
Genome:
Search all
Human
Assembly:
Dec. 2013 (GRCh38/hg38)
Query type:
BLAT's guess
Sort output:
query,score
Output type:
hyperlink
All Results (no minimum matches)
Submit
I'm feeling lucky
Clear
Paste in a query sequence to find its location in the the genome. Multiple sequences may be searched if separated by lines starting with '>' followed by the sequence name.
File Upload: Rather than pasting a sequence, you can choose to upload a text file containing the sequence.
Upload sequence:
Choose File
No file chosen
submit file
Only DNA sequences of 25,000 or fewer bases and protein or translated sequence of 10000 or fewer letters will be processed. Up to 25 sequences can be submitted at the same time. The total limit for multiple sequence submissions is 50,000 bases or 25,000 letters.
A valid example is GTCCTCGGAACCAGGACCTCGGCGTGGCCTAGCG (human SOD1).
The Search all checkbox allows you to search all genomes at the same time. Search all is only available for default assemblies and attached hubs with dedicated BLAT servers.The new dynamic BLAT servers are not supported, and they are noted as skipped in the output. See our BLAT All FAQ for more information.
The All Results checkbox disables minimum matches filtering so all results are seen. For example, with a human dna search, 20 is minimum matches required, based on the genome size, to filter out lower-quality results. This checkbox can be useful with short queries and with the tiny genomes of microorganisms.
For locating PCR primers, use In-Silico PCR for best results instead of BLAT.

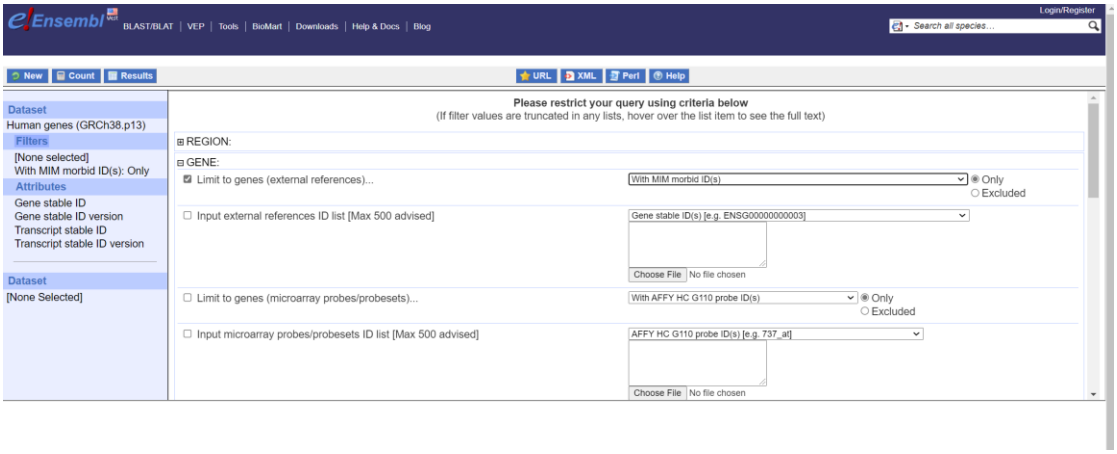
Output:


Genomes
Genome Browser
Tools
Mirrors
Downloads
My Data
Projects
Help
About Us
Human (hg19) BLAT Results
BLAT Search Results
Go back to chrX:15,578,261-15,621,068 on the Genome Browser.
Custom track name: blat YourSeq
Custom track description: blat on YourSeq
Build a custom track with these results
ACTIONS QUERY SCORE START END QSIZE IDENTITY CHROM STRAND START END SPAN
browser details YourSeq 1123 1 1127 1127 100.0% chr7 + 22766819 22771617 4799
browser details YourSeq 45 347 626 1127 90.6% chr1 - 6965981 7238975 272995
browser details YourSeq 25 835 859 1127 100.0% chr9 - 4178871 4178895 25
browser details YourSeq 25 719 751 1127 89.3% chr13 - 71492085 71492116 32
browser details YourSeq 22 1106 1127 1127 100.0% chr4 + 124365926 124365947 22
browser details YourSeq 21 521 541 1127 100.0% chr4 - 13939846 13939866 21
browser details YourSeq 21 1042 1062 1127 100.0% chrX + 53470821 53470841 21
browser details YourSeq 21 1069 1089 1127 100.0% chr1 + 86450246 86450266 21
browser details YourSeq 20 841 860 1127 100.0% chr1 - 174444164 174444183 20
Help
Missing a match?
What is chr_alt & chr_fix?

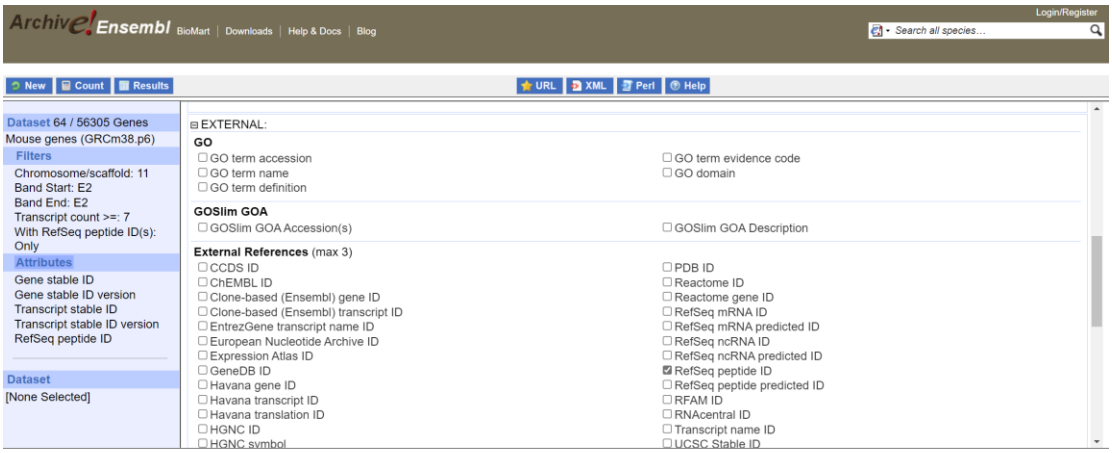
BioMart

BioMart tool is found in the top navigation panel of the **Ensembl** homepage. It allows users to download data sets based on certain criteria. The current build version (as of August 2021) is 104. “Filters” present on the left menu helps restrict the query by many parameters like region, gene, phenotype, gene ontology, variants etc. Clicking on the “Attributes” in the left menu shows us to select attribute categories. Here we can select the columns that we want in our data set. Once we have finalized on the filters and attributes, we can view the results by clicking on the “Results” button on the top left of the page. Options to download the data sets in TSV, HTML, CSV, XLS formats are available. The data set can also be previewed on the browser. Clicking on “Count” tells us the number of items in our data set.

A worked example of a scenario can be found **here.** This would familiarize one to the usage of BioMart. The example uses an older build version though.



The previous builds are archived and is available **here.**



R/Bioconductor and BiomaRt

R is an open-source programming tool to mine data from many different data sources. The **Bioconductor Project** is an open-source initiative to use R. Using R, genome scale data analysis can be accomplished more easily than by repeated lookups in the various genome browsers.

BiomaRt is a Bioconductor package that uses R to access data from data marts like Ensembl. An initial setup to use BiomaRt requires installation of R studio, Bioconductor tool biocLite, and use biocLite to install biomaRt.

The R commands are:

```
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install("biomaRt")
```

To use the library the following command needs to be run in R studio:

```
library(biomaRt)
```

One of the commonly used commands is getBM. It has four arguments – attributes, filters, values, and mart. Attributes are columns that we want in the output, filters are the limit the result set, values are specific for filters, and mart is the data source.

Example (Screen shot from R studio):

```
#use ensembl database
ensembl = useMart("ensembl")
listDatasets(ensembl)
ensembl = useDataset("hsapiens_gene_ensembl",mart=ensembl)
getBM(attributes=c("ensembl_transcript_id", "hgnc_symbol"), filters="entrezgene_id", values="6928",
                    mart=ensembl)
```

A worked example using BiomaRt is below:

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

First table: Entrez Gene ID, HGNC symbol, Ensembl Gene ID
Second table: HGNC symbol, Ensembl Gene ID, Ensembl Transcript ID

The first five entries in the OMIM database for the search string “huntington's disease” are shown in the screenshot below. Their record identifiers correspond to 603218, 604802, 143100, 606438, and 607136.

Screen shot of the first five records from the OMIM database search:

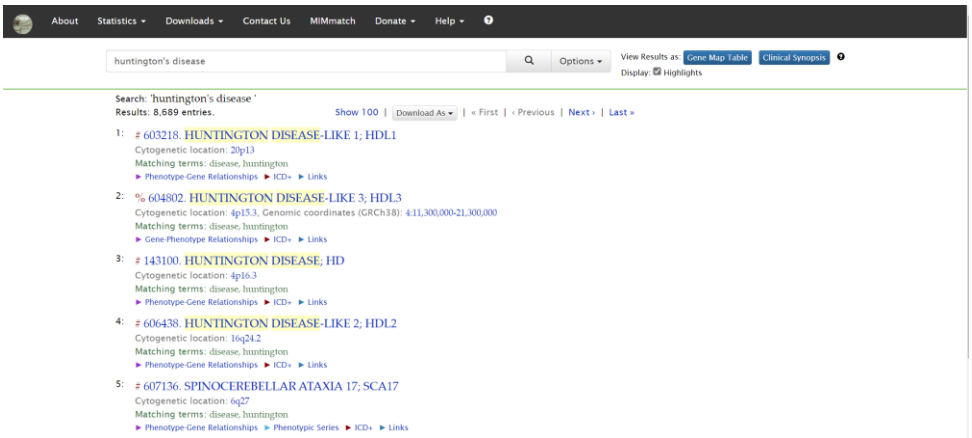


Table 1:

R code for Table 1:

```
library(biomaRt)
listMarts()

#use ensembl database
ensembl = useMart("ensembl")
listDatasets(ensembl)
ensembl = useDataset("hsapiens_gene_ensembl",mart=ensembl)

filters = listFilters(ensembl)
filters[1:100,]
attributes = listAttributes(ensembl)
attributes[1:300,]

#Table 1
omim_ids = c("603218", "604802", "143100", "606438","607136")
getBM(attributes=c("entrezgene_id","hgnc_symbol", "ensembl_gene_id"),
      filters=c("mim_morbid_accession"),
      values=list(omim_ids),
      mart = ensembl)
```

Table1 Output:

	entrezgene_id	hgnc_symbol	ensembl_gene_id
1	3064	HTT	ENSG00000197386
2	5621	PRNP	ENSG00000171867
3	57338	JPH3	ENSG00000154118
4	6908	TBP	ENSG00000112592

Table 2:

R code for table 2:

```
library(biomaRt)
listMarts()

#use ensembl database
ensembl = useMart("ensembl")
listDatasets(ensembl)
ensembl = useDataset("hsapiens_gene_ensembl",mart=ensembl)

filters = listFilters(ensembl)
filters[1:100,]
attributes = listAttributes(ensembl)
attributes[1:300,]

#Table 2
omim_ids = c("603218", "604802", "143100", "606438","607136")
getBM(attributes = c("hgnc_symbol","ensembl_gene_id","ensembl_transcript_id"),
      filters = c("mim_morbid_accession"),
      values = list(omim_ids),
      mart = ensembl)
```

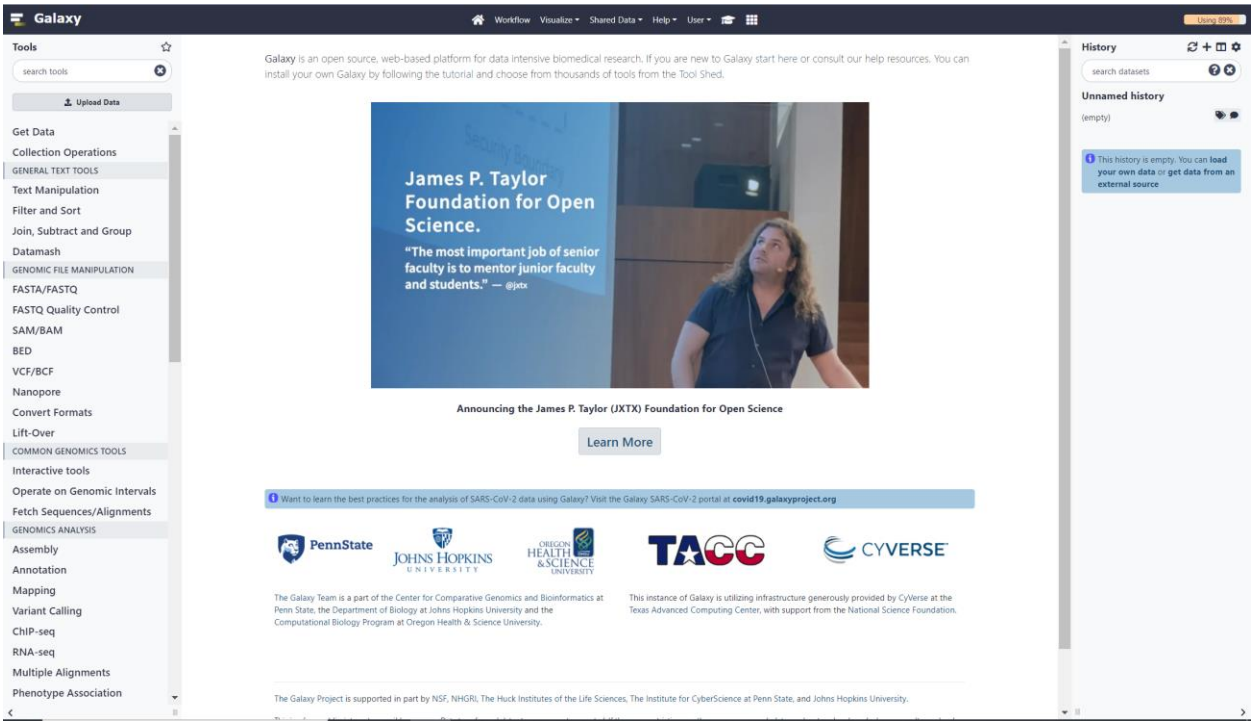
Table 2 output:

	hgnc_symbol	ensembl_gene_id	ensembl_transcript_id
1	HTT	ENSG00000197386	ENST00000680239
2	HTT	ENSG00000197386	ENST00000680956
3	HTT	ENSG00000197386	ENST00000680360
4	HTT	ENSG00000197386	ENST00000681528
5	HTT	ENSG00000197386	ENST00000647962
6	HTT	ENSG00000197386	ENST00000649900
7	HTT	ENSG00000197386	ENST00000680291
8	HTT	ENSG00000197386	ENST00000355072
9	HTT	ENSG00000197386	ENST00000648150
10	HTT	ENSG00000197386	ENST00000506137
11	HTT	ENSG00000197386	ENST00000512909
12	HTT	ENSG00000197386	ENST00000510626
13	HTT	ENSG00000197386	ENST00000649131
14	HTT	ENSG00000197386	ENST00000509618
15	HTT	ENSG00000197386	ENST00000650588
16	HTT	ENSG00000197386	ENST00000650595
17	HTT	ENSG00000197386	ENST00000513639
18	HTT	ENSG00000197386	ENST00000513326
19	HTT	ENSG00000197386	ENST00000509043
20	HTT	ENSG00000197386	ENST00000509751
21	HTT	ENSG00000197386	ENST00000512068
22	HTT	ENSG00000197386	ENST00000513806
23	HTT	ENSG00000197386	ENST00000508321
24	PRNP	ENSG00000171867	ENST00000430350
25	PRNP	ENSG00000171867	ENST00000379440
26	PRNP	ENSG00000171867	ENST00000424424
27	PRNP	ENSG00000171867	ENST00000457586
28	JPH3	ENSG00000154118	ENST00000537256
29	JPH3	ENSG00000154118	ENST00000301008
30	JPH3	ENSG00000154118	ENST00000284262
31	JPH3	ENSG00000154118	ENST00000563609
32	TBP	ENSG00000112592	ENST00000421512
33	TBP	ENSG00000112592	ENST00000230354
34	TBP	ENSG00000112592	ENST00000423353
35	TBP	ENSG00000112592	ENST00000636632
36	TBP	ENSG00000112592	ENST00000446829
37	TBP	ENSG00000112592	ENST00000392092
38	TBP	ENSG00000112592	ENST00000540980
39	TBP	ENSG00000112592	ENST00000616883

Galaxy

Galaxy is an open-source framework integrating command-line tools and database information giving the users the choice to perform various complex analysis on genomic data without the need to learn command line syntax. Users can create accounts, generate workflows of their analysis, and share them with peers. It is used by both beginners and seasoned programmers due to its ease of use and integration with other data sources like UCSC.

Galaxy interface looks like the one in the screen shot below:

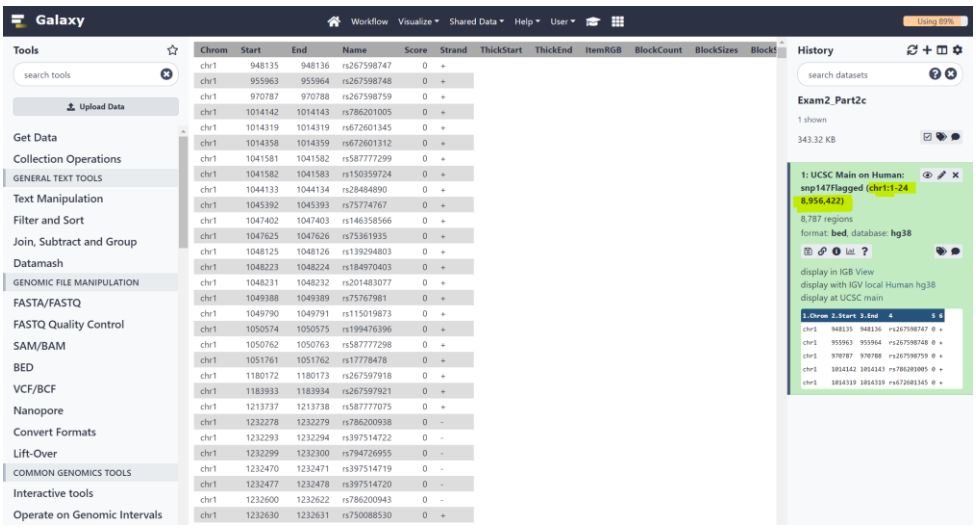


The tutorials on Galaxy also called screencasts can be found **here**. A detailed walk through of a scenario is found at **Galaxy 101**.

Example: Use Galaxy or any other tool (hg38) to find all UCSC flagged SNPs in (hg38/db147) on chromosome 1. Be sure to import the data as a BED file. Describe the result (number of lines, what's in columns, etc.).

Click on Get Data -> UCSC Table Browser. Put in the necessary details in the Table Browser and import the data into Galaxy. The BED file has 8787 lines. The columns in the BED file are chrom (name of the chromosome on which the feature is present), start (starting position

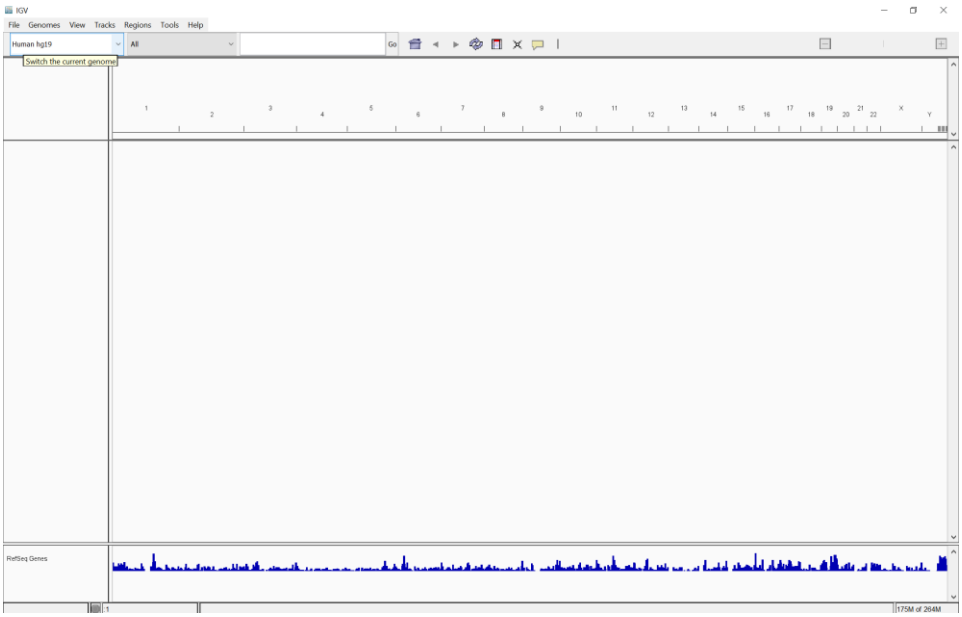
of the feature in the chromosome), end (ending position), name (name of the BED feature), score (range 0 to 100), and strand (either ‘+’ or ‘-‘).



IGV (Integrated Genome Viewer)

IGV available through the Broad Institute is like the other genome browsers, with the added advantage that it can be locally downloaded and can be run as a standalone application. It allows us to view at high resolution for small regions and low resolution at wider regions. IGV is designed to handle very large genomic-sized data sets. IGV allows for custom data to be uploaded in BED and WIG file formats.

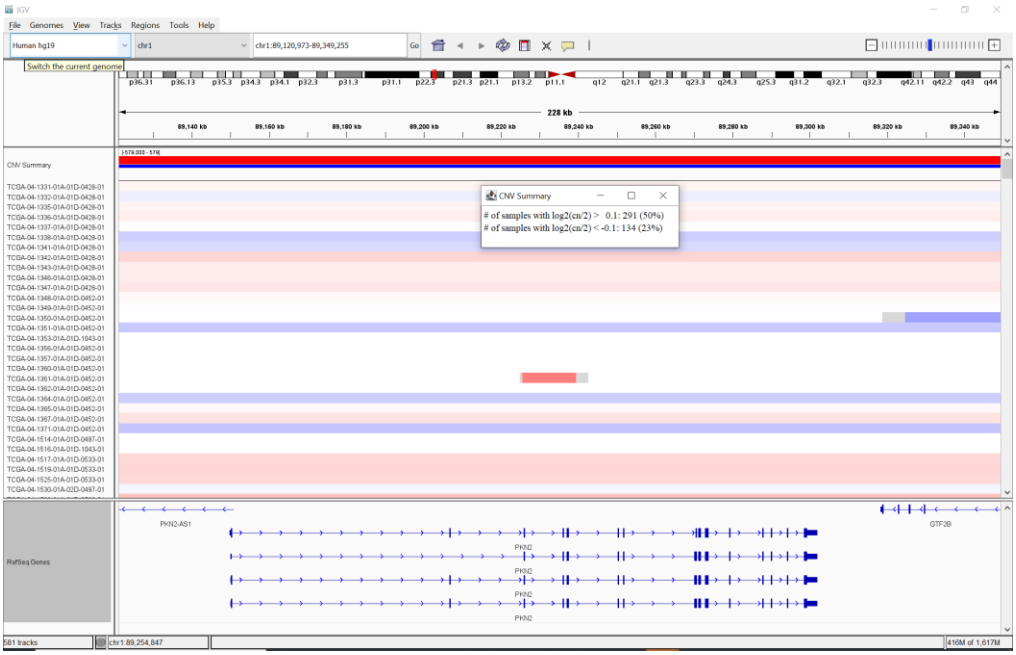
IGV desktop interface:



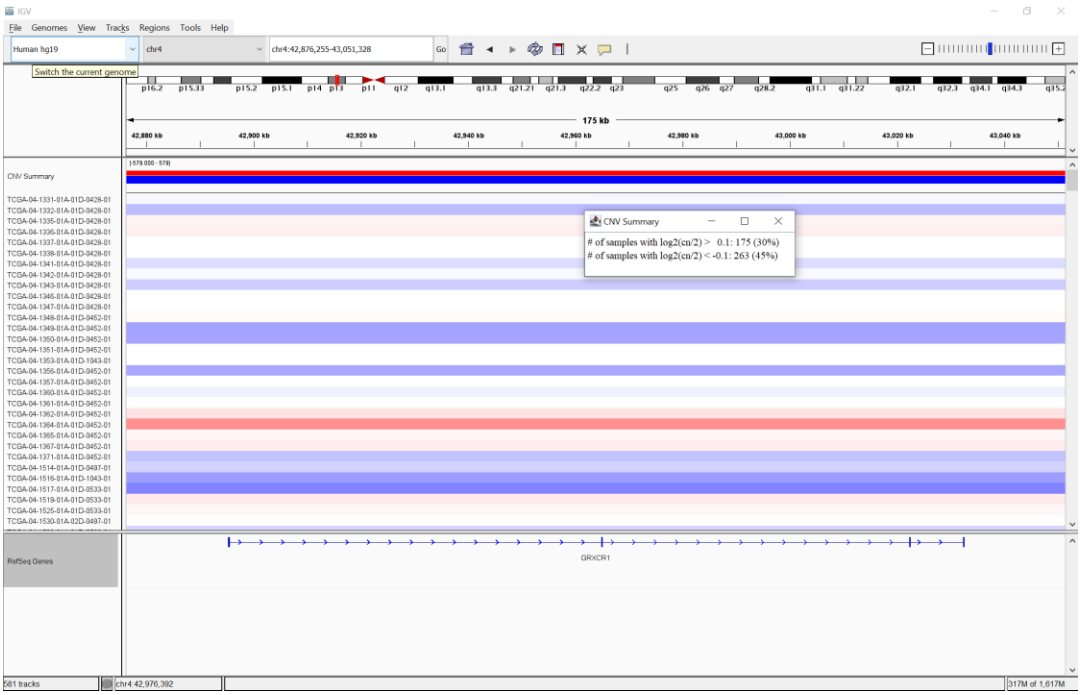
Example: Using IGV, load the Firehose (TCGA) data from January 28, 2016 for OV-TP (ovarian cancer) using hg19. The following four genes have been shown to be associated with CNV in some forms of ovarian cancer: PKN2, GRXCR1, PRKN (or PARK2), and PPIAL4A. In a table, qualitatively evaluate the CNV summary (minus germline) in each gene's region (e.g. "even blue and red", "twice as much blue as red", "overwhelmingly red").

Gene	Red %	Blue %	
PKN2	50	23	For the PKN2 gene, red is twice as much as blue, indicating that in 50% of patients the CNV increased, and 23% patients showed decrease in CNV.
GRXCR1	30	45	For GRXCR1gene, the blue is more than red, which means that there are more patients with decreased CNV (45%) than with an increase in CNV (30%).
PRKN	17	64	For PRKN gene, the CNV shows overwhelmingly blue, the blue is greater than red by more than 3 times, indicating that in majority of the patients (64%) there was a decrease in CNV compared to 17% of patients that showed increase in CNV.
PPIAL4A	60	2%	For the PPIAL4A gene, CNV shows overwhelmingly red with 60% patients showing an increase in CNV, and only 2% showing a decrease in CNV.

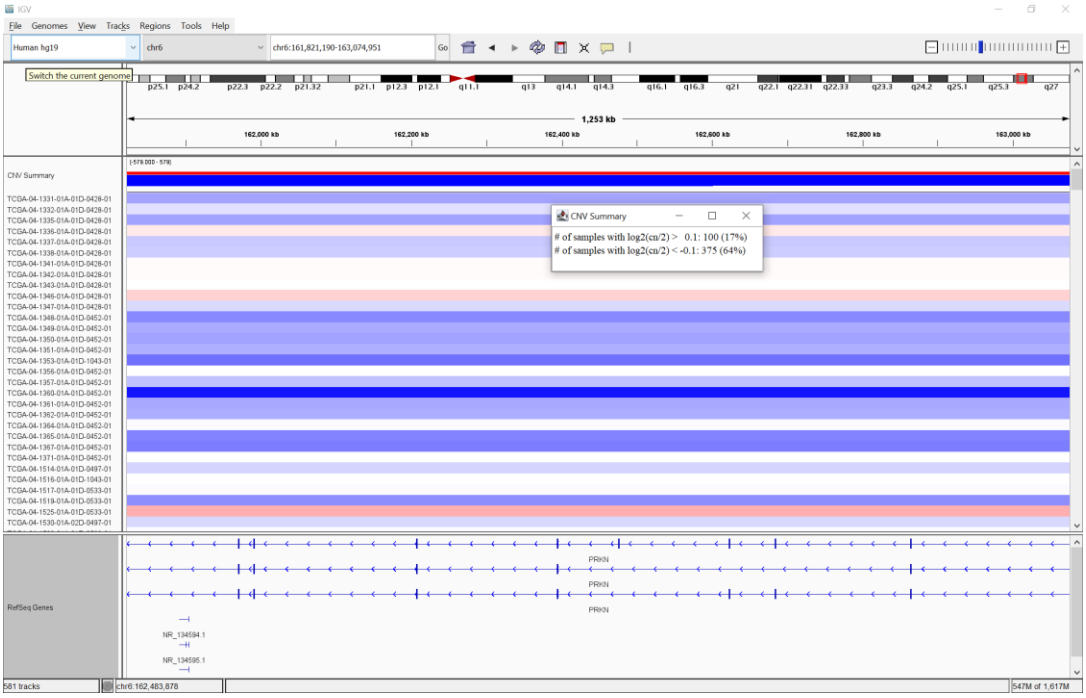
PKN2 gene CNV summary:



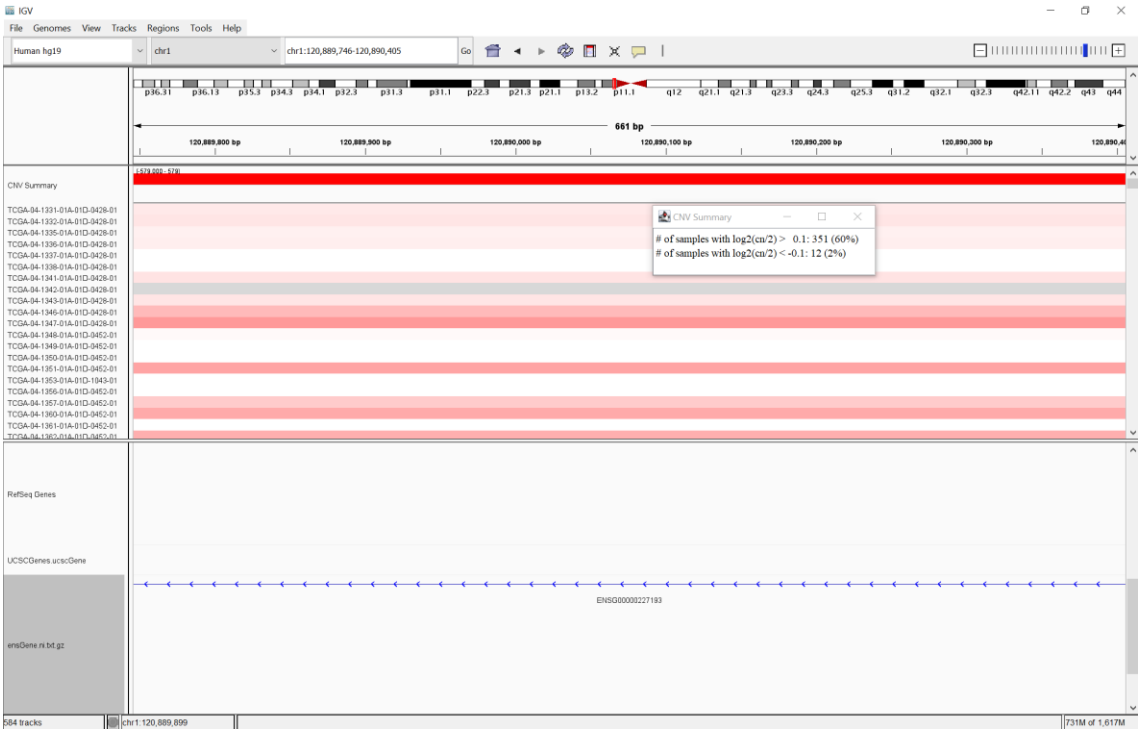
GRXCR1gene CNV summary:



PRKN gene CNV summary:



PPIAL4A gene CNV summary:



BEDtools

Bedtools is a popular collection of tools that allow us to perform operations on genomic data. Bedtools are available in Galaxy, but command-line version is also available. For Mac OS and Linux, instructions to install BEDtools can be found **here**.

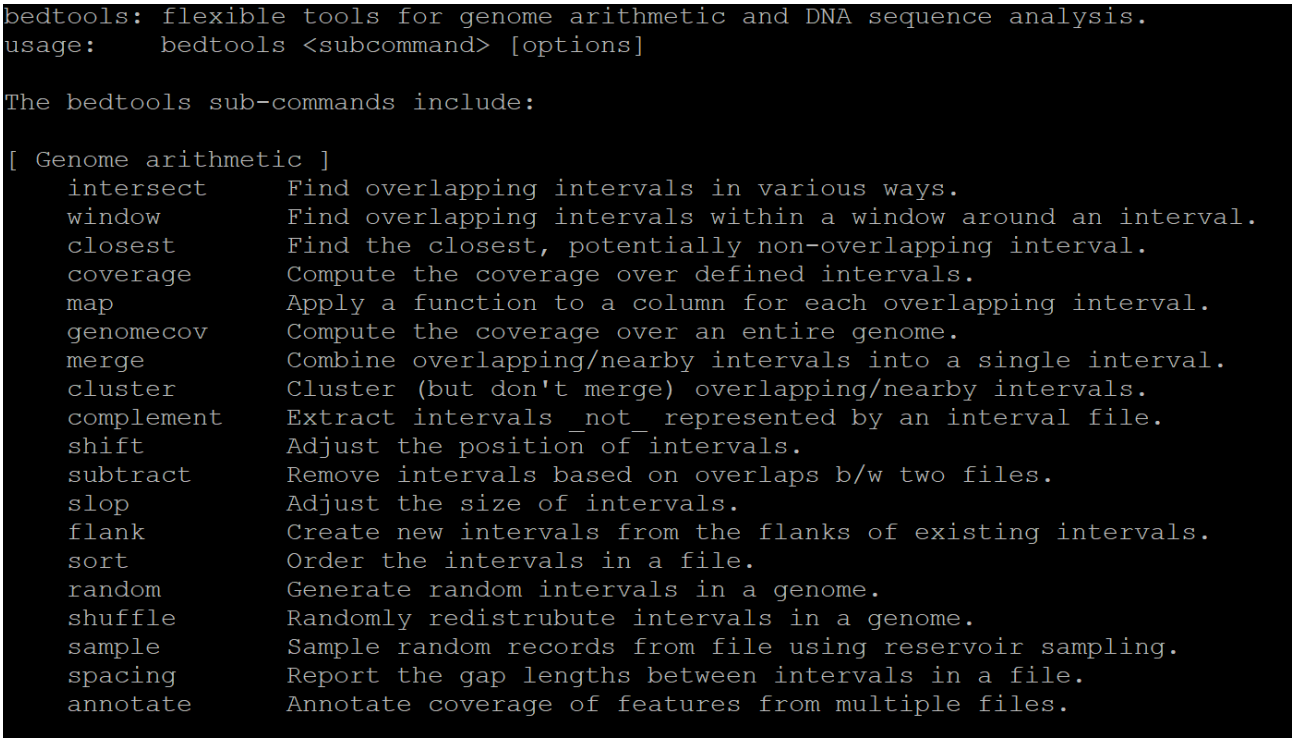
BEDtools intersect is a popular interval manipulation operation, with a detailed documentation found **here**.

Example: Find the closest exons to gaps on chromosome 20. To do so, run the following command.

bedtools closest -a hs_chr20_gaps.bed -b hs_chr20_refseq.bed > hs_chr20_nearest_exon_gap.bed

This creates an output file called hs_chr_20_nearest_exon_gap.bed

The operations that can be performed with bedtools are shown in the screen shot below:



SAMtools

SAMtools are tools for alignments in the SAM format. It is available to be downloaded **here**. The same is also available through Galaxy interface. SAMtools can be used to call SNPs in genomic alignments. BAM files can be displayed in IGV.

Some of the SAMtools commands are:

SAM-to-BAM conversion:

Step 1: Create a BAM file using *view* command. Below, -b specifies that the output is a BAM file, -S indicates that the input is a SAM file.

samtools view -bS filename.sam >filename.bam

Step 2: Sorting the BAM file using the *sort* command. In Galaxy, *SAMtools* does the sorting.

samtools sort filename.bam filename.sorted

Step 3: Indexing the sorted BAM file. The *index* command will create the BAM index file. Both the BAM and the BAI (BAM index file) must be downloaded. IGV will look for the .bai file with the same name as the BAM file in the same folder.

Below are the various operations that can be performed using command line SAMtools:

```
Usage:   samtools <command> [options]

Commands:
  -- Indexing
     dict           create a sequence dictionary file
     faidx          index/extract FASTA
     fqidx          index/extract FASTQ
     index          index alignment

  -- Editing
     calmd          recalculate MD/NM tags and '=' bases
     fixmate        fix mate information
     reheader       replace BAM header
     targetcut      cut fosmid regions (for fosmid pool only)
     addreplacerg   adds or replaces RG tags
     markdup        mark duplicates

  -- File operations
     collate        shuffle and group alignments by name
     cat            concatenate BAMs
     merge          merge sorted alignments
     mpileup        multi-way pileup
     sort           sort alignment file
     split          splits a file by read group
     quickcheck     quickly check if SAM/BAM/CRAM file appears intact
     fastq          converts a BAM to a FASTQ
     fasta          converts a BAM to a FASTA

  -- Statistics
     bedcov         read depth per BED region
     depth          compute the depth
     flagstat       simple stats
     idxstats       BAM index stats
     phase          phase heterozygotes
     stats          generate stats (former bamcheck)

  -- Viewing
     flags          explain BAM flags
     tview          text alignment viewer
     view           SAM<->BAM<->CRAM conversion
     depad          convert padded BAM to unpadded BAM
```

Next-Generation Sequencing (NGS) analysis

Protocol to perform NGS analysis to identify SNPs (variant calling) in NGS data from the 1000 Genomes Project (reference genome hg19). All the steps below can be performed within **Galaxy** or using command-line tools.

- Two FASTQ files from the 1000 Genome Project is used. The FASTQ files represent data from paired-end sequencing experiment. The forward reads usually are in the file ending in '_1', and the reverse reads are in the file ending in '_2'. Links to the sample data files are given below.
- **Data upload to Galaxy**: Load both files into Galaxy using the **Upload file** tool, choosing **Paste/Fetch data**, and pasting in the given ftp links. The data type should be set to 'fastq' and the genome should be set to 'hg19'.

  Forward reads:

  ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/phase3/data/HG00117/sequence_read/SRR044234_1.filt.fastq.gz

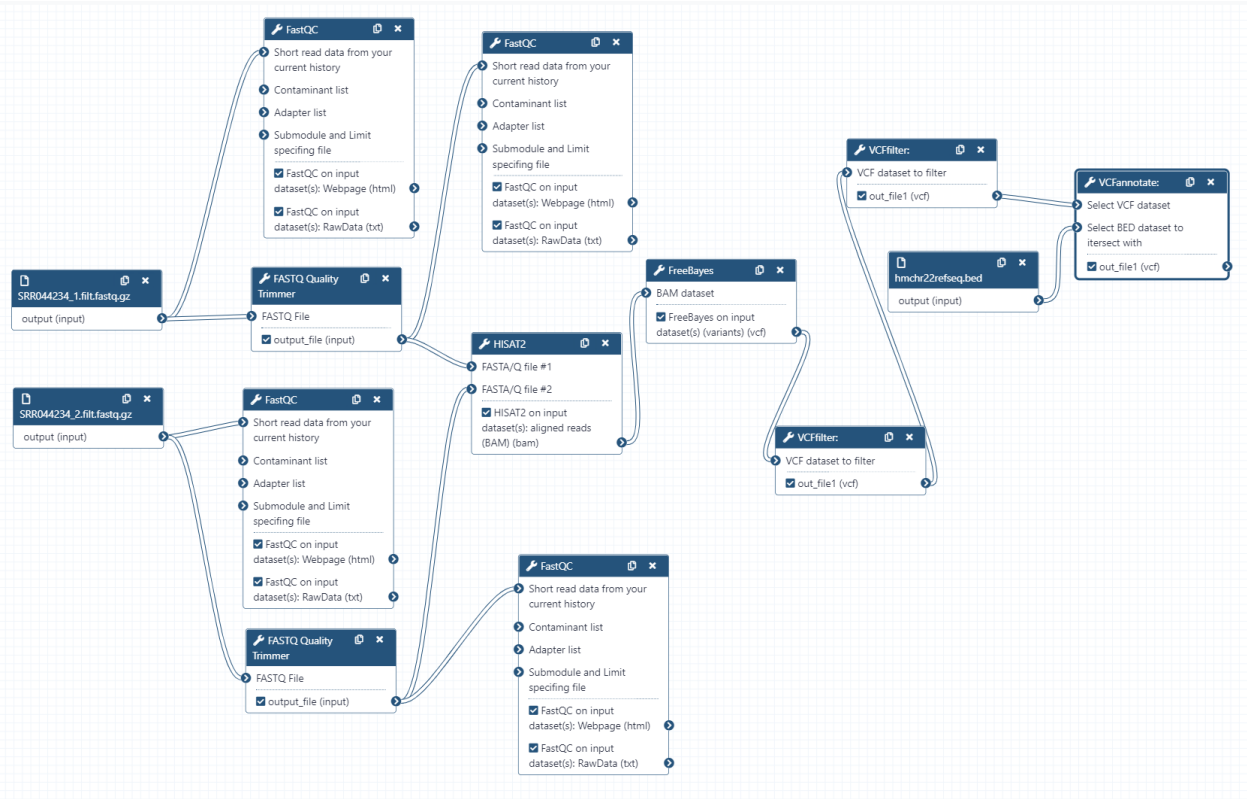
  Reverse reads:

  ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/phase3/data/HG00117/sequence_read/SRR044234_2.filt.fastq.gz

- **Determine quality encoding**: Run **FASTQC** on both files. If the quality encoding is found to be Sanger/Illumina 1.9, update the file type to 'fastqsanger'. If the quality encoding is found to be something other than Sanger/Illumina 1.9, use **FASTQ Groomer** to convert the files to Sanger/Illumina 1.9 encoding. At the end, we will get two FASTQ files in Sanger/Illumina 1.9 encoding.
- **Trim low-quality bases**: Either **FASTQ Quality Trimmer** or **Trimmomatic** tool is used to remove low quality bases from each file. A window size of 4 bases and average quality in the window can be set based on our requirement (20 or 30).
- **FASTQC** is rerun on the trimmed data to ensure that low quality bases were removed.
- **Align reads to reference genome**: There are many alignment tools to align the FASTQ files to the reference genome. **BWA**, **Bowtie2**, or **HISAT** are different alignment tools. Since data is from a paired-end experiment, we must align the reads as paired-end. This step generates the alignments in BAM format.
- **Identify variants**: There are many variant-calling approaches. Some of these tools are FreeBayes, SAMtools (both available in Galaxy), **GATK Unified Genotyper**, **GATK haplotype Caller**, and **Platypus**. The output can be restricted to a specific chromosome to make the result file more manageable. (Ex: Limit the output to chr22:0-51304566)

- **Filter and annotate variants**: Use the **VCFfilter** tool to filter for variants that show heterozygosity (estimated allele frequency = 0.5) and have more than 10 reads covering them (total read depth > 10). The tag IDs for these parameters can be found in the header of the VCF file. To annotate which genes the variants are in, we need to bring in RefSeq genes in BED format from **UCSC Main**. Then, using the **VCFannotate** tool we can intersect the filtered VCF file with the BED annotations.

Workflow diagram from Galaxy:



ChIP-seq data analysis

The raw ChIP-seq unaligned reads in the FASTQ format are uploaded into Galaxy. The data is then groomed (FASTQC) and trimmed (Trimmomatic) after which it is aligned to a reference genome. Bowtie2 and BWA are a couple of alignment programs that can perform the alignment. The next step is to perform 'peak calling' which identifies the peaks in the reads that are aligned to the reference genome. The peaks represent regions in the DNA that are bound to

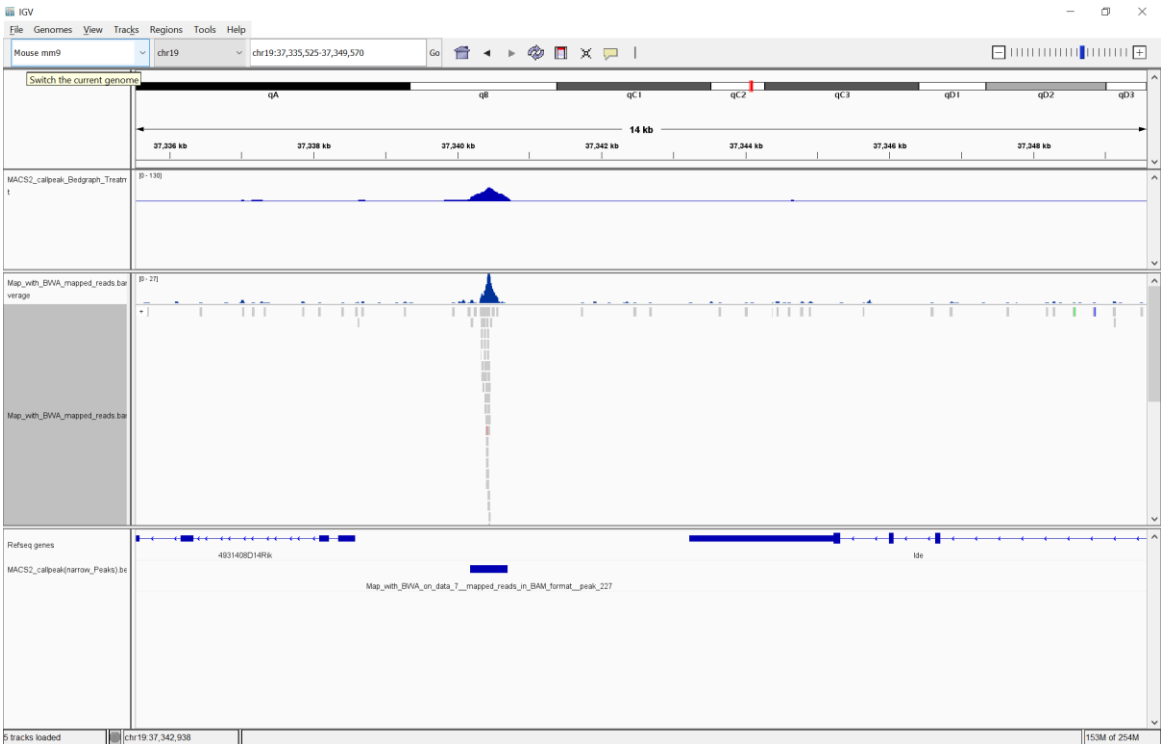
the proteins that was immunoprecipitated. MACS2 and Sicer are two popular peak callers. The input to the program is a BAM file, and the output produced is a BED (Browser extensible data) or WIG (Wiggle) format file. The peak caller used depends on factors such as presence of replicates, control experiments, single/paired end, and type of protein immunoprecipitated. The necessary tools are available in Galaxy or can be executed from command-line. If we need to determine the binding changes under certain conditions or cell types, differential peak calling is to be used. A ChIP-seq tutorial in Galaxy is available **here**.

Protocol to perform ChIP-seq data analysis (All the necessary tools are available in Galaxy):

- Import the FASTQ file of ChIP-seq reads from mouse (mm9). The full dataset was downsampled to the subset reads from chr19.
- Run **FASTQC** to determine the quality score encoding.
- Run **FASTQ Groomer** to convert the file to Sanger/Illumina 1.9 phred encoding ONLY IF NEEDED.
- Run **Trimmomatic** and set the minimum phred score in a 4 nt sliding window to 25.
- Re-run **FASTQC** to check the quality scores and encoding scheme.
- Run **Map with BWA** with default settings (make sure to select for single-end reads), aligning against the mouse genome version mm9.
- Run **MACS2 callpeak** on the BAM file, setting the Effective genome size to the mouse genome. Choose "Scores in bedGraph files (--bdg)" option under additional outputs. Use default settings for the rest of the parameters and leave the control field blank.
- Load the MACS2 Bedgraph Treatment file and narrow Peaks BED file from previous step and aligned BAM file to IGV or UCSC.
- Focus on chromosome 19 and look for a gene locus that has ChIP peaks nearby.

  The MACS2 results can be viewed in IGV browser. It shows the ChIP peaks. There are two genes in this vicinity – the Ide (insulin degrading enzyme) gene and the RIKEN cDNA 4931408D14 gene. We can analyze ChIP-seq data pertaining to histone modifications that has greater coverage over the entire gene region using the MACS2 'broad peaks' option. 'Narrow peaks' can be for transcription factor data since the region that is bound would be narrower (Mistry & Khetani, n.d.).

BEDgraph treatment file, MACS2 narrow peaks file, and the aligned BAM file viewed in the IGV browser:



RNA-seq data analysis

RNA-seq analysis is used to study the transcriptome of an organism that have a reference genome available. RNA-seq reads are aligned to the genome to determine the origination of the RNA molecule. Many tools in the TopHat suite are widely used to map and analyze RNA-seq data. TopHat suite also includes the Bowtie aligner. In case of organisms with no well-annotated genome, de novo assemblers need to be used (Ex: Trinity, available in Galaxy). These tools align the RNA-seq reads to each other based on overlapping segments. This generates a full RNA molecule.

Other tools like Cufflinks (available in Galaxy), assembles the transcripts, and can perform differential expression analysis and/or study the regulation in RNA-seq sample data. The accepted_hits.bam file from TopHat is provided as input to Cufflinks. The output is a GTF file with gene locations and names. There are two additional outputs – a table with gene expression and another with transcript expression levels.

Cuffcompare is used to compare two or more GTF files from Cufflinks. This allows one to compare the gene expression under various conditions like cell types, developmental stages etc.

Cuffdiff performs statistical analysis of differential expression between two datasets. It can also be used to perform differential expression of transcripts, exon expression, differential usage of transcription start sites.

Cuffcompare and Cuffdiff are part of the Cufflinks tools available in Galaxy.

DESeq2 – is a tool that determines differentially expressed features from count tables (obtained using featureCounts).

A Galaxy tutorial performing de novo transcriptome reconstruction with RNA-Seq is available **here**. The steps involved in the protocol are listed below:

- **Data upload**.
  file type: fastqsanger, genome:mm10
  https://zenodo.org/record/583140/files/G1E_rep1_forward_read_%28SRR549355_1%29
  https://zenodo.org/record/583140/files/G1E_rep1_reverse_read_%28SRR549355_2%29
  https://zenodo.org/record/583140/files/G1E_rep2_forward_read_%28SRR549356_1%29
  https://zenodo.org/record/583140/files/G1E_rep2_reverse_read_%28SRR549356_2%29
  https://zenodo.org/record/583140/files/Megakaryocyte_rep1_forward_read_%28SRR549357_1%29
  https://zenodo.org/record/583140/files/Megakaryocyte_rep1_reverse_read_%28SRR549357_2%29
  https://zenodo.org/record/583140/files/Megakaryocyte_rep2_forward_read_%28SRR549358_1%29
  https://zenodo.org/record/583140/files/Megakaryocyte_rep2_reverse_read_%28SRR549358_2%29

  file type: gtf, genome:mm10
  https://zenodo.org/record/583140/files/RefSeq_reference_GTF_%28DSv2%29

- Run FastQC on the forward and reverse read files to assess the quality of the reads.
- Run Trimmomatic to trim off low quality bases from the ends of the reads.
  - "Single-end or paired-end reads?": Paired-end (two separate input files)
    - "Input FASTQ file (R1/first of pair)": G1E_rep1 forward read
    - "Input FASTQ file (R2/second of pair)": G1E_rep1 reverse read
  - "Perform initial ILLUMINACLIP step?": No
- Re-run FASTQC on trimmed ends to inspect the differences.

Mapping – performed to understand the positions of the reads within the genome.

- Run HISAT2 on one forward/reverse read pair with the following settings:
  - "Source for the reference genome": Use a built-in genome

        - "Select a reference genome": Mouse (Mus Musculus): mm10
    - "Single-end or paired-end reads?": Paired-end
        - "FASTA/Q file #1": Trimmomatic on G1E_rep1 forward read (R1 paired)
        - "FASTA/Q file #2": Trimmomatic on G1E_rep1 reverse read (R2 paired)
        - "Specify strand information": Forward(FR)
    - "Advanced options"
        - "Spliced alignment options"
            - "Penalty for non-canonical splice sites": '3'
            - "Penalty function for long introns with canonical splice sites": 'Constant [f(x) = B]''
            - "Constant term (B)": '0.0'
            - "Penalty function for long introns with non-canonical splice sites": 'Constant [f(x) = B]''
            - "Transcriptome assembly reporting": 'Report alignments tailored for transcript assemblers including StringTie'

- Run HISAT2 on the remaining forward and reverse read pairs with the same parameters.

De novo transcriptome reconstruction:

- Run Stringtie on the HISAT2 alignments using the default parameters.
    - Use batch mode to run all four samples from one tool form.
    - "Specify strand information": Forward (FR)

Transcriptome assembly:

- Run Stringtie-merge on the Stringtie assembled transcripts along with the RefSeq annotation file.
    - "Transcripts": all four Stringtie assemblies.
    - "Reference annotation to include in the merging": RefSeq_reference_GTF
- Run GFFCompare on the Stringtie-merge generated transcriptome along with the RefSeq annotation file.
    - "GTF inputs for comparison": output of Stringtie-merge
    - "Use Reference Annotation": Yes
        - "Choose the source for the reference annotation": History
        - "Reference Annotation": RefSeq_reference_GTF
    - "Use Sequence Data": Yes
        - "Choose the source for the reference list": Locally cached
        - "Using reference genome": 'Mouse (Mus Musculus): mm10'

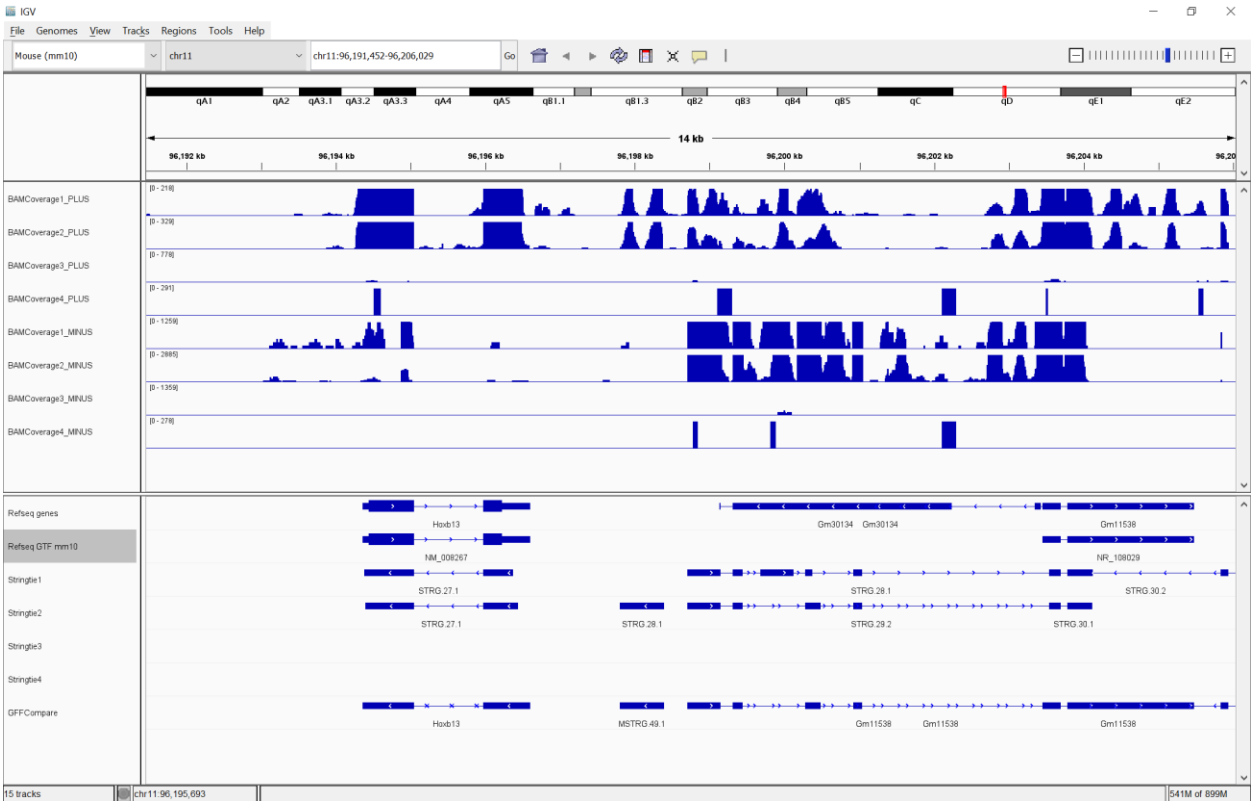
Differential gene expression testing:

- Run DESeq2 with the following parameters:
  - "1: Factor"
    - "1: Factor level": G1E
      - "Counts file(s)": featureCount files corresponding to the two G1E replicates
    - "2: Factor level": Mega
      - "Counts file(s)": featureCount files corresponding to the two Mega replicates

  Along with the list of genes, DESEQ2 produces a graphical summary file that is useful to determine the quality of the experiment.

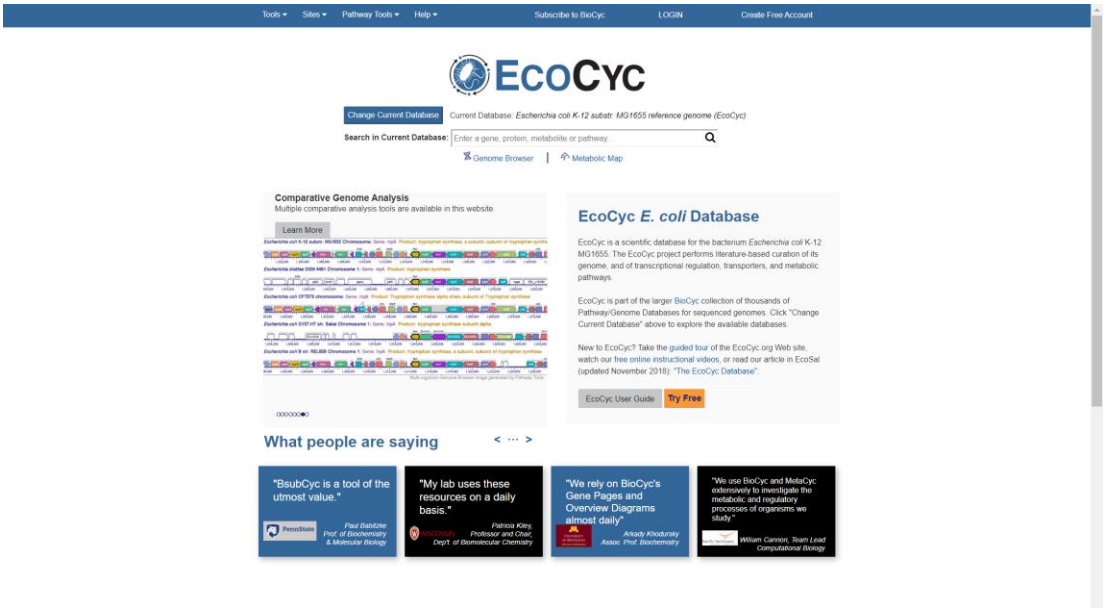
- **Visualization**: use the built-in genome browser in Galaxy, Trackster, or UCSC genome browser or IGV to view the BAM alignment file, Stringtie output files, GFFCompare file and the reference GTF file. The known and novel transcripts from G1E and Megakaryocytes can be viewed in the genome browser.

Below is the screenshot with the outputs from the above steps loaded in to IGV:

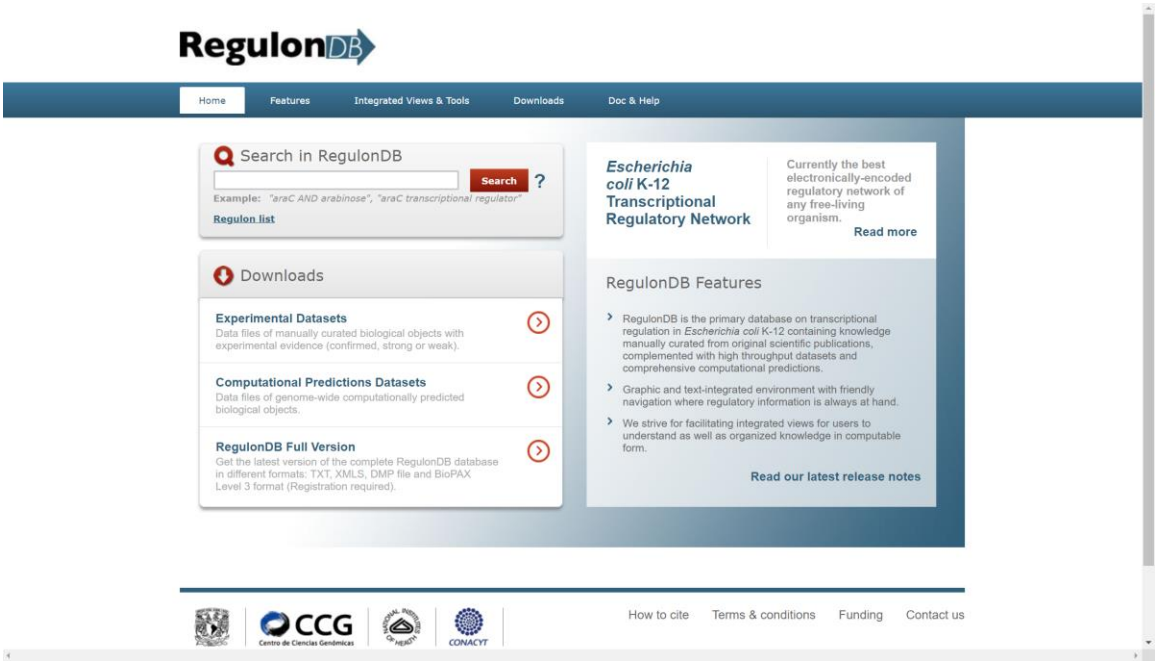


Genomic Databases and Genome browsers

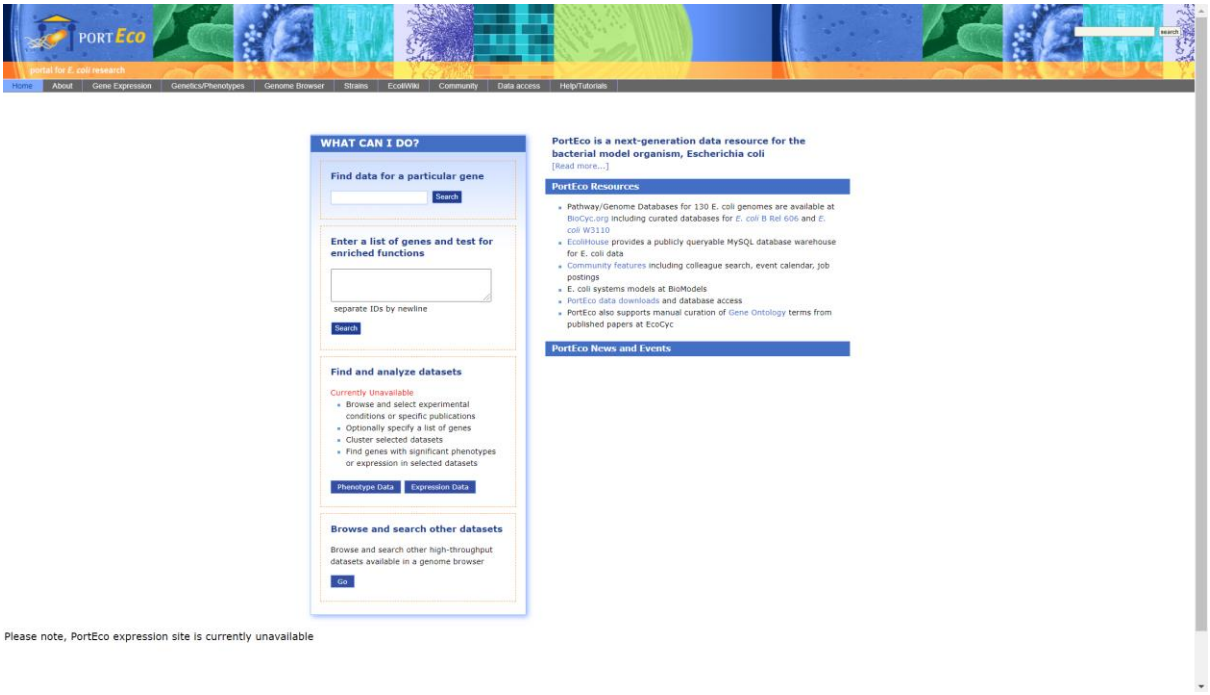
EcoCyc – database for bacterium *Escherichia coli* K-12 MG1655



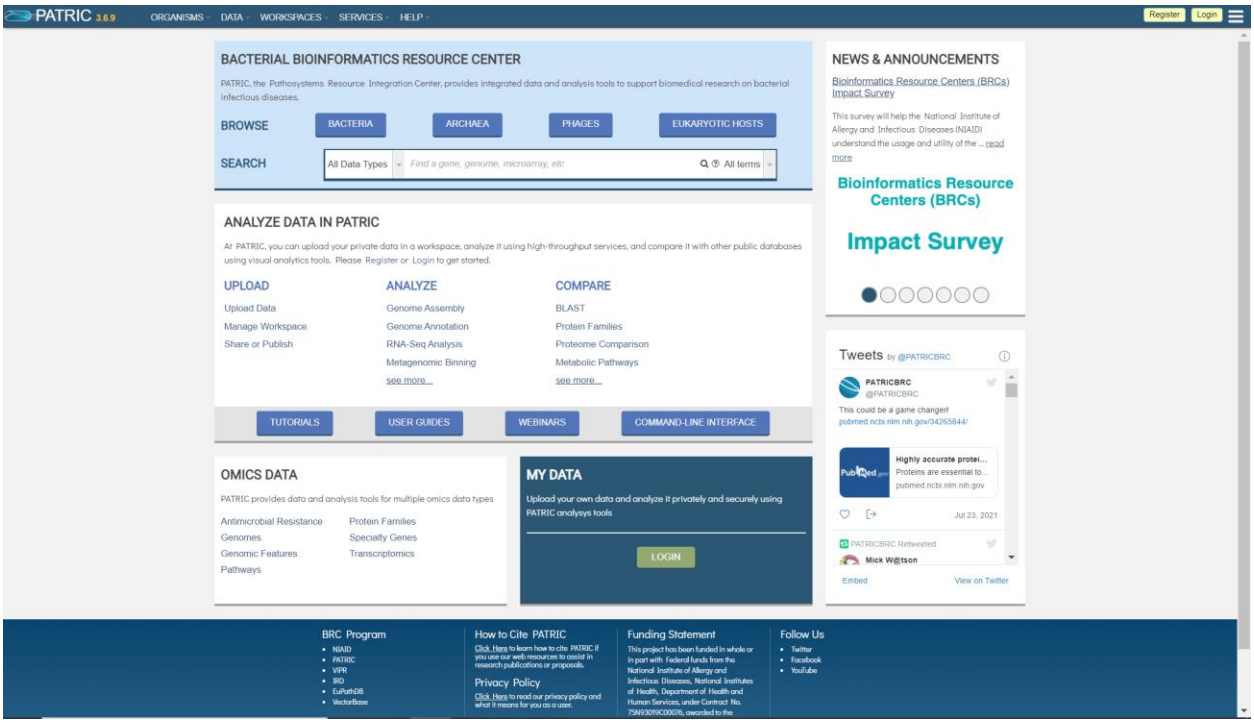
RegulonDB – is the primary database containing transcriptional regulation data in Escherichia coli k-12 manually curated from scientific publications.



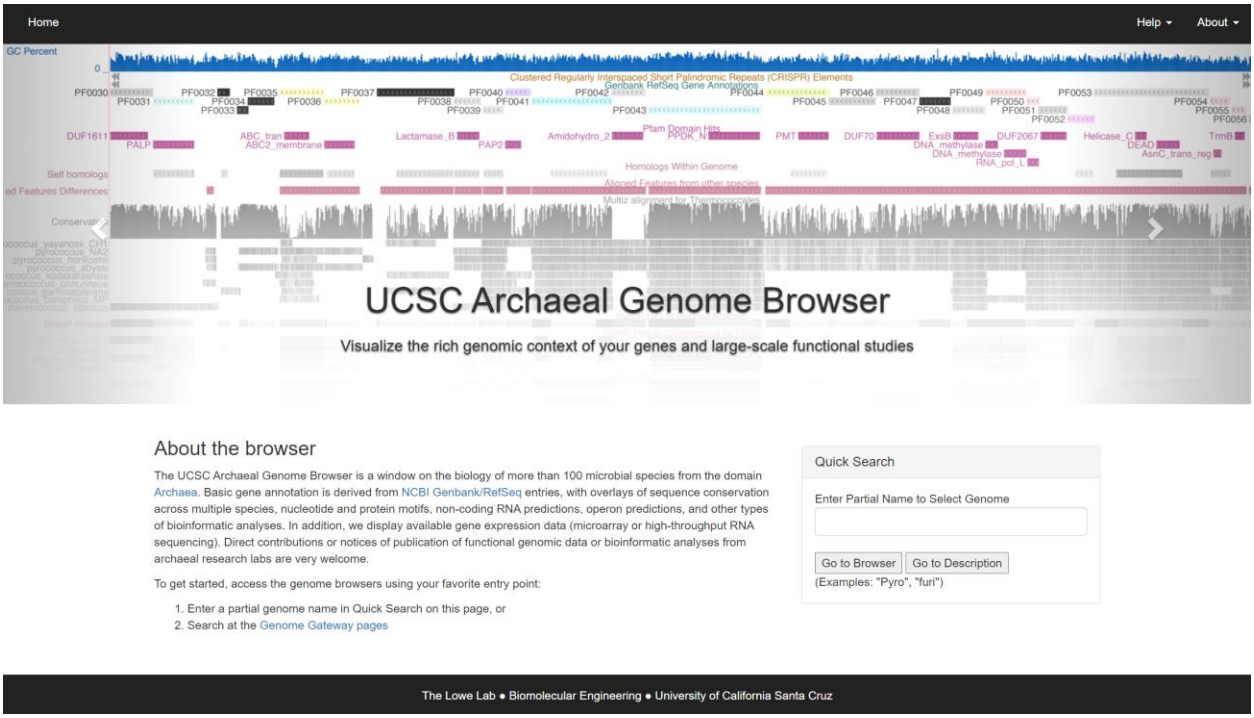
PORTEco – is a portal for *E. coli* research that enables viewing and downloading of datasets.



PATRIC (Pathosystems Resource Integration Center) – provides integrated data analysis tools that support biomedical research on infectious diseases caused by bacteria.

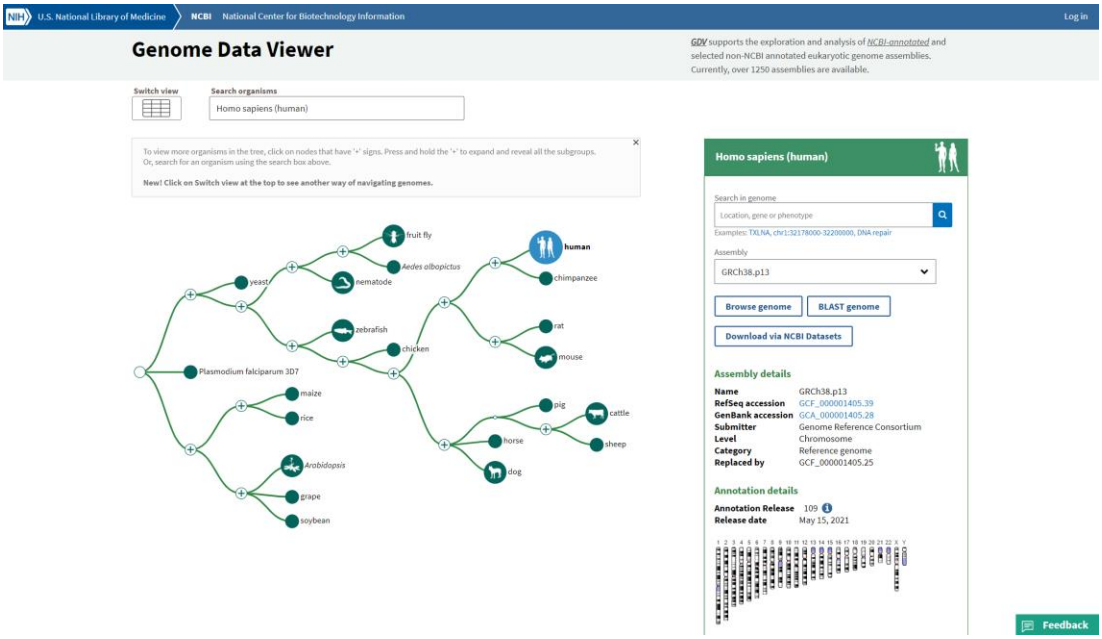


UCSC archaeal genome browser – a separate genome browser for archaea and certain bacterial genomes. It is linked to Galaxy that allows performing some genome analysis operations.

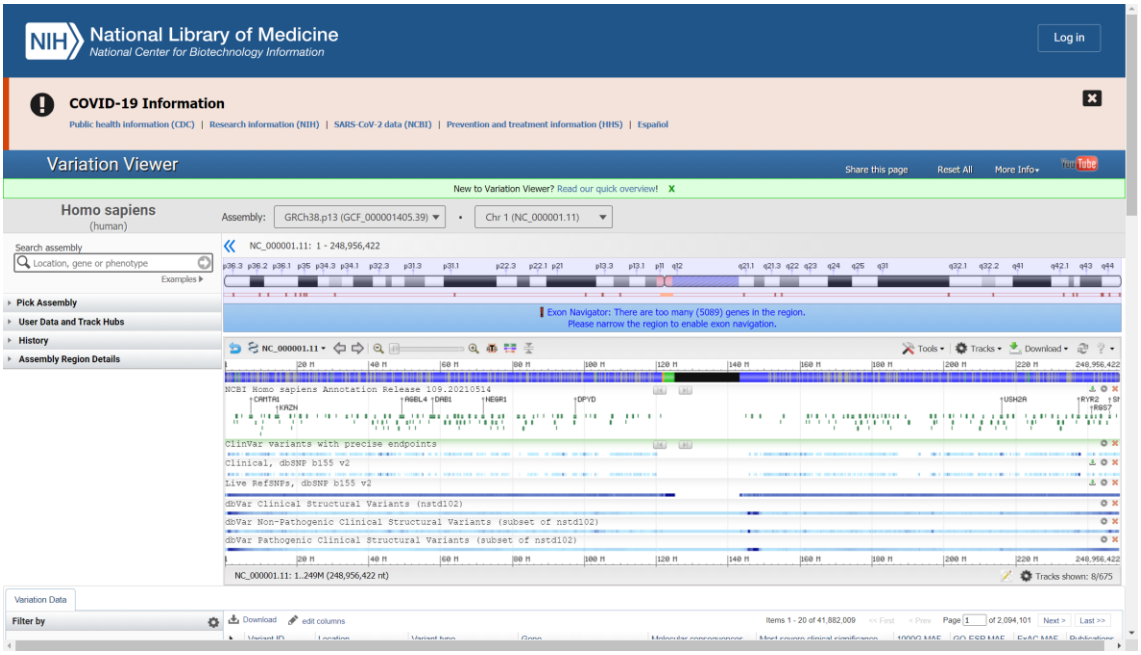


Greengenes – a repository for 16S rRNA sequences and tools. These sequences are used in phylogeny studies to compare organisms and for species identification.

NCBI Genome Data Viewer (GDV) – previously known as Map Viewer. GDV is a genome browser to view eukaryotic RefSeq genes.

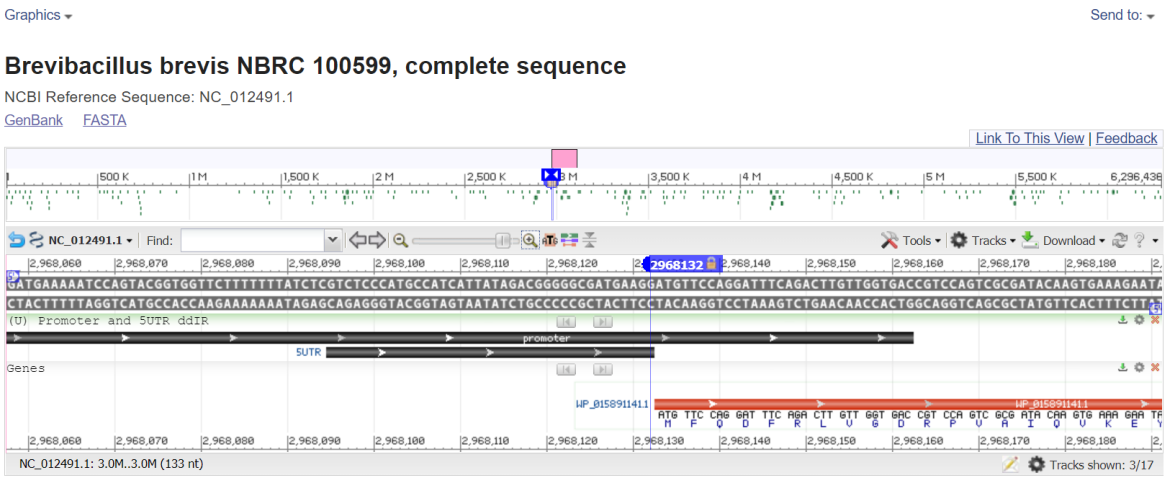


Variation Viewer: The NCBI variation viewer mostly focusses on human genome. Specific genes can be loaded on to the viewer. The many filters on the left panel helps the user choose the variation type to be displayed in the viewer. There is also an option to load user tracks with BED or WIG files. The NCBI genome browser can be used to view the contents of the BED files.



NCBI genome:

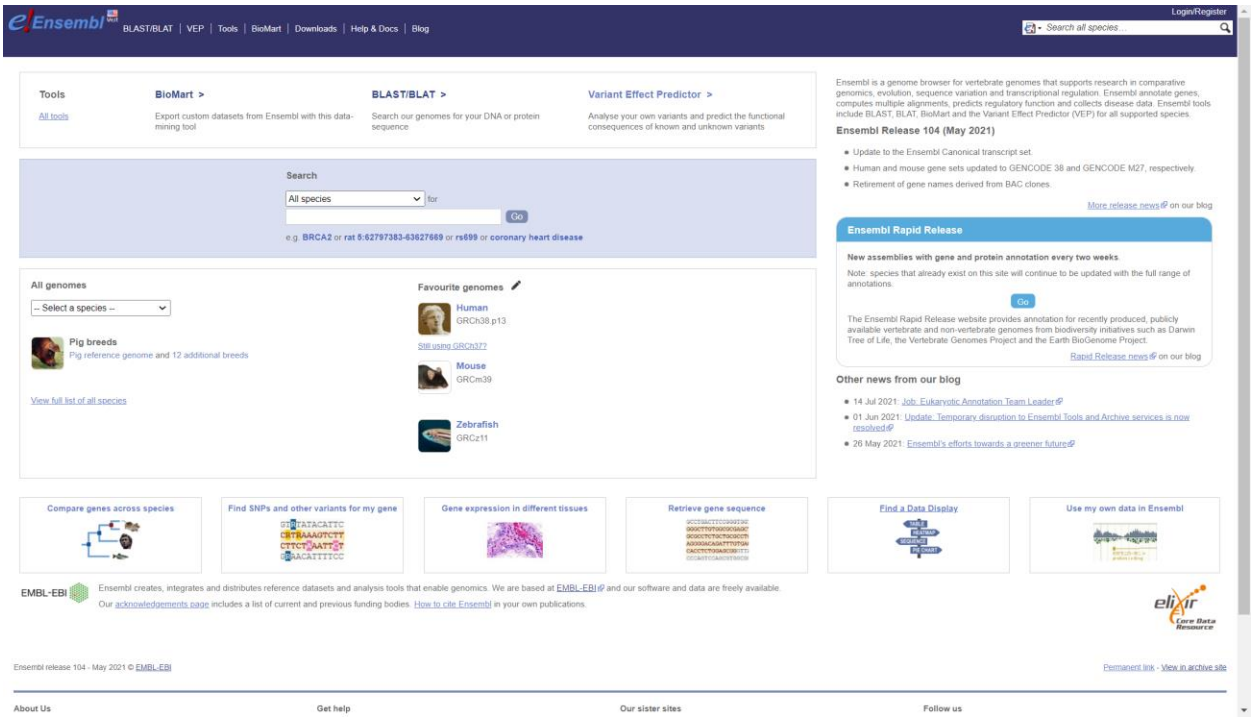
The **Genome** is a repository of information on genomes. It includes sequences, maps, annotations, assemblies, and chromosomes. We can browse the database by organisms and can even download the sequences of our interest. We can also upload custom data and view them as tracks.



Ensembl genome browser: It is a genome browser primarily for sequenced genomes of chordates. It includes few genomes of non-chordate model organisms, however, not prokaryotes. The database is updated about every four months. At the time of preparation of this document (August 2021), the current release is Ensembl release 104. The database is free to use for all, and we can create login accounts. This would allow us to share details of our research/analysis with other members of the team. The homepage allows us to search species-specific information.

There are many videos on the usage of various Ensembl tool. The link can be found **here**.

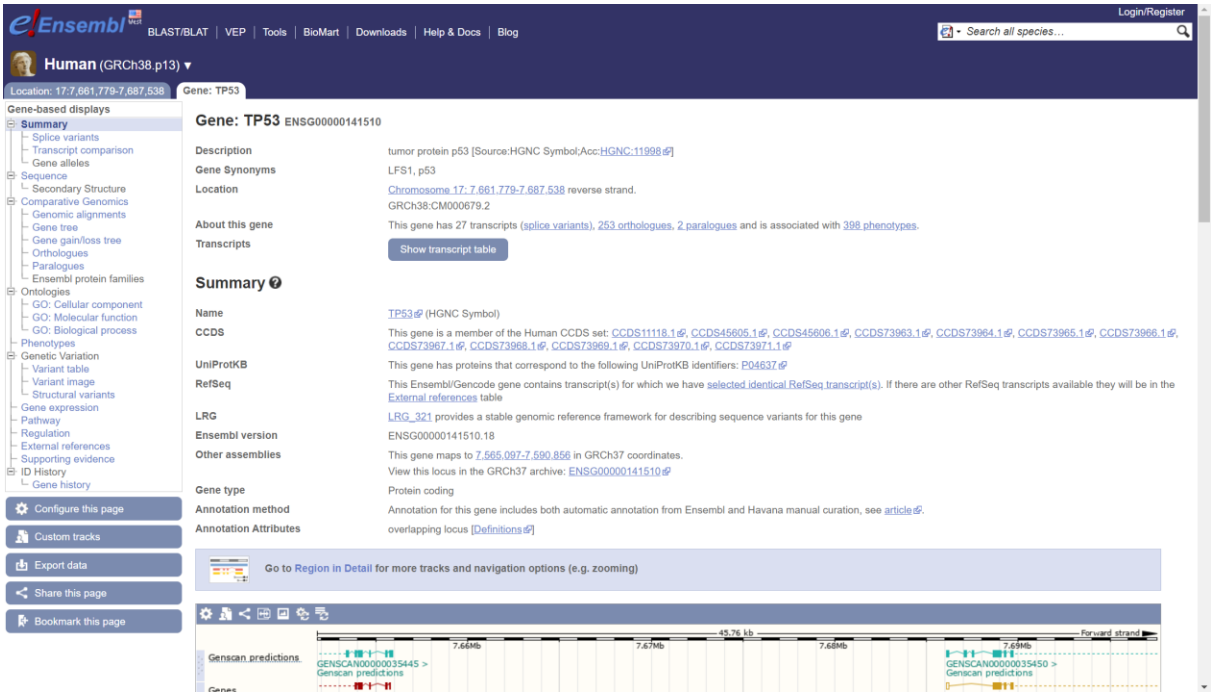
Ensembl Genomes is the database for non-vertebrates. It provides information on genomes of bacteria, protists, fungi, plants, and invertebrate metazoan. Using the bacterial portal, archaeal genomes can be accessed.



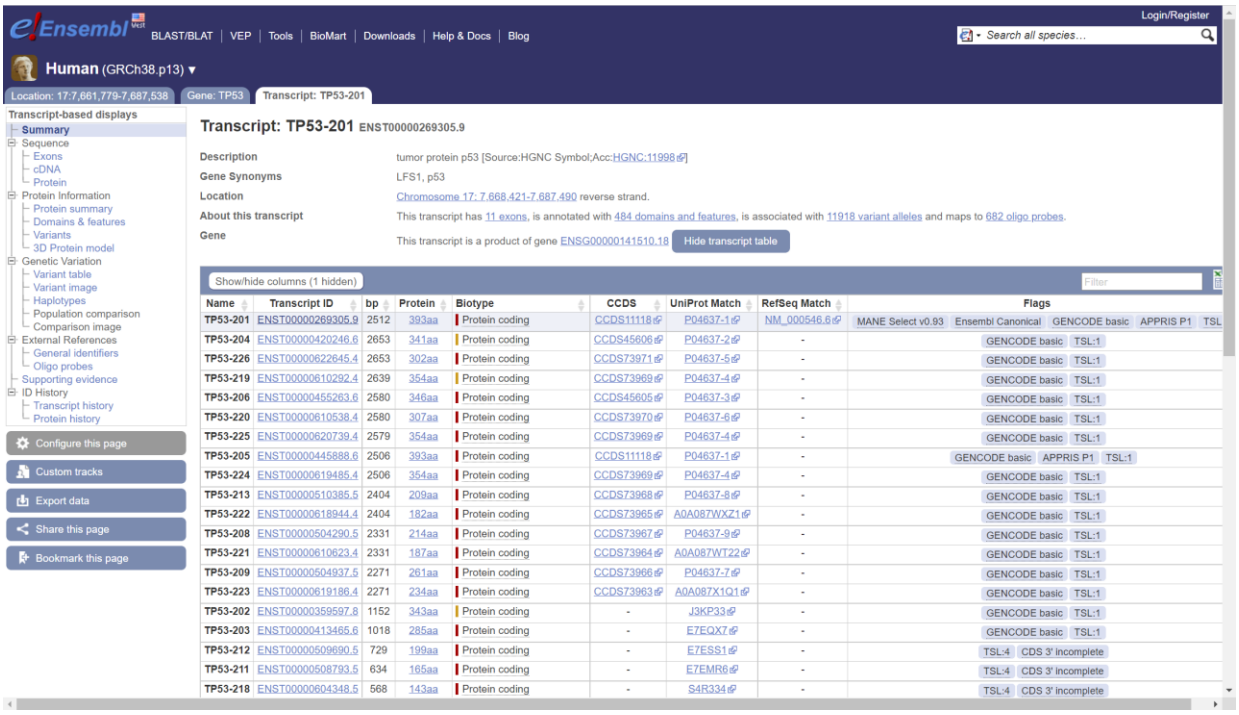
Searching for the Human TP53 in the homepage, leads us to the gene-specific page which provides information of the gene location on the chromosome, transcripts, orthologues, paralogues etc. On the left navigation panel, there are tools specific for gene data analysis. The region in detail view will allow us to view a region on the chromosome. We can see the coding and the non-coding transcripts, exons etc.

Clicking on a specific gene transcript, opens a new transcript tab that provides detailed information pertaining to the selected transcript. The exon data, variant, and protein information can be viewed by selecting the appropriate links from the left navigation panel. Below are the screenshots of the gene tab and the transcript tab from the Ensembl genome browser.

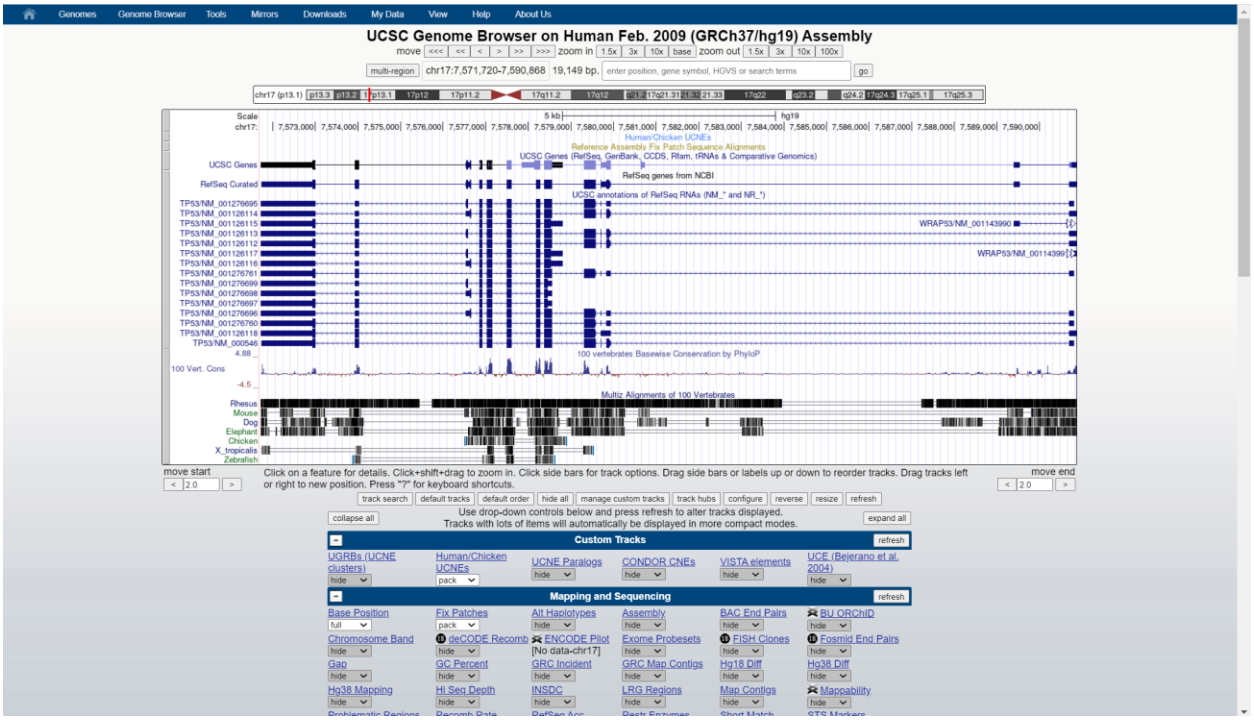
Gene tab:



Transcript tab:



UCSC Genome Browser: It is a genome browser developed by the University of Santa Cruz and provides options to view genomic data by configuring specific tracks. It is also possible to upload custom data and create custom tracks to view alongside the existing data for comparison and analysis. It is primarily a eukaryotic genome browser, with 97 eukaryotic genomes, including 49 mammals. The database holds assembly data, mRNA, EST, RefSeq alignments. It provides links to other applications like Ensembl, NCBI Map viewer, Galaxy etc. Data from ENCODE project is also available to be viewed.

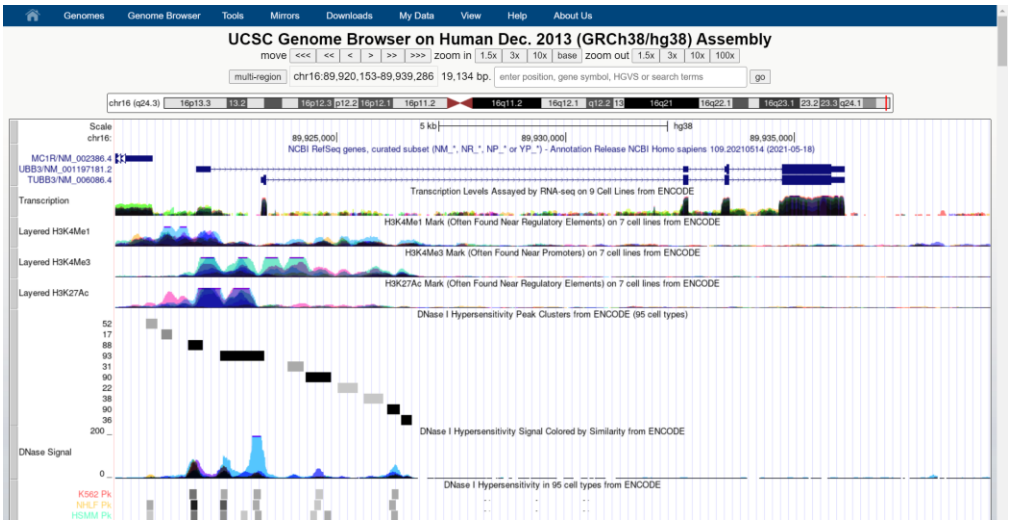


Example: Find the human TUBB3 gene using the UCSC Genome Browser (hg38). Turn on the Encode Regulation track (HINT: set display mode to “full” for these tracks) and NCBI RefSeq genes. In a few sentences, describe what you see at the TUBB3 locus in terms of the Encode Regulation tracks. Include in your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

At the TUBB3 locus in the Encode Regulation track, we can see the following tracks- Transcription Levels Assayed by RNA-seq on 9 Cell Lines from ENCODE, Layered H3K4me1 track, Layered H3K27Ac track, DNase 1 hypersensitivity peak clusters track, DNase 1 hypersensitivity signal track, track with DNase 1 hypersensitivity in 95 cell types from ENCODE, TF clusters track and Transcription Factor ChIP-seq Peaks track. H3K4me1, H3k4me3 and H3k27Ac modifications appear near the TSS of TUBB3 gene. These are represented by peaks in the track that are observed close to the TSS or in the region of the

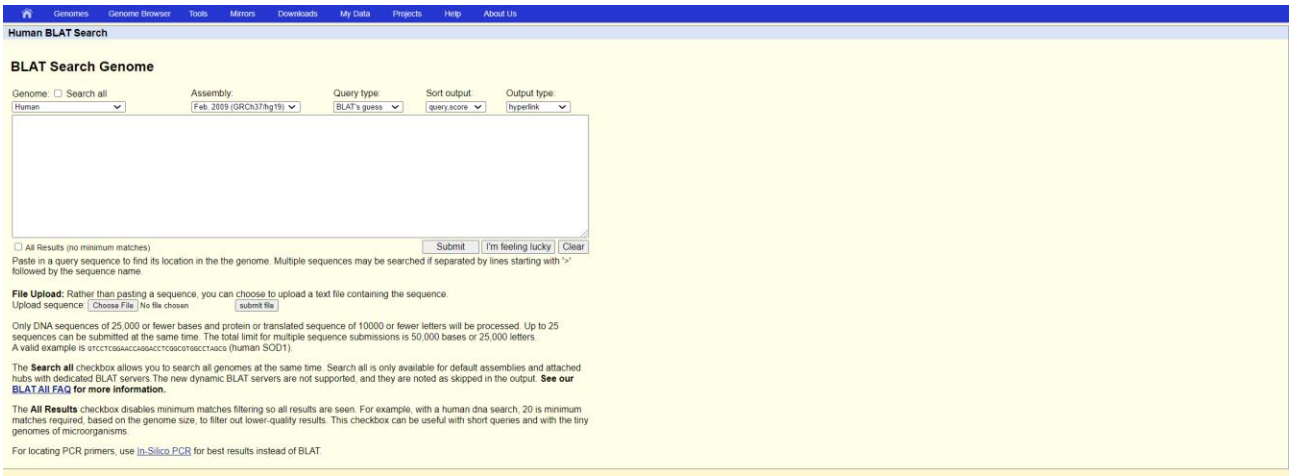
promoter of TUBB3 gene. The Layered H3K27Ac track show the modification of the histone proteins and is suggestive of enhancers. The peaks represent the histone modifications in HUVEC, NHEK, NHLF, K562 etc. cells from ENCODE.

Screen shot of the TUBB3 locus from UCSC genome browser:



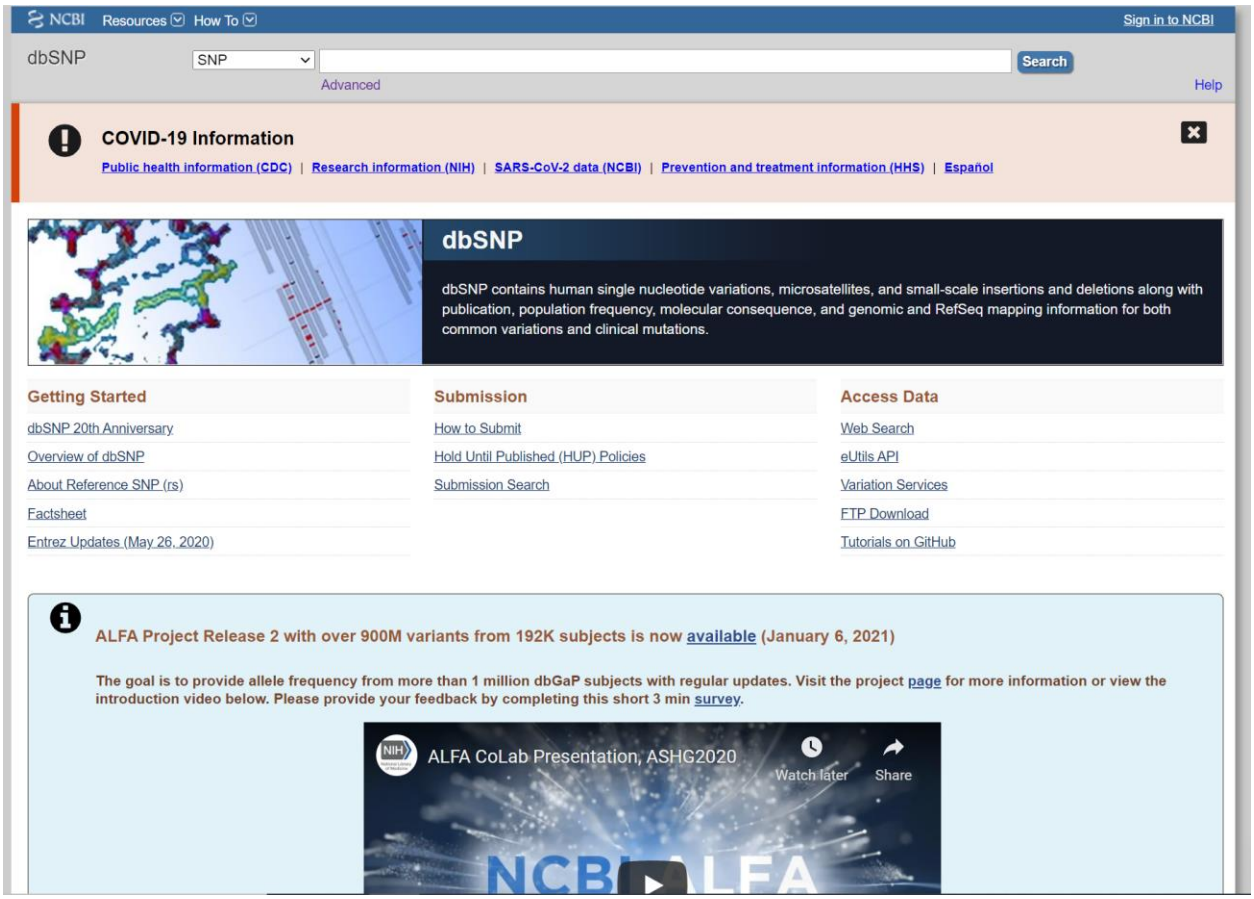
Using the Table Browser, we can download data from UCSC. There is option to download data in many formats like BED or send the results directly to Galaxy. The user guide for Table Browser is available **here.**

Another feature available at the UCSC Genome Browser is BLAT. This topic was covered in the previous section. The interface is shown below and can accessed by clicking on "Tools" from the top menu -> Blat

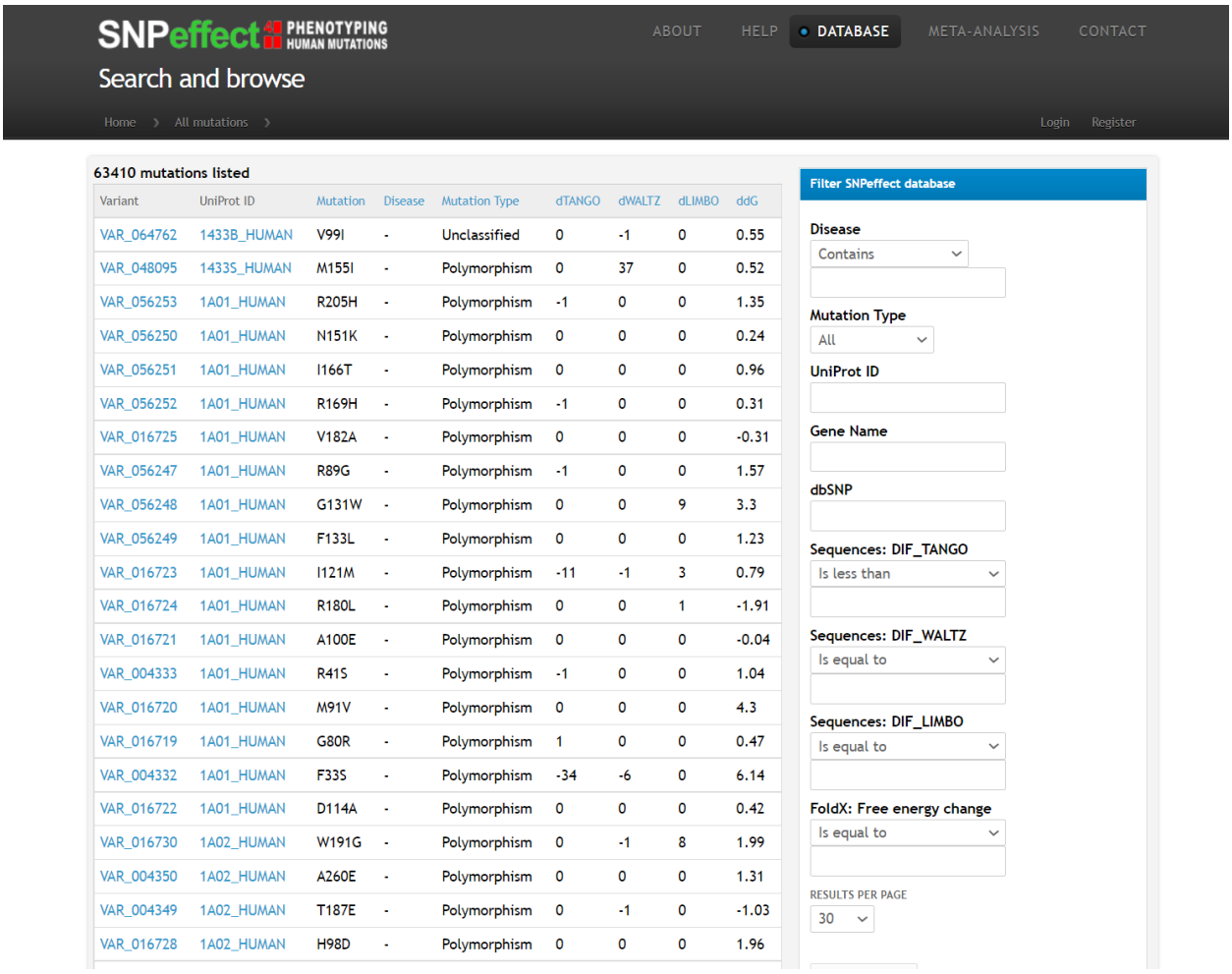


dbSNP database: The largest database of SNPs is hosted by NCBI. It holds submitted SNPs and curated reference SNPs. It also includes MNPs (multiple nucleotide polymorphisms) and short indels (insertions and deletions).

Web interface of dbSNP database:

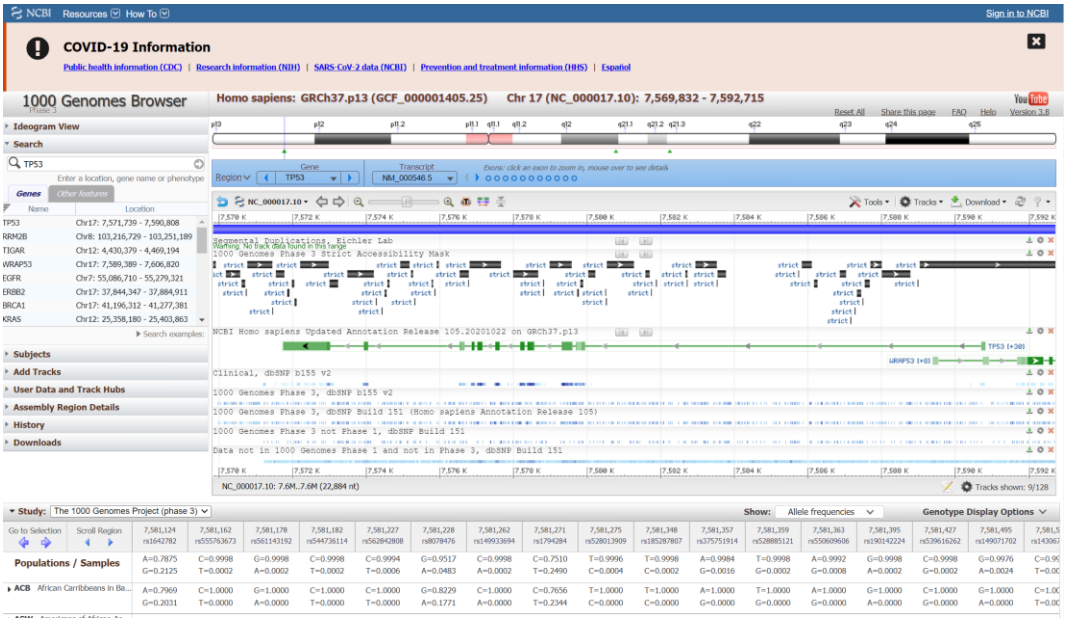


SNPeffect database: came into being in 2012, stores information pertaining to the effect of SNPs on proteins and their function. There are tools to predict protein stability and folding.

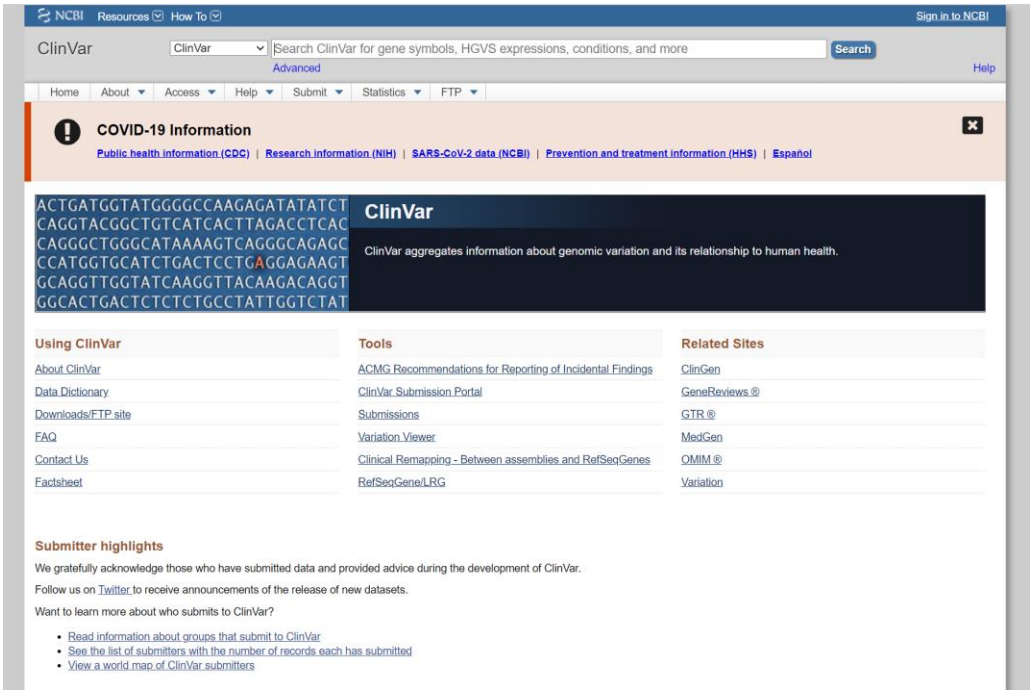


NCBI 1000 Genomes Browser: This allows access to the data from the 1000 Genomes Project. We can analyze specific genes /regions/ chromosome positions, add ClinVar and dbSNP tracks by clicking on the 'Tracks' button. For each variation in the variation table, genotype information can be accessed pertaining to specific ethnicities. Custom data can also be uploaded. The help page for the browser can be found **here**.

Below is a screen shot of the interface for the TP53 gene:

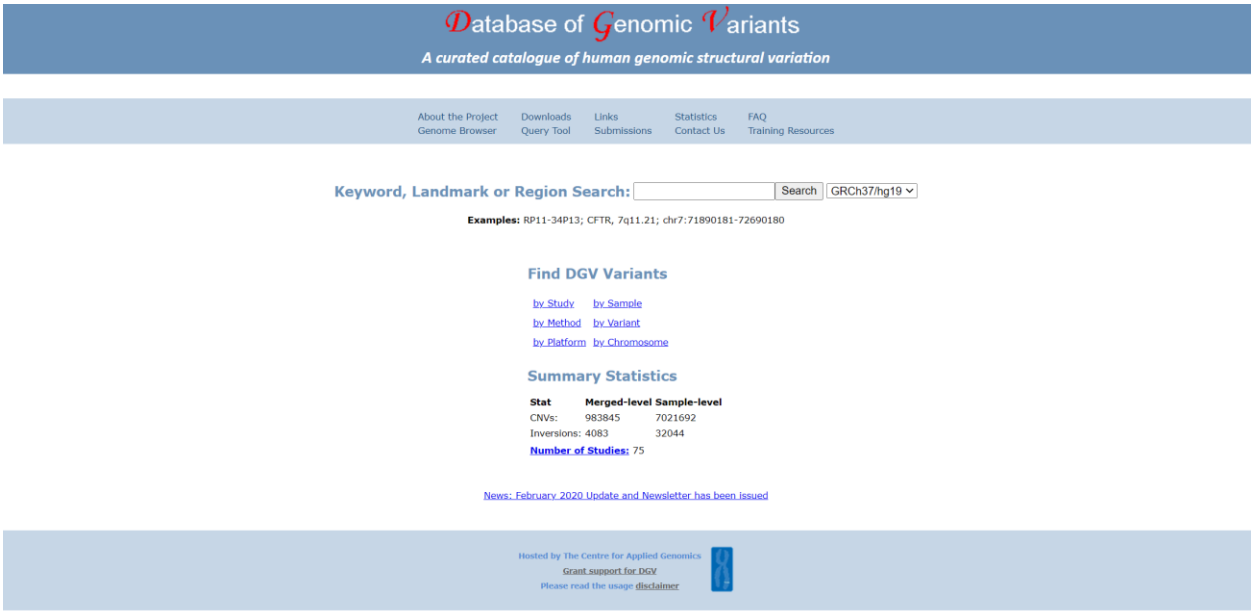


ClinVar: is a specialized database to hold a curated set of variations that have possible medical and clinical relevance. It focusses on human variations. It does not replace dbSNP. We can search the database using the gene symbol or disease. It is also available through UCSC Genome Browser under the group ‘Phenotype and Literature’ -> ‘ClinVar Variants’ and ‘ClinGen CNVs’ tracks.

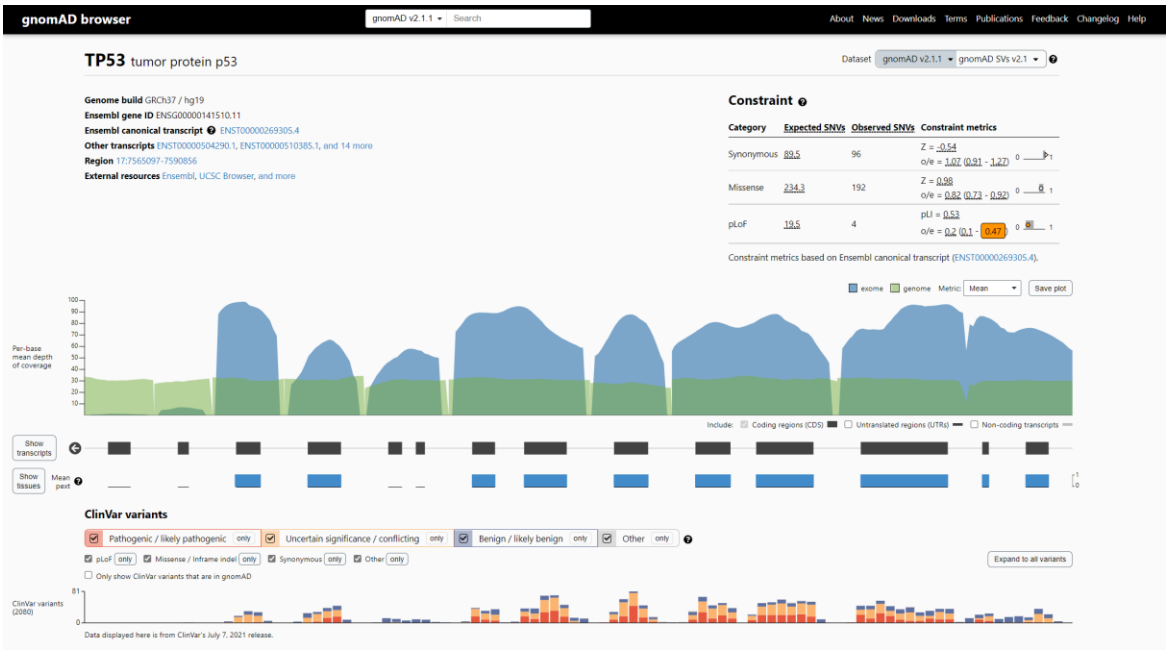


DGV (Database of Genomic Variants): is hosted by the Center for Applied Genomics, Toronto. By default, the hg19 assembly is mapped. In the UCSC Genome Browser, the same information can be accessed using the DGV Struct Var track, which may also be more user-friendly with gain in copy number indicated in blue and losses indicated in red.

DGV interface:

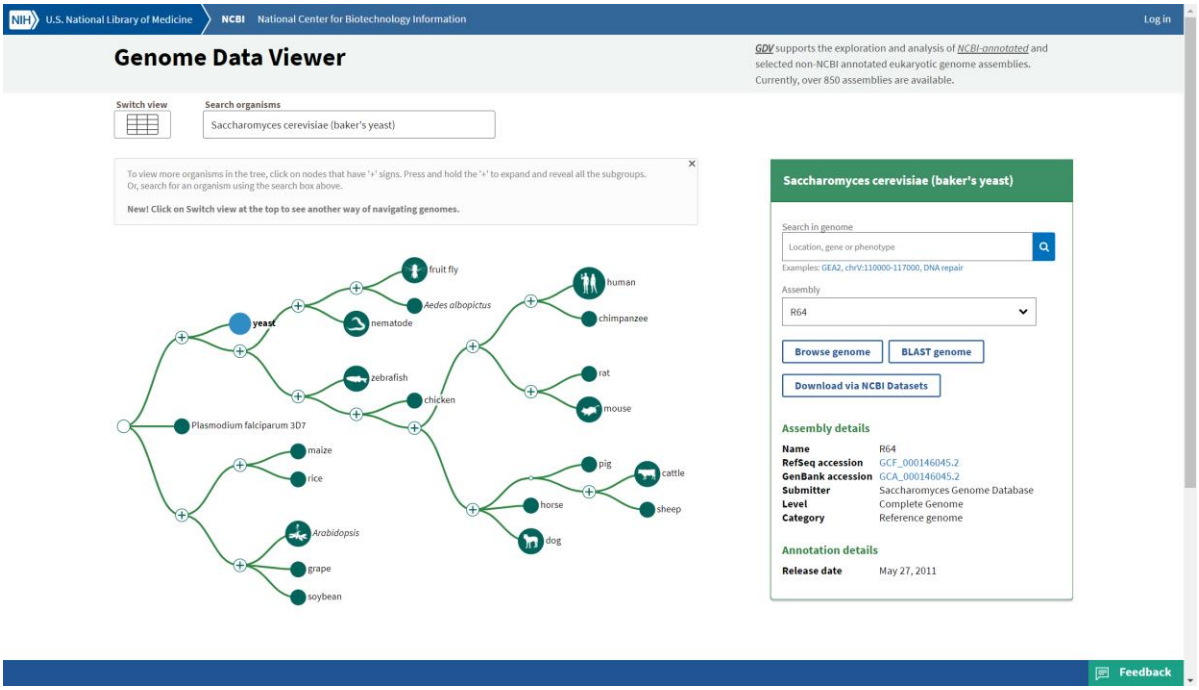


GnomAD (Genome Aggregation Database): It is a repository of exome and genome sequencing data from large-scale sequencing projects. It allows searches to performed based on gene names, ensemble transcript ID, dbSNP ID, or chromosomal region. Below is the search result for TP53 gene. The peaks indicate protein-coding SNPs in the region. The CNV data below is represented by brown (gains) and red (losses). Below it is the table of SNPs (not shown in the screen shot).

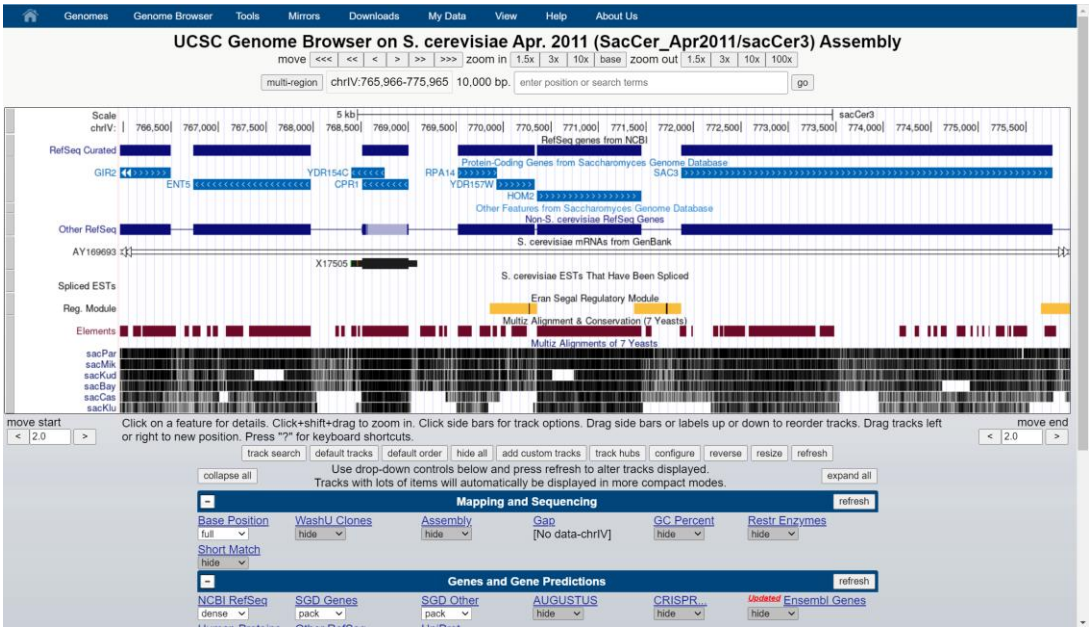


Yeast Genome Browsers: There are several browsers available for *Saccharomyces cerevisiae*. They are:

[NCBI Genome Data Browser](#) (NCBI Genomes) – Allows users to search by gene, location, or phenotype. It provides options to browse the genome, perform BLAST, and also download datasets.



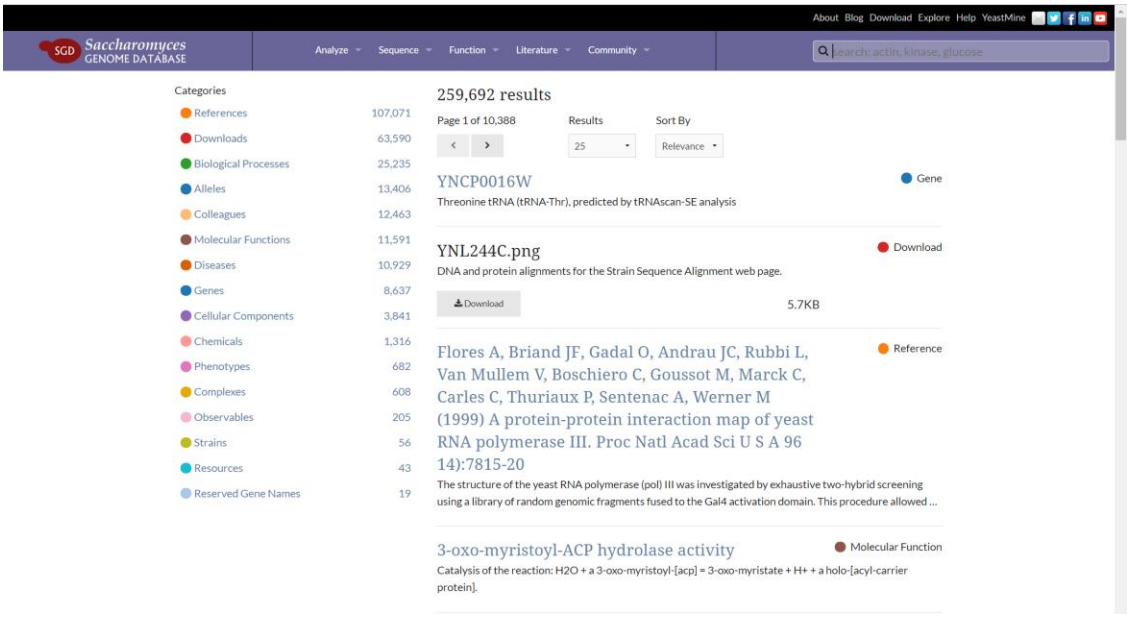
[S. cerevisiae Genome Browser] - sacCer3 assembly at UCSC



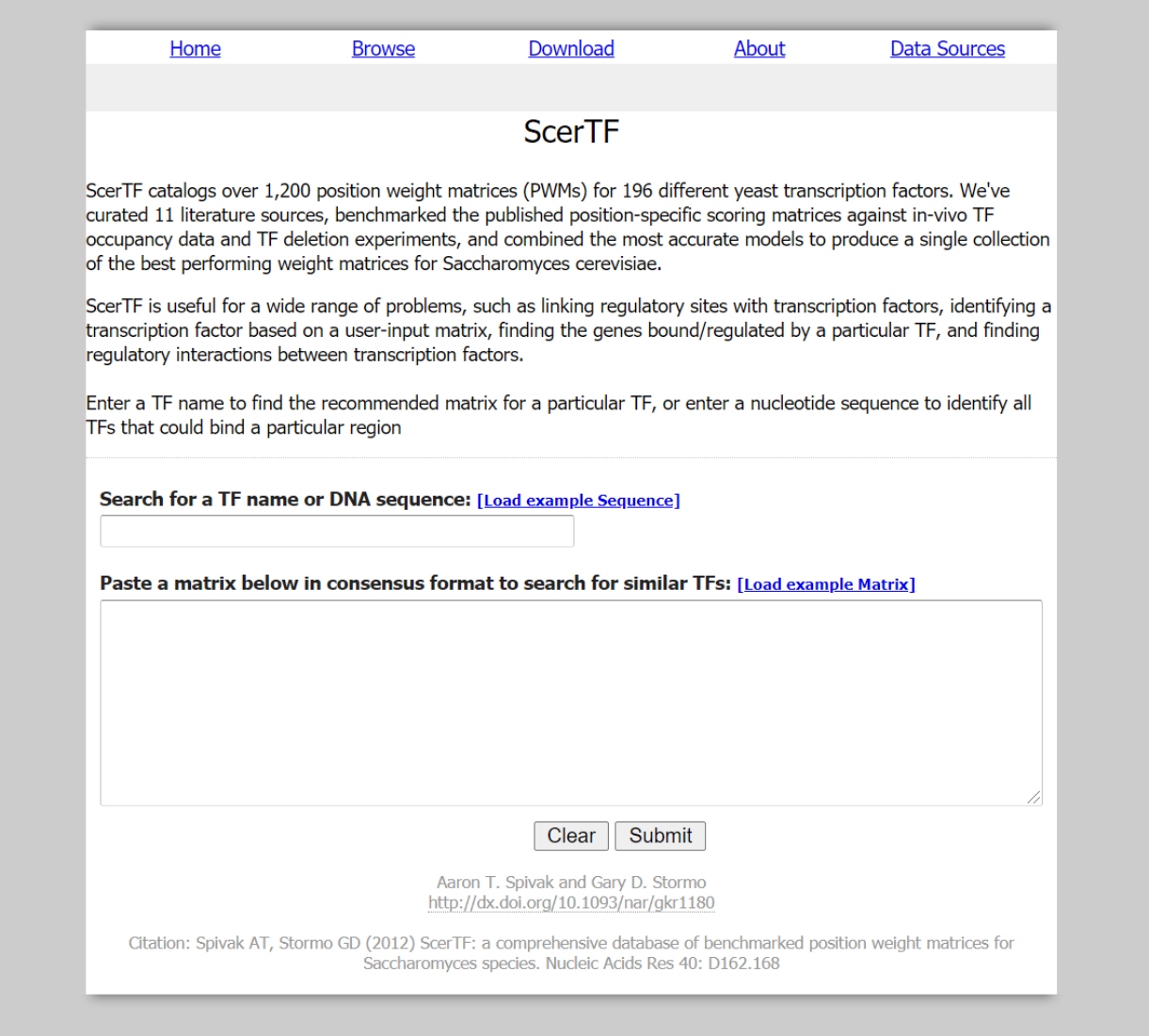
[SGD (Saccharomyces Genome Database)]: provides integrated biological information for budding yeast along with tools for data exploration to study the relationship between sequence and gene products in fungi and higher organisms.

Tutorials to help with the navigation of SGD is available [here].

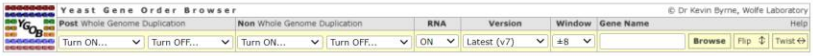
Homepage of SGD:



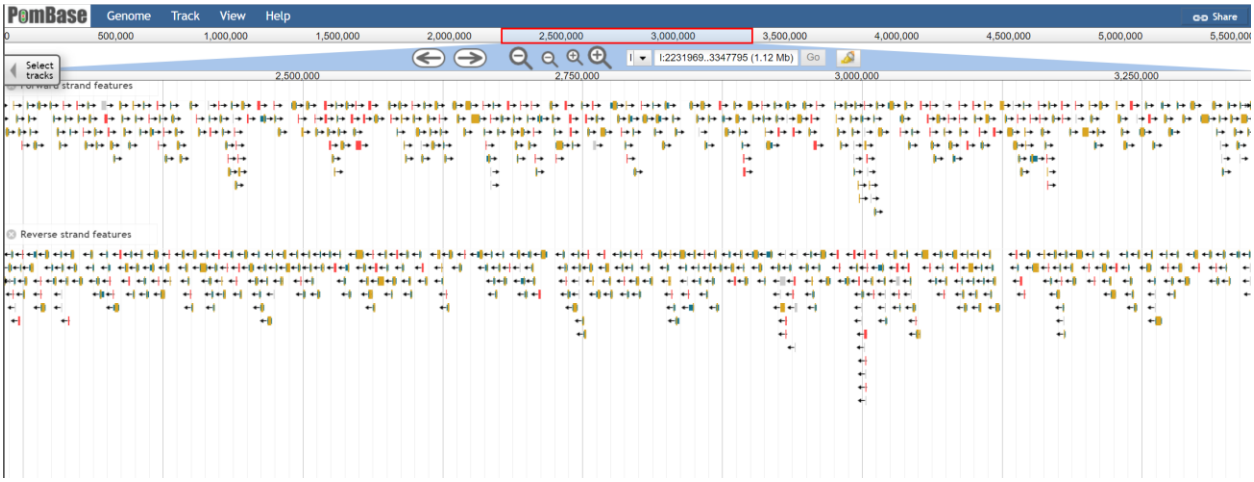
ScerTF : is a database for yeast transcription factors (TFs). This website is useful to find genes regulated by a particular TF, analyzing the regulatory interactions between different TFs.



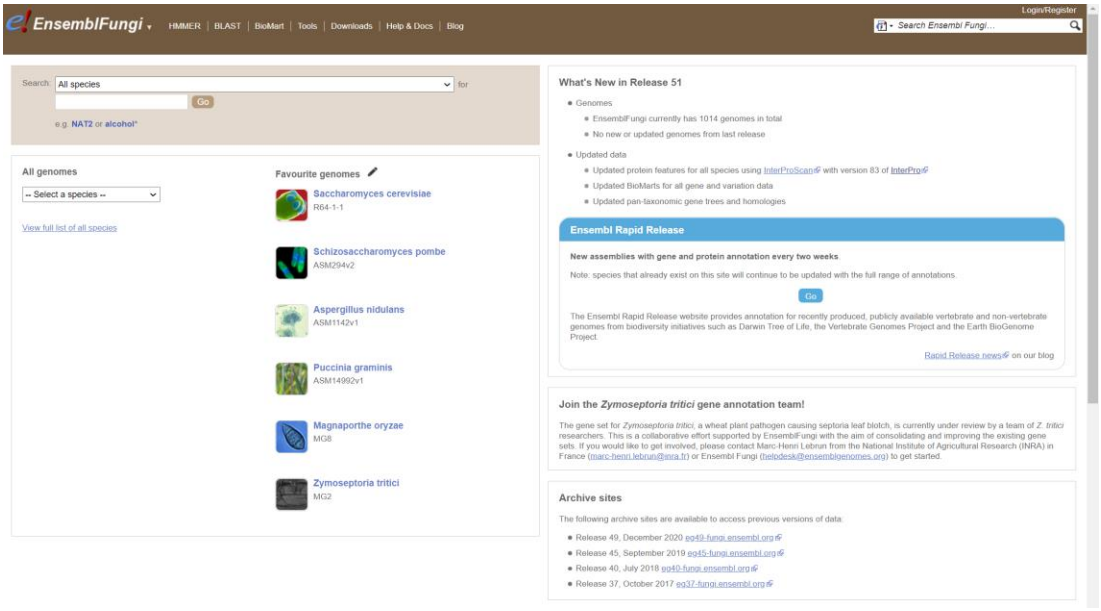
Yeast Gene Order Browser (YGOB): helps with visualizing the syntenic context of a gene from several yeast genomes.



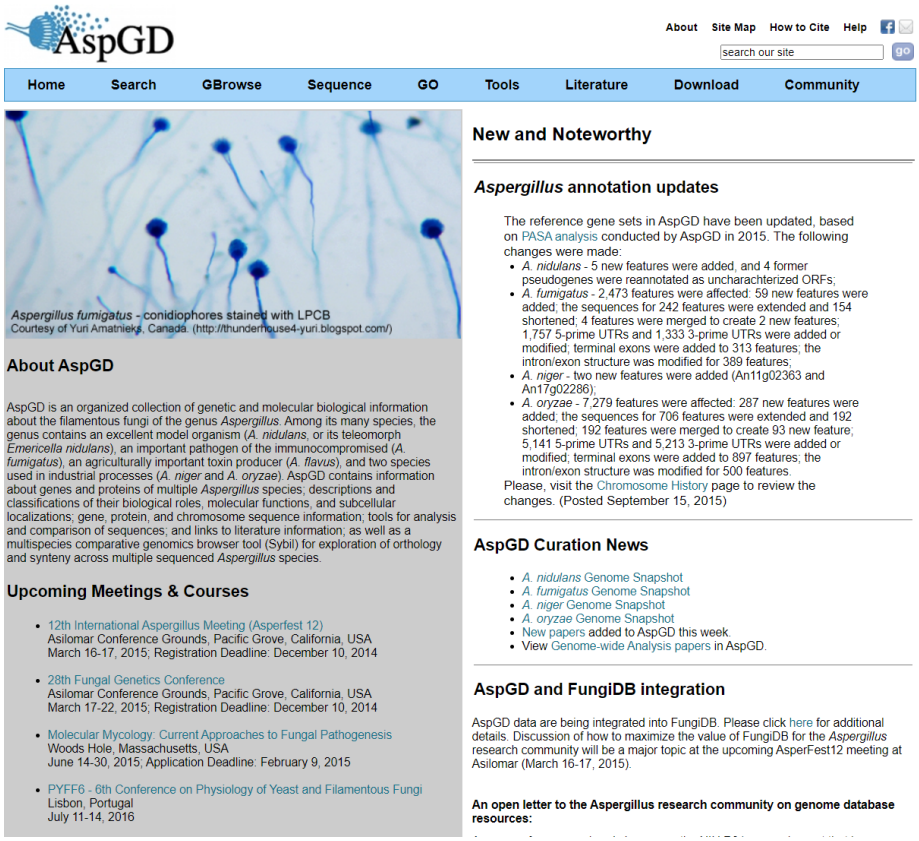
PomBase: It is a database for the fission yeast *Schizosaccharomyces pombe*, providing access to large data sets, structural and functional annotation information.



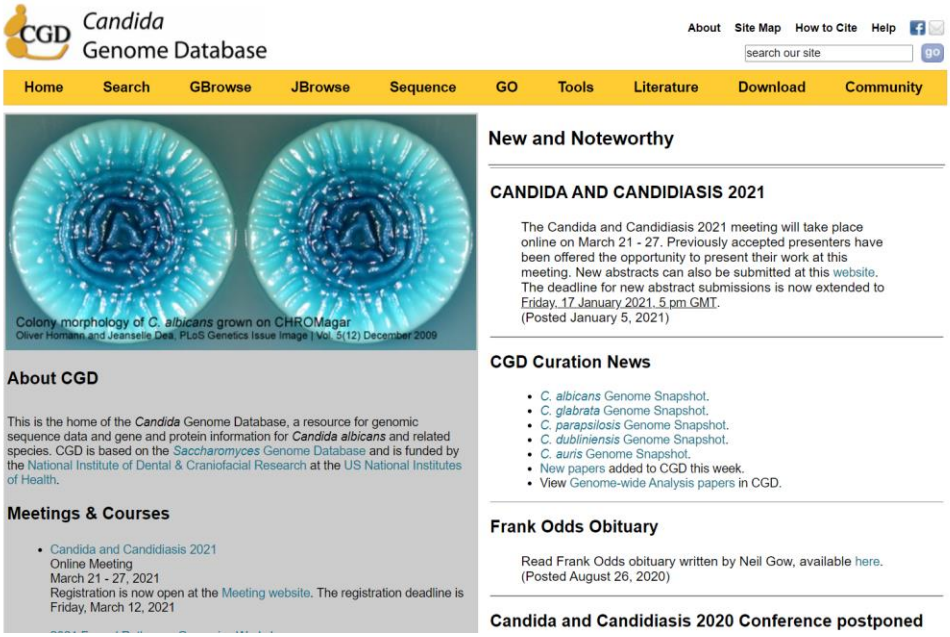
Ensembl Fungi: is a browser for fungal genomes.



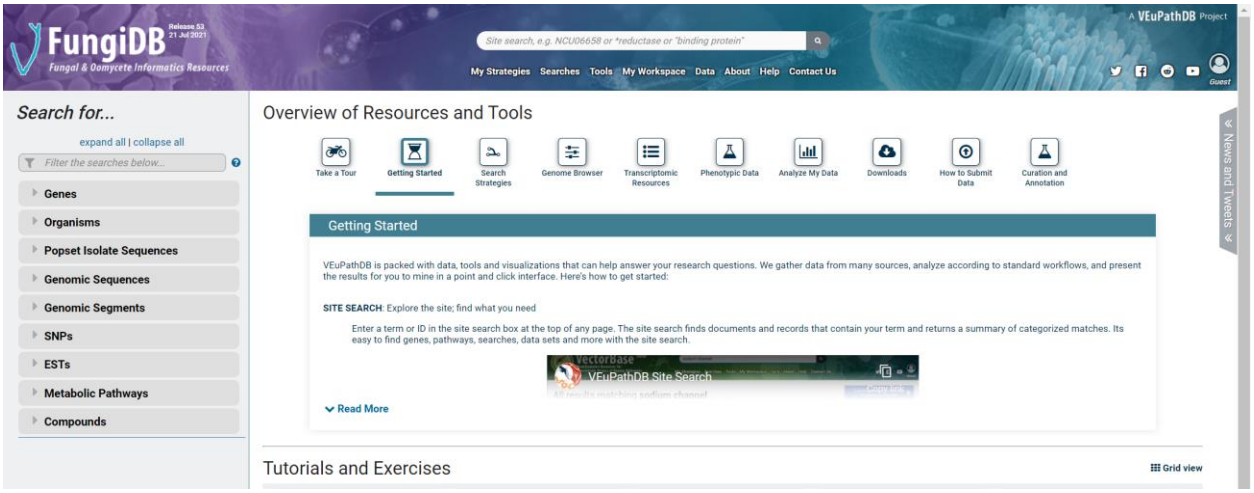
Aspergillus Genome Database (AspGD): It has complete genome information of two species - *Aspergillus nidulans and Aspergillus fumigatus*. It is maintained by the Broad Institute.



[Candida Genome Database (CGD)]: It primarily contains manually curated sequence information for Candida albicans. It also contains data pertaining to *Candida glabrata* and other species of *Candida*.



[FungiDB] : It is a data-mining interface focusing on comparative and functional genomics. It also allows for genomics comparison across species. The homepage provides many helpful tutorials.



Lower eukaryotic model systems: Giardia DB, Trypanocyc, Sanger Protozoan Genome Projects.

Other single-celled organisms : PlasmoDB, T. gondii database, Tetrahymena DB.

Plants:

TAIR – Arabidopisis database.

PGSB Plant Genome and Systems Biology - *Arabidopsis thaliana* project.

IRGSP – International Rice Genome Sequencing Project.

MaizeGDB – Maize genome database

Dictyostelium – Slime mold database.

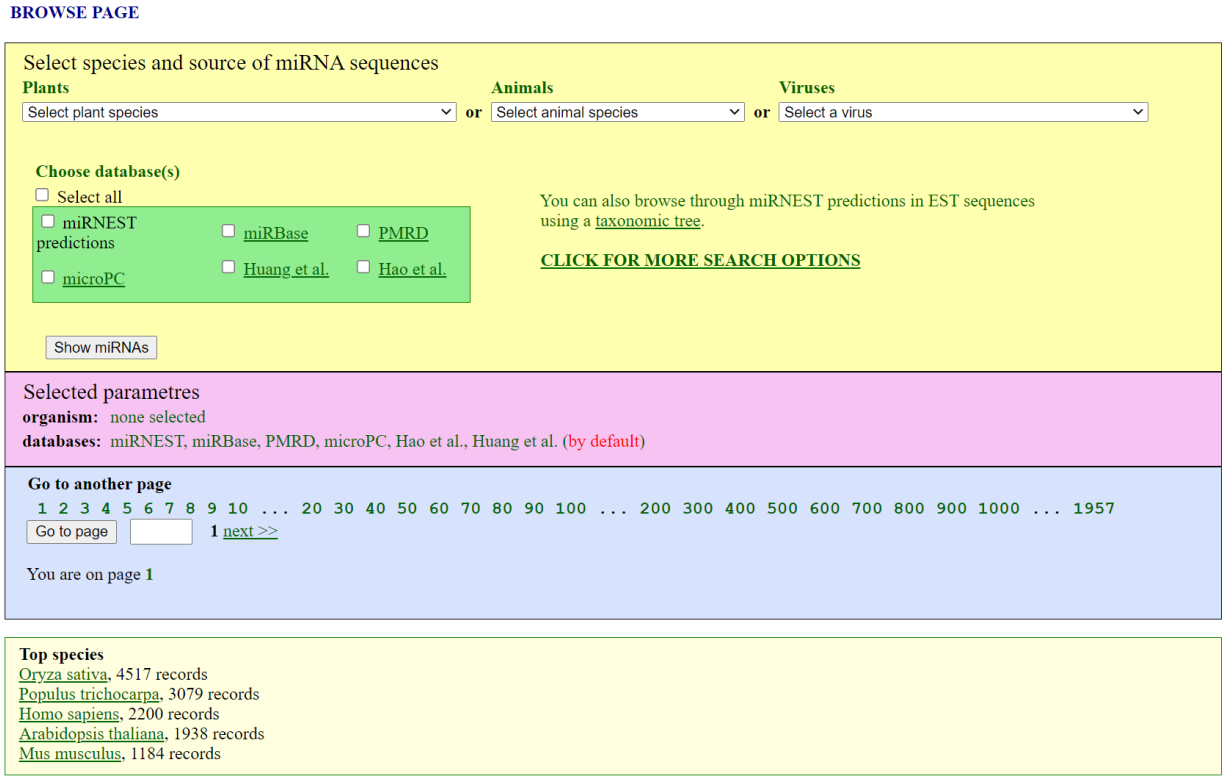
WormBase – Database for *Caenorhabditis elegans*

FlyBase - Database of Drosophila Genes & Genomes

MGI **(Mouse Genome Informatics)** – is a database that stores information of laboratory mouse, and provides genetic, genomic, and biological data to study human diseases.

TargetScan: is a database for predicted miRNA targets in mammals. We can search the database by species and gene symbol.

miRNEST 2.0 : is a collection of plant, animal, and virus miRNA information.



PolymiRTS Database 3.0 : has information of SNPs in miRNA target sites. It allows us to study how SNPs interfere with miRNA-based gene silencing.

miRDB : database for miRNA target predictions and functional annotations.

miRbase : repository of known miRNA sequences and annotations.

Rfam : resembles the Pfam database and is hosted by EBI. It provides information of ncRNA families.

NONCODE : database for ncRNAs (excluding tRNAs and rRNAs) especially lncRNAs.

ANCORA : database for highly conserved noncoding elements (HCNEs) in metazoan genomes.

cneViewer : displays conserved regions between human and zebra fish genomes. Users can search data based on gene name of CNE ID.

UCNEBase : repository of information pertaining to Ultra conserved non-coding elements (UCNEs) in 18 different vertebrate species.

Tools for analysis of metagenomes: **Real-Time metagenomics**, Rapid Annotation using Subsytem Technology (**RAST**), metagenomics analysis tool in Galaxy.

References

Galaxy. (n.d.). De novo transcriptome reconstruction with RNA-Seq. Retrieved from https://training.galaxyproject.org/training-material/topics/transcriptomics/tutorials/de-novo/tutorial.html

Glimmer. (n.d.). ABOUT GLIMMER. Retrieved from http://ccb.jhu.edu/software/glimmer/index.shtml

Mistry, M., & Khetani, R. (n.d.). Introduction to ChIP-Seq using high-performance computing. Retrieved from https://hbctraining.github.io/Intro-to-ChIPseq/lessons/05_peak_calling_macs.html

Yun, S., & Yun, S. (2021). AS.410.635.81.SU21 Bioinformatics: Tools for Genome Analysis-Course Materials. Retrieved from https://blackboard.jhu.edu/webapps/blackboard/content/listContent.jsp?course_id=_232465_1&content_id=_9712340_1&mode=reset