EXPERIMENT: 1

WORKING WITH NGS DATABASES AND NGS FILE FORMATS

AIM: To retrieve files of the from SRA database, obtain .fastq files using SRA toolkit and interpret results

PROCEDURE: Downloading data from SRA database: [RNA]

- Go to the website https://www.ncbi.nlm.nih.gov/sra
- Type any organism name or known accession number or (SRR)
- Collect library information

Name	b8h
Instrument	Illumina
Strategy	RNA-Seq
Source	TRANSCRIPTOMIC
Selection	RANDOM
Layout	PAIRED

- Click on the SRR number under Run. It takes to next page

Run	Spots	Bases	Size	GC content	Published	Access Type
SRR11797448	1.4M	210.4M	112.9MB	56.2%	2020-09-01	public

- Click on reads tab take print screen
- Download the data.
- fastq-dump: Convert SRA data into fastq format
- Below command should be used in the SRA toolkit folder

```
fastq-dump --split-files SRR11797448.sra
```

Quality Control analysis for raw data

Quality control or QC of short refers to the quality of the data before starting the experiment. The objective of the quality control is to find the Basic statistics, per base sequence quality, per sequence quality score, per base sequence content, per sequence GC content, per sequence GC content, sequence length distribution, sequence duplication level, overrepresented sequence.

FastQC

Modern high throughput sequencers can generate tens of millions of sequences in a single run. Before analysing this sequence to draw biological conclusions you should always perform some simple quality control checks to ensure that the raw data looks good and there are no problems or biases in your data which may affect how you can usefully use it. FastQC can be run in one of two modes. It can either run as a standalone interactive application for the immediate analysis of small numbers of FastQ files, or it can be run in a non-interactive mode where it would be suitable for integrating into a larger analysis pipeline for the systematic processing of large numbers of files.

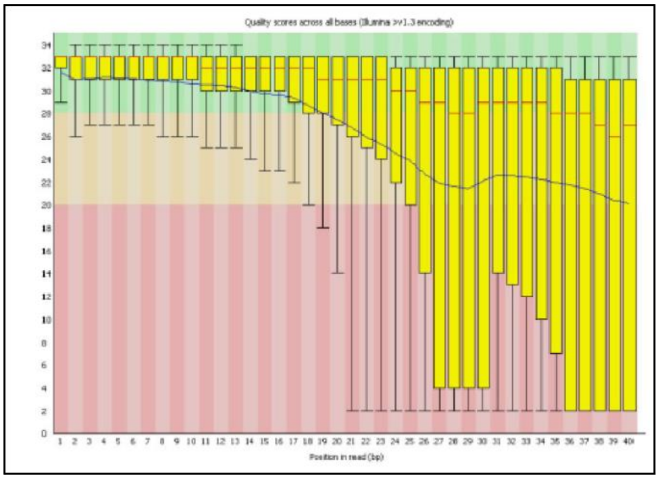
Basic Statistics

The Basic Statistics module generates some simple composition statistics for the file analyzed.

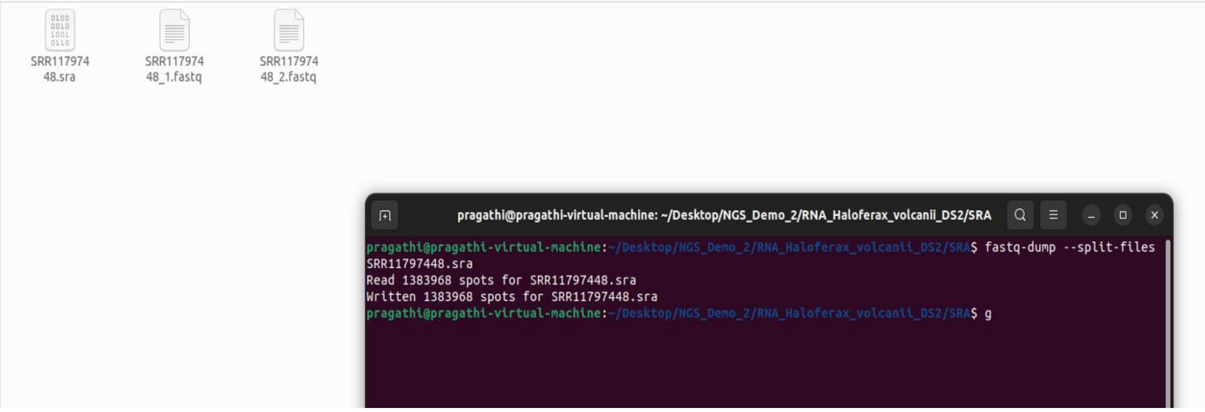
- **Filename:** The original filename of the file which was analyzed
- **File type:** Says whether the file appeared to contain actual base calls or colorspace data which had to be converted to base calls
- **Encoding:** Says which ASCII encoding of quality values was found in this file.
- **Total Sequences:** A count of the total number of sequences processed.
- **Filtered Sequences:** If running in Casava mode sequences flagged to be filtered will be removed from all analyses.
- **Sequence Length:** Provides the length of the shortest and longest sequence in the set. If all sequences are the same length only one value is reported.
- **%GC:** The overall %GC of all bases in all sequences.

Fig: SRA to fastq format conversion

Per Base Sequence Quality



- This view shows an overview of the range of quality values across all bases at each position in the FastQ file for each position a BoxWhisker type plot is drawn.
- The elements of the plot are as follows: The central red line is the median value
- The yellow box represents the inter-quartile range (25-75%)
- The upper and lower whiskers represent the 10% and 90% points
- The blue line represents the mean quality.
- The y-axis on the graph shows the quality scores. The higher the score the better the base call.
- The background of the graph divides the y axis into： Very good quality (green), Reasonable quality (orange), Poor quality (red).


SRR117974
48.sra
SRR117974
48_1.fastq
SRR117974
48_2.fastq
pragathi@pragathi-virtual-machine: ~/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/SRA

```
pragathi@pragathi-virtual-machine:~/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/SRA$ fastq-dump --split-files
SRR11797448.sra
Read 1383968 spots for SRR11797448.sra
Written 1383968 spots for SRR11797448.sra
pragathi@pragathi-virtual-machine:~/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/SRA$ g
```

Fig: Conversion from. sra to. fastq format

Per Sequence Quality Scores

The per sequence quality score report allows you to see if a subset of your sequences have universally low quality values. It is often the case that a subset of sequences will have universally poor quality, often because they are poorly images (on the edge of the field of view etc),

Per Base Sequence Content

Per Base Sequence Content plots out the proportion of each base position in a file for which each of the four normal DNA bases has been called. In a random library you would expect that there would be little to no difference between the different bases of a sequence run, so the lines in this plot should run parallel with each other.

Per Base N Content

If a sequencer is unable to make a base call with sufficient confidence then it will normally substitute an N rather than a conventional base call

THE ONLINE SERVERS AND TOOLS USED IN THIS EXERCISE ARE:

FASTQC	https://www.bioinformatics.babraham.ac.uk/projects/fastqc/

PROCEDURE

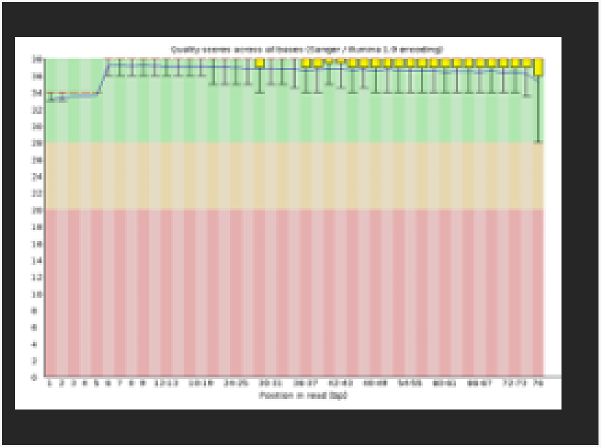
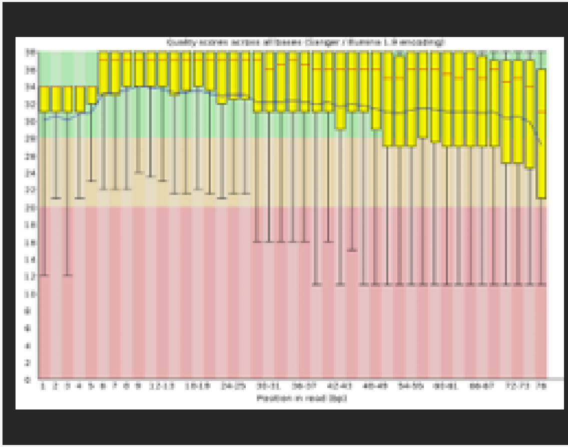
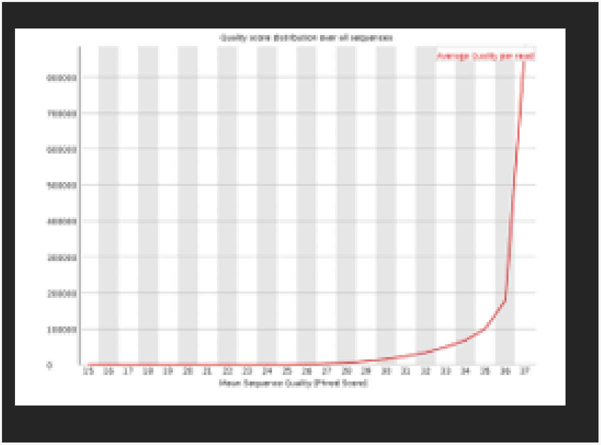
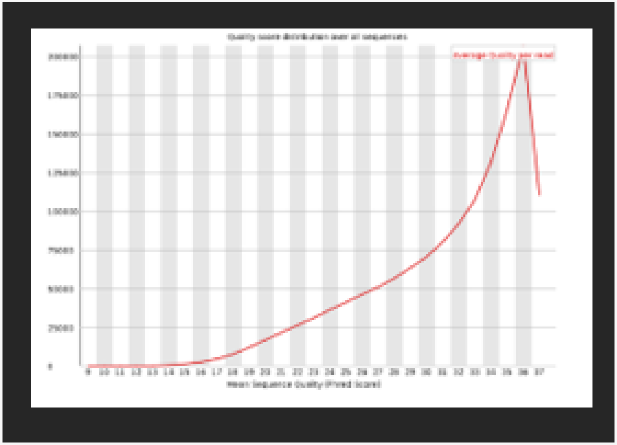
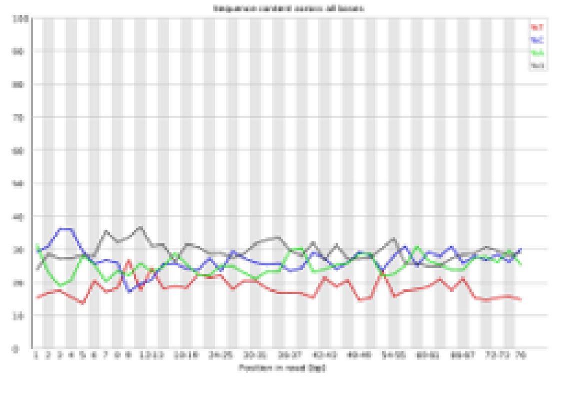
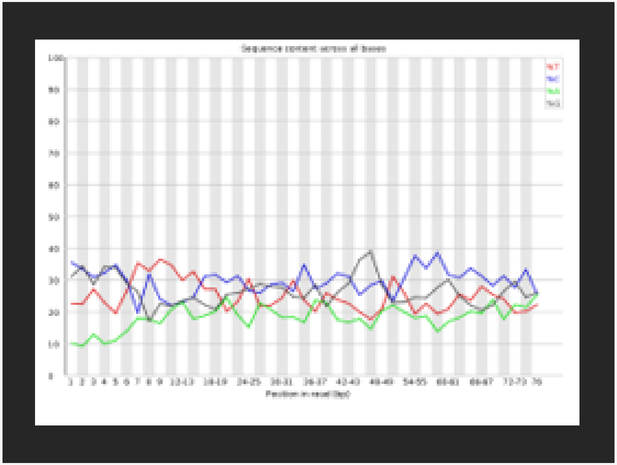
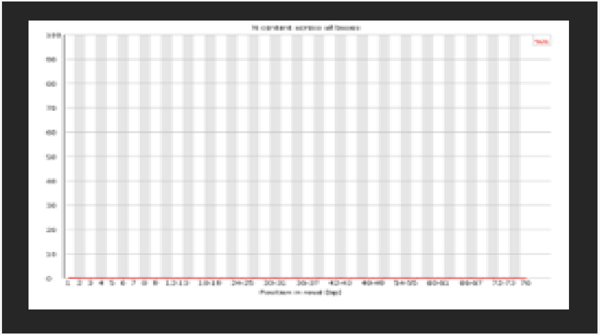
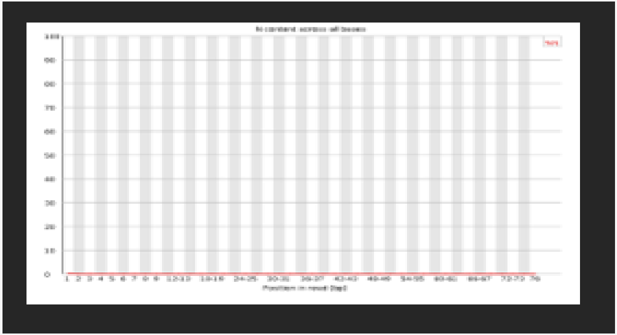
Collect basic statistics for selected data [RNA Data Set]

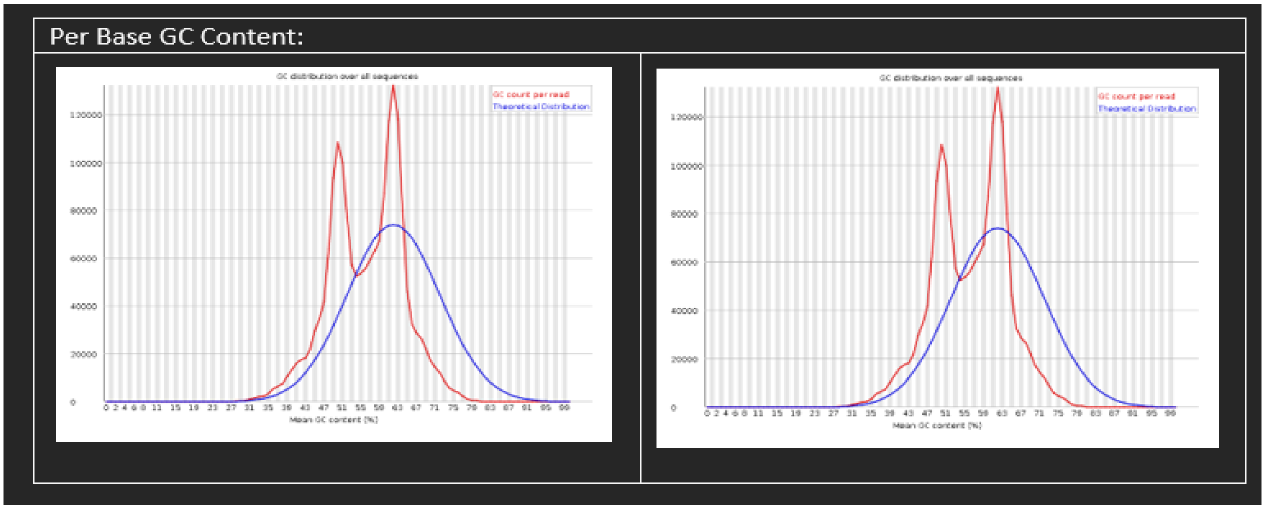
As the name implies Basic statistics is the primary characteristics of the sequence and that includes file name, file type, encoding total sequences, sequence flagged as poor quality, sequence length and %of GC content.

Measure	Read 1	Read 2
Filename	SRR11797448_1.fastq	SRR11797448_2.fastq
File type	Conventional base calls	Conventional base calls
Encoding	Sanger / Illumina 1.9	Sanger / Illumina 1.9

Total Sequences	1383968	1383968
Sequences flagged as poor quality	0	0
Sequence length	76	76
%GC	56	56

Save the report for read1 and read2 as *.html

SRR11797448_1_fastqc.html	SRR11797448_2_fastqc.html
Per Base Sequence Quality:	
	
Per sequence quality scores:	
	
Per base sequence content:	
	
Per base N content:	
	



Per Base GC Content:

CONCLUSION:

In read 1, all bases are in the green region, which signifies a good phred score and call quality. The average phred score is around 37, which indicates 99.98% base call accuracy with minimal errors, demonstrating excellent quality. N content is 0 throughout. Therefore, all these factors indicate trimming is not required

In read 2, bases lie in the green region with a considerable amount of them extending to the yellow region, signifying average call quality and phred scores, although there seems to be a significant amount of them of poor quality, indicated by a considerable amount of 90% whiskers in the orange and red regions. The average Per base N content still remains 0. Trimming can help improve the overall quality of the bases.

EXPERIMENT: 2

DE NOVO ASSEMBLY FOR RNA DATA SETS

Trinity

Trinity, developed at the Broad Institute and the Hebrew University of Jerusalem, represents a novel method for the efficient and robust de novo reconstruction of transcriptomes from RNA-seq data. Trinity combines three independent software modules: Inchworm, Chrysalis, and Butterfly, applied sequentially to process large volumes of RNA-seq reads. Trinity partitions the sequence data into many individual de Bruijn graphs, each representing the transcriptional complexity at a given gene or locus, and then processes each graph independently to extract full-length splicing isoforms and to tease apart transcripts derived from paralogous genes. Briefly, the process works like so:

- *Inchworm* assembles the RNA-seq data into the unique sequences of transcripts, often generating full-length transcripts for a dominant isoform, but then reports just the unique portions of alternatively spliced transcripts.
- *Chrysalis* clusters the Inchworm contigs into clusters and constructs complete de Bruijn graphs for each cluster. Each cluster represents the full transcriptonal complexity for a given gene (or sets of genes that share sequences in common). Chrysalis then partitions the full read set among these disjoint graphs.
- *Butterfly* then processes the individual graphs in parallel, tracing the paths that reads and pairs of reads take within the graph, ultimately reporting full-length transcripts for alternatively spliced isoforms, and teasing apart transcripts that correspond to paralogous genes.

THE ONLINE SERVERS AND TOOLS USED IN THIS EXERCISE ARE:

Trinity	https://github.com/trinityrnaseq/trinityrnaseq/wiki
Transdecoder	https://github.com/TransDecoder/TransDecoder/wiki

PROCEDURE FOR RNA DATASETS

Trinity Installation using Docker method

- Open terminal
- Make sure system is updated with latest packages, else update it with "**sudo apt-get update**" and "**sudo apt-get upgrade**" (Optional)

- Install docker tool--> **sudo apt install docker.io**
- Add current user to docker user group → **sudo gpasswd -a $USER docker**
- After this step please logout and login to the system
- Pull Trinity image for docker → **docker pull trinityrnaseq/trinityrnaseq** *(It may take time to pull images from source based on your internet speed)*

Running Trinity in docker method

Type below command in terminal

```
sudo docker run --rm -v "$(pwd)":"$(pwd)" trinityrnaseq/trinityrnaseq Trinity --
seqType fq --left
/home/pragathi/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/SRR1179
7448_1.fastq --right
/home/pragathi/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/SRR1179
7448_2.fastq --trimmomatic --max_memory 4G --CPU 2 --output
"$(pwd)"/trinity_out_dir
```

- What is docker? **It is a platform for running, shipping and developing applications in containers**
- What is --seqType? **It is an option used to specify type of sequencing data in Trinity**
- What is --left? **It is option used to specify the file containing the left (or forward) reads of paired-end RNA-Seq data**
- What is --right? **It is an option used to specify the file containing the right (or reverse) reads of paired-end RNA-Seq data**
- What is --trimmomatic? **It is a command that tells Trinity to use Trimmomatic to trim the adapters and unwanted sequences/bases before assembly**
- What is --max_memory? **It is used to signify the maximum amount of RAM Trinity can use during the process of assembly**
- What is –CPU? **It is used to signify the number of CPU cores that can be used during the assembly process**

Wait for minimum 30 min to complete trinity. Tirntiy.fasta file will be generated in output folder

- What is the size of Tirntiy.fasta? **1.5 MB**

TransDecoder:

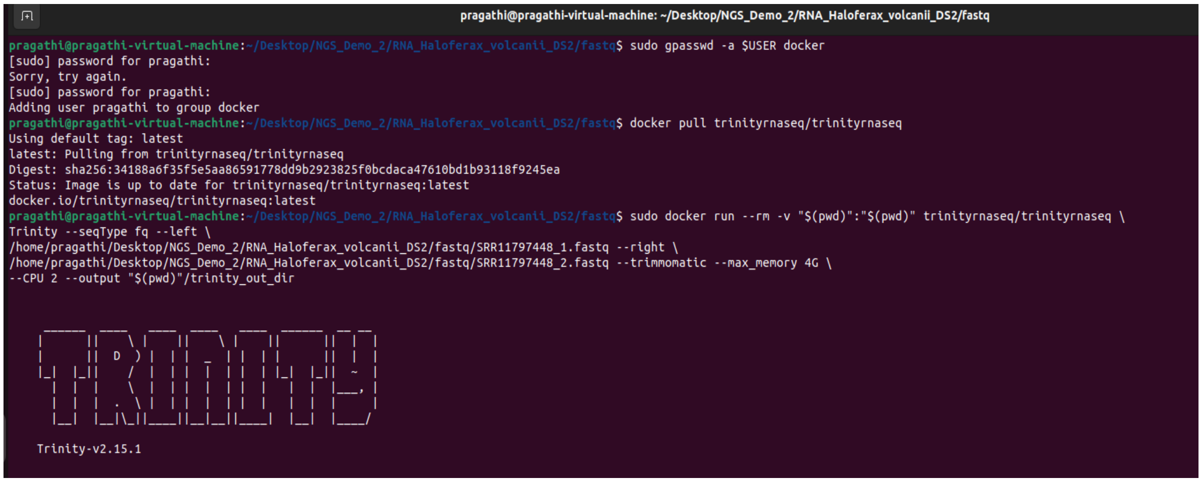
Running TransDecoder:

TransDecoder.LongOrfs -t Trinity.fasta

- What is the total no of CDS in longest orf cds file? **3653**
- What is the total no of Protein in pep file? **3653**

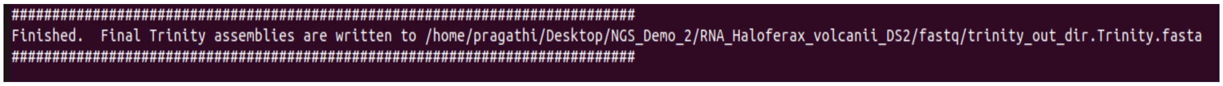
CONCLUSION:

Trinity Window:



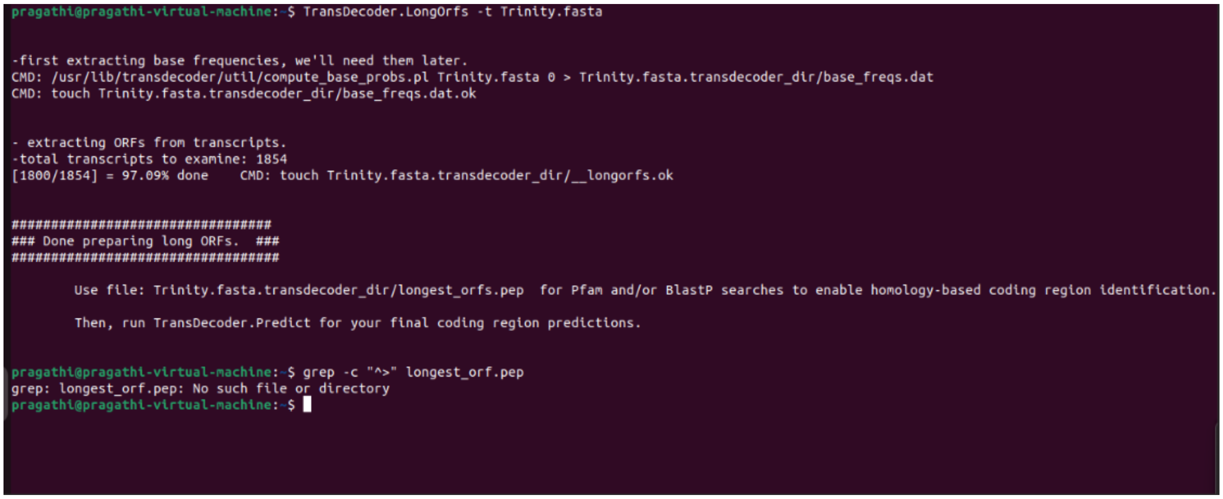
pragathi@pragathi-virtual-machine: ~/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq

```
pragathi@pragathi-virtual-machine:~/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq$ sudo gpasswd -a $USER docker
[sudo] password for pragathi:
Sorry, try again.
[sudo] password for pragathi:
Adding user pragathi to group docker
pragathi@pragathi-virtual-machine:~/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq$ docker pull trinityrnaseq/trinityrnaseq
Using default tag: latest
latest: Pulling from trinityrnaseq/trinityrnaseq
Digest: sha256:34188a6f35f5e5aa86591778dd9b2923825f0bcdaca47610bd1b93118f9245ea
Status: Image is up to date for trinityrnaseq/trinityrnaseq:latest
docker.io/trinityrnaseq/trinityrnaseq:latest
pragathi@pragathi-virtual-machine:~/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq$ sudo docker run --rm -v "$(pwd)":"$(pwd)" trinityrnaseq/trinityrnaseq \
Trinity --seqType fq --left \
/home/pragathi/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/SRR11797448_1.fastq --right \
/home/pragathi/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/SRR11797448_2.fastq --trimmomatic --max_memory 4G \
--CPU 2 --output "$(pwd)"/trinity_out_dir


     ______  ____   ____  ____   ____  ______  __ __
    |      ||    \ |    ||    \ |    ||      ||  |  |
    |      ||  D  ) |  | |  _  | |  | |      ||  |  |
    |_|  |_||    /  |  | |  |  | |  | |_|  |_||  ~  |
      |  |  |    \  |  | |  |  | |  |   |  |  |___, |
      |  |  |  .  \ |  | |  |  | |  |   |  |  |     |
      |__|  |__|\_||____||__|__||____|  |__|  |____/

    Trinity-v2.15.1
```



```
################################################################################
Finished.  Final Trinity assemblies are written to /home/pragathi/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/trinity_out_dir.Trinity.fasta
################################################################################
```

Running: TransDecoder



```
pragathi@pragathi-virtual-machine:~$ TransDecoder.LongOrfs -t Trinity.fasta


-first extracting base frequencies, we'll need them later.
CMD: /usr/lib/transdecoder/util/compute_base_probs.pl Trinity.fasta 0 > Trinity.fasta.transdecoder_dir/base_freqs.dat
CMD: touch Trinity.fasta.transdecoder_dir/base_freqs.dat.ok


- extracting ORFs from transcripts.
-total transcripts to examine: 1854
[1800/1854] = 97.09% done    CMD: touch Trinity.fasta.transdecoder_dir/__longorfs.ok


#################################
### Done preparing long ORFs.  ###
##################################

        Use file: Trinity.fasta.transdecoder_dir/longest_orfs.pep  for Pfam and/or BlastP searches to enable homology-based coding region identification.

        Then, run TransDecoder.Predict for your final coding region predictions.


pragathi@pragathi-virtual-machine:~$ grep -c "^>" longest_orf.pep
grep: longest_orf.pep: No such file or directory
pragathi@pragathi-virtual-machine:~$ █
```

In De novo transcriptome assembly using Trinity, both forward and reverse reads are used to construct help construct longer and accurate contigs (which are contiguous sequences constructed from overlapping strands), which helps in reduction of ambiguity and helps feed gaps between individual reads

The results on running Transdecoder indicate the presence of 3653 CDS (coding DNA sequences), which basically means that TransDecoder has predicted 3653 protein coding regions within the assembled transcripts. The corresponding .pep file consists of 3653 amino acid sequences that are derived from the CDS. Each of the CDS and amino acid sequences can be found by opening the FASTA file in a text editor and searching for the '>' symbol, which indicates the entry point of CDS/amino acid sequence.

EXPERIMENT: 3

GENOME ANNOTATION AND GENE ONTOLOGY (GO)

LITERATURE:

Panther:

The PANTHER (**P**rotein **AN**alysis **TH**rough **E**volutionary **R**elationships) Classification System was designed to classify proteins (and their genes) in order to facilitate high-throughput analysis.

THE ONLINE SERVERS AND TOOLS USED IN THIS EXERCISE ARE:

Panther	http://www.pantherdb.org/
BLAST+	Local Blast
Database	Uniprot

Counting Full Length Trinity Transcripts

#Full-length transcript analysis for model and non-model organisms using BLAST+

Useful protein databases to search include : SwissProt (ftp://ftp.uniprot.org/pub/databases/uniprot/current_release/knowledgebase/complete/uniprot_sprot.fasta.gz) download this data or collect from GPU1080 server

Download BLAST+ from given link ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/ or collect from GPU1080 server

Running BLASTx against (Uniprot sprot database)

Build a blastable database:

Type below command

```
makeblastdb -in uniprot_sprot.fasta -dbtype prot
```

- What is makeblastdb? **It is a command used to create a local BLAST database**
- What is -in? **It is a parameter used to specify the input file**
- What is uniprot_sprot.fasta? **It is the input file containing protein sequences in FASTA format, a subset of the UniProtKB , concerning the Swiss Prot section**

- What is prot? **It indicates the type of database to be generated is a protein database**

Perform the blast search, reporting only the top alignment:

Type below command

```
time blastx -query cds.fasta -db filename.fasta -out outputfilename -evalue  1e-20 -num_threads 6 -max_target_seqs 1 -outfmt 6
```

- What is -evalue? **It is parameter that sets the Expect (E) value threshold for saving hits ("hit" refers to a sequence similarity or alignment between the query sequence and a sequence in the database being searched.). In this case , only matches with E value less than 1e-20 will be reported**
- What is -num_threads ? **It is parameter specifies the number of CPU threads to use for the BLAST search (CPU threads are virtual processing units within a central processing unit (CPU) that can execute instructions independently.). Here it is 6**
- What is -max_target_seqs? **It is used to specify the maximum number of target sequences to report for each query, here in the above command , it is 1**
- What is -outfmt 6? **It is a parameter specifies the output format of the results. In this case, it is set to format 6, which is a tabular format that includes various fields such as query sequence ID, subject sequence ID, percentage of identity, alignment length, E value, and more.**
- Total no of sequences in the database? **570420**
- How much time does it tooks? **125 minutes 45.395 seconds**
- How many transcripts are annotated? **1542**

Running BLASTx against (uniprot – Mycobacterium (leprae) - Sprot database)

- Total no of sequences in the database? (depends on threads)

669

- How much time does it tooks? **58.907 seconds**
- How many transcripts are annotated? **227**

Running BLASTx against (uniprot – Mycobacterium (leprae) – trembl database)

- Total no of sequences in the database? **1603**
- How much time does it took? **1 minute 3.06 seconds**
- How many transcripts are annotated? **333**

Running BLASTx against (uniprot - uniprot Mycobacterium tuberculosis sprot) •

- Total no of sequences in the database? **2304**
- How much time does it took? **1 minute 1.517 seconds**
- How many transcripts are annotated? **392**

Running BLASTx against (uniprot - uniprot Mycobacterium tuberculosis trembl)

- Total no of sequences in the database? **3995**
- How much time does it took? **1 minute 36.232 seconds**
- How many transcripts are annotated? **434**
- Which database has more annotation? **Uniprot sprot database**

Make a table.

S.no	Database	Total no transcript	Total no of sequences in database	Total no of transcript annotated
1	Uniprot sprot database	**1554**	**570420**	**1542**
2	Mycobacterium (leprae) sprot database	**229**	**669**	**227**
3	Mycobacterium (leprae) trembl database	**343**	**1603**	**333**
4	Mycobacterium tuberculosis sprot	**402**	**2304**	**392**

5	Mycobacterium tuberculosis tremble	**444**	**3995**	**434**

Panther:

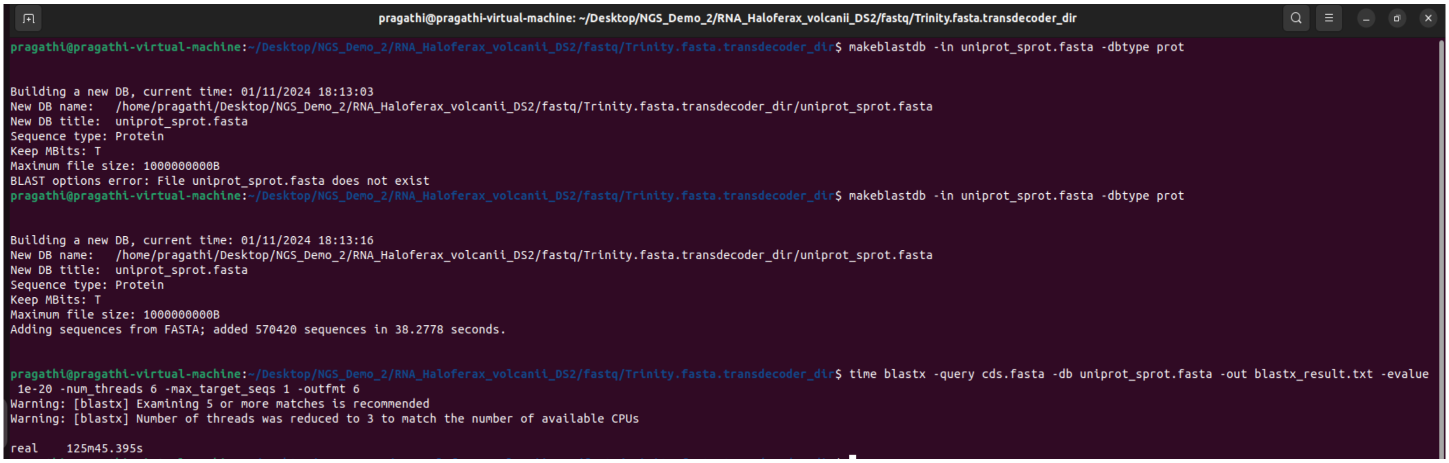
- Go to Panther website: http://www.pantherdb.org/
- Paste your uniprot accession no in to that search box (example: generated from BLASTx result)
  - How many accessions have you copied? **1**

(O32220)

- Select all the organisms and remaining parameters are default and click submit
  - How many gene ontology result came? **27**
- Click on pie chart and select MF pie chart
  - What is MF ? **Molecular Function**
- Take print screen and analysis the result
- Click on pie chart and select BP pie chart
  - What is BP? **Biological Process**
- Take print screen and analysis the result
- Click on pie chart and select CC pie chart
  - What is CC ? **Cellular Component**
- Take print screen and analysis the result

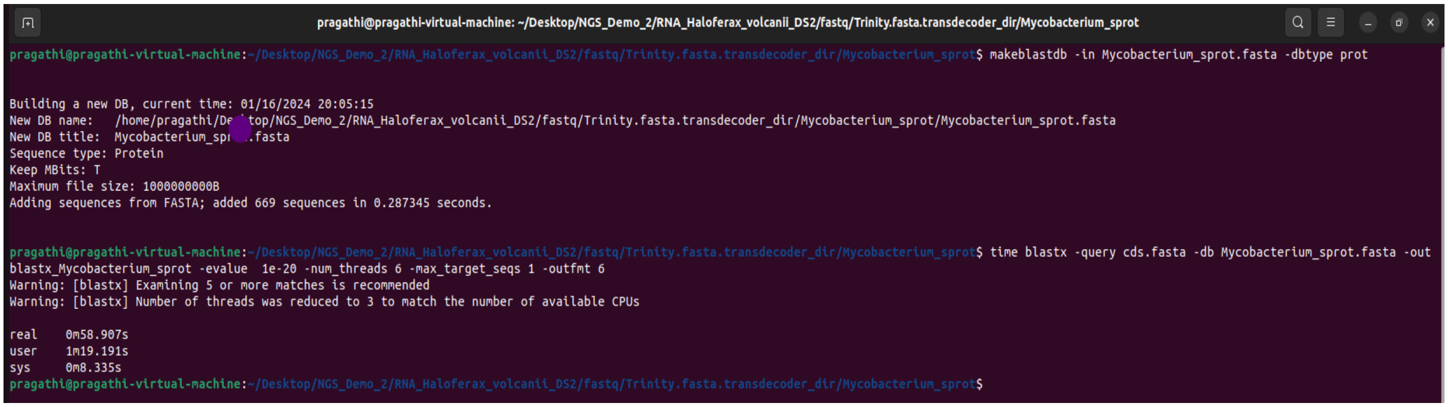
CONCLUSION:

Running BLASTX against various protein databases


pragathi@pragathi-virtual-machine: ~/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/Trinity.fasta.transdecoder_dir

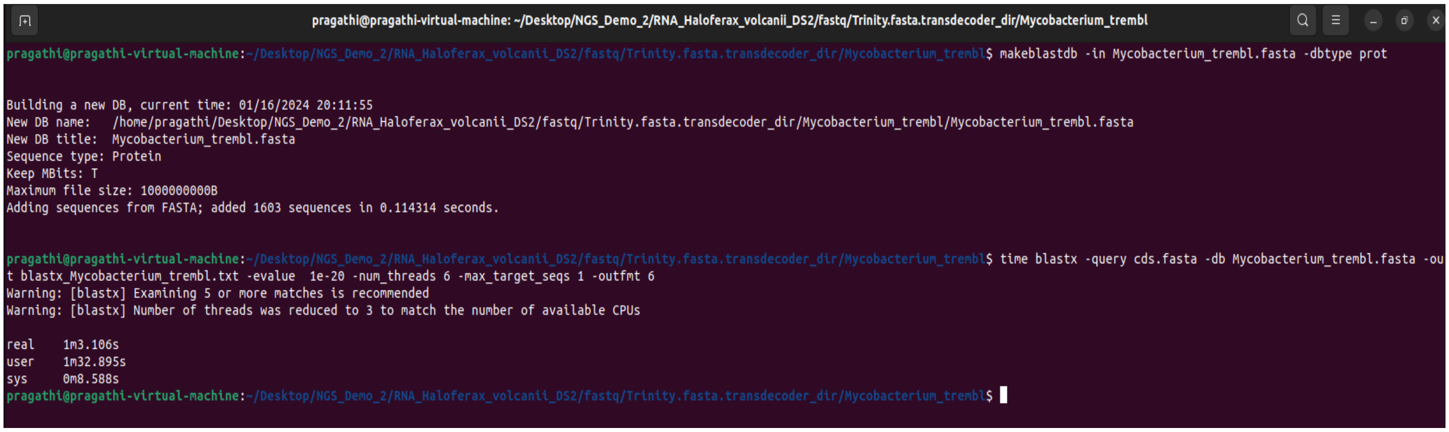
```
pragathi@pragathi-virtual-machine:~/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/Trinity.fasta.transdecoder_dir$ makeblastdb -in uniprot_sprot.fasta -dbtype prot


Building a new DB, current time: 01/11/2024 18:13:03
New DB name:   /home/pragathi/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/Trinity.fasta.transdecoder_dir/uniprot_sprot.fasta
New DB title:  uniprot_sprot.fasta
Sequence type: Protein
Keep MBits: T
Maximum file size: 1000000000B
BLAST options error: File uniprot_sprot.fasta does not exist
pragathi@pragathi-virtual-machine:~/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/Trinity.fasta.transdecoder_dir$ makeblastdb -in uniprot_sprot.fasta -dbtype prot


Building a new DB, current time: 01/11/2024 18:13:16
New DB name:   /home/pragathi/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/Trinity.fasta.transdecoder_dir/uniprot_sprot.fasta
New DB title:  uniprot_sprot.fasta
Sequence type: Protein
Keep MBits: T
Maximum file size: 1000000000B
Adding sequences from FASTA; added 570420 sequences in 38.2778 seconds.


pragathi@pragathi-virtual-machine:~/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/Trinity.fasta.transdecoder_dir$ time blastx -query cds.fasta -db uniprot_sprot.fasta -out blastx_result.txt -evalue
 1e-20 -num_threads 6 -max_target_seqs 1 -outfmt 6
Warning: [blastx] Examining 5 or more matches is recommended
Warning: [blastx] Number of threads was reduced to 3 to match the number of available CPUs

real    125m45.395s
```

Fig: Running BLASTX against uniport sprot database


pragathi@pragathi-virtual-machine: ~/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/Trinity.fasta.transdecoder_dir/Mycobacterium_sprot

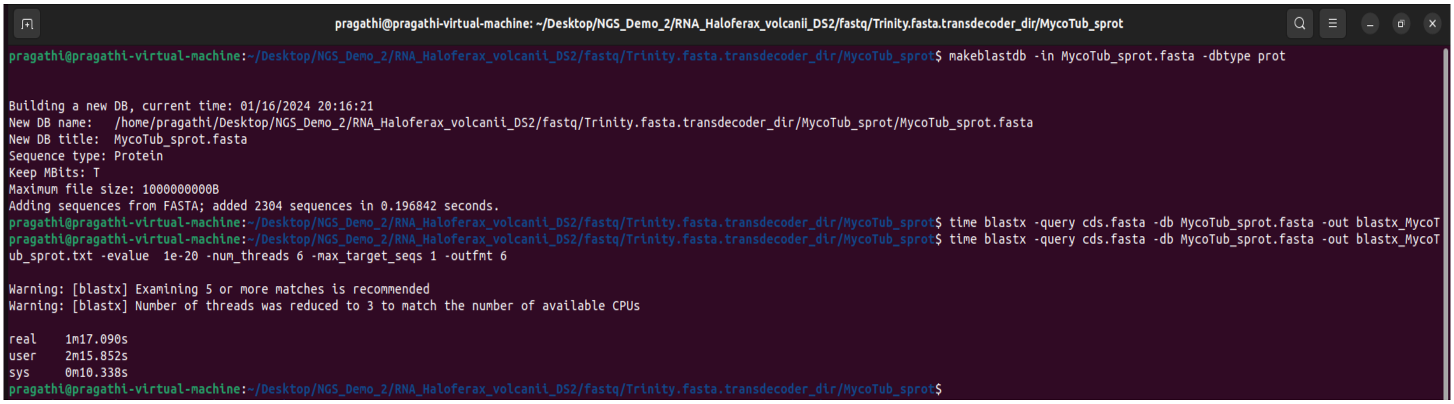
```
pragathi@pragathi-virtual-machine:~/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/Trinity.fasta.transdecoder_dir/Mycobacterium_sprot$ makeblastdb -in Mycobacterium_sprot.fasta -dbtype prot


Building a new DB, current time: 01/16/2024 20:05:15
New DB name:   /home/pragathi/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/Trinity.fasta.transdecoder_dir/Mycobacterium_sprot/Mycobacterium_sprot.fasta
New DB title:  Mycobacterium_sprot.fasta
Sequence type: Protein
Keep MBits: T
Maximum file size: 1000000000B
Adding sequences from FASTA; added 669 sequences in 0.287345 seconds.


pragathi@pragathi-virtual-machine:~/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/Trinity.fasta.transdecoder_dir/Mycobacterium_sprot$ time blastx -query cds.fasta -db Mycobacterium_sprot.fasta -out
blastx_Mycobacterium_sprot -evalue  1e-20 -num_threads 6 -max_target_seqs 1 -outfmt 6
Warning: [blastx] Examining 5 or more matches is recommended
Warning: [blastx] Number of threads was reduced to 3 to match the number of available CPUs

real    0m58.907s
user    1m19.191s
sys     0m8.335s
pragathi@pragathi-virtual-machine:~/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/Trinity.fasta.transdecoder_dir/Mycobacterium_sprot$
```

Fig: Running BLASTX against Mycobacterium (leprae) sprot database


pragathi@pragathi-virtual-machine: ~/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/Trinity.fasta.transdecoder_dir/Mycobacterium_trembl

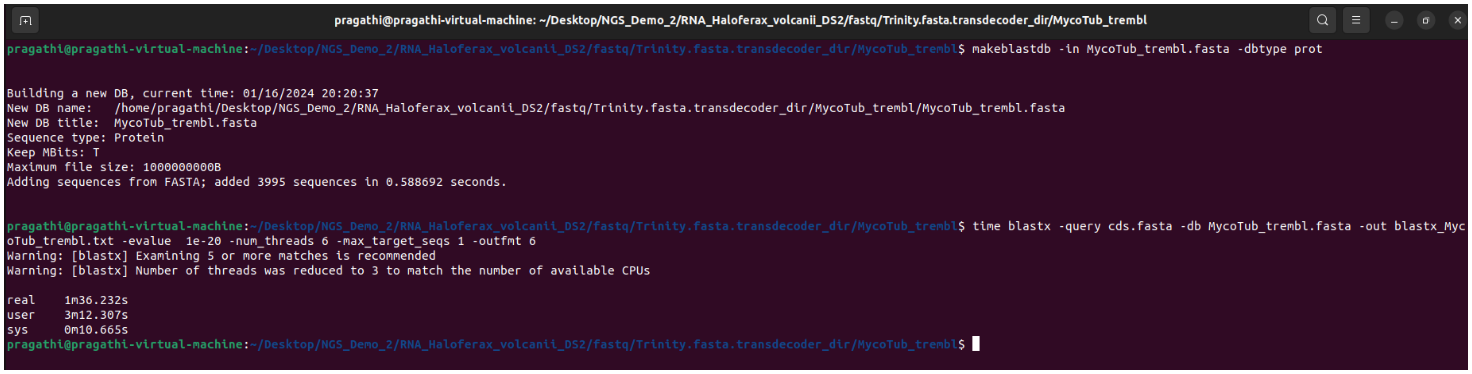
```
pragathi@pragathi-virtual-machine:~/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/Trinity.fasta.transdecoder_dir/Mycobacterium_trembl$ makeblastdb -in Mycobacterium_trembl.fasta -dbtype prot


Building a new DB, current time: 01/16/2024 20:11:55
New DB name:   /home/pragathi/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/Trinity.fasta.transdecoder_dir/Mycobacterium_trembl/Mycobacterium_trembl.fasta
New DB title:  Mycobacterium_trembl.fasta
Sequence type: Protein
Keep MBits: T
Maximum file size: 1000000000B
Adding sequences from FASTA; added 1603 sequences in 0.114314 seconds.


pragathi@pragathi-virtual-machine:~/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/Trinity.fasta.transdecoder_dir/Mycobacterium_trembl$ time blastx -query cds.fasta -db Mycobacterium_trembl.fasta -ou
t blastx_Mycobacterium_trembl.txt -evalue  1e-20 -num_threads 6 -max_target_seqs 1 -outfmt 6
Warning: [blastx] Examining 5 or more matches is recommended
Warning: [blastx] Number of threads was reduced to 3 to match the number of available CPUs

real    1m3.106s
user    1m32.895s
sys     0m8.588s
pragathi@pragathi-virtual-machine:~/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/Trinity.fasta.transdecoder_dir/Mycobacterium_trembl$ 
```

Fig: Running BLASTX against Mycobacterium (leprae) trembl database



```
pragathi@pragathi-virtual-machine: ~/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/Trinity.fasta.transdecoder_dir/MycoTub_sprot
pragathi@pragathi-virtual-machine:~/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/Trinity.fasta.transdecoder_dir/MycoTub_sprot$ makeblastdb -in MycoTub_sprot.fasta -dbtype prot


Building a new DB, current time: 01/16/2024 20:16:21
New DB name:   /home/pragathi/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/Trinity.fasta.transdecoder_dir/MycoTub_sprot/MycoTub_sprot.fasta
New DB title:  MycoTub_sprot.fasta
Sequence type: Protein
Keep MBits: T
Maximum file size: 1000000000B
Adding sequences from FASTA; added 2304 sequences in 0.196842 seconds.
pragathi@pragathi-virtual-machine:~/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/Trinity.fasta.transdecoder_dir/MycoTub_sprot$ time blastx -query cds.fasta -db MycoTub_sprot.fasta -out blastx_MycoT
pragathi@pragathi-virtual-machine:~/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/Trinity.fasta.transdecoder_dir/MycoTub_sprot$ time blastx -query cds.fasta -db MycoTub_sprot.fasta -out blastx_MycoT
ub_sprot.txt -evalue  1e-20 -num_threads 6 -max_target_seqs 1 -outfmt 6

Warning: [blastx] Examining 5 or more matches is recommended
Warning: [blastx] Number of threads was reduced to 3 to match the number of available CPUs

real    1m17.090s
user    2m15.852s
sys     0m10.338s
pragathi@pragathi-virtual-machine:~/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/Trinity.fasta.transdecoder_dir/MycoTub_sprot$
```

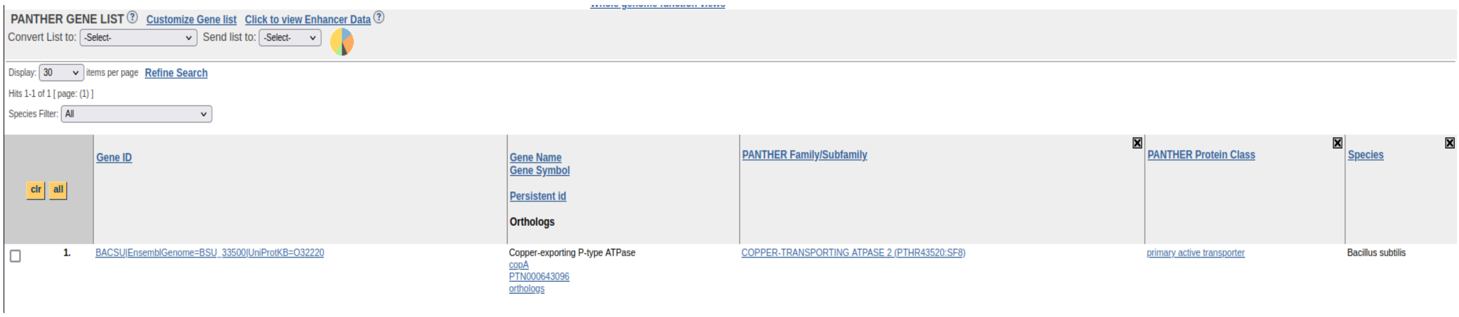
Fig: Running BLASTX against Mycobacterium tuberculosis sprot database



```
pragathi@pragathi-virtual-machine: ~/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/Trinity.fasta.transdecoder_dir/MycoTub_trembl
pragathi@pragathi-virtual-machine:~/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/Trinity.fasta.transdecoder_dir/MycoTub_trembl$ makeblastdb -in MycoTub_trembl.fasta -dbtype prot


Building a new DB, current time: 01/16/2024 20:20:37
New DB name:   /home/pragathi/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/Trinity.fasta.transdecoder_dir/MycoTub_trembl/MycoTub_trembl.fasta
New DB title:  MycoTub_trembl.fasta
Sequence type: Protein
Keep MBits: T
Maximum file size: 1000000000B
Adding sequences from FASTA; added 3995 sequences in 0.588692 seconds.

pragathi@pragathi-virtual-machine:~/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/Trinity.fasta.transdecoder_dir/MycoTub_trembl$ time blastx -query cds.fasta -db MycoTub_trembl.fasta -out blastx_Myc
oTub_trembl.txt -evalue  1e-20 -num_threads 6 -max_target_seqs 1 -outfmt 6
Warning: [blastx] Examining 5 or more matches is recommended
Warning: [blastx] Number of threads was reduced to 3 to match the number of available CPUs

real    1m36.232s
user    3m12.307s
sys     0m10.665s
pragathi@pragathi-virtual-machine:~/Desktop/NGS_Demo_2/RNA_Haloferax_volcanii_DS2/fastq/Trinity.fasta.transdecoder_dir/MycoTub_trembl$ 
```

Fig: Running BLASTX against Mycobacterium tuberculosis trembl database

Gene ontology



PANTHER GENE LIST ? Customize Gene list Click to view Enhancer Data ?

Convert List to: -Select- Send list to: -Select-

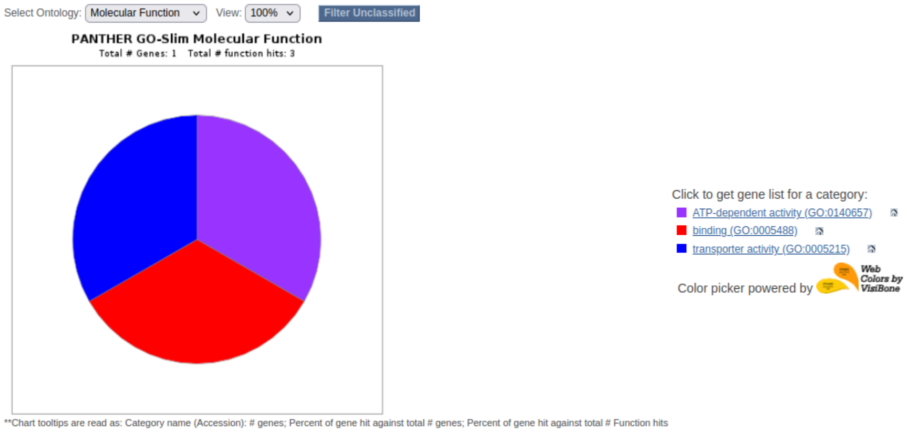
Display: 30 items per page Refine Search

Hits 1-1 of 1 [page: (1)]

Species Filter: All

clr all	Gene ID	Gene Name Gene Symbol Persistent id Orthologs	PANTHER Family/Subfamily ☒	PANTHER Protein Class ☒	Species ☒
□ 1.	BACSU|EnsemblGenome=BSU_33500|UniProtKB=O32220	Copper-exporting P-type ATPase copA PTN000643096 orthologs	COPPER-TRANSPORTING ATPASE 2 (PTHR43520:SF8)	primary active transporter	Bacillus subtilis

Fig: Gene list obtained for UniProt accession ID O32220, obtained from blastx_result_uniprot_sprot.txt



Select Ontology: Molecular Function View: 100%

PANTHER GO-Slim Molecular Function

Total # Genes: 1 Total # function hits: 3

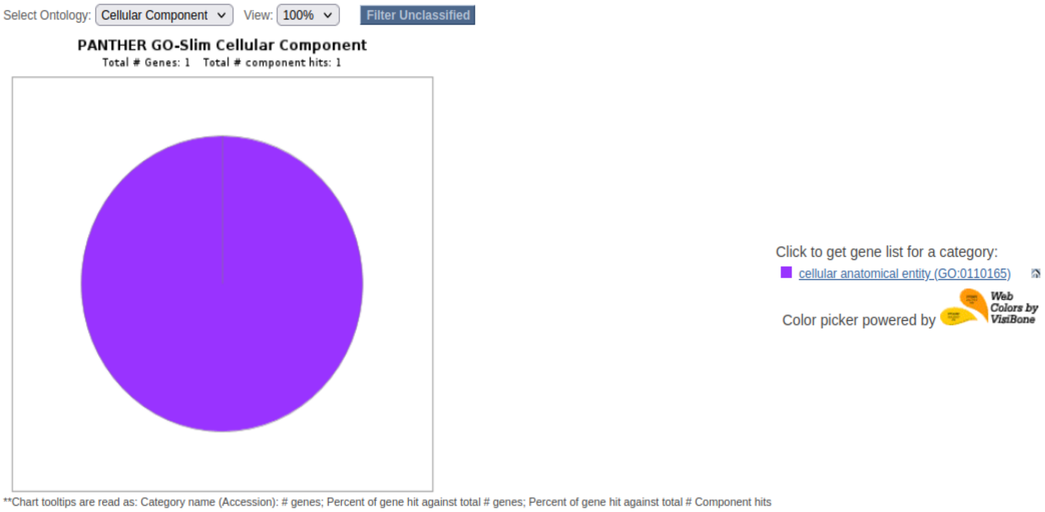
Click to get gene list for a category:

- 🟪 ATP-dependent activity (GO:0140657)
- 🟥 binding (GO:0005488)
- 🟦 transporter activity (GO:0005215)

Color picker powered by Web Colors by VisiBone

**Chart tooltips are read as: Category name (Accession): # genes; Percent of gene hit against total # genes; Percent of gene hit against total # Function hits

Fig: Pie chart of Molecular function



Select Ontology: Cellular Component View: 100%

PANTHER GO-Slim Cellular Component

Total # Genes: 1 Total # component hits: 1

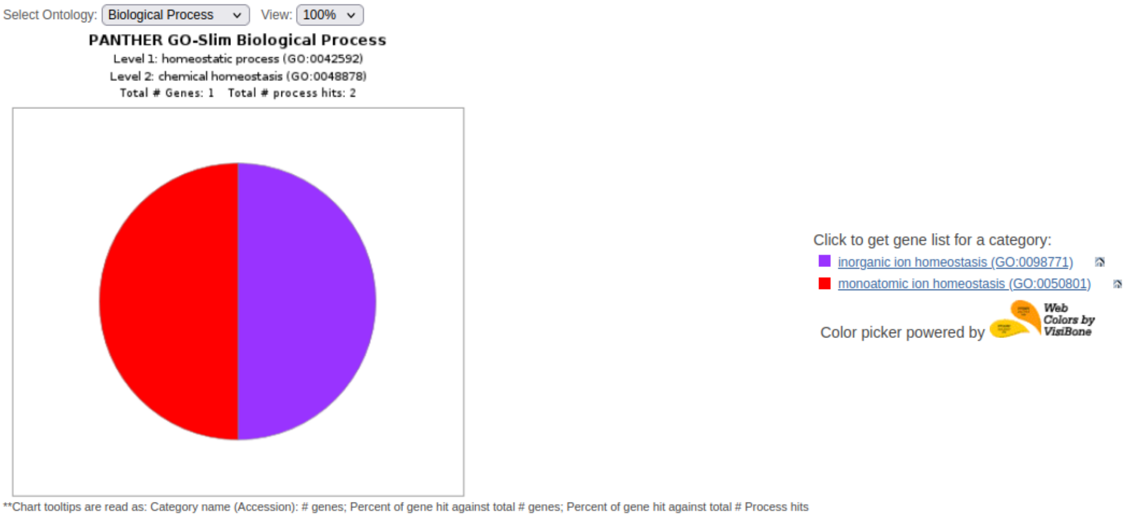
Click to get gene list for a category:

- 🟪 cellular anatomical entity (GO:0110165)

Color picker powered by Web Colors by VisiBone

**Chart tooltips are read as: Category name (Accession): # genes; Percent of gene hit against total # genes; Percent of gene hit against total # Component hits

Fig: Pie chart of Cellular Component



Select Ontology: Biological Process View: 100%

PANTHER GO-Slim Biological Process

Level 1: homeostatic process (GO:0042592)

Level 2: chemical homeostasis (GO:0048878)

Total # Genes: 1 Total # process hits: 2

Click to get gene list for a category:

- 🟪 inorganic ion homeostasis (GO:0098771)
- 🟥 monoatomic ion homeostasis (GO:0050801)

Color picker powered by Web Colors by VisiBone

**Chart tooltips are read as: Category name (Accession): # genes; Percent of gene hit against total # genes; Percent of gene hit against total # Process hits

Fig: Pie chart of Biological Process