1. **NTM.py** => creating a csv from ntm\_log.txt

Input : ntm\_log.txt

Command : **NTM.py**

1. **Cryptic\_compare.py** => compare the dst results with cryptic results

Input : Folder containing the dst tables

Command : **py crytpic\_compare.py -p /path/to/parent\_folder\_where\_pipeline\_results\_store -c /path/to/cryptic/csv -o path/to/output**

1. **p1p2.py** => compares the dst result of 2 set of dst results derived from different catalogs

Input : p1 input of folder containing all dst files from New Catalog; p2 input of folder containing all dst files from Old Catalog

Command :**python p1p2.py -p2 /home/alphabox0006/Deepthi/CRYPTIC/Input/Batch-cryptic\_oldcatalog\_dst\_tables -p1 /home/alphabox0006/Deepthi/CRYPTIC/Input/Cryptic\_DST\_tables -o /home/alphabox0006/Deepthi/CRYPTIC/Testing**

1. **Catalogue.py** => creating a catalog for mutations in TB using WHO-Mutation table

Input : Genome\_indices.tsv , Mutation\_catalog.tsv, Mycobacterium\_tuberculosis\_H37Rv.tsv => keep in same directory as script

Command : **Catalogue.py**

1. **32\_genes\_bench\_batc**h => script for mapping ref genome - variant calling steps and benchmarking

Input file : name of ref fasta file without .fasta and **change the path of input fastq files**

Command: **sh 32\_genes\_bench\_batch.sh <name of ref fasta without .fasta>**

1. **Alignmentpipeline\_bwa\_v3.p**y => edited the original script to make changes in names of output files

Input : fastq files

Command: **python ../alignmentpipeline\_bwa\_v3.py -f1 SRR3500411\_R1.fastq.gz -f2 SRR3500411\_R2.fastq.gz -r reference.fasta -t 8 -fq U**

1. **Merge\_catalogs.py** => This script was used to check for duplicate entries based on a "filter" column and extract unique values into a separate df. it was later merged to create the final catalog.

Input : TB catalogue generated from script “Catalogue.py” and Old\_catalog.csv

Command : **python Merge\_catalogs.py**

1. **Depth\_file\_check** => This script reads depth files and filters out those files where depth <50 ,it also filters summary files according to 2 conditions : Lineage is unknown and NO Mutation detected

Input : path to directory containing depth file and summary.csv

Command: **python check.py -d /path/to/dir/where/Batch60\_07012023**

1. **Indelcat.py** => script to create indel catalog from who\_mutation catalogs

Input : Genome\_indices.tsv , Mutation\_catalog.tsv, Mycobacterium\_tuberculosis\_H37Rv.tsv => keep in same directory as script

Command : **Catalogue.py**

1. **Cryptic\_select.py** => this script will take the cryptic\_reuse table and select only those rows which are required. Further renaming certain rows (filtering cryptic table according to your needs)

Input : 1. Cryptic \_reuse table

2. CRYpTIC PRJEB41116 .tsv (go to Ena Browser search for the project number and download report as tsv )

Command:**python cryptic\_select.py**

1. **log.sh** => It takes MTBseq log files and notes the start and stop time and makes a csv for all files given.

**Input : Log files**

Run the script inside the folder containing all log files

**Output : Csv file and text file having start and stop times**

**Command : sh log.sh**

1. **Depth.py** => Reads depth files and calculator length of probable deletion

Logic : Given a depth file, the script filters out positions with depth < 10 => probable deletions

Next consecutive positions are grouped into ranges

And then these ranges are considered probable deletion and length of probable deletion is determined.

**Input: depth.txt**

**Output : Csv file with Start, Stop and length of deletion**

**Command : python depth.py -f1 <depth file to be analyzed> -o <path to folder for storing output>**

Inside archive files

1. **match\_csv\_isin.py**=>this script compares 2 sorted tsv files and gives the difference between them as output using isin method

Input : two csv to compare (change the path for required input)

Command : match\_csv.py

1. **Matchcat.py** => this code reads a csv , sorts it by a column and then compares it linewise to check if both are the same. If same output in Resfile; if not the rows with diff values in change.csv

Input : two csv to compare (change the path for required input)

Command : Matchcat.py

1. **Check\_range.py** => this script is for the 32 genes of TB file to check which SNPs fall into gene position ranges .Output are those which do not fall into any of the ranges

Input : A csv containing all positions(change the path for required input) (start and stop or in this case left and right)

Command: check\_range.py

4. Inorder to compare two csv with multiple column having same column names and if shape of both df is same use **df1.compare(df2)**

**5.** Fit\_range.py => finds which all depth values ranges are present in a given range