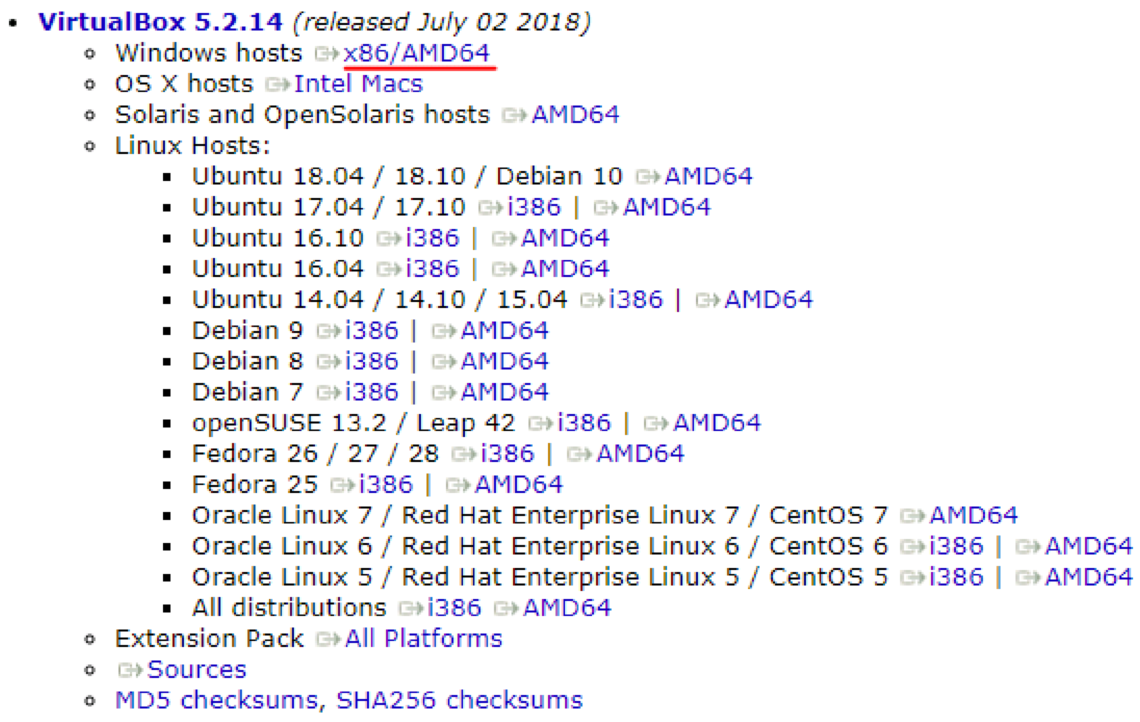
How to use the VirtualBox image to run the Gen2Epi version 0.1 pipeline

1) Download the VirtualBox 5.2.14 platform package (https://www.virtualbox.org/wiki/Downloads) appropriate for your host machine. Please Note:- To download the VirtualBox 5.2.14 version, please visit https://www.virtualbox.org/wiki/Download_Old_Builds_5_2 and download the 5.2.14 version as shown below:



- **VirtualBox 5.2.14** *(released July 02 2018)*
  - Windows hosts ⇨x86/AMD64
  - OS X hosts ⇨Intel Macs
  - Solaris and OpenSolaris hosts ⇨AMD64
  - Linux Hosts:
    - Ubuntu 18.04 / 18.10 / Debian 10 ⇨AMD64
    - Ubuntu 17.04 / 17.10 ⇨i386 | ⇨AMD64
    - Ubuntu 16.10 ⇨i386 | ⇨AMD64
    - Ubuntu 16.04 ⇨i386 | ⇨AMD64
    - Ubuntu 14.04 / 14.10 / 15.04 ⇨i386 | ⇨AMD64
    - Debian 9 ⇨i386 | ⇨AMD64
    - Debian 8 ⇨i386 | ⇨AMD64
    - Debian 7 ⇨i386 | ⇨AMD64
    - openSUSE 13.2 / Leap 42 ⇨i386 | ⇨AMD64
    - Fedora 26 / 27 / 28 ⇨i386 | ⇨AMD64
    - Fedora 25 ⇨i386 | ⇨AMD64
    - Oracle Linux 7 / Red Hat Enterprise Linux 7 / CentOS 7 ⇨AMD64
    - Oracle Linux 6 / Red Hat Enterprise Linux 6 / CentOS 6 ⇨i386 | ⇨AMD64
    - Oracle Linux 5 / Red Hat Enterprise Linux 5 / CentOS 5 ⇨i386 | ⇨AMD64
    - All distributions ⇨i386 ⇨AMD64
  - Extension Pack ⇨All Platforms
  - ⇨Sources
  - MD5 checksums, SHA256 checksums

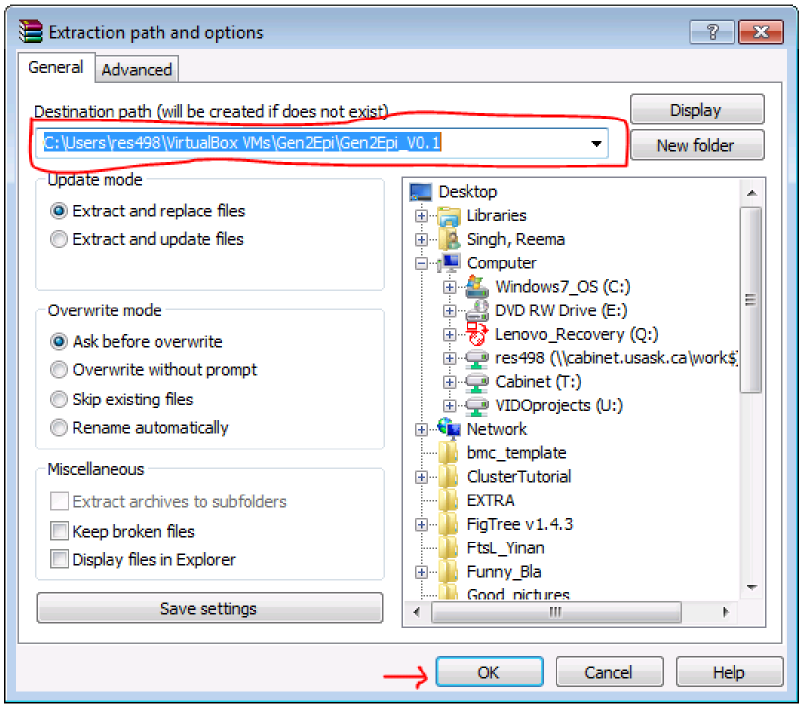
Please Note: - Gen2Epi VM image working fine with VirtualBox 5.2.22. However, if users encounter any problem then please use VirtualBox 5.2.14 from old build as shown above.

2) After installation, start VirtualBox, e.g. by clicking on the "Oracle VM VirtualBox" icon
3) Download "Gen2Epi_V0.1.rar" from ftp://www.cs.usask.ca/pub/combi. The downloaded file should be 13 GB in size. Extract the VM by using "unrar e Gen2Epi_V0.1.rar". Following message indicates that extraction is successful on a Linux machine:

```
"Extracting from Gen2Epi_V0.1.rar
 Extracting Gen2Epi.vdi                              OK
 All OK"
```

The extracted "Gen2Epi.vdi" is 28 GB in size.

To extract the VM on window machine please right click on the “Gen2Epi_V0.1.rar” file and click on “Extract Files” and click OK as shown below:


Extraction path and options

General | Advanced

Destination path (will be created if does not exist)

C:\Users\res498\VirtualBox VMs\Gen2Epi\Gen2Epi_V0.1

Update mode
- [✓ Extract and replace files]
- Extract and update files

Overwrite mode
- [✓ Ask before overwrite]
- Overwrite without prompt
- Skip existing files
- Rename automatically

Miscellaneous
- Extract archives to subfolders
- Keep broken files
- Display files in Explorer

Desktop, Libraries, Singh, Reema, Computer, Windows7_OS (C:), DVD RW Drive (E:), Lenovo_Recovery (Q:), res498 (\\cabinet.usask.ca\work$), Cabinet (T:), VIDOprojects (U:), Network, bmc_template, ClusterTutorial, EXTRA, FigTree v1.4.3, FtsL_Yinan, Funny_Bla, Good pictures

4) To import the Gen2Epi pipeline click on “New ->”. Type the name (“Gen2Epi”) of the machine in the prompted box. Allocate memory using the scroll bar. Click on “Use an existing virtual hard disk file” under the Hard Disk tab and select “Gen2Epi.vdi”. Click on the “create” button. Once everything is done start the imported VirtualBox image by clicking on the “start” button.
5) Username and password to access the virtual image are:

    Username = gen2epi

    Password = Gen2Epi

Superuser has the same password (Gen2Epi).

In order to turn off the automatic logoff option off please follow:

Applications -> System Tools -> Settings -> Privacy -> Screen Lock (Change to OFF) and show notifications (Change to OFF)

6) Memory parameters for the image are:

    VM image size = 29 GB

    VM memory = 2 GB

To analyze a large dataset both increase the memory (Please make sure the default memory of the VM is set to 2GB at least) and storage size of the VirtualBox VM OR make a shared folder in your Windows or Linux machine.

a. To share a folder between host OS and the VM, click on Devices->Shared folders->Shared folders settings.

b. To increase the size of the VM use procedure described in the post at http://www.ubergizmo.com/how-to/resize-virtualbox-disk/ followed by the additional steps:

   i. Login as root and type "*fdisk /dev/sda*" to start fdisk. To fdisk provide the following input:

      1. Type "p" for the current partition table
      2. Type "d" and delete the partition (2 in this case). This is the partition for which user wants to expand the size
      3. Now type "n" to create a new partition (use default)
      4. Change partition type – type "t" followed by "2", then "8e"

   ii. Reboot the VirtualBox image

   iii. Open the terminal and log in as Superuser

   iv. Type "*lvextend -l +100%FREE /dev/centos/root*"

   v. Type "*xfs_growfs /dev/centos/root*"

   Please note: If users face any problem with the above commands then please follow the description of above-mentioned commands in the post at https://www.ryadel.com/en/resize-extend-disk-partition-unallocated-disk-space-linux-centos-rhel-ubuntu-debian/

c. Test dataset is accessible under "*/home/gen2epi/Desktop/Test_DATA*" within the VM

7) Open terminal (Applications->Favorites->Terminal) and cd into
   "*/home/gen2epi/Desktop/Gen2Epi_Scripts* "

8) All the dependencies are working properly. However, in case of error in accessing these dependencies please set them in current working directory path by running the following command

   a. "*source /home/gen2epi/.bashrc*"

9) The following are quick commands to analyze the included sample data (in *"/home/gen2epi/Desktop/Test_DATA"*). (Note that in some cases the command has wrapped in this document. The command should be entered as a single UNIX shell command.)

   For Quality Check and Trimming:
   "perl WGS_SIBP_P1.pl ../Test_DATA/Input ../Test_DATA/WHO_Data both 3 3 4:15 30"

   For De-novo assembly:
   "perl WGS_SIBP_P2.pl ../Test_DATA/Input Trimming trimmed 2"

Please Note: if you are running this step with Default VM settings (2GB RAM and 2 processor) then it might take longer time (~8hrs for WHO Test_DATA) to finish *de-novo* assembly.

For Chromosome Scaffolding

```
_“Perl WGS_SIBP_P3-Chr-C1.pl /home/gene2epi/Desktop/Test_DATA/Input /home/gen2epi/Desktop/Test_DATA/WHO_Full_Reference_genome/Chromosome /home/gen2epi/Desktop/Gen2Epi_Scripts/Chrom_AssemblyTrimmedReads /home/gen2epi/Desktop/Test_DATA/WHO_Genome_Annotation/Chromosome 1 TXT”
```

Please Note: Chromosome scaffolding step may throw a java-warning message on some computers. Please ignore this message it has no effect on the outcome.

For Plasmid-type identification

```
“Perl WGS_SIBP_P3-Plas_C1.pl  ../Test_DATA/Input Plasmid_AssemblyTrimmedReads 1  ../Test_DATA/Plasmid.fasta”
```

For Epidemiological analysis and AMR prediction of the assembled scaffolds:
NOTE: Please make sure to delete the existing output file before running following commands.

NG-MAST

```
“perl WGS_SIBP_P4_Epi.pl ../Test_DATA/Input Chr_Scaffolds NGMAST”
```

NG-MLST

```
“perl WGS_SIBP_P4_Epi.pl  ../Test_DATA/Input Chr_Scaffolds MLST MLST-Genes.fasta MLST_alleles.fasta pubMLST_profile.txt”
```

Please Note: In case you encounter “BLAST database index” error then make sure to build the blast database for “MLST-Genes.fasta” using the following command:

```
“makeblastdb –in MLST-Genes.fasta –db nucl”
```

NG-STAR

```
“perl WGS_SIBP_P4_Epi.pl  ../Test_DATA/Input Chr_Scaffolds ngstar AMR-Genes-NgStar.fasta AMR-Genes-NgStar-alleles.fasta”
```

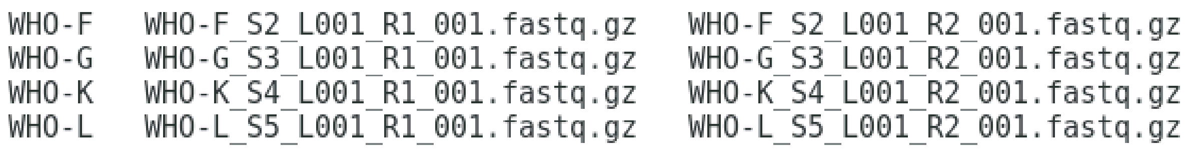
Chromosome-mediated Tetracycline Resistance

```
“perl TetRes.pl rpsJ.fasta Chr_Scaffolds/All_Sequences TetResOut”
```

“perl SeqProt.pl TetResOut”

Detailed description of the commands used for the WGS analysis

1) **Prepare a tab-limited input file describing the full name and the paired-end read files; e.g.**



```
WHO-F   WHO-F_S2_L001_R1_001.fastq.gz   WHO-F_S2_L001_R2_001.fastq.gz
WHO-G   WHO-G_S3_L001_R1_001.fastq.gz   WHO-G_S3_L001_R2_001.fastq.gz
WHO-K   WHO-K_S4_L001_R1_001.fastq.gz   WHO-K_S4_L001_R2_001.fastq.gz
WHO-L   WHO-L_S5_L001_R1_001.fastq.gz   WHO-L_S5_L001_R2_001.fastq.gz
```

First column = Sample ID

Second Column = First fastq read pair

Third Column = Second fastq read pair

Note: Make sure to put all the fastq reads in the same folder.

If you have thousands of samples then the input file in the above-mentioned format can be prepared by using the following script:

“p*erl Prepare_Input.pl <path-to-fastq-files> <number e.g 12>”*

Command line Arguments

<path-to-fastq-files> = Path of the folder/directory that has all the fastq files. Replace <path-to-fastq-files> with the actual path.
<number> = Number of strings that you would like to keep in sample name.

2) **Run the initial quality check on the raw dataset.**

“perl WGS_SIBP_P1 <Input> <path-to-fastq-files> qualitycheck”

Command line Arguments

<Input> = this is the tab-limited file as described in step 1. Replace <Input> with the actual filename.
<path-to-fastq-files> = Path of the folder/directory that has all the fastq files. Replace <path-to-fastq-files> with the actual path.
qualitycheck = Term used to let the program know that user wants to run the initial quality check on the raw readsets.

Output: This step will generate output in two folders

QualityControl/: Quality check results in .zip and .html files for individual samples

MultiQC-Raw/: Quality check results merged into one file for all samples.

3) Trimming, if needed

After checking the initial quality of the raw samples in the previous step, users can opt to go for read trimming by using the following command:

"perl WGS_SIBP_P1.pl <Input> <path-to-fastq-files> trimming <leading length> <trailing length> <sliding window> <minimum length>"

Command line Arguments

<Input> = this is the tab-limited file as described in step 1. Replace <Input> with the actual filename.
<path-to-fastq-files> = Path of the folder/directory that has all fastq files. Replace <path-to-fastq-files> with the actual path.
trimming = Term used to let the program know that the user wants to trim the raw readsets. In addition, users need to provide values for leading length, trailing length, sliding window (m:n) and minimum length, e.g. one can use 3 3 4:15 30.

Output: This step will generate output in the following folders:

Trimming/: fastq paired and unpaired files for each sample.
Trimmed_QC/: Quality check results in .zip and .html form for individual trimmed samples.
MultiQC-Trimmed/: Quality check results of all trimmed samples merged into one file.

Please note: Users can also run Step 2 and 3 by using one command.

"*perl WGS_SIBP_P1.pl <Input> <path-to-fastq-files> both <leading length> <trailing length> <sliding window> <minimum length>"*

4) **De-novo Assembly of Chromosome and Plasmid** : trimmed reads for chromosome and plasmid can be assembled into contigs using :

"perl WGS_SIBP_P2.pl <Input> <path-to-fastq-files> trimmed <processors>"

Command line Arguments

<Input> = this is the tab-limited file as described in step 1. Replace <Input> with the actual filename.
<path-to-fastq-files> = Path of the folder/directory that has all trimmed fastq files generated from step 3. Replace <path-to-fastq-files> with actual path.
trimmed = Term used to let program know that the fastq reads are trimmed.

<processors> = Number of processors. Replace <processors> with the available number of processors; e.g 1, 2…N, etc.

Please Note: - Users can increase the number of processors assigned to the VM image using the Processor tab under System from settings tab in VirtualBox Manager.

Output: This step will generate output in the following folders:

Chrom_AssemblyTrimmedReads/: assembled contigs for chromosomes
Plasmid_AssemblyTrimmedReads/: assembled contigs for plasmid
ChromContigAssemblyTrimmedStat/: Assembly statistics of the assembled contigs (chromosome)
PlasmidContigAssemblytrimmedStat/: Assembly statistics of the assembled contigs (plasmid)

Note: If users want to generate the assembly directly from raw fastq reads then the following command can be used:

"perl WGS_SIBP_P2.pl <Input> <path-to-fastq-files> raw <processors>"

5) **Scaffolding, annotation and quality check**
   a. **Chromosome:**
      i. Case1: When the strain type is known and full reference is available e.g., WHO reference strains.

"perl WGS_SIBP_P3-Chr-C1.pl <Input> <path-reference-genome> <path-assembled-contigs> <path-reference-genome-annotation> <processors> <annotation-format>"

Command line Arguments

<Input> = this is the tab-limited file as described in step 1. Replace <Input> with the actual filename.
<path-reference-genome> = Path of the folder/directory that has full reference genome for each sample. Replace <path-reference-genome> with the actual full path.
<path-assembled-contigs> = Assembled contigs from chromosome reads from step 4. Make sure to write the absolute path. Replace <path-assembled-contigs> with actual full path.
<path-reference-genome-annotation> = Path of the folder/directory that has full reference genome annotation for each sample. Replace <path-reference-genome-annotation> with actual full path

<processors> = Number of processors. Replace <processors> with the available number of processors; e.g 1, 2…N, etc.
<annotation-format> = Annotation format of the full reference genome i.e. .txt or .gff. Replace <annotation-format> with actual term (please make sure to remove the dot, e.g in case of .txt use TXT or txt and in case .gff use GFF or gff.

Output: This step will generate the following output:

Chr Scaffolds folder: This folder contains the full-assembled scaffolds, unplaced contigs (contigs that did not participate in the scaffolding process), annotation and the quality control results.

GenomeStateAll.txt: A text file with N50, GenFra, NA50 and NGA50 values from each sample.

Note: Use of preassembled genome is also possible with following command

Perl WGS_SIBP_P3-Chr-C1.pl <Input> <path-reference-genome> <path-assembled-contigs> <path-reference-genome-annotation> <processors> <annotation-format>

Change the <input> and *<path-assembled-contigs> file options with sample name in tab-limited file and current location of your current assembled contig paths*

ii. Case2: When do not know the strain type or do know the strain type but full reference genome is not available.

"perl WGS_SIBP_P3-Chr-C2.pl <Input> <path-reference-genome> <path-assembled-contigs> <path-reference-genome-annotation> <processors> <annotation-format>"

Command line Arguments

<Input> = this is the tab-limited file as described in step 1. Replace <Input> with the actual filename.
<path-reference-genome> = Path of the folder/directory that has full reference genome for each sample. Replace <path-reference-genome> with actual full path.
<path-assembled-contigs> = Assembled contigs from chromosome reads from step 4. Replace <path-assembled-contigs> with actual full path.
<path-reference-genome-annotation> = Path of the folder/directory that has full reference genome annotation for each sample. Replace <path-reference-genome-annotation> with actual path.

<processors> = Number of processors. Replace <processors> with the available number of processors; e.g 1, 2…N, etc.

Output: This step will generate the following output:

Chr_Scaffolds_folder: This folder contains the full-assembled scaffolds, unplaced contigs (contigs that did not participate in the scaffolding process), annotation and the quality control results.

GenomeStateAll.txt: A text file with N50, GenFra, NA50 and NGA50 values from each sample.

Note: Bug to fix – parsing the quality control results – working fine for the .txt files but for .gff file program is hard coded – at present, it is working fine for NCCP11945_NG.

b. Plasmid:

In order to get the plasmid types from assembled contigs, follow these steps:

A. Download *Neisseria gonorrhoeae* plasmids (Cryptic [NC_001377.1], Conjugative [CP020416.2], Conjugative TEM [NC_014105.1], Asia-type [NC_002098.1], Africa-type [MH140435], pFunnybla [MH140434], Toronto-type [NC_010881.1], Australian [NC_025191.1], and Johannesburg [NC_019211.1] from NCBI nucleotide database. Save the resulting fasta sequences in one file and name it "Plasmid.fasta". As an example, a file named "Plasmid.fasta" is already present in the *"/home/gen2epi/Desktop/Test_DATA"*.

Note: Users can add as many plasmids as they want

B. Make a blast-indexed database of "Plasmid.fasta" file using the following command.

```
"makeblastdb -in Plasmid.fasta -dbtype nucl"
```

C. To identify the type of plasmid present in the WGS read set, users need to run following command.

```
"perl WGS_SIBP_P3-Plas_C1.pl <Input> <path-assembled-contigs>
<processors> <path-plasmiddb>"
```

Command line Arguments

<Input> = this is the tab-limited file as described in step 1. Replace <Input> with the actual filename.

<path-assembled-contigs> = Assembled contigs for plasmid reads from step 4. Replace <path-assembled-contigs> with actual full path.

<processors> = Number of processors. Replace <processors> with the available number of processors; e.g 1, 2…N, etc.

<path-plasmiddb> = Path to local plasmid database generated in plasmid step B.

Output: This step will generate the following output:

Plasmid_Scaffolds_C1: This folder contains the blastn results.

To Dos: - annotation of the best contig

Note: This step will only give information about the plasmid type. However, if users want to generate the full scaffolds then they need to follow the next step.

D. Plasmid Scaffolding

"perl WGS_SIBP_P3-PlasScaf.pl <Plasmid_Scaffolds_C1> <path-plasmiddb> <path-assembled-contigs> <processors>"

Command line Arguments

<Plasmid_Scaffolds_C1> = The output directory with blastn search results in it.
<path-plasmiddb> = Path to local plasmid database generated in plasmid step 1.
<path-assembled-contigs> = Assembled contigs for plasmid reads from step 4.
<processors> = Number of processors

Output: This step will generate following output in Plasmid_Scaffolds_C1 folder

bestHits.txt: best plasmid hits for each plasmid.
Scaffolds: for samples in individual's folder.

Note: Bug to fix - This step generates "WARNING: "contigs" synteny blocks coverage" and "ERROR: Permutations file is empty" message that needs to be fixed.

6) Epidemiological analysis

"perl WGS_SIBP_P4_Epi.pl <Input> <Chr_Scaffolds> <type>"

Command line Arguments

<Input> = this is the tab-limited file as described in step 1.
<Chr_Scaffolds folder>= This folder contains the full-assembled scaffolds.
<Type>= NGMAST, NGSTAR, and NGMLST.

Example

"perl WGS_SIBP_P4_Epi.pl <Input> <Chr_Scaffolds> NGMAST"

"perl WGS_SIBP_P4_Epi.pl <Input> <Chr_Scaffolds> MLST <MLST-Genes.fasta> <MLST_alleles.fasta> <pubMLST_profile.txt>"

"perl WGS_SIBP_P4_Epi.pl <Input> <Chr_Scaffolds> ngstar <AMR-Genes-NgStar.fasta> <AMR-Genes-NgStar-alleles.fasta>"

OUTPUT:

NgMAST.txt
NgMLST.txt
NgStarSearchResults-WithST.txt
NgStarSearchResults-WithoutST.txt

7) Optional:

Read mapping:

"perl ReadMapping.pl <input> <reference-genome> <path-to-fastq-files> <Output-dir>"

Command line Argument

<Input> = this is the tab-limited file as described in step 1.
<Reference-genome> = path to the reference genome (bowtie index files should be prepared for the fasta file).
<path-to-fastq-files> = Path of the folder/directory that has all raw/trimmed fastq files
<Output-dir> = Output directory where all results will be saved.

NOTE: To build the bowtie index please run "bowtie2-build –f reference-genome reference-genome"

Read Binning/ Contamination check

“perl ReadBinning.pl <kraken-db> <path-to-fastq-files> <Output-dir>”

<u>Command line Arguments</u>

< *kraken-db* > = Kraken database path
<path-to-fastq-files> = Path of the folder/directory that has all raw/trimmed fastq files
<Output-dir> = Output directory where all results will be saved.

Tetracycline Resistance:

“*perl TetRes.pl <rpsJ.fasta> <Chr_Scaffolds/All_Sequences>”*
“*perl SeqProt.pl <TetResOut>*”

<u>Command line Argument</u>

<rpsJ.fasta>: rpsJ sequence in fasta format

<Chr_Scaffolds folder>: This folder contains the full-assembled scaffolds

<TetResOut>: Output directory where all results will be saved.

<u>OUTPUT</u>:

<u>Nucl_rpsJ.fasta</u>: Nucleotide sequences of all rpsJ.

<u>Prot_rpsJ.fasta</u>: Protein sequences of all rpsJ.

<u>Please Note</u>: Users can check the mutation by aligning the sequences in using multiple sequence aligner.