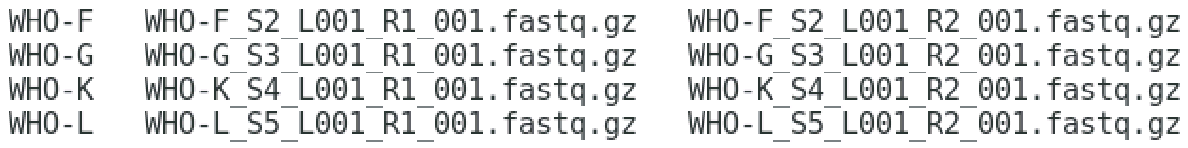
A detailed description of the commands implemented in Gen2Epi version 0.1 pipeline used for the WGS analysis

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. Also, 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 the 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 the 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 fully-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 the preassembled genome is also possible with the 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 fully-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.gff file program is hardcoded – at present, it is working fine for NCCP11945_NG.

b. Plasmid:

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 the "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 the 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 the following output in the Plasmid_Scaffolds_C1 folder

bestHits.txt: best plasmid hits for each plasmid.
Scaffolds: for samples in the 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 fully-assembled scaffolds.
<Type>= NGMAST, NGSTAR, and NGMLST.

<u>Example</u>

"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>"

<u>OUTPUT</u>:

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>"

<u>Command-line Argument</u>

<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.

<u>NOTE</u>: 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>”

Command line Arguments

< *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>*”

Command-line Argument

<rpsJ.fasta>: rpsJ sequence in fasta format

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

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

OUTPUT:

Nucl_rpsJ.fasta: Nucleotide sequences of all rpsJ.

Prot_rpsJ.fasta: Protein sequences of all rpsJ.

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