Genome Analysis on the Command Line

In this document, we have provided instructions for an initial analysis, including building a phylogenetic tree, of bacterial pathogen sequences. All of these analyses, and many others, require you to first generate and assemble consensus genomes from raw sequencing data.

Important notes for following this tutorial:

- Text with a gray background in `monospace font` represents commands to type in. Generally commands are one line, however, in this document, commands might wrap to the next line visually. We will add a blank line between commands to indicate when multiple commands are present.
- Bold text surrounded by **< >** is something you will have to replace with your own folder, path, or sample name.
- This tutorial assumes you have set up the bioinformatics pipeline directory on your computer, and have assembled consensus sequences in FASTA format. If this is **not** the case and you have unassembled raw sequencing data, follow the Command Line Setup and Genome Assembly on the Command Line instructions before proceeding.
- Phylogenetic analysis requires a background set of genomes to compare your sequences to. Background genomes can help you identify which lineages your sequences belong to, and to determine if newly generated sequences are similar to previously published ones. Careful consideration should be given to including genomes which have an appropriate temporal, geographical, and genetic diversity.

This tutorial will take you through an initial analysis of pathogen genome sequences. The process is comprised of two main steps:

1. Type newly generated sequences.
2. Construct a phylogenetic tree of newly generated and publicly available pathogen sequences.

This tutorial relies on the BACPAGE bioinformatics pipeline available at https://github.com/CholGen/bacpage. Instructions for setting up this pipeline and a project directory can be found in Command Line Setup and Genome Assembly on the Command Line. All commands in this tutorial should be run from the bioinformatics pipeline directory (typically at `~/bacpage`), and reference the configuration YAML file edited in the Genome Assembly on the Command Line (generally located at <**`project-path`**>`/config.yaml`).

STEP 0: Project Directory Setup

This analysis assumes you have completed the Genome Assembly on the Command Line and have created a project directory inside the pipeline directory. Additionally, you should have generated consensus sequences from raw sequencing reads for each of your samples.

1. Navigate to your project directory:

```
cd ~/bacpage/<project-path>
```

2. Confirm there are demultiplexed FASTQ files for each sample (i.e., two FASTQ files per sample) in your project directory by viewing the contents of the `input/` directory of your project directory.

```
ls input
```

3. Confirm there are consensus sequences for each sample (i.e, one FASTA file per sample) in your project directory by viewing the contents of the `results/consensus_sequences/` directory:

```
ls results/consensus_sequences/
```

STEP 1: Bacterial Typing

A first step for many analysis protocols is to broadly characterize isolates using their molecular information. We will do this using Multi-locus Sequence Typing (MLST), which detects allelic variants of seven ubiquitous housekeeping genes, and serotyping, which detects the presence or absence of six virulence factor genes.

Note: By default, the BACPAGE pipeline is set up to analyze *Vibrio cholerae* genomes. If you are studying another pathogen, you can change the "*mlst_profiling/scheme*" parameters in `<project-path>/config.yaml` to any typing scheme published by PubMLST (run `mlst -list` to view available typing schemes).

Additionally, you can provide your own directory of virulence factor genes in FASTA format rather than the *Vibrio cholerae*-specific genes provided by the pipeline. To do this, change the value of "*reference_genes*" in `<project-path>/config.yaml` to the directory containing FASTA files for these genes.

1. Navigate to the bioinformatics pipeline directory:

```
cd ~/bacpage
```

2. To perform MLST, run the following command:

```
snakemake --configfile <project-path>/config.yaml --cores
<number-of-processors> --keep-going --until mlst_profiling
```

This command will generate a single file <**project-path**>`/results/reports/mlst_types.csv`. The file is a CSV containing a row for each sample, describing which allelic variants it has for each of the housekeeping genes (*adk, gyrB, mdh, metE, pntA, purM, pyrC*). Using these alleles, MLST assigns a sequence type for each sample.

3. Isolates can also be classified by the presence or absence of five virulence factor genes. Run the following command to detect these genes for each of your samples:

```
snakemake --configfile <project-path>/config.yaml --cores
<number-of-processors> --keep-going --until
virulence_factor_profiling
```

This command will generate a single file <**project-path**>`/results/reports/typing_information.csv`, which will look something like the example below for three cholera genomes. Each row reports the fraction of the gene (indicated by the column) that is covered by the reads of an individual sample.

Sample	**ctxA**	**tcpA_classical**	**tcpA_eltor**	**toxR**	**wbeO1**	**wbfO139**
sample1	1.00	0.32	1.00	0.98	1.00	0.22
sample2	1.00	0.31	1.00	0.98	1.00	0.22
sample3	1.00	0.33	1.00	0.98	1.00	0.22

Genes found in a sample will have a fraction covered of near 1.00, indicating that reads map to the entire gene. Genes not found in this sample will have a lower fraction covered. In the example above, we can say that the first sample contains the *ctxA*, *wbeO1*, and *tcpA El Tor* genes, but not the *wbfO139*, or *tcpA classical* genes. We can therefore say that this is O1 *Vibrio cholerae*, and likely belongs to the El Tor biotype.

STEP 2: Multiple Sequence Alignment

We will now generate a phylogeny including your newly generated genomes. While it can sometimes be useful to make a phylogenetic tree using only the newly generated sequences, it is generally more useful to combine newly generated sequences with a set of previously published sequences, called a "background dataset."

Phylogenetic analysis requires a multiple sequence alignment as input. Because all of your newly generated sequences and the background dataset were both assembled by aligning against a common reference, we can easily generate an alignment by concatenating their individual files. If all of your sequences (including any previously published sequences you are including as part of your background dataset) were not assembled by aligning to the *same* reference genome, you will need to use a computationally intensive alignment software to align sequences against one another.

Note: We recommend that only sequences that cover at least 90% of the genome be included in the phylogenetic inference. By default, the pipeline will only include sequences in the alignment if they reach this coverage threshold. If you want a different stringency, change the "*tree_building/required_coverage*" value in <**project-path**>/`config.yaml` to the desired value. Additionally, you might consider using typing results to inform what sequences to include (e.g., only including sequences classified as a certain serotype).

1. Locate the background dataset you want to compare to your newly generated sequences. This dataset should be a single FASTA file containing separate entries for each sequence in your background dataset, also called a "multi-FASTA". We recommend placing the multi-FASTA file in the `resources/` directory of the bioinformatics pipeline (typically `~/bacpage/resources`). You can do this using the command line or by simply moving the file using the file browser on your computer.

2. Once you have placed your background dataset FASTA into the resources directory, determine the absolute path of the background dataset. If it was placed in the recommend location, you can find the absolute path by navigating to the resources directory and checking the output of `pwd`:

```
cd ~/bacpage/resources

pwd
```

   The absolute path of the background dataset will be the output of `pwd` plus the file name of the background dataset (ending with .fasta).

3. Navigate back to the bioinformatics pipeline directory:

```
cd ~/bacpage
```

4. Add the absolute path of your background dataset to <**project-path**>/`config.yaml`. Open the configuration file in a text editor, change the value of <**background-dataset-path**> on line 8 (see below), change the value of "*generate/phylogeny*" on line 15 to "True", and save the file.

```
background_dataset: "<background-dataset-path>"
```

5. Generate a multiple sequence alignment by running the following command:

```
snakemake --configfile <project-path>/config.yaml --cores
<number-of-processors> --keep-going --until concatenate_sequences
```

   This step simply concatenates your newly generated genomes, and the background dataset if its present, into a multi-FASTA located at:
   <**project-path**>/`intermediates/illumina/phylogeny/complete_alignment.fasta`

STEP 3: Build a Phylogenetic Tree

Having generated the multiple sequence alignment, we can proceed with inferring a phylogenetic tree.

1. Run the phylogenetic inference step of the pipeline by running the following command:

   ```
   snakemake --configfile <project-path>/config.yaml --cores
   <number-of-processors> --keep-going --until generate_rooted_tree
   ```

   Briefly, this step generates a phylogeny using *iqtree*. By default, this step will use the GTR substitution model and calculate branch support by conducting 1000 bootstraps. These options can be changed by modifying the value of "*tree_building/iqtree_parameters*" in `<project-path>/config.yaml`. This process is computationally intensive and may take a few hours to complete depending on the size and number of sequences being analyzed as well as the speed of your computer.

   The output of this command is a phylogeny in Newick format, `<project-path>/results/<project-directory-name>.ml.tree`.

2. While the tree file is a text file that can be opened and read in a text editor, it is generally not interpretable in this format. We will use a GUI tree viewer called FigTree to view tree files. From the applications directory on your computer, open FigTree. Click File -> Open, and select the newly generated phylogeny in the file browser that opens up. Alternatively, you can open the `<project-directory-name>.ml.tree` file directly from the file browser.

3. The phylogeny should now appear in the main FigTree window. Search through the tree for your samples, determine which samples they are closest to.