

I. Obtain viral genomes

- A. Obtain the H1N1 reference genome from NCBI: [LINK](#)
- B. Obtain genomes by segment from NCBI for multiple sequence alignment (download as fasta files):

- 1. H1N1 genome assembly ViralMultiSegProj274766 by segment:  
[https://www.ncbi.nlm.nih.gov/datasets/genome/GCF\\_001343785.1/](https://www.ncbi.nlm.nih.gov/datasets/genome/GCF_001343785.1/)  
RefSeq ID: GCF\_001343785.1
  - a) Segment 1 RefSeq ID: NC\_026438.1
  - b) Segment 2 RefSeq ID: NC\_026435.1
  - c) Segment 3 RefSeq ID: NC\_026437.1
  - d) Segment 4 RefSeq ID: NC\_026433.1
  - e) Segment 5 RefSeq ID: NC\_026436.1
  - f) Segment 6 RefSeq ID: NC\_026434.1
  - g) Segment 7 RefSeq ID: NC\_026431.1
  - h) Segment 8 RefSeq ID: NC\_026432.1
- 2. H5N1 genome assembly ASM3868529v1 by segment:  
[https://www.ncbi.nlm.nih.gov/datasets/genome/GCA\\_038685295.1/](https://www.ncbi.nlm.nih.gov/datasets/genome/GCA_038685295.1/)  
GenBank ID: GCA\_038685295.1
  - a) Segment 1 GenBank ID: KJ907628.1
  - b) Segment 2 GenBank ID: KJ907629.1
  - c) Segment 3 GenBank ID: KJ907630.1
  - d) Segment 4 GenBank ID: KJ907631.1
  - e) Segment 5 GenBank ID: KJ907632.1
  - f) Segment 6 GenBank ID: KJ907633.1
  - g) Segment 7 GenBank ID: KJ907634.1
  - h) Segment 8 GenBank ID: KJ907635.1
- 3. H1N2 genome assembly ASM3875498v1 by segment:  
[https://www.ncbi.nlm.nih.gov/datasets/genome/GCA\\_038754985.1/](https://www.ncbi.nlm.nih.gov/datasets/genome/GCA_038754985.1/)  
GenBank ID: GCA\_038754985.1
  - a) Segment 1 GenBank ID: KT225475.1
  - b) Segment 2 GenBank ID: KT225474.1
  - c) Segment 3 GenBank ID: KT225473.1
  - d) Segment 4 GenBank ID: KT225468.1
  - e) Segment 5 GenBank ID: KT225471.1
  - f) Segment 6 GenBank ID: KT225470.1
  - g) Segment 7 GenBank ID: KT225469.1
  - h) Segment 8 GenBank ID: KT225472.1

C. Generate multiple sequence alignments as bam files:

- 1. Install the following packages: **samtools**, **bwa**
- 2. Run the following commands to generate MSA bam files:
  - a. Concatenate swine and duck segment files together  
`cat h1n2_segment1.fa h5n1_segment1.fa > non_human.fa`
  - b. Index files using bwa and samtools  
`bwa index non_human.fasta`

```
bwa index h1n1_segment1.fa  
samtools faidx non_human.fa  
samtools faidx h1n1_segment1.fa
```

- c. Align merged fasta to “reference” (H1N1 human version of segment) to create a sam file

```
bwa mem h1n1_segment1.fa non_human.fasta >  
merged_seg#.sam
```

- I. Launch an AWS EC2 Instance
  - A. Log in to your AWS account and launch an EC2 instance:
    1. Name: Choose a name, such as bioe131\_final\_project.
    2. Application and OS Images (AMI): Select Ubuntu 22.04 as the operating system.
    3. Key Pair: Create a new key pair (e.g., bioe131) and save it securely.
    4. Network Settings:
      - a) Allow both HTTP and HTTPS traffic from the internet.
    5. Storage Configuration: Set storage to 30 GiB gp3.
  - B. Once the instance is launched, click **Connect** to get connection instructions.
- II. Set Up JBrowse2 in Terminal
  - A. Connect to Your Instance via SSH: Follow the instructions from the AWS EC2 console to connect to your instance via SSH.
  - B. Switch to Root User:
    1. **sudo su -**
  - C. Set a Password for the Default User:
    1. **passwd ubuntu**
    2. Enter a new password and write it down somewhere secure.
  - D. Exit Root User:
    1. **exit**
  - E. Install Linuxbrew:
    1. **/bin/bash -c "\$(curl -fsSL <https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh>)"**
    2. Press Return/Enter to continue when prompted.
    3. Enter the password you set earlier when prompted.
  - F. Add Linuxbrew to Your Execution Path:
    1. **echo 'eval "\$(/home/linuxbrew/.linuxbrew/bin/brew shellenv)'" >> ~/.bashrc**  
**eval "\$(/home/linuxbrew/.linuxbrew/bin/brew shellenv)"**
  - G. Install Node.js:
    1. Check if Node.js is installed:
      - a) **node -v**
    2. if not installed:
      - a) **sudo apt update**  
**sudo apt install unzip**  
**curl -fsSL https://fnm.vercel.app/install | bash**  
**source ~/.bashrc**  
**fnm use --install-if-missing 20**
    3. verify installation  
**node -v** # Should print `v20.18.1`  
**npm -v** # Should print `10.8.2`
  - H. Install npm
    1. **sudo apt update**  
**sudo apt install npm -y**
  - I. Install JBrowse CLI:

1. **sudo npm install -g @jbrowse/cli**  
**jbrowse --version**
- J. Install Apache2 and dependencies:
  1. **sudo apt install wget apache2**  
**brew install samtools htlib**
- III. Configure Apache2 and JBrowse2
  - A. Start the Apache Server:
    1. **sudo service apache2 start**
  - B. Install JBrowse2:
    1. Create a temporary working directory:
      - a) **mkdir ~/tmp**  
**cd ~/tmp**
    2. Create a JBrowse2 app:
      - a) **jbrowse create output\_folder**
    3. Move the JBrowse app to the Apache root directory:
      - a) **sudo mv output\_folder /var/www/html/jbrowse2**  
**sudo chown -R \$(whoami) /var/www/html/jbrowse2**
  - C. Verify the Installation:
    1. Open your browser and go to:
      - a) **http://<public\_IP\_of\_your\_instance>/jbrowse2/**
    2. You should see the message "JBrowse 2 is installed."
- IV. Add files to AWS server
  - A. Download fastas
    1. Add them to the same directory as the pem file that you've downloaded
  - B. Combine and index each subtype if needed
    1. **Cat f1.fasta f2.fasta.... > combined.fasta**
  - C. Upload to server
    - 1.
- V. Upload files to JBrowse
  - A. Index the files
    1. **samtools faidx combined.fasta**
  - B. Upload fasta to JBrowse
    1. **sudo jbrowse add-assembly <combined.fasta> --out**  
**\$APACHE\_ROOT/jbrowse2 --load copy**
  - C. Upload annotations
    1. **sudo jbrowse add-track <annotations..gz> assemblyNames**  
**<corresponding\_fasta> --out /var/www/html/jbrowse2 --load copy**
- VI. Launch JBrowse
  - A. **http://<public\_IP\_of\_your\_instance>/jbrowse2/**