Design Document:

RT Covid Lineage Classification with MinION

Last updated: 202105251233

Overview

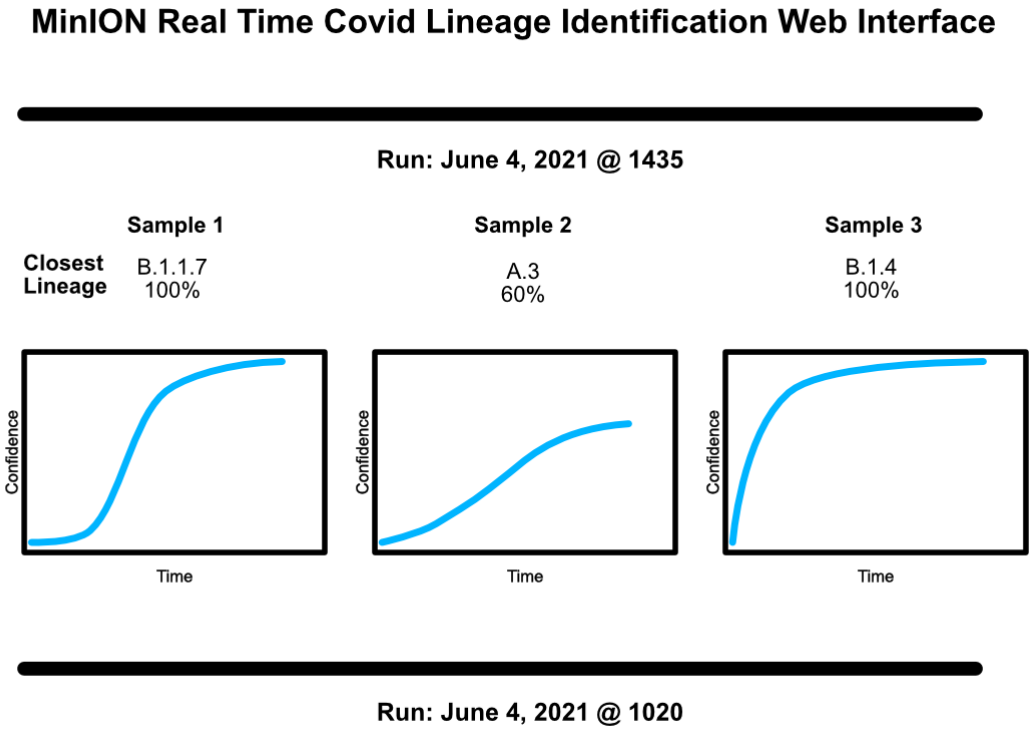
1. Objective
2. Pipeline, High Level
3. Considerations
4. Implementation Details
5. Questions

Objective

Build a platform that performs real time Pango lineage identification with a MinION to help operators determine when to end a run while improving sequencing throughput and flow cell efficiency.

Pipeline, High Level

→ Start MinION run
→ Sequencing files on server updated every 5-10 minutes
  → Assemble genome with Arctic
  → Run through PangoLEARN classifier
→ Report lineage and confidence over time on web interface



Considerations

1. The pipeline presented here is the simplest implementation possible. It merely passes output files from one pre-developed analysis tool to another. Once the framework has been built, custom analysis tools can be explored - such as identifying new spike protein mutations.

2. The output of PangoLEARN is not very informative, but plotting the change in confidence (aka "support") in real time could be useful for identifying diminishing returns with continued sequencing. An example output is provided below:

Taxon	Lineage	support	pangoLEARN_version	status
Virus1	B.1	80	2020-04-27	passed_qc

3. Pango lineage classification can be updated at anytime, so it's important to pull changes from the database at the start of each run. Also, viruses mutate all the time, so having less than 100% confidence may be common.

Implementation Details

MinION

1. New files from starting a MinION run starts a cron job that syncs file changes to the server every 5 minutes.
2. File syncing stops when there are no changed files after 10 minutes

Server:

1. When a new run is started, files are synced to the server. New directories launch a script that:
    - updates the PangoLEARN lineage classifier,
    - allocates space for run details on the web interface,
    - and starts the pipeline.
2. Trigger the Arctic pipeline
    - Basecall with Guppy
    - Demultiplex with Guppy
    - Filter reads with Arctic
    - Run the MinION pipeline with Arctic to generate consensus sequences
    - Aggregate consensus sequences into one file
3. Upon completing consensus aggregation, start Pagolin pipeline
    - Run pangolin
    - Wait for output .csv
4. CSV is parsed, data is appended to database, and web page is updated
5. Script terminates when no file changes are made after 10 minutes

Questions

1. Does the MinION produce large files that take 5+ minutes to complete?

Git is useful for appending new lines to large files over the network that take long to complete. If the MinION produces many small files that take less than 5 minutes to complete, rsync could be better.

2. Are the MinION runs multiplexed or is it one sample per run?

If a single sample is run each time, that could simplify the implementation. The Arctic protocol is for multiplexed runs, so that's what I'm currently expecting.

2. Is there a database that lists the characteristic mutations for each lineage?

Outbreak.info has a great mutation table that explicitly lists out characteristic mutations of each lineage. However, I haven't found a database that lists **individual mutations** that I can scan programmatically. This would be useful for later identifying mutations in the spike protein that don't match the best fitting lineage in future iterations.

Characteristic mutations of B.1.1.7

gene	amino acid
S	D614G
ORF1a	T1001I
S	D1118H
S	A570D
ORF1a	A1708D
ORF1b	P314L
ORF8	Y73C
N	S235F
ORF8	Q27*
S	S982A
S	T716I
ORF8	R52I
S	P681H
N	R203K