# Implementation of Nextstrain – Use in a State Public Health Lab
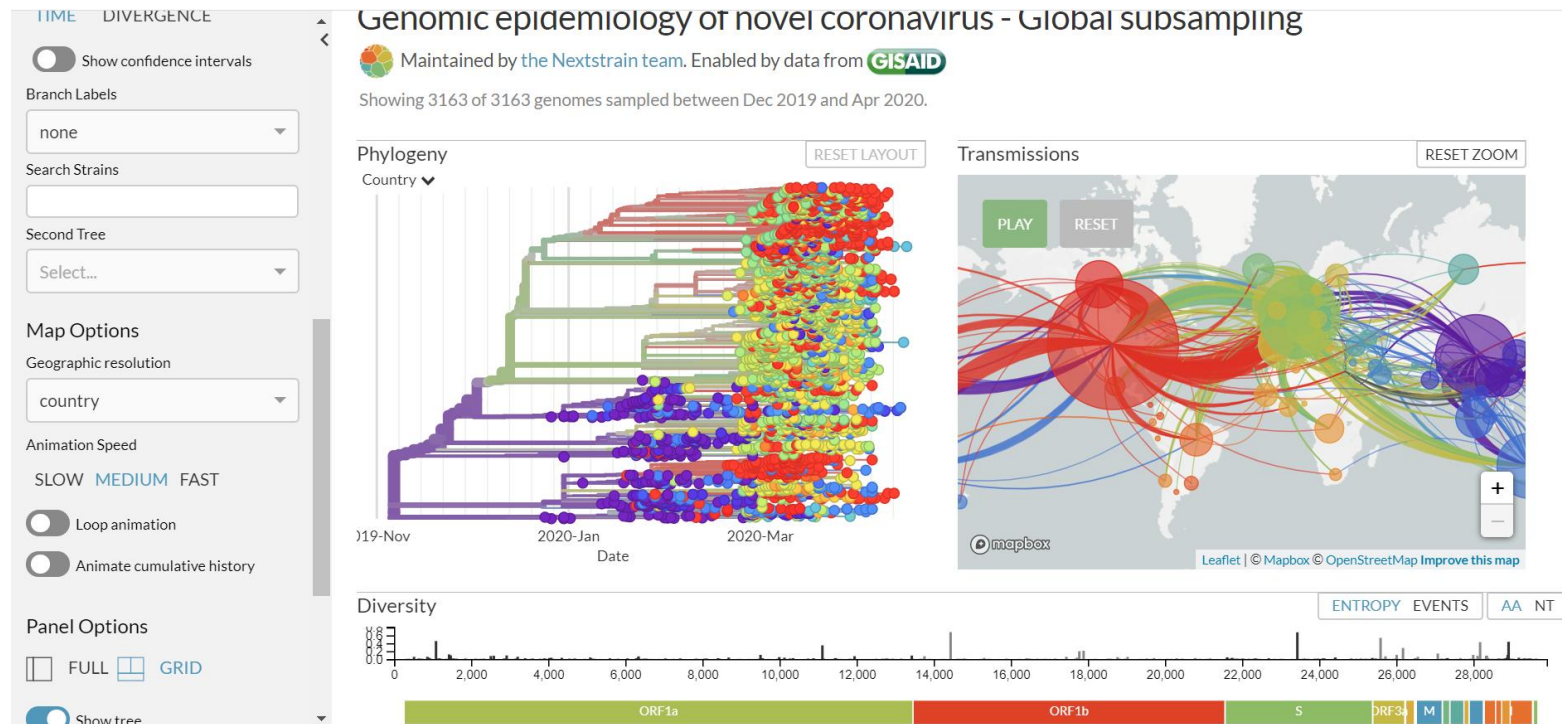
Heather Blankenship, PhD

MDHHS Molecular Microbiologist/Bioinformatics Specialist

MDHHS
Bureau of Laboratories

# What is Nextstrain?

- Real-time tracking and evolution of pathogens
- Interactive visualization platform
- Visualization power to examine geographic, metadata, and microbial variants

# Why Design a Local Build?

- <u>COMMUNICATION!! and GENOMIC EPIDEMIOLOGY</u>

  ▫ What potential entry points and from which countries or states do we share related isolates?
  ▫ Transmission within the state, can we examine at the county and regional level?
  ▫ Do we see a spread from a hot spot to other places within the state?
  ▫ Can we overlay any metadata and get a preliminary idea of association of clusters with demographics or clinical outcomes?
  ▫ How can we visually understand the variants that are present and what genes are we see variants?

# Necessary Dependents

- Python 3

- Pip

- Docker

```
# Python 3
$ python3 --version
$ sudo apt-get install python3.6

# pip3
$ sudo apt install python3-pip

# Docker
https://doc.docker.com/install
https://github.com/StaPH-B/scripts/blob/master/image-
information.md#docker-ce
```

# Installation of NextStrain CLI and Docker

```
# Install Nextstrain CLI
$ pip3 install nextstrain-cli==1.16.2

$ nextstrain version
Nextstrain.cli 1.16.2

$ nextstrain check-setup

# Docker Nextstrain Environment
$ nextstrain update
```
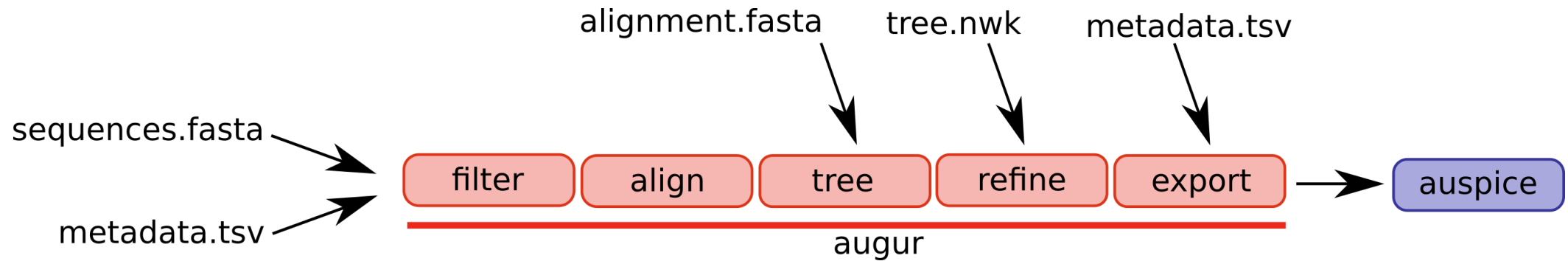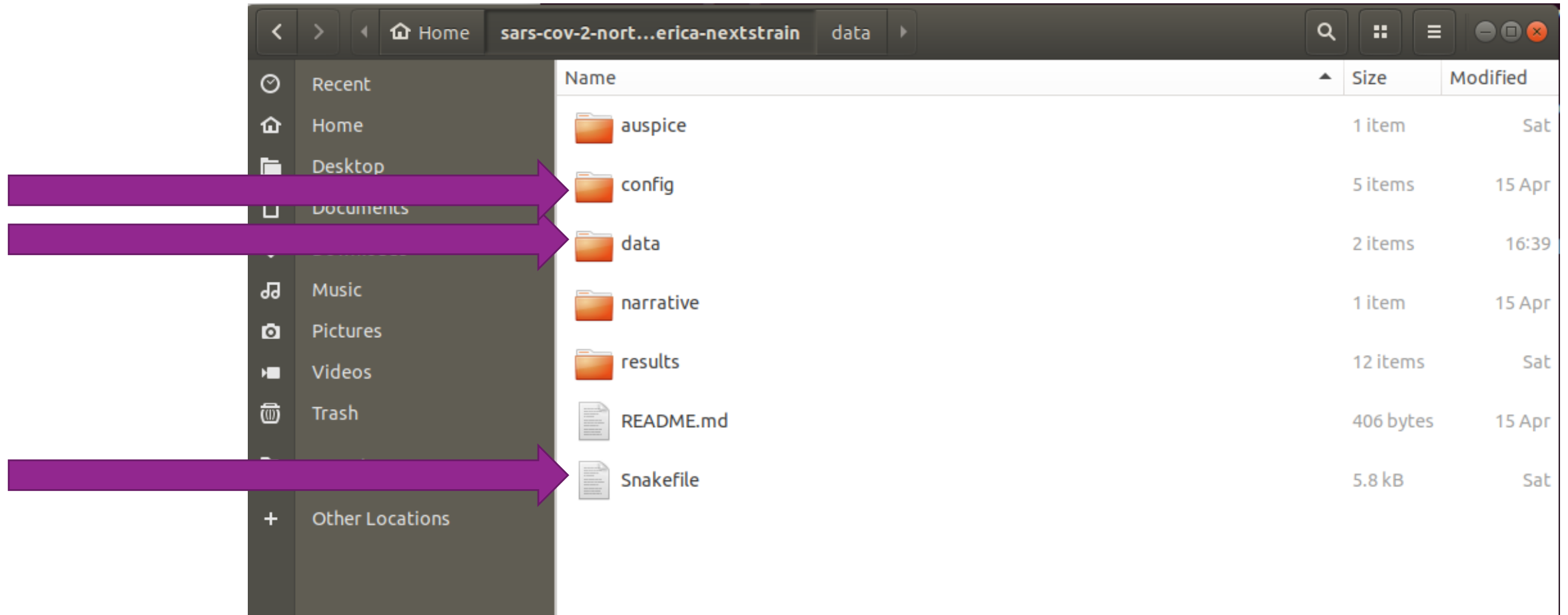
# Nextstrain Pipeline
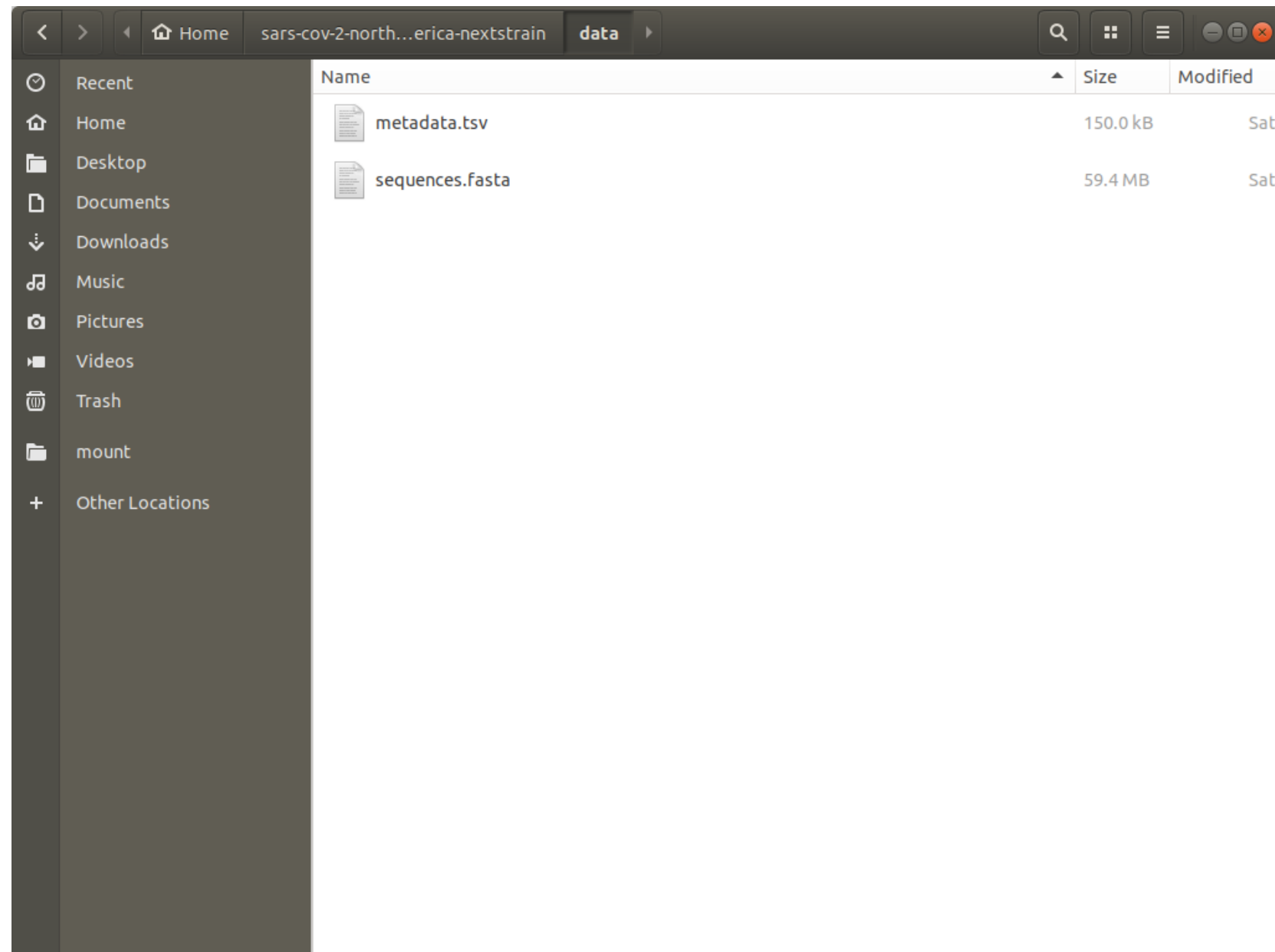


- Two main files
  - sequences.fasta
  - metadata.tsv

# General Overview of Files

# Data Folder

# Metadata

- Isolate name must match the same as it is in the sequencing file
- Must have a virus identified
- Include a date of collection in the format (YYYY-MM-DD)
- Include a location

| strain | virus | date | country | state_province | region |
|--------|-------|------|---------|----------------|--------|
| hCoV-19/Canada/BC_5306970/2020 | sars_cov_2 | 2020-03-11 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_5282984/2020 | sars_cov_2 | 2020-03-11 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_5275718/2020 | sars_cov_2 | 2020-03-11 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_4799711/2020 | sars_cov_2 | 2020-03-09 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_6129127/2020 | sars_cov_2 | 2020-03-10 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_6004567/2020 | sars_cov_2 | 2020-03-10 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_5979789/2020 | sars_cov_2 | 2020-03-11 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_5522039/2020 | sars_cov_2 | 2020-03-11 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_4540462/2020 | sars_cov_2 | 2020-03-10 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_4143868/2020 | sars_cov_2 | 2020-03-04 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_4143842/2020 | sars_cov_2 | 2020-03-04 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_3972884/2020 | sars_cov_2 | 2020-03-10 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_3968175/2020 | sars_cov_2 | 2020-03-10 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_3842755/2020 | sars_cov_2 | 2020-03-09 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_4122951/2020 | sars_cov_2 | 2020-03-10 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_4118226/2020 | sars_cov_2 | 2020-03-10 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_4078583/2020 | sars_cov_2 | 2020-03-03 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_3989992/2020 | sars_cov_2 | 2020-03-09 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_3808524/2020 | sars_cov_2 | 2020-03-09 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_1318414/2020 | sars_cov_2 | 2020-03-06 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_0554880/2020 | sars_cov_2 | 2020-03-13 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_0443574/2020 | sars_cov_2 | 2020-03-11 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_8622445/2020 | sars_cov_2 | 2020-03-13 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_8606204/2020 | sars_cov_2 | 2020-03-13 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_8897642/2020 | sars_cov_2 | 2020-03-11 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_8896915/2020 | sars_cov_2 | 2020-03-11 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_8894200/2020 | sars_cov_2 | 2020-03-11 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_8718874/2020 | sars_cov_2 | 2020-03-07 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_9574898/2020 | sars_cov_2 | 2020-03-13 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_9446031/2020 | sars_cov_2 | 2020-03-13 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_9345715/2020 | sars_cov_2 | 2020-03-13 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_9345042/2020 | sars_cov_2 | 2020-03-13 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_8486790/2020 | sars_cov_2 | 2020-03-13 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_6502001/2020 | sars_cov_2 | 2020-03-12 | Canada | British Columbia | British Columbia |
| hCoV-19/Canada/BC_7493713/2020 | sars_cov_2 | 2020-03-12 | Canada | British Columbia | British Columbia |

# Metadata

```
hCoV-19/USA/MI-SC2-0005/2020    sars_cov_2    2020-03-09    USA    Michigan
hCoV-19/USA/MI-SC2-0003/2020    sars_cov_2    2020-03-11    USA    Michigan
hCoV-19/USA/MI-SC2-0007/2020    sars_cov_2    2020-03-12    USA    Michigan
hCoV-19/USA/MI-SC2-0008/2020    sars_cov_2    2020-03-12    USA    Michigan
hCoV-19/USA/MI-SC2-0001/2020    sars_cov_2    2020-03-09    USA    Michigan
hCoV-19/USA/MI-SC2-0002/2020    sars_cov_2    2020-03-10    USA    Michigan
hCoV-19/USA/MI-SC2-0006/2020    sars_cov_2    2020-03-11    USA    Michigan
hCoV-19/USA/MI-SC2-0004/2020    sars_cov_2    2020-03-10    USA    Michigan
```
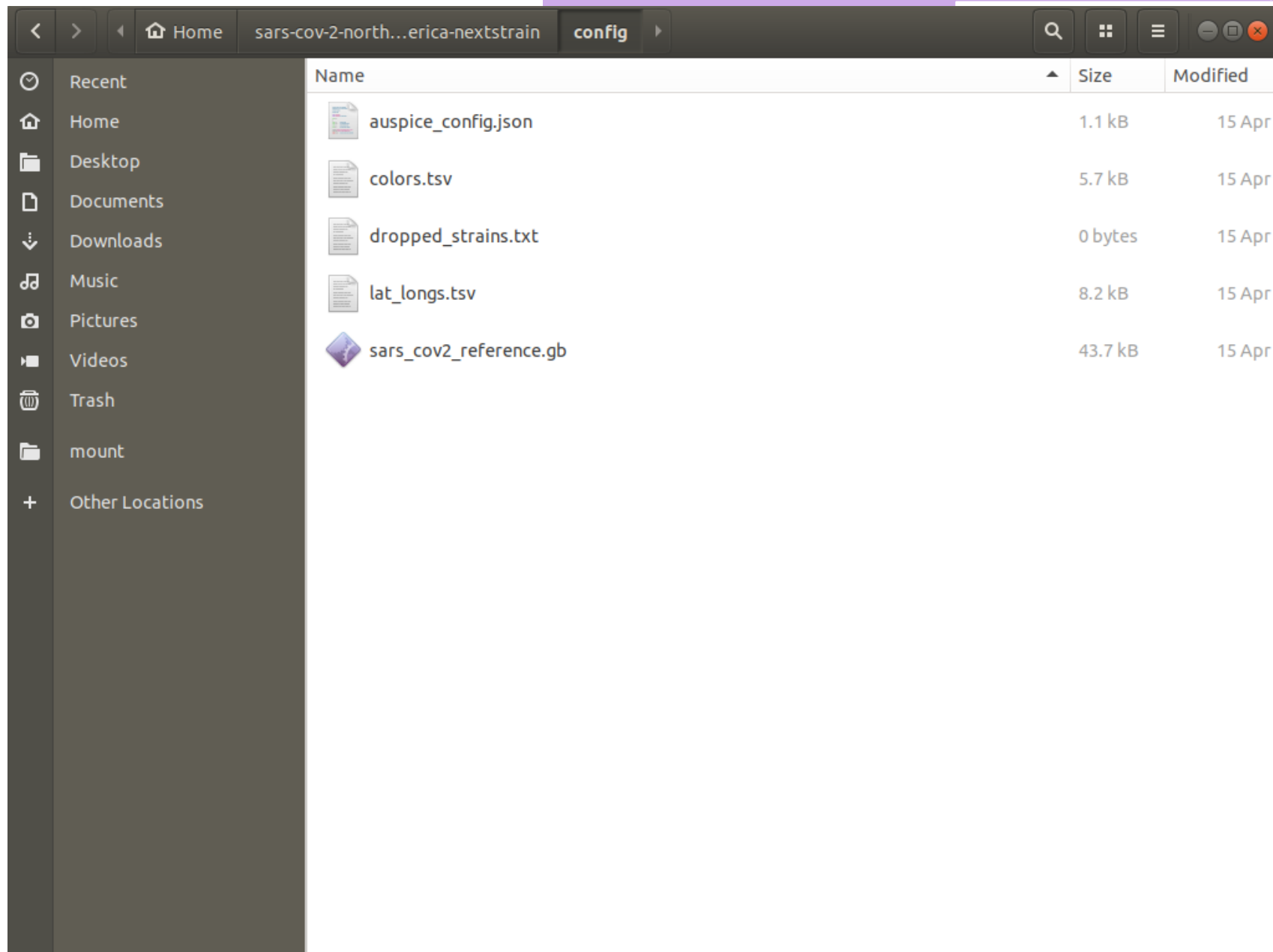
**COUNTY/ ZIP CODE**

# Sequencing Data

- For SARS-CoV-2 this is a concatenated fasta file

- Nextstrain can start with VCF files as the input data

- Ensure that your sequencing name here matches that with the metadata file

- Metadata information can be included in the sequencing header

# Config File

# Colors.tsv

- The variables that you are assigning colors are the ones that were identified in the metadata
- All colors are in HEX Color Code
- There are color schemes that range up to 500 colors in a scheme on: https://github.com/nextstrain/ncov/blob/master/config/color_schemes.tsv

# Lat and Long File (lat_longs.tsv)

- Identify which metadata variable the location is found

- Assign the latitude and longitude to each position that you want geographic resolution present

```
Open ▾          

region   Leelanau         45.146182         -86.051574
region   Lenawee 41.895915          -84.066853
region   Livingston       42.602532         -83.911718
region   Luce     46.940602         -85.582368
region   Mackinac         46.167981         -85.303756
region   Macomb  42.671467          -82.910869
region   Manistee         44.350385         -86.602967
region   Marquette        46.656597         -87.584028
region   Mason    43.996636         -86.750814
region   Mecosta 43.635295          -85.332751
region   Menominee        45.544174         -87.509892
region   Midland 43.648378          -84.37922
region   Missaukee        44.325424         -85.085471
region   Monroe  41.916097          -83.487106
region   Montcalm         43.312782         -85.149468
region   Montmorency      45.024134         -84.130107
region   Muskegon         43.289258         -86.751892
region   Newaygo 43.562709          -85.791423
region   Oakland 42.660452          -83.38421
region   Oceana  43.647255          -86.807575
region   Ogemaw  44.33328           -84.128073
region   Ontonagon        47.216604         -89.500461
region   Osceola 43.997552          -85.322283
region   Oscoda   44.685121         -84.124894
region   Otsego   45.021794         -84.576597
region   Ottawa  42.942346          -86.655342
region   Presque Isle     45.489515         -83.384019
region   Roscommon        44.339517         -84.611272
region   Saginaw 43.328267          -84.05541
region   St Clair         42.928804         -82.668914
region   St Joseph        41.911488         -85.52287
region   Sanilac 43.449155          -82.642815
region   Schoolcraft      46.020758         -86.199352
region   Shiawassee       42.951545         -84.146352
region   Tuscola 43.487902          -83.436618
region   Van Buren        42.283986         -86.305697
region   Washtenaw        42.252327         -83.844634
region   Wayne    42.284664         -83.261953
region   Wexford 44.331375          -85.570046
region   Cruise   35.829309         -58.410552
```

# Reference.gb

- GenBank file for reference strain of choice

- All Michigan local builds are using reference: MN908947

Open ⏷

```
LOCUS       MN908947                    29903 bp ss-RNA     linear   VRL 17-JAN-2020
DEFINITION  MN908947
ACCESSION   MN908947
VERSION     MN908947
KEYWORDS    .
SOURCE      Wuhan seafood market pneumonia virus
  ORGANISM  Wuhan seafood market pneumonia virus
            Viruses; Riboviria; Nidovirales; Cornidovirineae; Coronaviridae;
            Orthocoronavirinae; Betacoronavirus; unclassified Betacoronavirus.
REFERENCE   1  (bases 1 to 29903)
  AUTHORS   Wu,F., Zhao,S., Yu,B., Chen,Y.-M., Wang,W., Hu,Y., Song,Z.-G.,
            Tao,Z.-W., Tian,J.-H., Pei,Y.-Y., Yuan,M.L., Zhang,Y.-L.,
            Dai,F.-H., Liu,Y., Wang,Q.-M., Zheng,J.-J., Xu,L., Holmes,E.C. and
            Zhang,Y.-Z.
  TITLE     A novel coronavirus associated with a respiratory disease in Wuhan
            of Hubei province, China
  JOURNAL   Unpublished
REFERENCE   2  (bases 1 to 29903)
  AUTHORS   Wu,F., Zhao,S., Yu,B., Chen,Y.-M., Wang,W., Hu,Y., Song,Z.-G.,
            Tao,Z.-W., Tian,J.-H., Pei,Y.-Y., Yuan,M.L., Zhang,Y.-L.,
            Dai,F.-H., Liu,Y., Wang,Q.-M., Zheng,J.-J., Xu,L., Holmes,E.C. and
            Zhang,Y.-Z.
  TITLE     Direct Submission
  JOURNAL   Submitted (05-JAN-2020) Shanghai Public Health Clinical Center &
            School of Public Health, Fudan University, Shanghai, China
COMMENT     On Jan 17, 2020 this sequence version replaced MN908947.2.

            ##Assembly-Data-START##
            Assembly Method        :: Megahit v. V1.1.3
            Sequencing Technology :: Illumina
            ##Assembly-Data-END##
FEATURES             Location/Qualifiers
     source          1..29903
                     /organism="Wuhan seafood market pneumonia virus"
                     /mol_type="genomic RNA"
                     /isolate="Wuhan-Hu-1"
                     /host="Homo sapiens"
                     /db_xref="taxon:2697049"
                     /country="China"
                     /collection_date="Dec-2019"
```

# Auspice_config.json

- This file will help with how to configure your auspice visualization

- Identify which coloring choices you want to include, geographical resolutions, layout of build, and filter strains highlighted

```
"title": "Michigan Nextstrain Build for SARS-CoV-2",
"maintainers": [
  {"name": "Heather Blankenship"}
],
"colorings": [
  {
    "key": "gt",
    "title": "Genotype",
    "type": "categorical"
  },
  {
    "key": "date",
    "title": "Date of Collection",
    "type": "continuous"
  },
  {
    "key": "county",
    "title": "County",
    "type": "categorical"
  },
  {
    "key": "region",
    "title": "Epidemiology Region",
    "type": "categorical"
  },
  {
    "key": "state_province",
    "title": "US state and Canadian provinces",
    "type": "categorical"
  },
  {
    "key": "country",
    "title": "North America country",
    "type": "categorical"
  }
],
"geo_resolutions": [
  "state_province",
  "region"
],
"panels": [
  "tree",
  "map",
  "entropy"
],
"display_defaults": {
  "map_triplicate": true
},
"filters": [
  "state_province",
  "country",
  "region",
  "date"
]
}
```

# Snakemake file

Filter – filter out any data and subsample the data based on the grouping and number per group

```
rule all:
    input:
        auspice_json = "auspice/sars_cov_2.json",

input_fasta = "data/sequences.fasta",
input_metadata = "data/metadata.tsv",
dropped_strains = "config/dropped_strains.txt",
reference = "config/sars_cov2_reference.gb",
colors = "config/colors.tsv",
lat_longs = "config/lat_longs.tsv",
auspice_config = "config/auspice_config.json"

rule filter:
    message:
        """
        Filtering to
          - {params.sequences_per_group} sequence(s) per {params.group_by!s}
          - from {params.min_date} onwards
          - excluding strains in {input.exclude}
        """
    input:
        sequences = input_fasta,
        metadata = input_metadata,
        exclude = dropped_strains
    output:
        sequences = "results/filtered.fasta"
    params:
        group_by = "state_province",
        sequences_per_group = 250,
        min_date = 2019
    shell:
        """
        augur filter \
            --sequences {input.sequences} \
            --metadata {input.metadata} \
            --exclude {input.exclude} \
            --output {output.sequences} \
            --group-by {params.group_by} \
            --sequences-per-group {params.sequences_per_group} \
            --min-date {params.min_date}
        """
```

# Snakemake file

Align – multi-sequence analysis with mafft and fill in all gaps with N

Tree – phylogenetic analysis and tree generation with IQ-Tree, this can be changed to RAxML and FastTree as alternatives

```
rule align:
    message:
        """
        Aligning sequences to {input.reference}
          - filling gaps with N
        """
    input:
        sequences = rules.filter.output.sequences,
        reference = reference
    output:
        alignment = "results/aligned.fasta"
    shell:
        """
        augur align \
            --sequences {input.sequences} \
            --reference-sequence {input.reference} \
            --output {output.alignment} \
            --fill-gaps \
            --nthreads 30
        """

rule tree:
    message: "Building tree"
    input:
        alignment = rules.align.output.alignment
    output:
        tree = "results/tree_raw.nwk"
    shell:
        """
        augur tree \
            --alignment {input.alignment} \
            --output {output.tree}
        """
```

# Snakemake file

Refine – infer a time tree and adjustment of branch lengths and assigns confidence values to the tree using Treetime

```
rule refine:
    message:
        """
        Refining tree
          - estimate timetree
          - use {params.coalescent} coalescent timescale
          - estimate {params.date_inference} node dates
          - filter tips more than {params.clock_filter_iqd} IQDs from clock expectation
        """
    input:
        tree = rules.tree.output.tree,
        alignment = rules.align.output,
        metadata = input_metadata
    output:
        tree = "results/tree.nwk",
        node_data = "results/branch_lengths.json"
    params:
        coalescent = "opt",
        date_inference = "marginal",
        clock_filter_iqd = 4
    shell:
        """
        augur refine \
            --tree {input.tree} \
            --alignment {input.alignment} \
            --metadata {input.metadata} \
            --output-tree {output.tree} \
            --output-node-data {output.node_data} \
            --timetree \
            --coalescent {params.coalescent} \
            --date-confidence \
            --date-inference {params.date_inference} \
            --clock-filter-iqd {params.clock_filter_iqd}
        """
```

# Snakemake file

Traits – Infer ancestral traits

Ancestral – Infer ancestral
sequences at each node

Translate – used to identify amino
acid mutations

Export – export all of the data
that is needed to visualize the
build into the FILE.json file

```
rule traits:
    message: "Inferring ancestral traits for {params.columns!s}"
    input:
        tree = rules.refine.output.tree,
        metadata = input_metadata
    output:
        node_data = "results/traits.json",
    params:
        columns = "state_province region country"
    shell:
        """
        augur traits \
            --tree {input.tree} \
            --metadata {input.metadata} \
            --output-node-data {output.node_data} \
            --columns {params.columns} \
            --confidence
        """
```

# Running the Docker Image

# Auspice                          # Results

# Turn it into a local build you can share!

- Website to visualize json file - https://auspice-us.herokuapp.com/

- Any json files that are created from nextstrain can now be password protected and shared with state epidemiologists and laboratorians
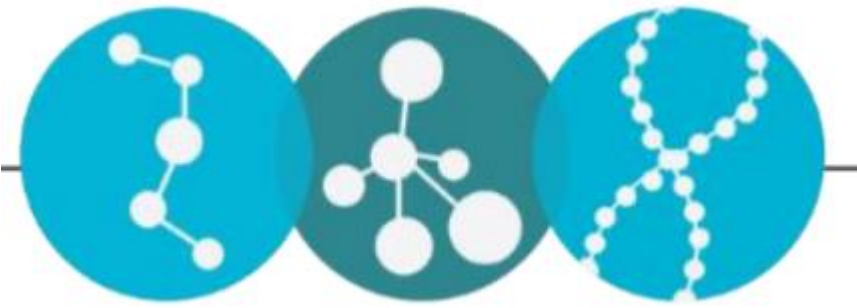
# Additional Considerations

- Additional metadata can be added into a build
  - Demographic information
  - Higher geographic resolution (zip code)
  - Submitter information (which hospital or long-term care facility)
  - Clinical outcomes (hospitalized, death, asymptomatic)

# To-Do

- Automate the updates of sequence data file and the metadate file for Michigan only build
  - A large amount of this work is pulling together the metadata for each isolate

# Discussion