Instituto
Todos
pela
Saúde

Nextstrain tutorial

In this tutorial we are going to generate a nextstrain build. To do so, we will run a bioinformatics pipeline that will prepare the metadata, perform sequence alignment, phylogenetic reconstruction, infer discrete ancestral states (for division, location, host, and mutations), and finally combine these results in a JSON file that can be visualized using the online tool `https://auspice.us/`.

(1) **Visit** the GitHub repository where the template pipeline is located, and **read** the basic instructions:

`https://github.com/InstitutoTodosPelaSaude/flexpipe`

(2) **Create** a folder named `analyses` in any directory of preference: it will be your working directory. Access this newly created folder using the Terminal (tip: `cd analyses`).

(3) Inside the `analyses` folder, **run** the command below to **download** the template pipeline:

```
> git clone https://github.com/InstitutoTodosPelaSaude/flexpipe.git
```

(4) **Check** if a conda environment called “nextstrain” is correctly installed. **Access** the folder flexpipe (tip: `cd flexpipe`), and **run** the command `conda env list`. If nextstrain is not in the list, **run** the command below to install the conda environment.

```
> mamba env create -n nextstrain --file config/nextstrain.yml
```

(5) **Activate** the conda environment, and check the packages that were installed for running the pipeline.

```
> conda activate nextstrain
  conda list
```

(6) The template pipeline contains all the required files to run the analyses. Let’s now **open** some of them and check their content:

`config/keep.txt`	→ List of contextual genomes to be included in the analyses
`config/ignore.txt`	→ List of contextual genomes to be explicitly ignored
`config/name2hue.tsv`	→ Hue degrees or matplotlib colormaps used by auspice
`config/clades.tsv`	→ List of tags to be assigned to internal nodes of the tree

`data/metadata.tsv`	→ Metadata associated with contextual genomes
`data/sequences.fasta`	→ Sequences of contextual genomes, named as in keep.txt
`data/new_metadata.xlsx`	→ Metadata associated with newly sequenced genomes
`data/new_sequences.fasta`	→ Sequences of newly sequenced genomes

(7) **Add** any newly sequenced genomes in `data/new_sequences.fasta`, and add their associated metadata lines in `data/new_metadata.xlsx`. Make sure that the name of the sequences in the FASTA file is precisely the same listed under the column ‘strain’ of the metadata.

(8) **Open** the file `config/auspice_config.json` using a text editor. **Edit** the lines that show the content below, and replace it with any appropriate information:

- ‘Add your project title here’
- ‘Add your name here’
- ‘Add your webpage URL here’

(9) **Open** the `Snakefile` located in the flexpipe folder. This file controls the steps of the workflow and lists all the input files and parameters that will be used in the analyses. By changing these elements users have the flexibility to use the same pipeline to run analyses for other pathogens.

(10) It’s time to **run** the first steps of the pipeline. First, let’s prepare some extra input files, such as `latlongs.tsv`, `colour_scheme.tsv`, among other required files:

```
> snakemake --cores all prepare
```

(11) **Watch** for potential error messages: they may indicate issues with file formatting in previous steps.

(12) **Check** the extra files generated inside `config` and `results`. In `results` you will find the final FASTA and metadata files combining contextual and newly sequenced genomes.

(13) Let’s now run the remaining steps of the pipeline. **Run** the command below to align the sequences using `mafft`:

```
> snakemake --cores all align
```

(14) **Check** the outputs that were generated inside `results/alignments`. Now, **run** the command below to infer a phylogenetic tree using `iqtree`:

```
> snakemake --cores all tree
```

(15) **Run** the command below to infer a time-calibrated phylogenetic tree using `treetime`. At this step you may note that some genomes were likely removed ("pruned") as molecular clock outliers:

```
> snakemake --cores all refine
```

(16) If no issues were flagged in the previous steps, **run** the last steps to export the final results:

```
> snakemake --cores all export
```

(17) If no issues were flagged, visit `https://auspice.us/`, drag and drop the file `auspice/results.json` onto the webpage to visualize the results.

(18) One can also visualize the results in offline mode. To do so, run the command below, and access the URL displayed in the output. To resume the visualization on your terminal, press Ctrl + C.

```
> nextstrain view --conda auspice
```

(19) Nextstrain also allows users to host their results on GitHub and share the interactive results via a custom URL. To do so, you need to create an account and a public repository on `https://github.com/`. Once that is done, rename the JSON file as specified below, upload the folder auspice to the public repository, as and visit a URL similar to this one:

```
https://nextstrain.org/community/GitHubUsername/RepoName/ProjectName
```

Where:

- `GitHubUsername` = your GitHub account name;
- `RepoName` = name of the repository where the auspice folder is located;
- `ProjectName` = name of your project. It should be included as a suffix of the JSON file name, preceded by the repository name (for example: `RepoName_ProjectName.json`).