**Question: How to create a new Nextclade Dataset?**

**2023-04-06**

A Nextclade dataset is really a zipped folder of files that are organized similar to

* <https://github.com/nextstrain/nextclade_data/tree/master/data/datasets/flu_vic_ha/references/KX058884/versions/2021-08-27T15%3A51%3A42Z/files>
* <https://docs.nextstrain.org/projects/nextclade/en/stable/user/datasets.html>

A quick way to generate a Nextclade dataset is by using our template script

* <https://github.com/nextstrain/nextclade_dataset_template>
* [reference set](https://github.com/nextstrain/nextclade/tree/master/data)

From Jover:

* This does not generate the tree.json, it only copies over the template tree.json. So you have to create the tree separately.
* It only creates reference.fasta and genemap.gff based on the GenBank reference passed.

Ergo, yes you will need to create your reference tree and use augur export v2 to convert it to tree.json. (Originally had a 9K PRRSV reference newick tree but have reduced the tree down to 10 reference sequences for testing)

* <https://docs.nextstrain.org/projects/augur/en/stable/usage/cli/export.html>

To call Nextclade using the dataset, check our Monkeypox Ingest Snakemake rule

* <https://github.com/nextstrain/monkeypox/blob/master/ingest/workflow/snakemake_rules/nextclade.smk#L20-L37>
* Jason mentioned that the alignment parameters probably need to be tweaked

In summary, could go through steps similar to:

| # Pull the template script  git clone <https://github.com/nextstrain/nextclade_dataset_template.git> cd [nextclade\_dataset\_template](https://github.com/nextstrain/nextclade_dataset_template.git)  # Generating the nextclade dataset for PRRSV python generate\_from\_genbank.py \  --reference NC\_001961 \  --output-dir PRRSV  # Okay, turns out the tree.json file is copied over from a template, so it will need to be replaced # <https://docs.nextstrain.org/projects/augur/en/stable/usage/cli/export.html>  # Replace the PRRSV/files/tree.json  augur export v2 \  —tree your\_reference\_tree.nwk \  —json tree.json  # [edit: jump to 2023-04-13 notes below] In development to drop the -node-data requirement ([thread](https://github.com/nextstrain/augur/issues/273))  # augur align to check references, since internal to nextclade  # Zip the files  # Use the zipped file in a Nextclade  nextclade run \  -D PRRSV.zip \  -j 4 \  --retry-reverse-complement \  --output-fasta output\_alignment.fasta \  --output-translations translations\_{gene}.zip \  --output-insertions insertions.csv \  sequences.fasta |
| --- |

—--

**Question: Some clarification questions**

**2023-04-13**

* The augur export v2 has a 3rd required parameter for a node json with properties.
  + What extra data can we put through into this JSON?
* We have also noticed figtree will export a JSON format, is this compatible?

Might need augur traits to create the node.json file

* <https://docs.nextstrain.org/projects/augur/en/stable/usage/cli/traits.html>

May need a metadata file with "strain", "clade\_membership" based on view of [tree.json](https://github.com/nextstrain/nextclade_data/blob/master/data/datasets/flu_vic_ha/references/KX058884/versions/2021-08-27T15%3A51%3A42Z/files/tree.json) data

| "children": [  {  "branch\_attrs": {  "mutations": {}  },  "name": "B/Brisbane/60/2008",  "node\_attrs": {  "clade\_membership": {  "value": "1A"  },  "continent": {  "value": "Oceania"  },  "country": {  "value": "Australia"  },  "div": 0.0,  "fluSeason": {  "value": "Southern Hemisphere"  },  "num\_date": {  "value": 2008.4971539162116  },  "accessionID": {  "value": "KX058884"  }  }  }, |
| --- |

| augur traits \  --tree your\_tree.newick \ # Think this is newick based on [src/traits.py](https://github.com/nextstrain/augur/blob/master/augur/traits.py#L51)  --metadata your\_metadata.tsv \ # strain, clade\_membership  > clade\_membership.json  augur export v2 \  --tree your\_reference\_tree.nwk \  --node-data clade\_membership.json \  --output tree.json |
| --- |

For their purposes, will be expanding out to build Nextclade datasets for other swine pathogens (e.g. PED, rotavirus)

Question: Can they run their own nextclade web instance to provide a graphical pre-processed service to clients?

Docs for using custom Nextclade datasets through URL parameters: <https://docs.nextstrain.org/projects/nextclade/en/stable/user/nextclade-web.html#url-parameters>

Regarding the Reference tree, Richard brought up some points about properly rooting the tree

* Mostly explaining the necessity of this snakemake rule: <https://github.com/neherlab/nextclade_data_workflows/blob/master/flu/Snakefile#L386-L417>

—-

Jen - Might be useful, seems to be the internal workflow to create flu/rsv/other datasets but requires access to their HPC  
  
\* <https://github.com/neherlab/nextclade_data_workflows>

Michael - Draft for PRRSV

* <https://github.com/mazeller/nextclade_test>

Script

| #! /usr/bin/env bash  set -euv  INFILE="sequences.fasta"  REF="reference.fasta"  REF\_GFF="genemap.gff"  [[ -d "results" ]] || mkdir results  echo "strain|date|clade\_membership" | tr '|' '\t' > metadata.tsv  grep ">" ${INFILE} \  | sed 's/>//g' \  | awk -F'|' '{print $0"\t"$3"\t"$4}' >> metadata.tsv  augur align \  --sequences ${INFILE} \  --reference-sequence ${REF} \  --output results/prrsv\_aln.fasta \  --fill-gaps \  --nthreads 1  augur tree \  --alignment results/prrsv\_aln.fasta \  --output results/tree.nwk \  --nthreads 1  augur refine \  --tree results/tree.nwk \  --alignment results/prrsv\_aln.fasta \  --metadata metadata.tsv \  --output-tree results/refined\_tree.nwk \  --output-node-data results/branch\_labels.json  augur ancestral \  --tree results/refined\_tree.nwk \  --alignment results/prrsv\_aln.fasta \  --output-node-data results/nt-muts.json \  --inference joint  augur translate \  --tree results/refined\_tree.nwk \  --ancestral-sequences results/nt-muts.json \  --reference-sequence ${REF\_GFF} \  --output results/aa-muts.json  augur traits \  --tree results/refined\_tree.nwk \  --metadata metadata.tsv \  --output results/clade\_membership.json \  --columns clade\_membership  augur export v2 \  --tree results/refined\_tree.nwk \  --node-data \  results/branch\_labels.json \  results/clade\_membership.json \  results/nt-muts.json \  results/aa-muts.json \  --output results/tree.json  # --validation-mode skip : optional, but want a valid tree for clients to view in the web  echo "Done! look at results/tree.json"  # Copy in the meta entries or maybe use something like  # https://github.com/nextstrain/dengue/blob/main/config/auspice\_config\_all.json  #  # zip into PRRSV.zip  # nextclade run -D PRRSV.zip -j 4 \  # --output-fasta output\_alignment.fasta \  # --output-translations translations\_{gene}.zip \  # --output-insertions insertions.csv \  # sequences.fasta  # Validation error |
| --- |

Still hits some warnings

| Validating produced JSON  Validating schema of 'results/tree.json'...  .meta.genome\_annotations {"nuc": {"end": 603, "start": 1, "strand": "+"},…} failed additionalProperties validation for false  .tree {"name": "NODE\_0000000", "node\_attrs": {"div": 0…} failed oneOf validation for [{"$ref": "#/$defs/tree"}, {"type": "array", "minItems": 1, "items": {"$ref": "#/$defs/tree"}}]  validation for arm 0: {"$ref": "#/$defs/tree"}  .tree.children[…].branch\_attrs.mutations {"nuc": ["A39T", "T50C"], "25kD major envelope p…} failed additionalProperties validation for false  .tree.children[…].branch\_attrs.mutations {"nuc": ["G79A", "T83C", "A381G", "G523A"], "25k…} failed additionalProperties validation for false  .tree.children[…].branch\_attrs.mutations {"nuc": ["C35T", "T44C", "A89G", "A95G", "A97G",…} failed additionalProperties validation for false  .tree.children[…].branch\_attrs.mutations {"nuc": ["A11G", "C77T", "T83C", "C84T", "G85A",…} failed additionalProperties validation for false  .tree.children[…].branch\_attrs.mutations {"nuc": ["G29T", "T44C", "A95G", "A173G", "T198C…} failed additionalProperties validation for false  .tree.children[…].branch\_attrs.mutations {"nuc": ["T59C", "G81A", "A101G", "T222A", "T272…} failed additionalProperties validation for false  .tree.children[…].branch\_attrs.mutations {"nuc": ["T86C"], "25kD major envelope protein":…} failed additionalProperties validation for false  .tree.children[…].branch\_attrs.mutations {"nuc": ["G8A", "A101G"], "25kD major envelope p…} failed additionalProperties validation for false  .tree.children[…].branch\_attrs.mutations {"nuc": ["G51C", "A95C", "A98G", "C166T", "A172C…} failed additionalProperties validation for false  .tree.children[…].branch\_attrs.mutations {"nuc": ["G8T", "C30T", "C80T", "C133T", "G219A"…} failed additionalProperties validation for false  .tree.children[…].branch\_attrs.mutations {"nuc": ["C77T", "A176C", "A177C", "A189G", "C20…} failed additionalProperties validation for false  .tree.children[…].branch\_attrs.mutations {"nuc": ["C30T", "T47C", "C80T", "A95G", "A100G"…} failed additionalProperties validation for false  .tree.children[…].branch\_attrs.mutations {"nuc": ["G29A", "T144C", "A177G", "C207T", "C36…} failed additionalProperties validation for false  .tree.children[…].branch\_attrs.mutations {"nuc": ["G270A", "A321C", "T362C", "C474T", "T4…} failed additionalProperties validation for false  .tree.children[0].branch\_attrs.mutations {"nuc": ["G29T", "T44C", "G66A", "A98G", "T171C"…} failed additionalProperties validation for false  .tree.children[1].branch\_attrs.mutations {"nuc": ["C30T", "T68C", "C77T", "A101G", "G138T…} failed additionalProperties validation for false  .tree.children[2].branch\_attrs.mutations {"nuc": ["C183T", "T225C", "T234C", "T246C", "T2…} failed additionalProperties validation for false  validation for arm 1: {"type": "array", "minItems": 1, "items": {"$ref": "#/$defs/tree"}}  .tree {"name": "NODE\_0000000", "node\_attrs": {"div": 0…} failed type validation for "array"  Validation of 'results/tree.json' failed.  ------------------------  Validation of results/tree.json failed. Please check this in a local instance of `auspice`, as it is not expected to display correctly.  ------------------------ |
| --- |

Copy some items into the "meta" field. Make sure there are no duplicates.

| "meta": {  "title": "PRRSV nextclade reference tree",  "updated": "2023-04-18",  "build\_url": "https://github.com/neherlab/nextclade\_dataset\_template",  "data\_provenance": [  {  "name": "fictional data"  }  ],  "maintainers": [  {  "name": "optional: Your Name",  "url": "optional url to your website"  }  ],  "display\_defaults": {  "color\_by": "clade\_membership",  "distance\_measure": "div",  "branch\_label": "clade"  },  "colorings": [  {  "key": "clade\_membership",  "title": "Clade",  "type": "categorical"  },  {  "key": "gt",  "title": "Genotype",  "type": "categorical"  }  ], |
| --- |

—-

"It will be a moment until I can touch this again, but one thing I am noticing with the IAV[Influenza A Virus] set, is the amino acid positioning appears to be all over the place. I will vet the sequences though and let you know what I find. Thanks so much for the help so far"

* Failing samples seem to have <75 sequence identity with provided root
* <https://docs.nextstrain.org/projects/nextclade/en/stable/user/algorithm/01-sequence-alignment.html> "Can maybe lower this threshold in Nextclade CLI, but not for web
* Ah, and the failing samples might be the "alpha-deletion" clade, which has 2 or 3 amino acids deleted in the middle
* GitHub Issue related to failing samples due to low sequence identity <https://github.com/nextstrain/nextclade/issues/297>