Pipeline with Augur to generate the clade designations

This document outlines how we will relate the influenza virus sequences we have to the WHO clade designations. This in turn will help us to relate the predictions we make to known vaccines in recent years, and to understand how our predictions would translate into vaccine strain selection.

(1) Align sequences

- Command:

```
python3 -m augur align --sequences /path/to/FASTA2018-5.fa --reference-sequence
/path/to/h3n2_ha_outgroup.gb --output /path/to/aligned.fasta --fill-gaps
```

- Input:

--sequences /path/to/FASTA2018-5.fa : Just the regular fasta file with the DNA sequences
--reference-sequence /path/to/h3n2_ha_outgroup.gb : Here we used the augur reference sequence, which can be downloaded at
https://github.com/nextstrain/seasonal-flu/blob/master/config/reference_h3n2_ha.gb
for h3n2 and at corresponding links below for other strains.
--output /path/to/aligned.fasta : all the DNA sequences from FASTA2018-5.fa aligned based on the sequence "h3n2_ha_outgroup.gb" or the corresponding reference for other strains

Remarks:
- To run this command, it is necessary to install MAFFT in addition to Augur.
- If the sequences in the file "ASR_dna.fa" were correctly aligned with the reference sequence that they use, we could skip this alignment step

(2) Infer ancestral sequences

- Command:

```
python3 -m augur ancestral --tree /path/to/flutree2018-5.nwk --alignment /path/to/aligned.fasta
--output /path/to/nt_muts.json --inference joint
```

- Input:

--tree /path/to/flutree2018-5.nwk : the Newick tree obtained from "flutree2018-5.Rdata". It must have labels in the internal nodes, which can be solved in R with "makeNodeLabel(flutree)".
--alignment /path/to/aligned.fasta : the aligned DNA sequences, either obtained from Augur in step (2.1) or from some other way.
--inference joint : it can be either "joint" or "marginal". It calculates joint or marginal maximum likelihood ancestral sequence states. I chose "joint"

--output /path/to/nt_muts.json : it is a JSON file with the following information for each node in the tree: "sequence", the inferred sequence; "muts", a list of all mutations that occurred between the parent and the current node, obtained by comparing the sequence in the parent node with the one in the current node. For instance, "muts": ["T861C", "T930C"] means that a mutation from T to C occurred at positions 861 and 930 respectively.

- Remarks:

- We needed to change a bit the names of the sequences in the file "aligned.fasta" in order to make them match *exactly* the names used in the Newick file. For instance, instead of ">JQ988042,A/Moscow/26/2009,2009/02/25", it should be just ">JQ988042", otherwise "augur ancestral" can't match the leaves in the Newick tree with the DNA sequences in the fasta file. We attached the Python script "labelRenamer.py", which receives in "fname" the name of the fasta file containing the sequences with extended names (e.g. ">JQ988042,A/Moscow/26/2009,2009/02/25") and creates another file with the short version of the names, in order to match with the names used in the Newick tree.

(3) Assign clades (Python script)

Priscila Biller wrote a Python script to assign clades. It receives as input the sequences (a FASTA file properly aligned as above using the specified outgroup), and the file "clades_h3n2_ha.tsv" (or the corresponding file for the other influenza virus variants), which defines the criteria used to determine if a sequence belongs to a certain clade or not. The sequence is assigned with a clade only if it satisfies all criteria of the clade. The output is a TSV file where each line corresponds to a sequence in the FASTA file and the clades that have all criteria satisfied by the sequence.

- **Command:** python3 identifyClades.py

--> The script is very simple and doesn't receive any inputs. The beginning of the script needs to be changed according to the dataset (or maybe someone could make a more general version that receives these things from input parameters instead).

For a given sequence, the nucleotide sequence is cut in 3 pieces: "SigPep" (nucleotides 1..48), "HA1 protein" (nucleotides 49..1035), and "HA2 protein" (nucleotides 1036..1698), then each piece is translated into an AA sequence. During the translation of nucleotide sequences to amino-acids, the uncertain nucleotides are taken into account. For example, the codon "ATN" can be either "ATA", "ATC", "ATG", or "ATT". Then the constraints are processed. For example, "3b	HA1	223	I" checks if the 223th amino-acid of "HA1 protein" is "I".

It is possible that a sequence satisfies the criteria of more than one clade. For instance, the sequence "KU289613_A/Corsica/33-02/2015_2015/02/20" satisfies the clades "3c","3c3", and "3c3.B". In this scenario, the most specific clade is assigned ("3c3.B" in the example). However, there are still some cases where there was more than one "specific clade". For instance, the

sequence "KY273051_A/Xiamen/s175/2016_2016/03/14" satisfies "3c,3c2,A1,A1a". To resolve these we proceed visually (CC did this manually) using the trees at nextflu.org
The script is provided as identifyClades.py. BUT it will have to be customised because certain things are hard coded.

Input files from the augur/nextflu people

The relevant files for four influenza virus types are listed here.

Augur aligns to a distant reference / outgroup for H3N2, H1N1pdm, Vic and Yam. This doesn't have anything to do with what is chosen for seasonal vaccines.This reference sequence doesn't need updating from year-to-year, and sequences are available at:

* https://github.com/nextstrain/seasonal-flu/blob/master/config/reference_h3n2_ha.gb
* https://github.com/nextstrain/seasonal-flu/blob/master/config/reference_h1n1pdm_ha.gb
* https://github.com/nextstrain/seasonal-flu/blob/master/config/reference_vic_ha.gb
* https://github.com/nextstrain/seasonal-flu/blob/master/config/reference_yam_ha.gb

Clade definition files for all four lineages are available in the same directory:

* https://github.com/nextstrain/seasonal-flu/blob/master/config/clades_h3n2_ha.tsv
* https://github.com/nextstrain/seasonal-flu/blob/master/config/clades_h1n1pdm_ha.tsv
* https://github.com/nextstrain/seasonal-flu/blob/master/config/clades_vic_ha.tsv
* https://github.com/nextstrain/seasonal-flu/blob/master/config/clades_yam_ha.tsv

It is possible to annotate sequences with clades using "augur clades" as referenced here but we found that Priscila’s custom script was effective and did not require the ancestral sequence reconstruction. The augur instructions are at:
https://github.com/nextstrain/seasonal-flu/blob/master/Snakefile_base#L711