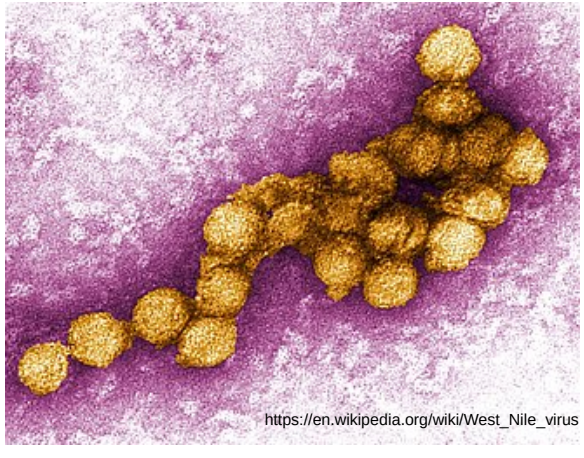
Nextstrain analysis on West Nile virus

Nextstrain



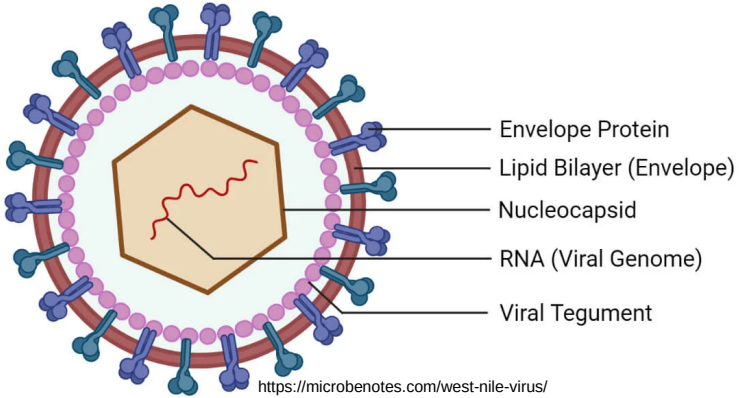
https://en.wikipedia.org/wiki/West_Nile_virus

Index

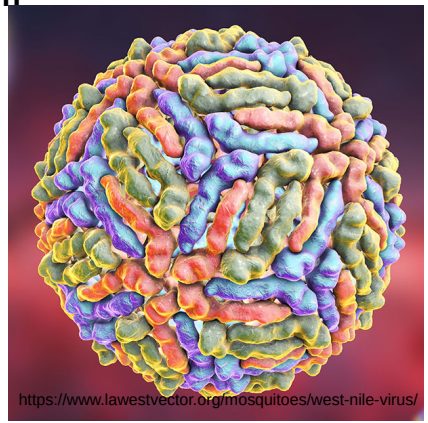
- **What?** West Nile virus and Nextstrain
- **How?** Workflow
- **Why?**
- **Result**
- **Further analyses**

What? West Nile virus (WNV)

- Single-stranded RNA (**class IV**) virus that causes West Nile fever
- Family: flaviviridae
- It infects mainly birds and mosquitoes, it can be transmitted to humans through mosquito bite
- Enveloped virus with icosahedral symmetry
- 45–50 nm virion covered with a relatively smooth protein shell



https://microbenotes.com/west-nile-virus/

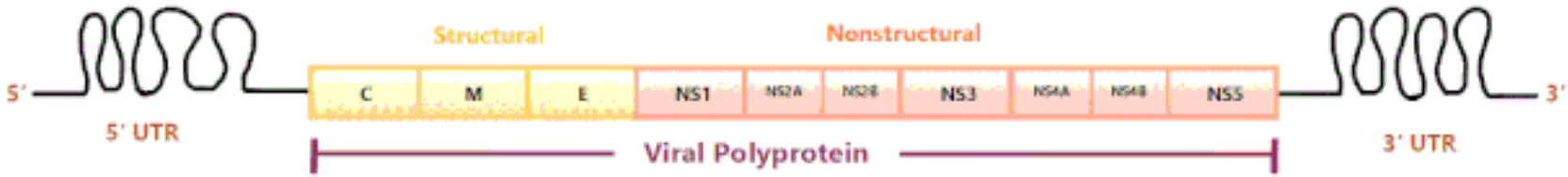


https://www.lawestvector.org/mosquitoes/west-nile-virus/

What? West Nile virus (WNV)

Genome:

- Approximately 11,000 nucleotides long and is flanked by 5′ and 3′ non-coding stem loop structures.
- Three structural proteins and seven nonstructural (NS) proteins.
- First translated into a polyprotein and later cleaved by virus and host proteases into separate proteins (i.e. NS1, C, E)



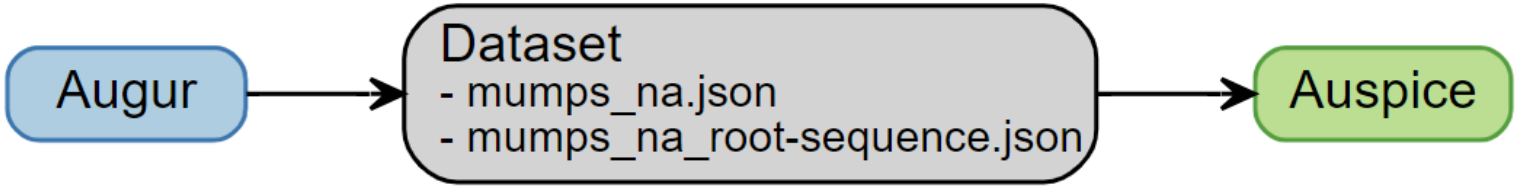
https://en.wikipedia.org/wiki/West_Nile_virus

What? Nextstrain

Nextstrain is a collection of open-source tools for **visualising** the **genetics** behind the **spread of viral outbreaks**. The two core parts are Augur and Auspice.

Augur is a series of composable, **modular bioinformatics tools**. We use these to create recipes for different pathogens and different analyses, which can be reproduced given the same input data and replicated when new data is available.

Auspice is a **web-based visualization program**, to present and interact with phylogenomic and phylogeographic data.



https://nextstrain.org/

Why?

1. Track the **spread** and **evolution** of WNV

2. Use the generated data for further analyses

How? Workflow

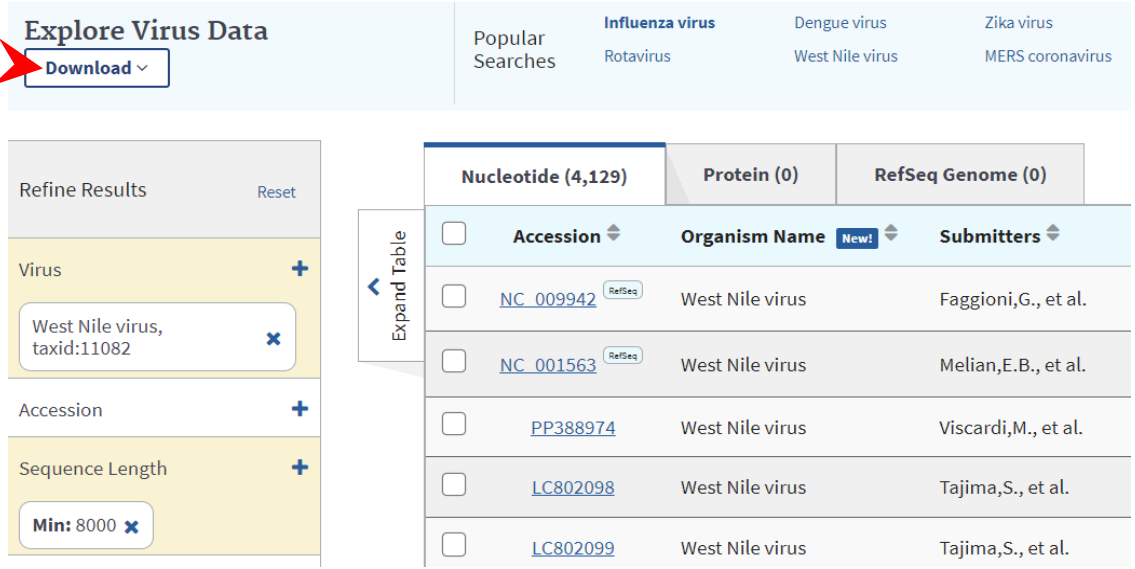
Workflow = sequence of data, commands and parameters which work together to **reproducibly** generate an output.

A workflow manager (e.g. Snakemake) allows to use bioinformatics tools rigorously, defining each subsequent manipulation performed on the input data.

How? Workflow

Workflow = sequence of data, commands and parameters which work together to **reproducibly** generate an output.

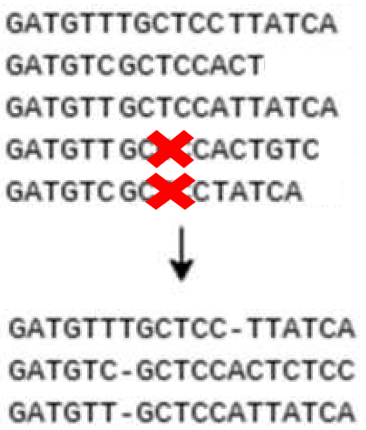
1. **Sequences and metadata**



How? Workflow

2. Filter: filter sequences and metadata to **exclude strains** from subsequent analysis and **subsample** the remaining strains to a fixed number of samples per group (e.g. country).

How? Workflow



2. Filter: filter sequences and metadata to **exclude strains** from subsequent analysis and **subsample** the remaining strains to a fixed number of samples per group (e.g. country).

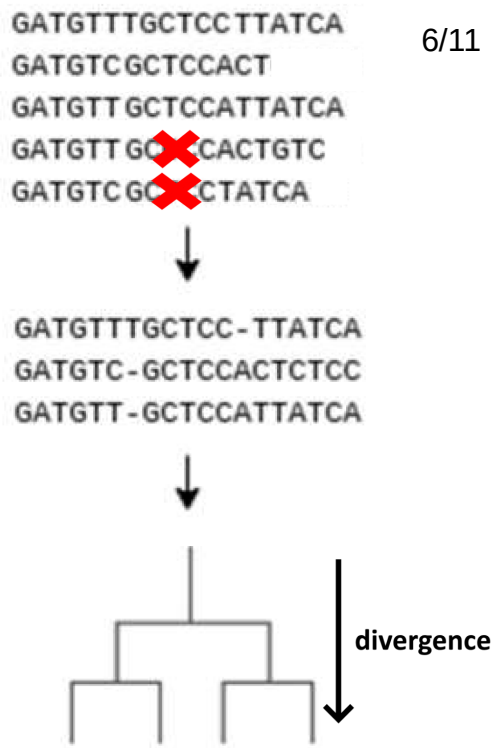
3. Align: Create a multi-sequence alignment.

How? Workflow

2. **Filter**: filter sequences and metadata to **exclude strains** from subsequent analysis and **subsample** the remaining strains to a fixed number of samples per group (e.g. country).

3. **Align**: Create a multi-sequence alignment.

4. **Tree**: Infer a phylogenetic tree from the multi-sequence alignment.



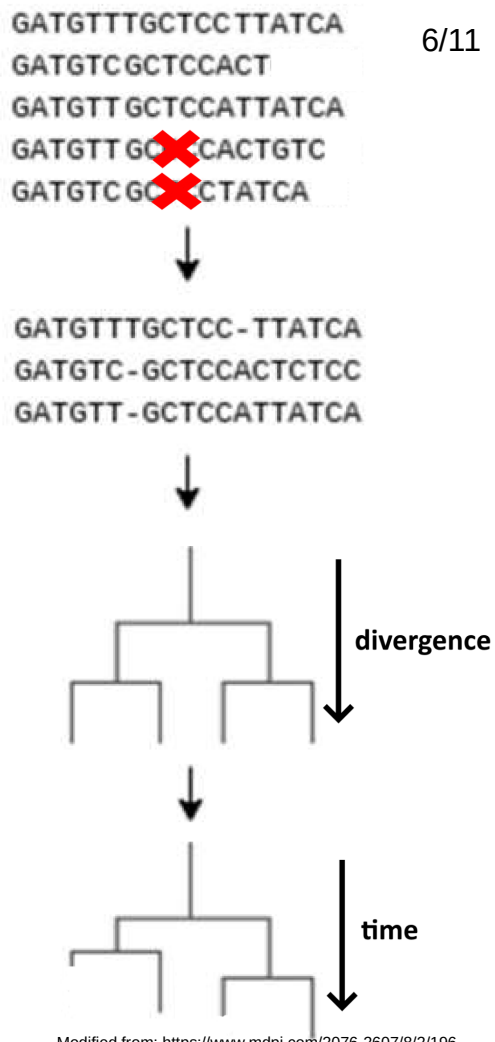
How? Workflow

2. Filter: filter sequences and metadata to **exclude strains** from subsequent analysis and **subsample** the remaining strains to a fixed number of samples per group (e.g. country).

3. Align: Create a multi-sequence alignment.

4. Tree: Infer a phylogenetic tree from the multi-sequence alignment.

5. Refine (time-resolved tree): adjust branch lengths in this tree to position tips by their sample date and infer the most likely time of their ancestors (thanks to metadata) (using TreeTime)



Modified from: https://www.mdpi.com/2076-2607/8/2/196

How? Workflow

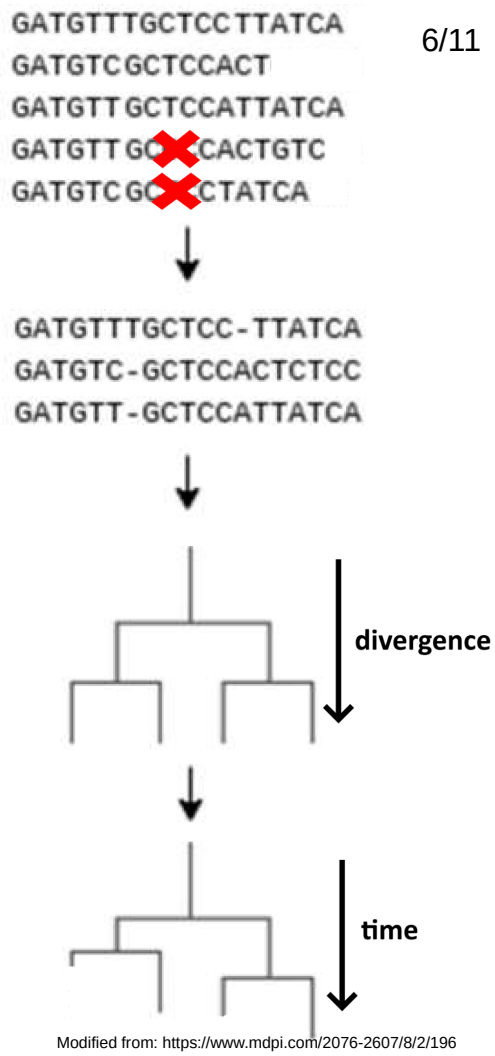
2. Filter: filter sequences and metadata to **exclude strains** from subsequent analysis and **subsample** the remaining strains to a fixed number of samples per group (e.g. country).

3. Align: Create a multi-sequence alignment.

4. Tree: Infer a phylogenetic tree from the multi-sequence alignment.

5. Refine (time-resolved tree): adjust branch lengths in this tree to position tips by their sample date and infer the most likely time of their ancestors (thanks to metadata) (using TreeTime)

Now we have a tree!! But it can get much better!!



Modified from: https://www.mdpi.com/2076-2607/8/2/196

How? Workflow

Node annotations:

6. **Ancestral**: infer the ancestral sequence and metadata of each internal node and identify any nucleotide mutations on the branches leading to any node in the tree (using TreeTime).

7. **Translate**: Identify amino acid mutations from the nucleotide mutations and a reference sequence with gene coordinate annotations.

8. **Traits**: define metadata interpretation (e.g. column x contains country data).

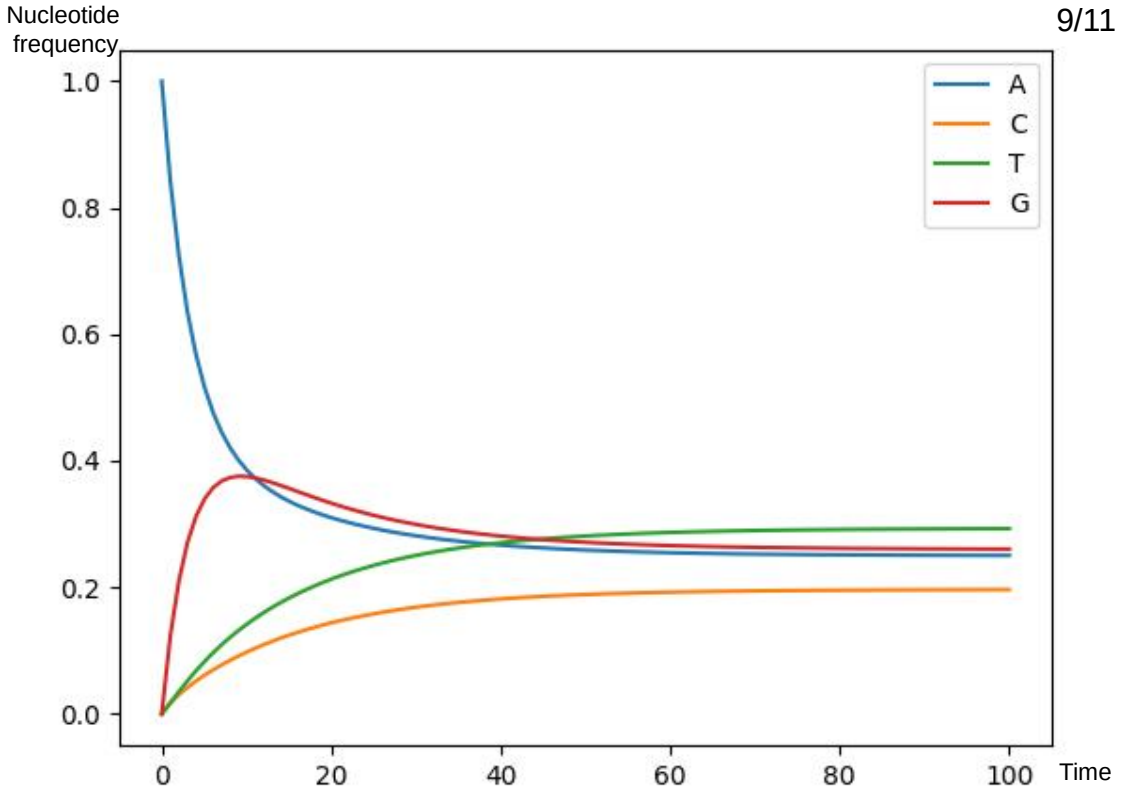
Result

10. Export: collect the tree, all the node annotations and metadata and export it in Auspice’s JSON format.

View the results in Auspice…

Further analyses

GTR model: simple substitution model



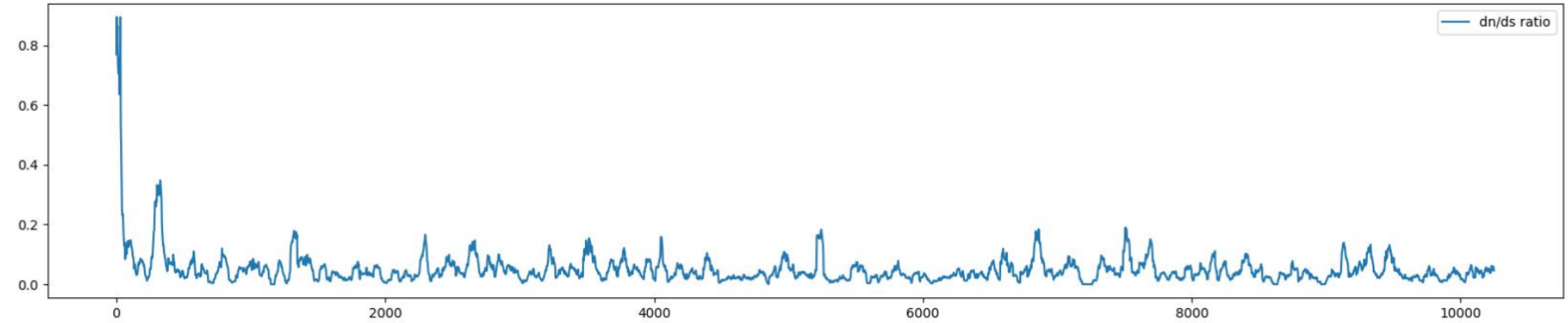
Estimated frequencies: A=0.25, C=0.19, T=0.29, G=0.26

Empirical frequencies: A=0.24, C=0.26, T=0.19, G=0.29

Natural selection is acting on the sequence

Further analyses

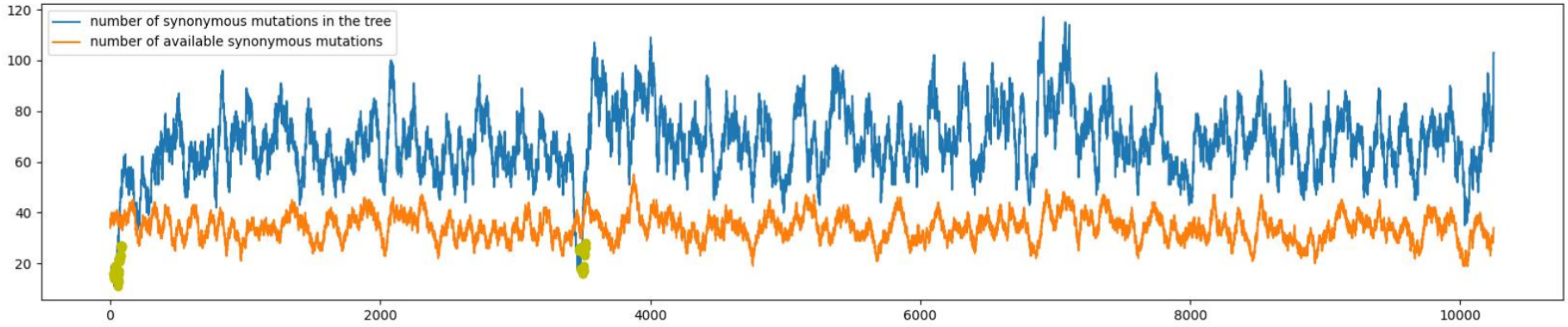
dN/dS: ratio between non-synonymous and synonymous mutations



dN/dS is low over the whole genome, every change in the sequence is counterselected

Further analyses

Secondary structure analysis



Some spots are poor in synonym mutations, secondary structures in RNA are probably present

Thank you!