Coding Session – day 4

At this point, you should have generated your first genome sequences, congratulations!

The sequences themselves come in the form of “fasta-files”. To keep practicing your biopython skills, the first assignments in this doc are about ambiguous sites in your fasta-file(s). Both nextclade and irma will provide you with information about ambiguous sites too, so you can easily check whether your scripts work as intended.

If you have more time, you can try to analyze some of the other output-files provided by nextclade and irma (**optional**). With a bit of programming, you can analyze these files exactly the way you want!

Remember: add your scripts to your github repository at the end of the day!

## Python script for listing ambiguous sites

In the python module (Day 2), you created a biopython script that prints the length of the sequences in a fasta-file. You can use this script as your starting template for your new script.

**Note**: Start by testing your original script, making sure its bug-free (and you have the necessary conda environment loaded).

Instead of calculating the length of your sequence, you need to save the sequence as a variable (a “string”). From here, it is back to standard python. Your task will be to count all characters that are **not** a normal base, iterating through the string.

## Python script for counting ambiguous sites

Instead of listing each ambiguous base and its position, you may want to summarize how many ambiguous sites there are in total.

Write a script that counts all characters which are **not** a normal base (A, C, T, G) and N bases.

This would give us an idea of how many sites per sequence contains ambiguous/undetermined bases.

## Python script for calculating the proportion of undetermined bases

When using PCR enrichment for sequencing a genome, there is always a risk of “amplicon dropout” (regions of the genome where the PCR doesn’t work properly, and you therefore don’t have sufficient coverage for base-calling), Therefore, an important marker of genome quality would be the proportion of undetermined bases (missing sites, normally denoted as ‘N’).

Specifically, we would like to know the number of undetermined bases per sequence, and the proportion compared to the total length of the sequence.

For example:

The segment PB1 of Sample1 has 10 missing sites.   
Calculate the length of the segment and divide the number of missing sites by the length. Multiply this number by 100 to get the percentage of missing sites.

## Optional exercises

Here are some ideas for practicing your coding skills. If there is some other task you would like to write a script for, feel free to do that instead.

1. Write a python script that reads a nextclade output-file and calculates the number of substitutions found in each sequence (relative to the reference genome). Think about what kind of output format you would like. Perhaps sample-id and substitution-count on one line? Tab-delimited? And a nice header?
2. Write a python script that works on an interesting outfile generated by IRMA, to summarize some relevant stats

## Acknowledgements

This coding session was developed by Kirsten Ellegaard (SSI) and Martha Ciucani (SSI).