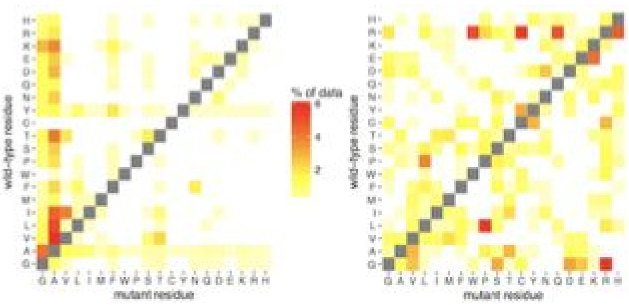
NGS for protein variants - exercise 1

Goals: become familiar with DNA variants, translating them into protein space, and summarising the observations
Tools: (you can work in the programming language of your choice but we can help best with these two)
R: Bioconductor, Biostrings, MSA
Python3: BioPython https://biopython.org/wiki/

- ❏ Download native_DNA.fa from Absalon
- ❏ Read in - you can use existing FASTA reader functions
- ❏ Translate the DNA sequence to protein
    - ❏ Again, you can use existing translate() functions
    - ❏ For `R/Biostrings`, you may want `toString(translate(DNAString(s)))`
- ❏ What does the * at the end of the sequence mean?
- ❏ Mutagenesis - generating synthetic variant data
    - ❏ Introduce a single mutation (randomly choose a position, then randomly choose one of the 4 nucleotides to insert)
    - ❏ Translate the new DNA sequence and look for differences to the wild type (WT) **in protein space**
- ❏ Repeat the random mutagenesis above 1000 times to generate a diverse set of sequences. Record either the sequences or their differences to the WT
    - ❏ There may be mutations that introduce a premature STOP codon. You should either remove any amino acids after the stop codon, or fully exclude those sequences from your analysis.
- ❏ Create a 20x20 matrix showing all possible combinations of wt/mutant amino acids, to summarise the changes observed in your sequence set



    - ❏ You could e.g. pre-initialise a matrix with all the possible 20x20 combinations and initialise counts to zero, then iterate over the sequences, compare each to the original sequence and +1 whenever you spot a difference
    - ❏ Python: If you are not familiar with data frames (pandas) a common trick is to write your data to a csv file and read that file back in with a csv reader
    - ❏ See illustration for examples of how such matrices might look - the data is different though, so yours will look different
    - ❏ If you want to use `ggplot()` for visualisation, you'll need `melt()` from the `reshape2` package to reformat your matrix into a long, narrow dataframe first:

    https://seananderson.ca/2013/10/19/reshape/ - and then `geom_tile()` for the actual heat map visualisation
    - ❏ In python you can plot with matplotlib/seaborn. Heatmaps are easy to generate from numpy arrays or pandas data frames
- ❏ What substitutions do you observe? Are there amino acid changes that are never observed? Discuss why that would be.