# Homework 2 for computer scientists
# 1-INF-301: Methods in bioinformatics
# Deadline: Wednesday December 3, 2025 22:00

**Comparative genomics.** Implement this task in one of the languages Java, C, C++ or Python. If you want use another language, please consult with the instructors first. Use only the standard libraries in your language. The exception is that you may use Numpy library in Python (but not other specialized libraries) and you may use arbitrary libraries or other visualization software for final data analysis and visualization in part e) and f).

Submit your source code in one zip file and the results or discussion in questions a), b), d), e), or f) as a pdf file. Data for this task (files `tree.txt`, `exons.txt` and `cftr.txt`) can be found on the course website. Depending on the efficiency of your implementation, the calculation in part d) may be take long, so leave enough time.

Felsenstein's algorithm to be implemented in part a) was covered on tutorials November 13. You can find some notes on the course website or in many sources, including this online textbook by Luke J. Harmon.

a) Implement a function, which gets a rooted tree with branch lengths, mutation rate $\alpha$ and one column of a multiple alignment with rows corresponding to the leaves of the tree. The function should use the Felsenstein's algorithm to compute the probability that on this tree the Jukes-Cantor model generates this particular alignment column. Interpret alignment gaps (denoted by minus signs) as missing data, that is, the Felsenstein algorithm should use $A[v,a]=1$ for each base $a$ for leaf $v$ with a gap. Character N indicates an unknown base and is also treated as missing data.

   In the Jukes-Cantor model, the probability that the base stays the same after time $t$ is $P(A|A,t)=(1+3e^{-\frac{4}{3}\alpha t})/4$ and that it changes to a different particular base (e.g. from A to C) $P(C|A,t)=(1-e^{-\frac{4}{3}\alpha t})/4$. The probability of each base in the equilibrium is $\frac{1}{4}$, which is used as probability $q_a$ of generating base $a$ in the root.[1]

   In you pdf file, describe how you store the tree in your program and how you implement the dynamic programming algorithm on the tree.

b) What probability will your function return for the tree stored in file `tree.txt` and a column of alignment with all bases equal? Show the result for $\alpha=1$ and $\alpha=0.2$.

   The tree is specified in the file as a sequence of triplets (vertex, parent, edge length). The vertex named Root is the root of the tree and therefore does not have a parent.

c) Implement a function that gets the alignment of several sequences and a phylogenetic tree and finds the value of the parameter $\alpha$, for which this alignment achieves the highest probability. Assume that individual alignment columns evolve independently according to the Jukes-Cantor model with the same value of $\alpha$. To find the optimal value of $\alpha$, simply try the values of $\alpha$ from 0.1 to 2 with step 0.1 and select $\alpha^*$ for which you get the highest probability of generating the input alignment. (In practice, more complex and accurate optimization methods are used.)

   As you will run the function on relatively long sequences, the probability of generating the whole alignment can be very small and numerical problems with numbers that are too small can occur. To avoid this, take the logarithm of the probability of generating each column, and combine column probabilities in the logarithmic form (multiplication of probabilities is replaced by summation of their logarithms).

d) Write a program that loads a long multiple alignment, splits it into non-overlapping windows of length $w=100$, and for each window calculates the optimal value of $\alpha^*$ using the function from part c).

   Run your program on the alignment in file `cftr.txt`. This file contains the alignment of a portion of the CFTR region of 14 mammals. The species name is given on a line beginning with >, followed by a series of lines with the sequence into which the gaps (minus signs) were inserted.

   In the text part of the homework, please list the found value of $\alpha^*$ for the first 10 windows (the first window contains columns $1,\dots,100$, the second window columns $101,\dots,200$, etc.).

e) File `exons.txt` contains start and end positions of protein coding regions (exons) in the human sequence included in our alignment. The bases in the human sequence are numbered starting from 1 and gaps are not

[1] Beware, in many programming langauges $4/3=1$, which is not what you want.

counted as positions (but bases N are counted). For example, interval 2,4 in sequence `A-T--CN-T` corresponds to bases `TCN`. Split the windows from task d) into two groups: those that overlap some exon (even by a single base) and those that do not overlap any exon.

For each group of windows make a histogram of optimal values of $\alpha^*$ and plot it (using Excel or some other statistical or visualization tool or library). Comment on your results: what are the differences between the two histograms? Do the histograms look as you would expect or is there anything surprising? What do you think causes the observed trends?

f) For at most 3 bonus points, analyze the data in some other way. You can, for example, try a smaller number of species, change the window size or look at properties of windows with very small or large values of $\alpha^*$. Describe what you did, what results you obtained and comment on them.