readMe

randomIC.R

Given the number of states and the size of the alphabet, generates random initial conditions for the model parameters, to be used in the fwd/bwd HMM learning algorithm.

INPUT:
K: number of states in the model
M: the size of the alphabet (4 in the case of {A, G, T, C})

OUTPUT:
Uniformly random HMM parameters
pi0: initial state cmf for the model
A: the transition matrix for the model
E: the emission matrix for the model

likelihood.R

Given the HMM, outputs the log-likelihood of the observed sequence, as well as the log-likelihood per sequence step

INPUT:
K: number of states in the model
M: the size of the alphabet (4 in the case of {A, G, T, C})
bpSequence: the sequence of emitted nuleotides
pi0: initial state cmf for the model
A: the transition matrix for the model
E: the emission matrix for the model

OUTPUT:
Log[P(bpSequence | HMM)]: the log-probability of observing the emitted sequence, given the model parameters
Log[P(bpSequence | HMM)} / length(bpSequence): the log-probability of observing a single base-pair from the emitted sequence

NOTES:
log-transform is used throughout the calculations to avoid underflow issues
special treatment is given to arithmetic operations with log-s (+, *, …)

functions.R

Contains a single function that converts the input nucleotide sequence into a set of numbers {1, 2, 3, 4}.

bwLearning.R

Given the number of model states, the alphabet, the observed sequence and initial conditions for the model parameters, performs a bwd/fwd learning and returns the locally optimal values of the model parameters.

INPUT:
K: number of states in the model
M: the size of the alphabet (4 in the case of {A, G, T, C})
bpSequence: the sequence of emitted nucleotides
pi0: initial condition for the initial state cmf for the model
A: initial condition for the transition matrix for the model
E: initial condition for the emission matrix for the model
eps: the minimum difference between the log-likelihoods of observing the bpSequence in consecutive fwd/bwd iterations required for termination of the loop
bwMax: the maximum number of fwd/bwd iterations in the EM algorithm

OUTPUT:
pi0.log: the log of the inferred initial state cmf of the model
A.log: the log of the inferred transition matrix for the model
E.log: the log of the inferred emission matrix for the model
LogLikelihoods: the log likelihoods of oberserving the bpSequence at each fwd/bwd iteration

bwLearningGlobal.R

Repeats the local learning through bwLearning.R and returns the values of model parameters which lead to the highest likelihood.

consistencyCheck.R

Used to check the consistency of the bwLearning algorithm. The condition of consistency is that the likelihood of observing the synthetic nucleotide sequence under the inferred parameters should be higher than that under the initially chosen parameters.

main.R

Here we implement the fwd/bwd learning algorithm on a known bacterium reference sequence and learn the HMM paramaters for the K=4, Alphabet={A, T, C, G} model. We then input 4 different reads from our sample dataset, 1 of which was identified as an almost 100% match by Muther pipeline, while the other 3 were identified as mismatches. We calculate the likelihood of observing these sequences under the HMM learned from the reference sequence. The likelihoods per single nucleotide turn out to be all very close to 25%, suggesting that the HMM method does not yield practical results for the 16sRNA sequences, likely because of the short length of the reads and the lack of variability in sequences.