Assignment 4 Report

Brief Write up Of Code

This code is written in C# and reads the first sequence from the FASTA file corresponding to Methanocaldococcus jannaschii and implements HMM Viterbi algorithm and traceback using this sequence.

The output of this program is available in output_actual_data_wo_path.txt. This file contains :

1. Language in which the program is written
2. Prints for all 10 iterations
    a. Transition State Probabilities
    b. Emission State Probabilities
    c. Overall Log Probability
    d. Number of hits
    e. 1-based start and end index and length of first 5 hits for each of the first 9 iterations and all hits information for the 10^{th} iteration.
3. At the very end prints the total run time in seconds and this excludes the last iteration

The total run time 20.677s and it was run on i7-6600U CPU @2.60GHz with 16GB RAM

There is another file output_actual_data_with_path.txt that has the all the information listed above and additionally for every iteration includes the Viterbi path determined by traceback with '0' mapping to state1 and '1' to state2.

Matches for First 10 Hits with length > 50

For Methanocaldococcus jannaschii, the found subsequences match pretty close to the tRNAs. Listed below overlap found with first 10 hits with length > 50 output by my code against gff.

Start	End	Length	Gff Start	Gff End	Gff Attributes
97326	97541	216	97426	97537	GeneID:1450942;Name=MJ_RS00515
97627	97823	197	97629	97716	GeneID:1450943;Name=MJ_RS00520
111764	111856	93	111768	111852	GeneID:1450955;Name=MJ_RS00580
118079	118179	101	-	-	-
138345	138419	75	138344	138419	GeneID:1450982;Name=MJ_RS00730
154610	157697	3088	154662	157639	GeneID:1451001;Name=MJ_RS00815
157782	159591	1810	157847	157919	GeneID:1451002;Name=MJ_RS00820
186974	187067	94	186978	187066	GeneID:1451036;Name=MJ_RS00980
190831	190907	77	190832	190908	GeneID:1451045;Name=MJ_RS01025
215200	215296	97	215210	215297	GeneID:1451075;Name=MJ_RS01175

Extra Credit

2) Explore convergence of Viterbi and/or Baum-Welch on this data. Do the answers get better with substantially more iterations? E.g., if you run 100 passes or 1000, do you get the same answers, or does the solution slowly drift to something very different? How sloppy can we be with starting parameter values and still have it converge to an interesting solution? I picked "10 iterations" by eye; how might you automate convergence (and divergence) detection?

More iterations do no drift to yield better/different results. I tried upto 1000 passes and saw that the answers are the same at the end of the iterations, I see that the overall log probability hits a maximum value at a certain iteration and continues to be at that value for the rest of the iteration. The Viterbi algorithm converges faster.

Sloppiness in transition probabilities were better tolerated, however in the case of emission probabilities, if the probabilities for C&G were high, the algorithm converged to the correct values.

Another approach to determine convergence could be done by comparing the overall log probability of the current iteration with the previous iteration and see if the difference is within a set epsilon(~0.001) and does not go more than that. This is implemented in the code inside EMWithEpsilon() where it automatically stops when the log probability value converges.