Exercises 1-3
Statistics in Genetics

Alfons Edbom Devall

March 1, 2022

Contents

1 Chapter 4

1.1 Exercise 1

1.1.1 Background

Hidden Markov Models (HMMs) are often used when analyzing genetic sequences. What is unique and useful with HMMs is that within a single sequence S, the probability of the element at position i depends on k elements that comes before it in S. The number k is called the order of the HMM.
If we have a sequence S with L elements (in DNA, the number of bases). Then to construct a HMM we need to define the following:

The **symbols** (also called the alphabet) q which the sequence can contain, in a DNA-sequence that would correspond to the number of bases in the DNA (G, C, A, T).

The number of **hidden states** N the HMM will be made up of. If we are modeling a DNA-sequence with 2 states, these states could be: "High GC-content" and "High AT-content" respectively.

The **transition matrix** T which defines the probabilities to go between the different states. In the above example, it would then correspond to the probability to go from the state "High GC-content" to "High AT-content" or to stay in the same state.

The **starting vector** V which defines the probability for S to start in a given state.

The **emission probabilities** E which defines the probability for each state to emit/produce a given symbol. Using the above example of a HMM with 2 states, the probability that we get a G, C, A or T at a specific position when we are in the state "High GC-content", and the same for all the bases when we are in the state "High AT-content".

A common workflow when using HMMs is to start by approximating the values in T, V and E and then use either the Viterbi - or Baum-Welch algorithm to find the optimal parameters.

In this exercise, the lambda phage genome (NC_001416) will be analyzed by using two different HMMs.
The first being a 2 state HMM, with the states "High GC-content" and "High AT-content" having a bias towards the bases G and C as well as the bases A and T respectively.
The second HMM will be a 4 state HMM with one state biased towards each nucleotide in DNA, i.e. the states "High G", "High C", and so on...

1.1.2 Methods

The complete genome of the lambda phage was downloaded from GenBank. The genome was then analyzed by using the programming language R.
First the FASTA-file was loaded using the seqinr package
Then the 2-state HMM was initialized by using the function "initHMM" in the HMM package. The states used was: "High GC-content" and "High AT-content" as well as the starting vector *V2*, transition matrix *T2* and the emission matrix *E2*. After the 2-state HMM-model was initialized it was optimized using the "baumWelch" function in the HMM package for 10 iterations, using the DNA sequence from the lamda phage as the model sequence
When the 2-state model was done, the 4-state HMM was initialized in a similar way. Using the states: "High G", "High C", "High A", "High T". As well as the starting vector *V4*, transition matrix *T4* and the emission matrix *E4*. After it had been initialized, this model to was optmized using the "baumWelch" function in the HMM package for 10 iterations, this time also using the lamda phage as the model sequence.

To view the code for this exercise, see Appendix A.A

1.1.3 Results and Discussion

For the 2 state HMM, the initial values for the starting vector *V2*, transition matrix *T2* and the emission matrix *E2* was:

$V2 =$

HGC	HAT
0.5	0.5

$T2 =$

	to	
from	HGC	HAT
HGC	0.75	0.25
HAT	0.25	0.75

$E2 =$

	to			
from	A	T	C	G
HGC	0.05	0.05	0.45	0.45
HAT	0.45	0.45	0.05	0.05

I decided to set *V2* to have equal chance to start in either state.. For *T2* I decided to choose a pretty high bias for staying in the same state, since we expect the genome to have longer stretches of either "High GC-content" or "High AT-content", however the specific number of 75% was chosen rather arbitrarily. If one wanted to be even more accurate a possible approach could have been to use the lamda phage genome as a reference and calculate the number of times we transition from a G or a C to a A or T as well as the other way around and use that instead. For *E2* I also arbitrarily chose a

large bias for the "High GC-content" state to emit either G's or C's, as well as the other way around. Similar to *T2* it could have been approximated by calculating the number of times we transition from a G or a C to a G or a C and so on...

After 10 iterations of the Baum Welch algorithm with the DNA sequence from the lamda phage as the model the following values for the starting vector *V2*, transition matrix *T2* and the emission matrix *E2* was (rounded to 2 decimals):

$V2 =$

HGC	HAT
0.5	0.5

$T2 =$

	to	
from	HGC	HAT
HGC	0.58	0.42
HAT	0.43	0.57

$E2 =$

	to			
from	A	T	C	G
HGC	0.10	0.11	0.37	0.42
HAT	0.41	0.39	0.09	0.11

It can be seen that the starting vector *V2* is unchanged after all of the iterations. The transition vector *T2* changed from having a very high bias for staying in the same state (75%) to having only about a 55% chance for staying in the same state and conversely about a 45% chance for changing state. The emission matrix *E2* changed from having a total of 90% (45 % of each base) to emit a base of its bias to closer to a total of about 80 % for the corresponding bases, it can also be noted that the probability to get a G when in the "High GC-content" state the probability of getting a G is slightly higher than an C and when in the "High AT-content" state the probability for getting a A is slightly higher however this in not as pronounced as for the other state. I am inclined to conclude that these results are reasonable and correct, since they are still keeping the emission probabilities high for their respective states. But while they are still clearly biased towards their respective direction they went more towards "the middle" than what I arbitrarily set when initializing it, which probably makes sense from a biological perspective since it would be very highly improbable for sequences generated by natural selections that regions would be that skewed.

Similar to the 2 state HMM, the initial values for the starting vector *V4*, transition matrix *T4* and the emission matrix *E4* was decided using the same reasons as for the 2 state HMM, with the addition of arbitrarily saying that the probability of having a C after a G is higher than having a A or T and so on. The values I decided on can be seen

below:

$V4 =$

HG	GC	HA	HT
0.25	0.25	0.25	0.25

$T4 =$

	to			
from	HG	HC	HA	HT
HG	0.80	0.10	0.05	0.05
HC	0.10	0.80	0.05	0.05
HA	0.05	0.05	0.80	0.10
HT	0.05	0.05	0.10	0.80

$E4 =$

	to			
from	HG	HC	HA	HT
HG	0.80	0.10	0.05	0.05
HC	0.10	0.80	0.05	0.05
HA	0.05	0.05	0.80	0.10
HT	0.05	0.05	0.10	0.80

After 10 iterations of the Baum Welch algorithm with the DNA sequence from the lamda phage as the model the following values for *V4*, *T4* and *E4* was (rounded to 2 decimals):

$V4 =$

HG	GC	HA	HT
0.25	0.25	0.25	0.25

$T4 =$

	to			
from	HG	HC	HA	HT
HG	0.66	0.12	0.14	0.08
HC	0.21	0.58	0.11	0.10
HA	0.10	0.11	0.64	0.15
HT	0.11	0.12	0.10	0.68

$E4 =$

	to			
from	HG	HC	HA	HT
HG	0.41	0.24	0.16	0.20
HC	0.27	0.39	0.19	0.16
HA	0.17	0.16	0.52	0.14
HT	0.18	0.16	0.16	0.49

The results for the 4 state HMM are quite similar to that of the 2 state HMM. *T4* seems to still be quite biased to stay in the same state, however, my assumption that it would be more likely to go from "High G" to "High C" than either "High A or T" seems not to be quite valid. Since for all states except "High C" the probability to change to any of the other states seems to be approximately the same. I cannot really determine why this is the case and further studies into the actual structure of the DNA sequence would be needed to determine if this is correct or not. Also note that not all of the rows in *T4* and *E4* does sum up to 1, this is not an error but is due to the rounding (See A.A for the code used and rerun it to get the more accurate numbers)

2 Chapter 5

2.1 Exercise 1

2.1.1 Background

To determine how closely related different species are to each other, it is common to compare their mitochondrial DNA (mtDNA). The reasons for using mtDNA when comparing evolutionary distance are many, but here a few: Since each cell contains many mitochondria, we get multiple copies of each mtDNA from a single cell, which makes the extraction of the DNA much easier and makes it possible to get accurate information about even really old DNA-samples. We inherit our mtDNA only from our mother, so we have only one version of it, opposed to the nuclear DNA where we have two different copies of each chromosome (one from the mother and one from the father). The mtDNA contains some regions that are highly variable, in particular the region called the D-loop. When comparing the mtDNA of different species it is therefore very common to simply this region from the analysis as reduce some of the unnecessary noise.

In this exercise the mtDNA of mammoth, African and Indian elephants as well as a member of the Elephantulus genus (NC_007596,NC_000934, NC_005129, NC_004921) will be investigated to determine how closely related they are to each other.

2.1.2 Methods

To start, all of the above sequences was downloaded from Genbank. Using the information from Genbank the D-loop region was removed from the downloaded files. When the D-loop was removed from all the files, they were open in Jalview 2.11.1.7, and a multiple sequence alignment (MSA) was done. Based upon the MSA a phylogenetic tree was generated by using the "Neighbour Joining tree" algorithm, the similarity was calculated based on the "Percentage Identity" (PID), which is essentially the number of identical bases per 100 base pairs.

To view the MSA for this exercise, see Appendix A.B

2.1.3 Results and Discussion

The tree obtained in this exercise can be seen in figure 1. It can quickly be seen that the species that is the furthest away from the rest of the rest is the Elephantulus. It can also be seen that the elephants probably share some common ancestor after the elephants diverged from the mammoth. From figure 1 it can be seen that the mammoth and the elephants share some common ancestor which had a distance of 28.98 to the mammoth and 7.36 with the species that the elephants descended from. From that ancestor it seems that the distance to the African elephant is 18.38 and 24.66 to the Indian elephant. From that it follows that of the two elephant species the African elephant is more closely related to the mammoth.

To do the MSA, a multitude of different tools and packages could have been used, as well as different methods for calculating the distances between the species, but since I was familiar with Jalview since before I decided to use it to solve this exercise.

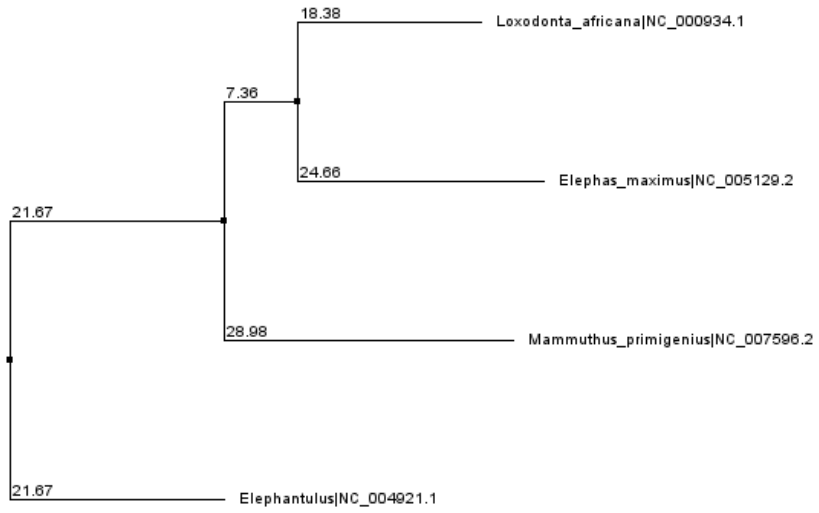

Figure 1 – Phylogenetic tree showing distance in PID between Elephantulus (Elephantulus|NC_004921.1), Mammoth (Mammuthus_primigenius|NC_007596.2), Indian elephant (Elephas_maximus|NC_005129.2) and African elephant (Loxodonta_africana|NC_000934.1)

The mtDNA sequences of blue whale, hippopotamus, and cow have accession numbers respectively NC 001601, NC 000889, NC 006853. Is the whale genetically closer to cows or to hippos?

Appendices

A R-code

The sections below shows the R-code used to solve the exercises in this report. All code can also be found at this github-link: `https://github.com/AlfonsEdbom/Exercises_Statistics_in_Genetics.git`

A.A Exercise 4.1

Ex4_1.R

Source on Save

```
require(pacman)  # Gives a confirmation message.
pacman::p_load(pacman, here, HMM, seqinr)

#Get path to the FASTA file
virus_file <- here("sequences", "NC_001416.fasta")

#load the file
virus_seq <- read.fasta(virus_file,forceDNAtolower = FALSE)

#The states

Symbols <- c("A", "T", "C", "G")

#Define parameters for 2 state - HMM
States <- c("H_GC", "H_AT")
start <- c(.5, .5) #probability to start in either state
trans <- matrix(c(.75, .25, .25, .75), 2) #probability to transition from a state to another
emission <- matrix(c(.05, .45, .05, .45, .45, .05, .45, .05), 2) #for each state, the probability to emit a given symbol

#Create 2 initial state HMM
HMM_2state <- initHMM(States, Symbols, startProbs = start, transProbs = trans, emissionProbs = emission)

#Simulate sequence of length 100 with HMM
simHMM(HMM_2state, 100)

#Train HMM to find optimal parameter settings
obs <- unlist(getSequence(virus_seq))
HMM_2new <- baumWelch(HMM_2state, obs, 10, delta=1E-9)

#Create 4 state HMM
States <- c("H_G", "H_C","H_A", "H_T")
Symbols <- c("G", "C", "A", "T")
start <- c(.25, .25, .25, .25)
trans <- matrix(c(.8, .1, .05, .05, .1, .8, .05, .05, .05, .05, .8, .1, .05, .05, .1, .8), 4)
emission <- matrix(c(.8, .1, .05, .05, .1, .8, .05, .05, .05, .05, .8, .1, .05, .05, .1, .8), 4)

#Create HNMM
HMM_4state <- initHMM(States, Symbols, startProbs = start, transProbs = trans, emissionProbs = emission)

#Simulate sequence of length 100 with HMM
simHMM(HMM_4state, 100)

#Train HMM to find optimal parameter settings
obs <- unlist(getSequence(virus_seq))
HMM_4new <- baumWelch(HMM_4state, obs, 20, delta=1E-9)


```

Figure 2 – The code used to create the HMMs in Exercise 4.1

A.B Exercise 5.1

To get the MSA file, please visit the Github link above. The file can be found in the Exercises 4-5 and is called Ex5_1_MSA.fasta.