A Rudimentary Application of Profile Hidden Markov Models in Multiple Sequence Alignment

Allison Hui[1], **Ireanne Cao**[2], **and Irfan Azidan**[3]

[1]Computer Science, Junior, alh322
[2]Computer Science, Sophomore, ixc3
[3]Computer Science, Junior, ib262

ABSTRACT

This project presents a rudimentary reimagination of multiple sequence alignment (MSA) methodologies through the implementation of Profile Hidden Markov Models (HMMs). Traditional MSA techniques often grapple with limitations in handling insertions, deletions, and diverse sequence conservation, which are crucial for accurate biological interpretation. Our approach leverages the statistical robustness and flexibility of Profile HMMs to provide a more nuanced, probabilistic framework for sequence alignment. We intend to develop a novel alignment program that constructs and utilizes profile HMMs derived from input sequence data. This program is designed to model each column of an MSA as a distinct state within the HMM, effectively capturing the position-specific conservation patterns and accommodating variable-length insertions and deletions. This approach allows for a more accurate representation of the evolutionary relationships and structural nuances within the sequences. Our primary goal is to successfully implement a program that can construct a profile HMM from an input MSA and take in further sequences to align to the model. This basic implementation can then act as a springboard for us to explore the ways by which we may improve our program's performance by augmenting the profile HMM with various methods.

Keywords: multiple sequence alignment, hidden Markov model, protein sequence, probabilistic models, biological sequence analysis

Project type: re-implementation

Project repository: irfanznz/CS4775-final-proj

1 Introduction

Sequence alignment (SA) is a critical and highly researched process in bioinformatics. SA compares nucleotide and protein sequences by looking at their functional, structural, and evolutionary components. SA has the power to find conserved sequence motifs, evaluate the evolutionary differences of sequences, and even make predictions about the relationships between genes and species through a historical lens.

To thoroughly understand biological patterns of related sequences, the role of MSAs is crucial. Uncovering the function, evolution, and structure of closely similar sequences, MSA prompts a variety of analyses, from database searching to metagenomic analysis.

The sequence database consists of three sections: gene, protein, and comprehensive. When it comes to searching, FASTA and BLAST are the key components of in identifying sequences and studying their functions by matching sequence fragments against databases. In present times, BLAST is more widely used, for it incorporates a threshold for fragment matching and expansion. Matching new sequences to old sequences in the database is essential for analyzing their functions. MSA is also used in phylogenetic analysis, which looks closely at relationships of species and evolutionary transitions. The alignments obtained from MSA are applied when constructing phylogenetic trees, which give a summary of what species are related. Furthermore, MSA takes the lead in genomic, metagenomic, and protein analysis. With the 2020 surge of SARS-CoV-2, or COVID-19, researchers incorporated MSA and respective phylogenetic trees in genomic analysis to investigate the specific spike proteins that led to infection in humans. The examination of virus variants is also heavily reliant on the outcomes of MSA. With the help of base substitutions, deletions, variations, and single nucleotide polymorphisms, vaccine research is significantly enhanced with the ability to find evolutionary pathways and categorize variants. Expanding further, MSA aids metagenomic analysis in studying the genome of all microbial communities living in a certain environment by looking at genetic

relationships between organisms. For example, the closest related disease to COVID-19, RmYN02, was discovered in this way with metagenomes of 227 bat samples. RmYN02 was found to have 93.3 percent nucleotide homology with SARS-SoV-2. [1] Metagenomic analysis is frequently combined with other MSA-influenced algorithms, including k-mers and Markov model. [2] Lastly, MSA is employed in predicting protein structure. By identifying conserved regions and motifs, scientists can make educated guesses about the structure of a protein. This is frequently combined with deep learning, specifically natural language processing, aiming to improve prediction accuracy.

SA aligns the most similar regions of each sequence and equalizes the lengths of the aligned sequences by inserting gaps as necessary. SA can be divided into global SA (end-to-end alignment of entire sequences) and local SA (alignment of sub-regions of sequences) based on the range of alignment. Conversely, SA can be divided based on the number of sequences to be aligned, into pairwise sequence alignment (PSA), which has only two sequences to be aligned, or MSA, which involves more than two sequences to be aligned. There already exist algorithms for PSA that are able to exactly align the sequences as optimally as possible using dynamic programming and backtracing methods, such as the Needleman-Wunsch algorithm (for global SA) and the Smith-Waterman algorithm (for local SA). However, MSA is much more difficult than PSA and has been proven to be a NP-hard problem [3]. This means that there is no known efficient algorithm to find the exact solution for MSA, and as such, current MSA algorithms either have trade-offs in accuracy or efficiency. [4]

Profile HMM based methods for MSA offer significant advantages over traditional alignment techniques, particularly in their ability to handle sequence variability and complexity with high accuracy. Unlike progressive or pairwise alignment methods that can struggle with insertions, deletions, and divergent regions, profile HMMs capture the position-specific conservation and variability within a sequence family. This probabilistic approach allows for a more nuanced representation of alignments, effectively accommodating gaps and varying sequence lengths. This is especially crucial when aligning distantly related sequences or sequences with complex evolutionary histories. Additionally, profile HMMs can integrate additional biological information, such as structural or functional annotations, enhancing the biological relevance of the alignments. This robustness of profile HMMs has intrigued us and provided the primary motivation for our project.

2 Methods

2.1 Hidden Markov Models

HMMs are a class of statistical models used for modeling stochastic processes that are assumed to have the Markov property, that is, the future of the states of the processes only depend on the current state and not the ones that preceded it. In other words, the future is independent of the past, given the present. More formally,

$$P(X_{t_n} = x | X_{t_1} = x_1, X_{t_2} = x_2, \ldots, X_{tn-1} = x_{n-1}) = P(X_{t_n} = x | X_{t_{n-1}} = x_{n-1}).$$

where X_t is a random variable at time t.

An HMM can be defined by the following:

The set of states, $S = \{\pi_1, \pi_2, \ldots, \pi_N\}$
The set of observations, $\Sigma = \{x_1, x_2, \ldots, x_N\}$
The transition probabilities, $\mathbf{A}$
The emission probabilities, $\mathbf{E}$
The initial state probabilities, $\mathbf{a_0}$

An example of a stochastic process that can be modeled with an HMM is the following: a signal marks when a pedestrian should stop or cross the road. If we suppose the signal, being one of "stop" or "go", is hidden from us and that the only thing we can observe is whether a pedestrian crosses the road or not, the state of the signal corresponds to the hidden states in the HMM and the observation of a pedestrian crossing the road or not corresponds to the observations in the HMM. The initial probabilities tell us how likely it is that we start with a hidden state. The example can be visualized as in Figure 1.

2.2 Profile Hidden Markov Models [5]

Profile HMMs are named after standard profiles which are used to model sequence families. Standard profiles such as Position-Specific Scoring Matrices (PSSM) can be constructed from the frequencies of amino acids at each alignment column. PSSMs, however, do not model insertions and deletions as well as being unable to encapsulate variable-length sequences due to the strict position-based modeling. Profile HMMs provide a framework in which insertions and deletions are robustly modeled and hence allow for variable-length sequence analyses.

Similar to PSSMs, we can construct a profile HMM from an MSA. The question that is raised then is how a profile HMM should look like.

2.2.1 HMM Topology

A naive approach would be to model the sequences as a simple HMM with one "match" state for each alignment column. Transitions are trivially modeled with probability 1 of transitioning from one state to the next; this is due to the HMM having the same number of states as there are alignment columns in our MSA. The only nontrivial parameters then are the emission probabilities, which can be straightforwardly calculated from the frequencies of symbols (in the example, amino acids) at each alignment column. Lastly, since biological sequences are bounded sequences, we also need to include the begin and end states.

Given the following MSA,

```
VGA-EY
V---EV
VKG--D
IAGDGY
```

we can obtain a profile HMM as that in Figure 2.

The model evidently needs to be revised to handle insertions–as of now, it cannot represent anything other than sequences of length of the fixed number of alignment columns. We can add insertion states I_t for every match state M_t for a profile HMM with t match states. Note that a match-insert transition at time t occurs as $M_t \to I_t$, and an insert-match transition as $I_t \to M_{t+1}$. Furthermore, we can add self-transitions for the insertion states so as to provision different probabilities for entering the insertion state and staying in it. This effectively models affine gap penalties.

Emission probabilities for insertion states are usually taken from background frequencies of residues. Background frequencies are obtained from representative sets of sequences for the sequence that one is studying; for example, the background frequency of nucleotides in human DNA can be obtained from a broad analysis of the human genome. The choice of background frequencies can be tuned for desired properties of the profile HMM, such as detecting more distant homologs. Furthermore, each insertion state can have different emission probabilities, allowing for position-specific choices for the background probabilities or any other choice of residue emission probabilities. Indeed, this hints at the strengths of a profile HMM-based multiple alignment program over other methods.

As we iterate through the alignment columns in the source MSA, the problem of distinguishing between match and insertion states arises. This is important as the "length" of our profile HMM depends on the number of match states. For now, we can a heuristic of taking columns half of which is comprised of gaps as insertion states. This is part of the parameterization process of the HMM–the number match states is a hyperparameter one can tune to obtain differently performing models. Applying this to our example MSA, we then obtain a topology shown in Figure 3.

We also need a way to represent deletions. One such way would be to add a transition from any match state to any other match state following itself. This would allow for "skips" that model deletions, as seen in Figure 4.

However, it is clear from this small example that such a representation does not scale very well if we wish to preserve simplicity. The introduction of non-linear jumps would complicate the model, particularly in determining transition probabilities, as it increases in size. Therefore, we could instead opt for the addition of "delete" states in a similar fashion to the insertion states, as in Figure 5. Deletion states do not emit any symbols and do not have emission probabilities.

In all, we now have a profile HMM with a topology shown in Figure 6.

2.2.2 HMM Parameterization

With the general topology of the model, we now move on to obtaining the parameters of the profile HMM: the length of the model and the transition and emission probabilities. The length of the model can be obtained by classifying columns into match and insert states and taking the number of insert states. In our implementation, we have used a simple heuristic of taking columns with more than half gaps to be insertions. Though this simple heuristic is not unfavorable, more optimal models can be determined using algorithms such as the MAP match-insert assignment. The number of match states will determine the final topology of the model.

The transition probabilities can be determined using the counts of transitions between states in the MSA once match, insert, and delete states have been assigned. Gaps corresponding to match columns are delete states while those in insertion columns are ignored as padding. However, especially for small MSAs, not all transitions present in the profile HMM topology are observed. This can cause zero probability issues where some states present in the HMM topology are impossible to reach. This causes poor generalization and is detrimental to the performance of the model. Hence, absent transitions are given a pseudocount to rectify this issue.

The emission probabilities differ for match and insertion states. For match states, one can simply calculate the probability of emission for each residue from the residue frequencies. Similar to before, we add pseudocounts for absent residues. For insertion states, emission probabilities are often based on background frequencies of amino acids; however, the nature of the profile HMM allows for individual variation of insertion states for the emission probabilities. In our implementation, we used a single emission distribution for all insertion states.

Once the parameters of the HMM are obtained, the model can now be applied to uses such as multiple sequence alignment.

2.3 Multiple Sequence Alignment with profile HMMs [5]

If we have another sequence or more, we can "align" each of the sequences to the profile HMM by taking the sequence of states π^* that maximizes the conditional probability $P(\pi|\mathbf{x})$ where π is some state path in the profile HMM and $\mathbf{x}$ is the observations in the sequence we are aligning. In short, we are finding the Viterbi path of the profile HMM given the input sequence. Using the Viterbi algorithm to find the best state path, the input sequence can modified to align with the profile HMM.

If we have a sequence `MENACING` that we wish to align to our profile HMM (working in log probabilities), the Viterbi matrix would be initialized similar to the following:

$$\begin{bmatrix} . & \hat{} & \texttt{M} & \texttt{E} & \texttt{N} & \texttt{A} & \texttt{C} & \texttt{I} & \texttt{N} & \texttt{G} \\ B & 0 & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty \\ I_0 & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty \\ M_1 & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty \\ I_1 & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty \\ D_1 & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty \\ M_2 & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty \\ I_2 & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty \\ D_2 & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty \\ \vdots & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty \\ E & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty \end{bmatrix}$$

A start character has been added and the probability of being in the begin state at the character's column is set to 0; the matrix is initialized to a very small number otherwise. As we progress through the Viterbi algorithm, the matrix will fill up with probabilities. Upon termination, we can find the largest probability and the state associated with it and start constructing the path from a backpointer matrix–another matrix with a similar topology as the one above but that which stores the likeliest path that has led to all of the states for all of the residues.

2.4 Our Implementation

We implemented a program for multiple sequence alignment in Python. A multiple sequence alignment to construct a profile HMM is passed in as an input along with sequences to be aligned to the constructed profile HMM. The program then produces an alignment. The full implementation can be found in the project repository.

3 Results

We were able to successfully construct a profile HMM from any multiple sequence alignment. As an example, the following MSA [5] produces the transitions and emissions matrices as shown in figures 7 and 8 using a pseudocount of 1.

```
VGA--HAGEY
V----NVDEV
VEA--DVAGH
VKG------D
VYS--TYETS
FNA--NIPKH
IAGADNGAGV
```

We were unable to implement the Viterbi algorithm correctly. Upon further testing, our initial Viterbi implementation did not scale to larger sequences and sequences of varying lengths. We have since revised the algorithm several times to no avail; the time constraint of this endeavor has prohibited us from converging to a solution. As such, were unable to produce sufficiently correct alignments for use in benchmarking.

4 Discussion

4.1 Future Directions

Algorithm Correctness We would like to fully implement the Viterbi algorithm so as to complete the MSA program. We did not anticipate the incorrectness of our initial implementation and the time and complexity of debugging, therefore we could not complete what we set out to do. We would still like to see it to the end and implement a full multiple sequence alignment program by fixing our Viterbi algorithm implementation.

Benchmarking Benchmarking would be the next step after fully implementing the program. Using benchmarking data and tools such as BAliBase [6], we can measure our program's performance quantitatively and also compare it with that of other sequence alignment programs.

Optimization Our first angle in optimization would be computational optimization. We have made little use of Python libraries that could significantly increase computational performance, nor have we regarded in detail the algorithmic complexity of our code. Multiple sequence alignment is an important step in biological sequence analysis and often involves a large amount of data; therefore, reducing computational overhead is paramount for a multiple sequence alignment program.Secondly, the model itself could be improved. For example, using the Forward-Backward algorithm, we could compute a posterior distribution associated with the profile HMM to better fit input sequences. This method can also be used to construct profile HMMs from unaligned sequences, which could also be a future direction for our program. The selection of match and insert states, as mentioned before, can be optimized with the MAP construction algorithm. Parameters such as pseudocounts and background probabilities for insert states also provide ample room for model improvement, as well as more complex processes such as model surgery. In all, there are a lot of avenues for improvement to fit specific needs of our program.

4.2 Other Multiple Sequence Alignment Methods

There are five general types of widely used MSA algorithms: progressive alignment, iterative, heuristic, machine learning and divide-and-conquer.

Progressive alignment is a less complex algorithm that trades accuracy for speed and simplicity. The first step is to construct a guide tree, or a clustering of the sequences produced by clustering methods such as neighbor-joining and used as a base for the MSA. This step has generally been the limiting factor in creating large alignments using progressive alignment algorithms. The second step is to gradually construct the MSA based on the guide tree, combining pairwise alignments from the most to least similar pairs. The main issue with progressive alignment is that any potential errors are propagated, leading to less accurate results.

To address this issue, iterative algorithms continuously realign previously combined sequences as well as growing the MSA with new sequences. Iterative methods are certainly more accurate since they repeatedly re-optimize the alignment in each iteration until the accuracy converges, and they are robust regardless of the number of sequences. However, a disadvantage is that iterative re-optimization may converge to local optima rather than an improved global optimum.

A heuristic algorithm is a stochastic iterative algorithm. The core basis of a heuristic algorithm is to compute feasible solutions within improved bounds of time and space complexity, so heuristic algorithms are suitable for solving NP-hard problems such as MSA. However, the results of a heuristic algorithm may vary due to randomness, as opposed to a traditional deterministic iterative algorithm. Heuristic algorithms for MSA include simulated annealing, genetic algorithm, swarm intelligence, and hidden Markov models (HMMs). The first three algorithms solve the MSA problem by simulating MSA as other processes, which respectively are solid annealing, biological evolution, and natural behavior of various biological groups.

Application of machine learning to solving MSA is still relatively new, but shows promise based on demonstrably improved performance of some models [7]. This method uses similar techniques to progressive alignment and then applies reinforcement learning to build the optimal alignment. Recent developments improved the time required for the alignments to converge and accuracy of global alignment. However, some challenges to ML applications to MSA include lack of optimal samples for training models, no suitable loss function, and how many possible lengths and quantities of sequences there may be.

Divide-and-conquer is a widely used method for MSA, which produces multiple independent sub-problems of smaller sizes from an initial group of sequences, and then splices the results. The performance of divide-and-conquer methods have shown significant improvement over older methods, and can be combined with other methods since the sub-problems are independent and can be solved as such. [4]

The combination of some of these methods with profile HMMs in an ensemble multiple sequence alignment program is an interesting notion that we would like to explore in the future.

Author Contributions

The nature of our project required close contribution between group members, as such, little delegation or separation of tasks have been made.

- Study design: Allison, Ireanne, Irfan
- Coding: Allison, Ireanne, Irfan
- Experiments: Allison, Ireanne, Irfan
- Analyses: Allison, Ireanne, Irfan
- Writing:
  - *Introduction:* Allison, Ireanne, Irfan
  - *Methods:* Allison, Ireanne, Irfan
  - *Results:* Allison, Ireanne, Irfan
  - *Discussion:* Allison, Ireanne, Irfan

References

1. Zhou H, Chen X, Hu T, Li J, Song H, Liu Y, Wang P, Liu D, Yang J, Holmes EC, *et al.*, 2020. A novel bat coronavirus closely related to sars-cov-2 contains natural insertions at the s1/s2 cleavage site of the spike protein. *Current Biology*, 30(11).

2. Storato D, Comin M, 2021. K2mem: Discovering discriminative k-mers from sequencing data for metagenomic reads classification. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 19:220–229.

3. Caucchiolo A, Cicalese F, 2023. Hardness and approximation of multiple sequence alignment with column score. *Theoretical Computer Science*, 946(113683).

4. Zhang Y, Zhang Q, Zhou J, Zou Q, 2022. A survey on the algorithm and development of multiple sequence alignment. *Briefings in Bioinformatics*, 23(3):bbac069.

5. Durbin R, Eddy SR, Krogh A, Mitchison G, 1998. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press.

6. Thompson JD, Plewniak F, Poch O, 1999. BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics*, 15(1):87–88.

7. Kinattinkara Ramakrishnan R, Singh J, Blanchette M, 2018. Rlalign: A reinforcement learning approach for multiple sequence alignment. In *2018 IEEE 18th International Conference on Bioinformatics and Bioengineering (BIBE)*, pages 61–66.

Figures

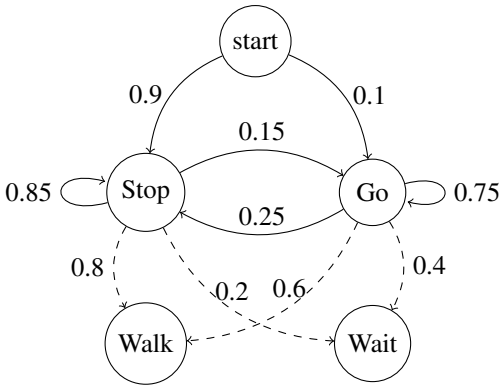

Figure 1. A graphical representation of the example HMM.

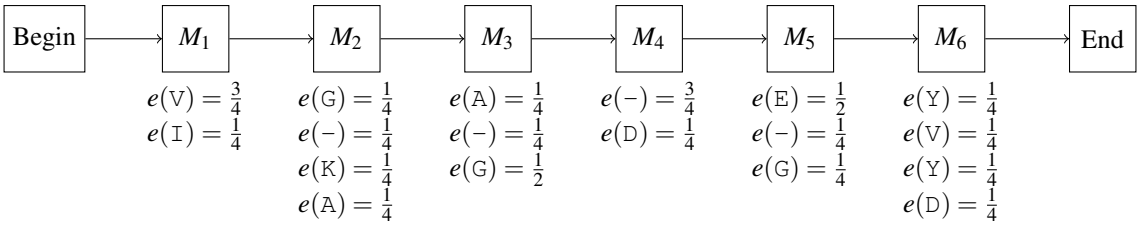

Figure 2. A basic topology of a profile HMM based on the given multiple sequence alignment. The emission probabilities of the symbols are shown under the corresponding match states.

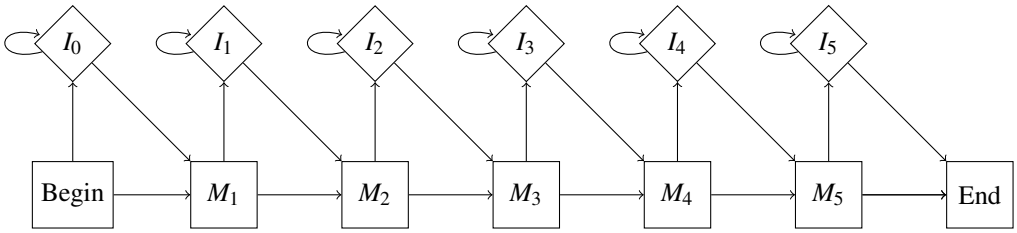

Figure 3. A profile HMM topology with insertions modeled.

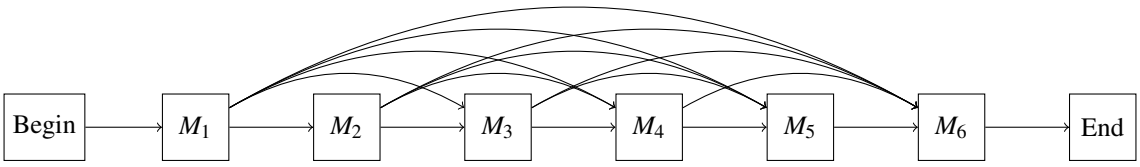

Figure 4. A profile HMM topology with skips to model deletions.

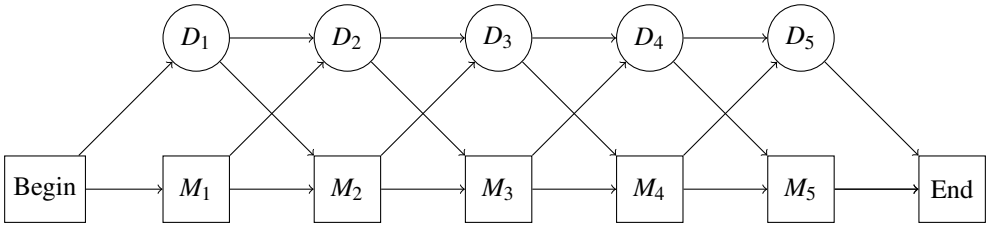

Figure 5. A profile HMM topology with deletion states to model deletions.

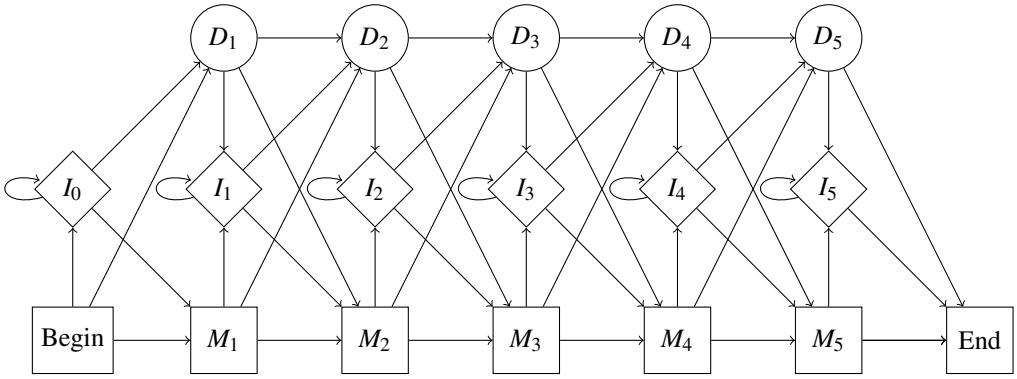

Figure 6. The complete topology of the profile HMM based on the example multiple sequence alignment.

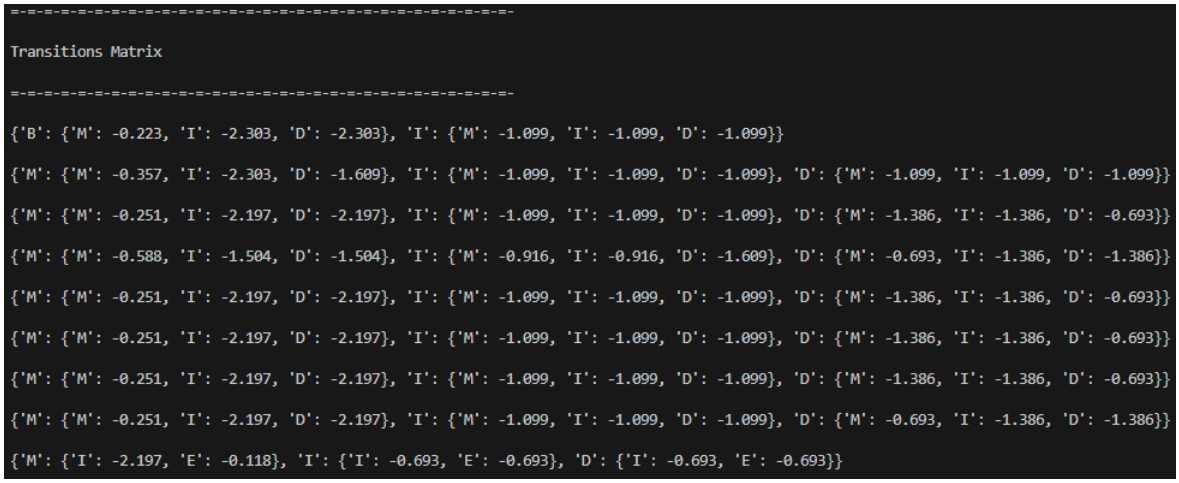

Figure 7. The transitions matrix.

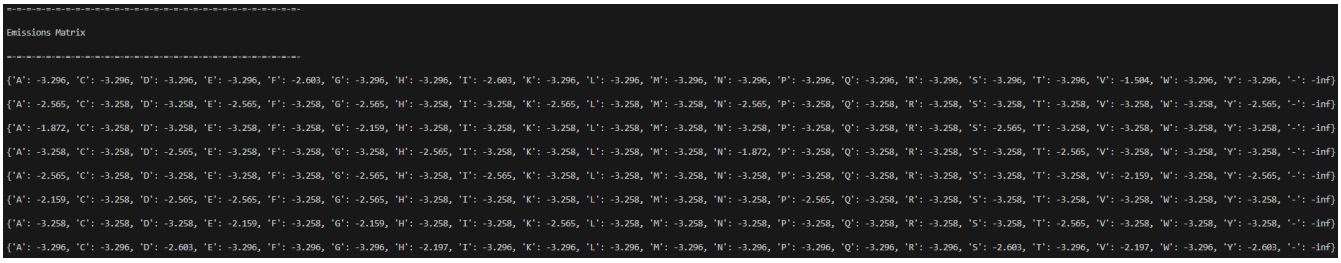

Figure 8. The emissions matrix.