CSI5126. Algorithms in bioinformatics

Hidden Markov Models

Marcel Turcotte

School of Electrical Engineering and Computer Science (EECS)
University of Ottawa

Version October 31, 2018

Summary

This module is about **Hidden Markov Models**.

General objective

- **Describe** in your own words Hidden Markov Models.
- **Explain** the **decoding**, **likelihood**, and **parameter estimation** problems.

Reading

- A. Krogh (1998) An introduction to hidden Markov Models for biological sequences. In S.L. Salzberg, D.B. Searls, S. Kasif (Eds.), *Computational Methods in Molecular Biology*, Elsevier Science. §4, 45–63.
- Pavel A. Pevzner and Phillip Compeau (2018) *Bioinformatics Algorithms: An Active Learning Approach*. Active Learning Publishers.
  `http://bioinformaticsalgorithms.com`
  Chapter 10.
- Yoon, B.-J. Hidden Markov Models and their Applications in Biological Sequence Analysis. *Curr. Genomics* **10**, 402–415 (2009).
- A. Krogh, R. M. Durbin, and S. Eddy (1998) *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids.* Cambridge University Press.

Plan

1. Introduction
2. Motivational example
3. Formal definitions
4. Worked example
5. Applications

Introduction

- **Twilight** zone (database search)
- **Gene** finding
- Indentifying **transmembrane** proteins

Modeling biological sequences

- Sequence alignment techniques, such as Needleman & Wunsch or Smith & Waterman, assume that positions along the sequence are independent and **identically distributed** (i.i.d.):
  - Indeed, the same substitution matrix (PAM250, BLOSUM62, etc.) is used for weighting all the substitutions of an alignment;
  - Clearly, anyone looking at a multiple sequence alignment can see that the amino acid distribution varies greatly from one position to another. Some positions are clearly biased towards hydrophobic, charged or aromatic residues, for example.

Modeling biological sequences (cont.)

- **Regular expressions** (RE) can be used to model these variations, `[FAMILY][KREND][ILV][PG] ... [ST]`. However, REs can be **too rigid**. Being deterministic, a sequence either match or not a regular expression.

⇒ **Probabilistic motifs**, in particular Hidden Markov Models (HMMs), elegantly combine the advantages of these two approaches.

Motivational **example**

Based on:
A. Krogh (1998) An introduction to hidden Markov Models for biological sequences. In S.L. Salzberg, D.B. Searls, S. Kasif (Eds.), *Computational Methods in Molecular Biology*, Elsevier Science. §4, 45–63.

Motivational **example**

Consider the following aligned DNA sequences.

```
ACA---ATG
TCAACTATC
ACAC--AGC
AGA---ATC
ACCG--ATC
```

A regular expression representing the above motif could be:

```
[AT][CG][AC][ACGT]*A[GT][CG]
```

The expression matches all 5 sequences, the shortest possible sequence is of length 6, and a match must have an A three positions from its end.

Motivational **example** (contd)

Consider the following aligned DNA sequences.

```
ACA---ATG
TCAACTATC
ACAC--AGC
AGA---ATC
ACCG--ATC
```

Which of the following two sequences is the **least likely** to be a member of the above family and **why**?

```
TGCT--AGG
ACAC--ATC
```

Motivational **example** (contd)

```
[AT][CG][AC][ACGT]*A[GT][CG]
```

First of all, both sequences are recognized by the above RE!

```
TGCT--AGG
ACAC--ATC
```

Therefore, both sequences are good candidate for being a member of this family.

Regular expressions are deterministic: a sequence is a member of the family or not! In itself, this formalism does not provide information for **ranking** the sequences.

Motivational **example** (contd)

Consider the following aligned DNA sequences.

```
ACA---ATG
TCAACTATC
ACAC--AGC
AGA---ATC
ACCG--ATC
```

```
TGCT--AGG (least likely)
ACAC--ATC (most likely)
```

However, notice that the top sequence has been constructed by selecting the "**least likely**" symbol at each position (i.e. the one that appears only once in that column), whilst the second one has been constructed by selecting the "**most likely**" nucleotide at each position, it is therefore a consensus sequence.

Motivational **example** (contd)

A natural way to score a match would be to use the frequencies of occurrence at each position of the motif as estimates of the probabilities of occurrence.

For the first sequence, this would this means

$$\frac{1}{5} \times \frac{1}{5} \times \frac{1}{5} \times \ldots$$

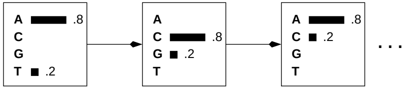
As for the second one, its probability would be

$$\frac{4}{5} \times \frac{4}{5} \times \frac{4}{5} \times \ldots$$

After the third position, our calculation has to take into account insertions and a diagram would be useful.

Motivational example (contd)

```
A C A - - - A T G
T C A A C T A T C
A C A C - - A G C
A G A - - - A T C
A C C G - - A T C
```



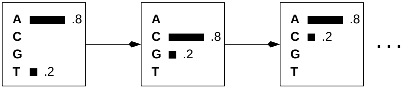
Let's create a diagram to represent the sequence alignment. Each **conserved column** of the alignment (i.e. each column that has no gaps) is associated with a **box**, called a (**match**) **state**.

A state **emits a symbol** with a certain probability.

Finite state machines that produce an output for each state are called **Moore** machines.

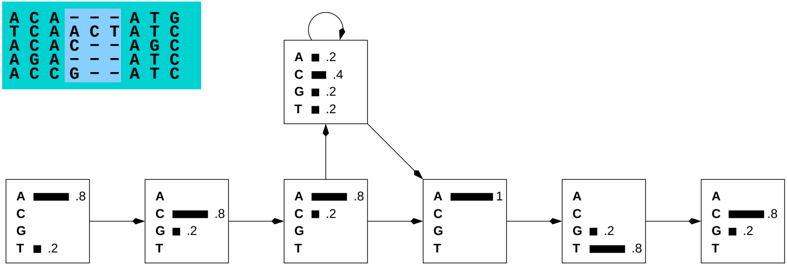
Motivational **example** (contd)

```
A C A - - - A T G
T C A A C T A T C
A C A C - - A G C
A G A - - - A T C
A C C G - - A T C
```



After the third position, sequence 1 and 4 have no insertion at all, in terms of the regular expression `[ACGT]*` does not match any position, sequence 3 and 5 have one insertion, match `[ACGT]*` once and, finally, sequence 2 matches `[ACGT]*` three times.

Motivational **example** (contd)



```
A C A - - - A T G
T C A A C T A T C
A C A C - - A G C
A G A - - - A T C
A C C G - - A T C
```

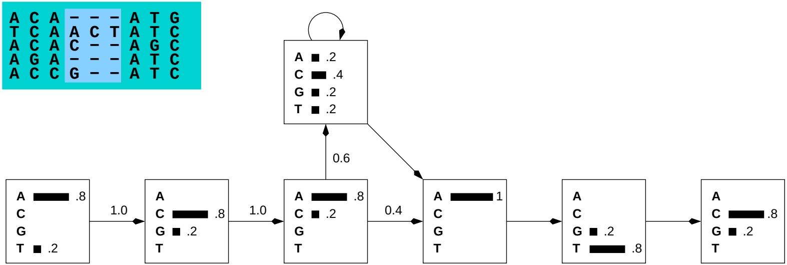
Let’s create a new node (state) to model `[ACGT]*`.

Sequences 1 and 4 do not need to visit that state. Sequences 3 and 5 visit this state once, whilst the second sequence visits the state three times.

Make sure to understand how the (emission) probabilities for that state are computed.

Motivational **example** (contd)

```
A C A - - - A T G
T C A A C T A T C
A C A C - - A G C
A G A - - - A T C
A C C G - - A T C
```

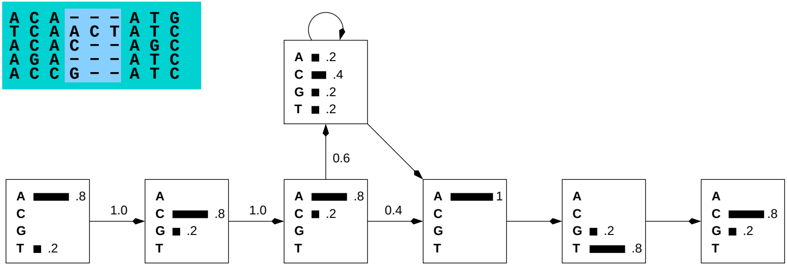


Transitions 1 → 2, and 2 → 3 occur with probability 1.0.

2 out of 5 sequences do not visit state 4 and are going directly from state 3 to state 5. Let's assign a (transition) probability $\frac{2}{5}$ to that edge, and $\frac{3}{5}$ for the other outgoing edge so that the sum of all the probabilities is 1.

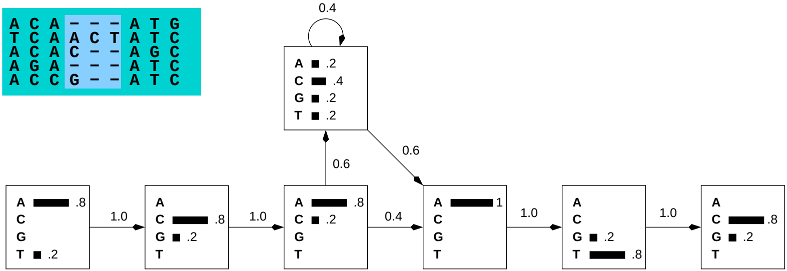
Motivational **example** (contd)

```
A C A - - - A T G
T C A A C T A T C
A C A C - - A G C
A G A - - - A T C
A C C G - - A T C
```



Once in state 4, 5 events occur before state 5 is reached. Sequences 3 and 5 are making a transition immediately to state 5 after the C an G has been emitted/matched. In the case of sequence 2, after the first A has been matched, two transitions to state 4 are made, matching C and T, before the transition to state 5 is made. Therefore, 2 out of 5 transitions are made to state 4 and 3 to state 5.

Motivational **example** (contd)



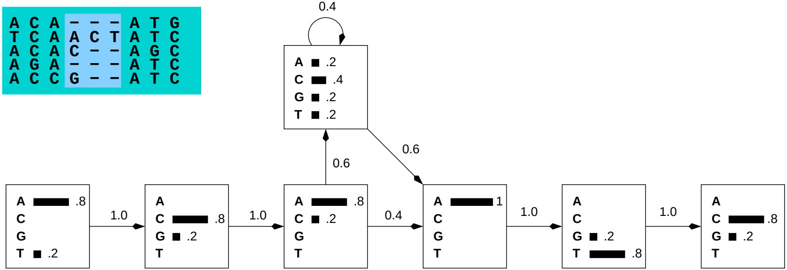
```
A C A - - - A T G
T C A A C T A T C
A C A C - - A G C
A G A - - - A T C
A C C G - - A T C
```

Finally, transitions from states 5 to 6, and 6 to 7 occur with probability 1.

This is the basic idea behind Hidden Markov Models (HMMs), as applied to model sequence motifs.

Motivational **example** (contd)

```
A C A - - - A T G
T C A A C T A T C
A C A C - - A G C
A G A - - - A T C
A C C G - - A T C
```



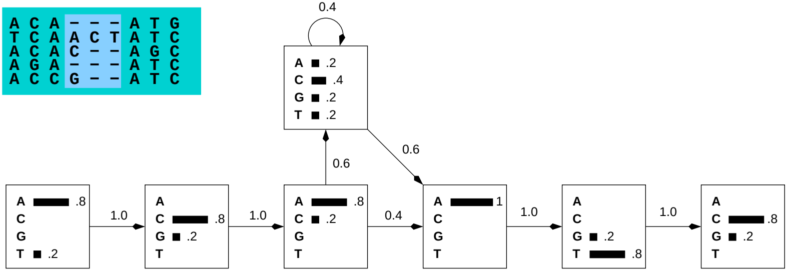
It's now easy to score the probability of any sequence.

In particular the consensus sequence,

$$
\begin{aligned}
P(ACACATC) &= 0.8 \times 1 \times 0.8 \times 1 \times 0.8 \times 0.6 \times 0.4 \times 0.6 \times 1 \times 1 \times 0.8 \times 1 \times 0.8 \\
&= 0.0472
\end{aligned}
$$

Motivational example (contd)

```
A C A - - - A T G
T C A A C T A T C
A C A C - - A G C
A G A - - - A T C
A C C G - - A T C
```

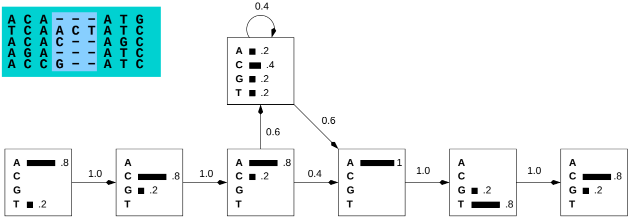


And, the exceptional one,

$$\begin{aligned} P(TGCTAGG) &= 0.2 \times 1 \times 0.2 \times 1 \times 0.2 \times 0.6 \times 0.2 \times 0.6 \times 1 \times 1 \times 0.2 \times 1 \times 0.2 \\ &= 0.000023 \end{aligned}$$

The **consensus** sequence is 2,052 times more likely than the **exceptional** one.

Remarks



```
A C A - - - A T G
T C A A C T A T C
A C A C - - A G C
A G A - - - A T C
A C C G - - A T C
```

We used to assume that all the positions are identically distributed, not anymore!

We used to model the length the gaps only, now distribution of the symbols is also modeled.

There is one small problem to be fixed, the computed probability highly depends on the sequence length (number of times the insertion state is used).

Length dependency

To remove the length dependency of the score, the probability of a sequence given the model just described is compared to the probability of that sequence given **a random (NULL) model**, as usual it's convenient to express the ratio as a **log-odds score**. For our random model, let's assume that nucleotides are equiprobable, for a sequence S of length n, the log-odds score becomes,

$$\log_2 \frac{P(S|M)}{P(S|R)} = \log_2 \frac{P(S|M)}{0.25^n}$$

Since the two models have the same length and that the log of a

product is the sum of logs, each probability value in the hidden Markov model can be transformed to a log-odd score, in our case dividing the probability values by 0.25 or using the probability estimates from actual data. This would also help avoiding underflow problems.

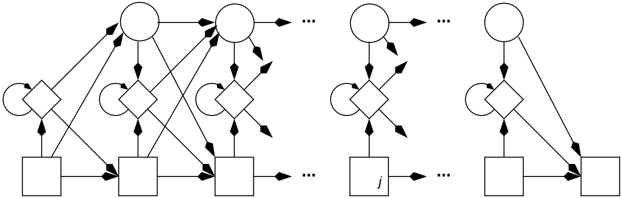
Length dependency (contd)

	S	$P(S\|M) \times 100$	Log-odds
Consensus	`ACAC--ATC`	4.7	6.7
Sequence motifs	`ACA---ATG`	3.3	4.9
	`TCAACTATC`	0.0075	3.0
	`ACAC--AGC`	1.2	5.3
	`AGA---ATC`	3.3	4.9
	`ACCG--ATC`	0.59	4.6
Exception	`TGCT--AGG`	0.0023	-0.97

Notice that two matches cannot be compared in the probability space, because of the length dependency, consider the scores for the second sequence and the exceptional one, their raw probability scores are almost identical but their log-odd scores are quite different.

In the case of the exceptional sequence, its log-odd score is negative, indicating that the **null model is a better fit**.

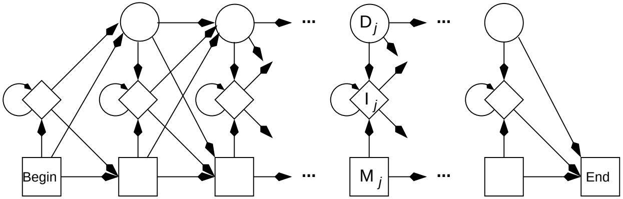
Profile-**HMM**



There is a particular type of HMM that is often used to model families of sequences they are called profile-HMMs.

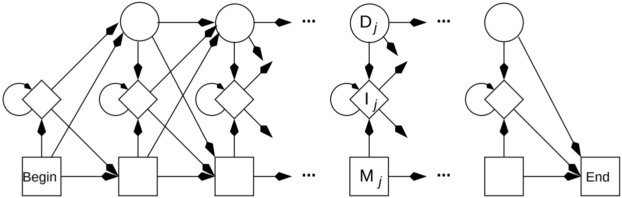
They resemble normal sequence profiles and allow to **model insertions and deletions in a position specific manner**.

Profile-**HMM**



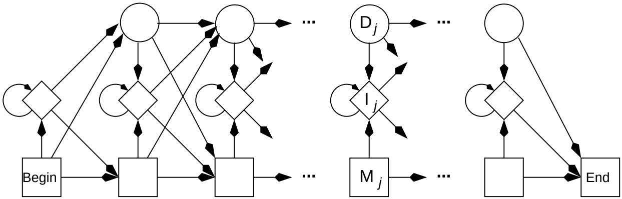
The bottom nodes are called **main** or **match states**, each M_j corresponds to a particular column in a multiple sequence alignment, **the probability distribution at that node corresponds to the probability distribution of the column it models in the alignment**.

Profile-HMM



The diamond shaped nodes are called **insertion states**, noted I_j, they allow to model variable regions, the amino acid probability distribution at those nodes could be set to the overall probability distribution of amino acids, for example.

Profile-**HMM**

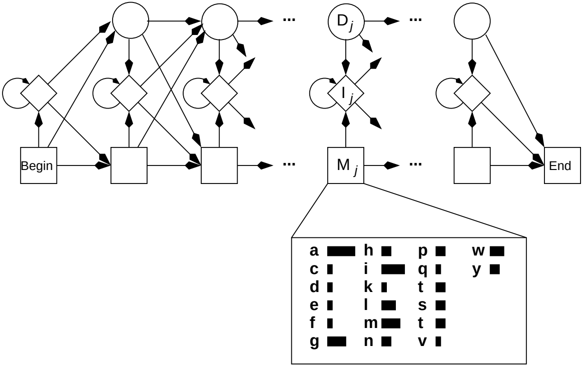


Finally, the round nodes are called **delete** or **silent states**, noted D_j, they allow to model deletions, i.e. to skip certain columns of the alignment.

As you can see, insertions and deletions are modelled separately.

Also, their respective probabilities are allowed to vary along the profile.

Profile-HMM



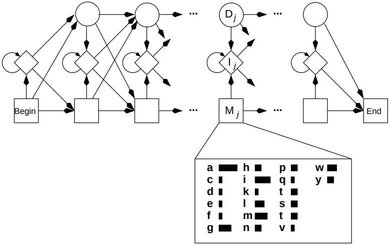
⇒ Emission probabilities are now associated with each M_j, in the case of profile HMMs they correspond to a column in a profile (alignment).

Remarks

The Profile-HMM topology (states and interconnections) is specific to bioinformatics.

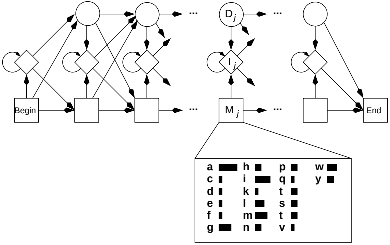
In bioinformatics, many other topologies are used, including specific topologies for modeling eukaryotic **gene structures** (exons and introns), **sequence alignments** and **trans-membrane proteins**. Examples will be seen later.

An HMM can be seen as a **generative model**



⇒ Starting from the **begin** state, move to an adjacent state, *i*, according to some **transition probability distribution**, emit a symbol according to the **emission probability distribution** of that state, move to an adjacent state, again according to the transition probabilities, repeat until the **end** state has been reached.

What's **hidden**?



Seen as a generative model, at each step this abstract machine moves to a new state and produces a symbol.
The observer only sees the sequence of symbols; not the sequence of state transitions, which are hidden.

What is Markovian?

References

Pensez-y!

L’impression de ces notes n’est probablement pas nécessaire!