COSC 348:
Computing for Bioinformatics

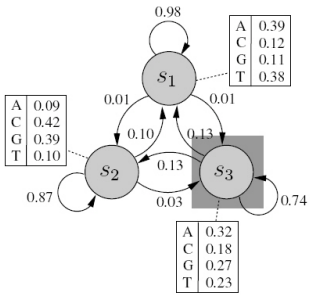
Lecture 10:
Hidden Markov models: applications in bioinformatics

Lubica Benuskova

http://www.cs.otago.ac.nz/cosc348/

Example: 3-state HMM of DNA

- A 3-state HMM, with observation probabilities associated with each state and state transition probabilities.
- We can use the HMM to generate and score a new sequence **X**
- But how do we know number of states and all the probabilities?
- *Training of HMM*: if we are given *n* aligned sequences we can infer the underlying HMM from them.



$\mathbf{X}$ = TAACGGCAGA...

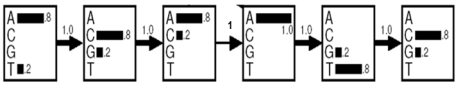
Inference/training of HMM based on alignment

```
1) A C A A T G
2) T C A A T C
3) A C A A G C
4) A G A A T C
5) A C C A T C
```

First we perform global alignment of *n* sequences, we assume there are as many states as letters

Observation probability of each letter at a given position is derived from the frequency. If these frequencies are the same at several positions, then we can collapse two or more states into one.

Transition probability: in our simple case $P(X_t \mid X_{t-1}) = 1.0$



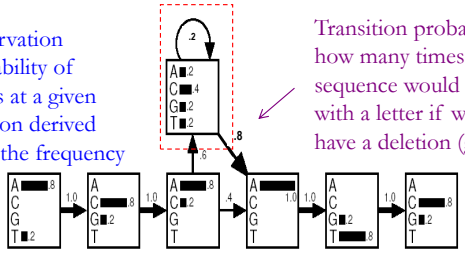
Inference/training of HMM based on alignment

```
1) A C A - - - A T G
2) T C A A C T A T C
3) A C A C - - A G C
4) A G A - - - A T C
5) A C C G - - A T C
```

First we perform global alignment of *n* sequences, we assume there are as many states as letters + the state that represents gaps

Observation probability of letters at a given position derived from the frequency

Transition probability: how many times the sequence would continue with a letter if we did not have a deletion (gap)

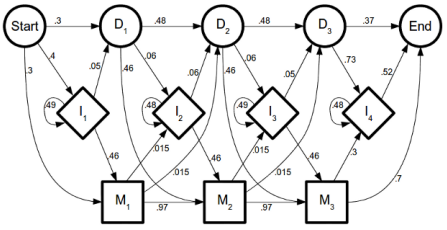


Inference/training of HMM for biosequences

- We choose & align particular set of *n* **related** training sequences.
- Initial number of M states equals number of times there is a letter (any letter) at a given position in **all** aligned sequences.
  - We fill in the I and D states according to initial number of M states.
- Then we estimate the symbol emission probabilities in each M & I state from a set of training sequences by observing the number of times each emission occurs in the training set and dividing by the *n*.
  - If emission probabilities for 2 or more states are the same, then we merge them into one hidden state. We re-calculate the number of I and D states accordingly.
- Then we estimate all state-to-state transition probabilities from a set of training sequences by observing the number of times each transition occurs in the training set and dividing by the *n*.

Example of trained HMM for DNA

- **Left:** state transition model; **Right:** emission model for M and I states.
- We have 4 insertion states, 3 match states and 3 deletion states (# of D and I states depends on # of M states).



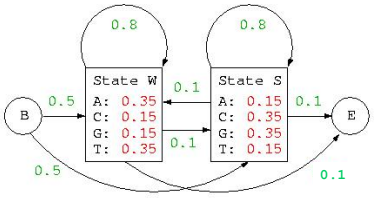
	A	C	G	T
M1	0.10	0.80	0.04	0.06
M2	0.40	0.55	0.04	0.01
M3	0.73	0.18	0.02	0.07
I1	0.15	0.29	0.33	0.23
I2	0.44	0.51	0.02	0.03
I3	0.70	0.08	0.19	0.03
I4	0.01	0.01	0.08	0.90

Applications of HMM in bioinformatics

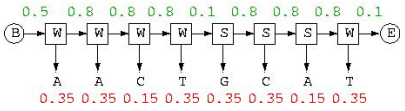
- Once a HMM has been successfully derived from a family of sequences, it can be used for a number of tasks, including
  - Multiple alignments
  - Database mining and classification of sequences
  - Structural analysis and pattern discovery
- All these tasks are based on computation of any given sequence,
  - of its probability (i.e. score) according to derived HMM,
  - Computation/inference of the most likely path of states and
  - on the analysis of the HMM structure itself.

Analysis of HMM

- The desired or ideal HMM is a ***minimal*** model against which all the sequences in the training set will have the highest scores compared to if they were generated by any other HMMs.



Score = product of concrete emission probabilities



Inference of the most likely state path

- **Most likely state sequence**: given all evidence to date, we want to find the sequence of states that is most likely to have generated all the evidence up to date, i.e. argmax $_{\mathbf{x}_{1:t}} P(\mathbf{x}_{1:t} \mid \mathbf{e}_{1:t})$.
  - In the weather example, if it rained on each of the first three days and it does not rain on the fourth day, then the most likely explanation is that the atmospheric pressure was low on the first three days and was high on the fourth.
  - Algorithms for this task are useful in many applications, including speech recognition, i.e. to find the most likely sequence of words, given utterance, or **the reconstruction of state sequences in bioinformatics,** that is to infer the most likely sequence of abstract states from a concrete sequence of letters.

Viterbi formula for the most likely path

- Let us denote by $\mathbf{m}_{1:t}$ the probability of the ***best*** sequence reaching each state at time t *given particular evidence* $\mathbf{e}_{1:t}$

$$\mathbf{m}_{1:t} = \max_{\mathbf{x}_1 \ldots \mathbf{x}_{t-1}} \mathbf{P}(\mathbf{x}_1 \wedge \ldots \wedge \mathbf{x}_{t-1} \wedge \mathbf{X}_t \mid \mathbf{e}_{1:t})$$

- Then the recursive relationship between most likely paths to each state $\mathbf{x}_{t+1}$ and most likely paths to each state $\mathbf{x}_t$, reads

$$\mathbf{m}_{1:t+1} = \alpha \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}) \max_{\mathbf{x}_t} \left(\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{x}_t) m_{1:t} \right)$$

- This is the *Viterbi formula* for the most likely sequence of states.

Calculation for t =1 and t = 2

- General formula: $\mathbf{m}_{1:t+1} = \alpha \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}) \max_{\mathbf{x}_t} \left(\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{x}_t) m_{1:t} \right)$
- Substitution t+1 = 1 and t =0 (we must know the prior $\mathbf{P}(\mathbf{X}_0)$):

$$\mathbf{m}_{1:1} = \alpha \mathbf{P}(\mathbf{e}_1 \mid \mathbf{X}_1) \max_{\mathbf{x}_0} \left(\mathbf{P}(\mathbf{X}_1 \mid \mathbf{x}_0) P(\mathbf{x}_0) \right)$$

- Substitution t+1 = 2 and t = 1:

$$\mathbf{m}_{1:2} = \alpha \mathbf{P}(\mathbf{e}_2 \mid \mathbf{X}_2) \max_{\mathbf{x}_1} \left(\mathbf{P}(\mathbf{X}_2 \mid \mathbf{x}_1) m_{1:1} \right)$$

- Etc. We record the sequence of states which maximizes $m_{1:t}$.

Multiple sequence alignments (MSA)

- Computing Viterbi path of most probable states is also called "aligning sequence to its (hidden Markov) model".
- MSA can be derived, in an efficient way, by aligning the Viterbi paths to each other.
  - We do not align actual sequences, but the sequences of states instead.
  - First we infer these state sequences and then we align them.
  - This is based on an assumption that letters at a given position may not be the same, but the state sequence may be the same.
- So first, HMM is derived from a family of sequences aligned by a different method and then it is used to re-align them or to align a new query sequence to a database.

Database mining

- Given a trained HMM, the likelihood of **any** given sequence as well as likelihood associated with the Viterbi path can be calculated.
- That is, not only for the family of sequences, the HMM was derived from and for, but for **any** sequence.
- These probability scores can be used in discrimination tests in database searches to separate sequences associated with the training family from those that are from different families.
  - This is applicable to both the whole sequences and to their fragments (e.g. genes, promoter regions, motifs, etc.)
  - Similar idea like hash functions but more sophisticated.

Classification of sequences

- HMMs can also be used in classification, for instance across protein families or subfamilies of a single protein family.
  - Based on a principle that these similar sequences have a similar likelihood and/or similar Viterbi path.
- This can be done by training HMM for each class (if class-specific training sets are available).
- A global protein classification system with roughly one HMM per superfamily is under way.
  - There are hundreds of thousands of proteins but it is estimated there might be only 1000 or so superfamilies.

Pattern discovery: information

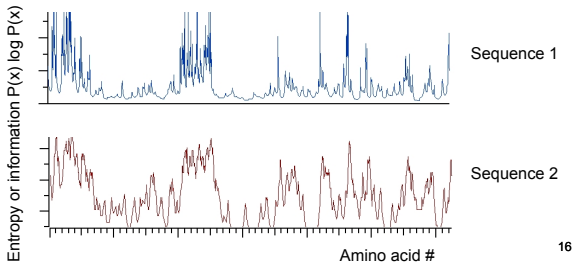
- Patterns can be discovered by examining the structure of trained HMM.
- Shannon's information (in bits) is a measure of the information content associated with the outcome of a random variable X, which assumes one of N values x_i:
- If the possible values x_i of variable X have probabilities $P(x_i)$ then *entropy* of the whole sequence is:

$$H(P(x_1),\ldots,P(x_N)) = -\sum_{i=1}^{N} P(x_i)\log_2 P(x_i)$$

- Similar sequences will have similar entropies.

Entropy profile of the emission probability

- Let the two sequences produce these entropy profiles of the emission probability distributions associated with the M states of underlying HMM.
  - If we know regions with low entropy are associated with some property, we can predict the second sequence has at least one possibly two such regions.



Pattern discovery and analysis of structure

- High emission and transition probabilities are associated with conserved regions and consensus patterns that may have structural or functional significance.
- One convenient way of detecting such patterns is to plot entropy of the emission distributions along the backbone of the model.
- Various patterns in entropy are then associated with corresponding features (structure, function) and we can build a corresponding library of these associations.
- There are number of tools now available that use HMM for gene finding, protein classification and even the structure/function prediction.