

CSE182-L9

Modeling Protein domains using
HMMs

Profiles Revisited

- Note that profiles are a powerful way of capturing domain information
- $\Pr(\text{sequence } x \mid \text{profile } P) = \Pr(x_1 \mid P_1) P(x_2 \mid P_2) \dots$
- $\Pr(\text{AGCTTGTA} \mid P) = 0.9 * 0.2 * 0.7 * 0.4 * 0.3 * \dots$
- Why do we want a different formalism?

	1	2	3	4	5	6	7	8
A	0.9	0.4	0.3	0.6	0.1	0.0	0.2	1.0
C	0.0	0.2	0.7	0.0	0.3	0.0	0.0	0.0
G	0.1	0.2	0.0	0.0	0.3	1.0	0.3	0.0
T	0.0	0.2	0.0	0.4	0.3	0.0	0.5	0.0

Profiles might need to be extended

- A Domain might only be a part of a sequence, and we need to identify and score that part
 - $\Pr[x|P] = \max_i \Pr[x_i \dots x_{i+l} | P]$
- A sequence containing a domain might contain indels in the domain
 - EX: AACTTCGGA

	1	2	3	4	5	6	7	8
A	0.9	0.4	0.3	0.6	0.1	0.0	0.2	1.0
C	0.0	0.2	0.7	0.0	0.3	0.0	0.0	0.0
G	0.1	0.2	0.0	0.0	0.3	1.0	0.3	0.0
T	0.0	0.2	0.0	0.4	0.3	0.0	0.5	0.0

A A C T T C G G A
 A A C T T C G G A

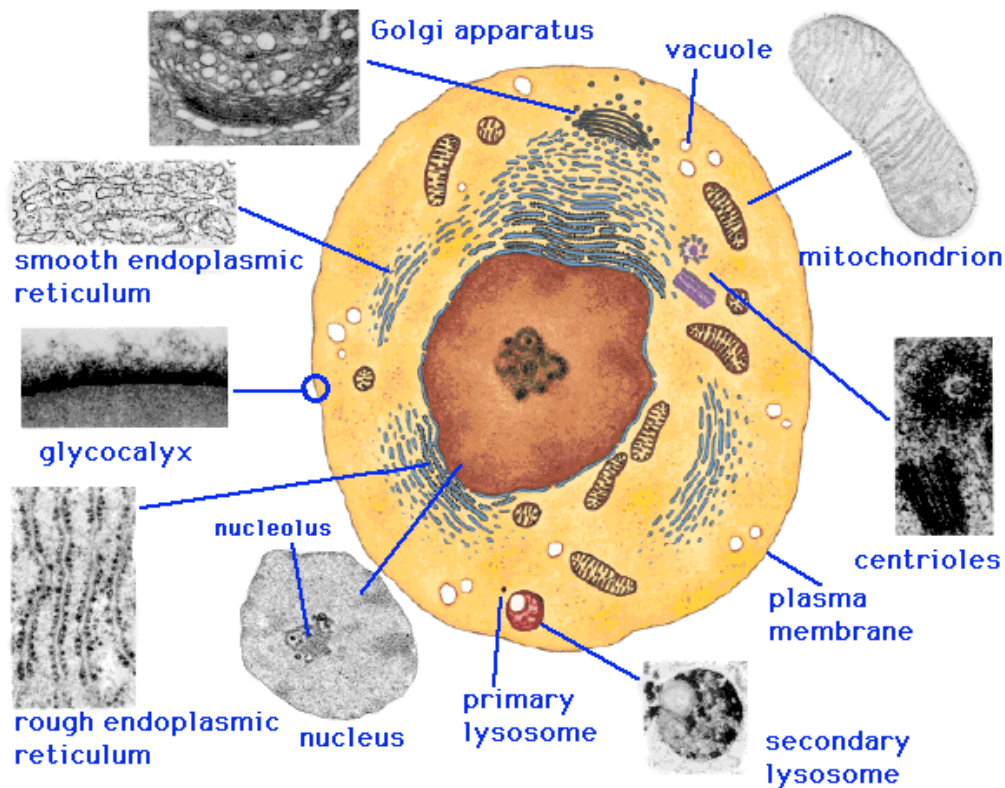
Profiles & Indels

- Indels can be allowed in matching sequences to a profile.
- However, how do we construct the profile in the first place?
 - Without indels, multiple alignment of a sequence is trivial
 - Multiple alignments in the presence of indels is much trickier. Similarly, construction of profiles is harder.
 - The HMM formalism allows you to do both, alignment of single a sequence to the HMM, and construction of the HMM given unaligned sequences.

Profiles and Compositional signals

- While most protein domains have position dependent properties, other biological signals do not.
- Ex:
 1. CpG islands,
 2. Signals for protein targeting, transmembrane etc.

Protein targeting



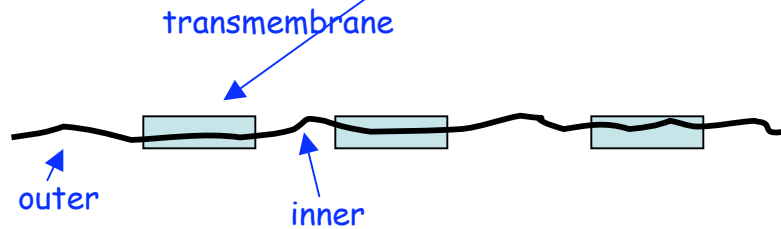
- Proteins act in specific compartments of the cell, often distinct from where it is 'manufactured'.
- How does the cellular machinery know where to send each protein.

Protein targeting

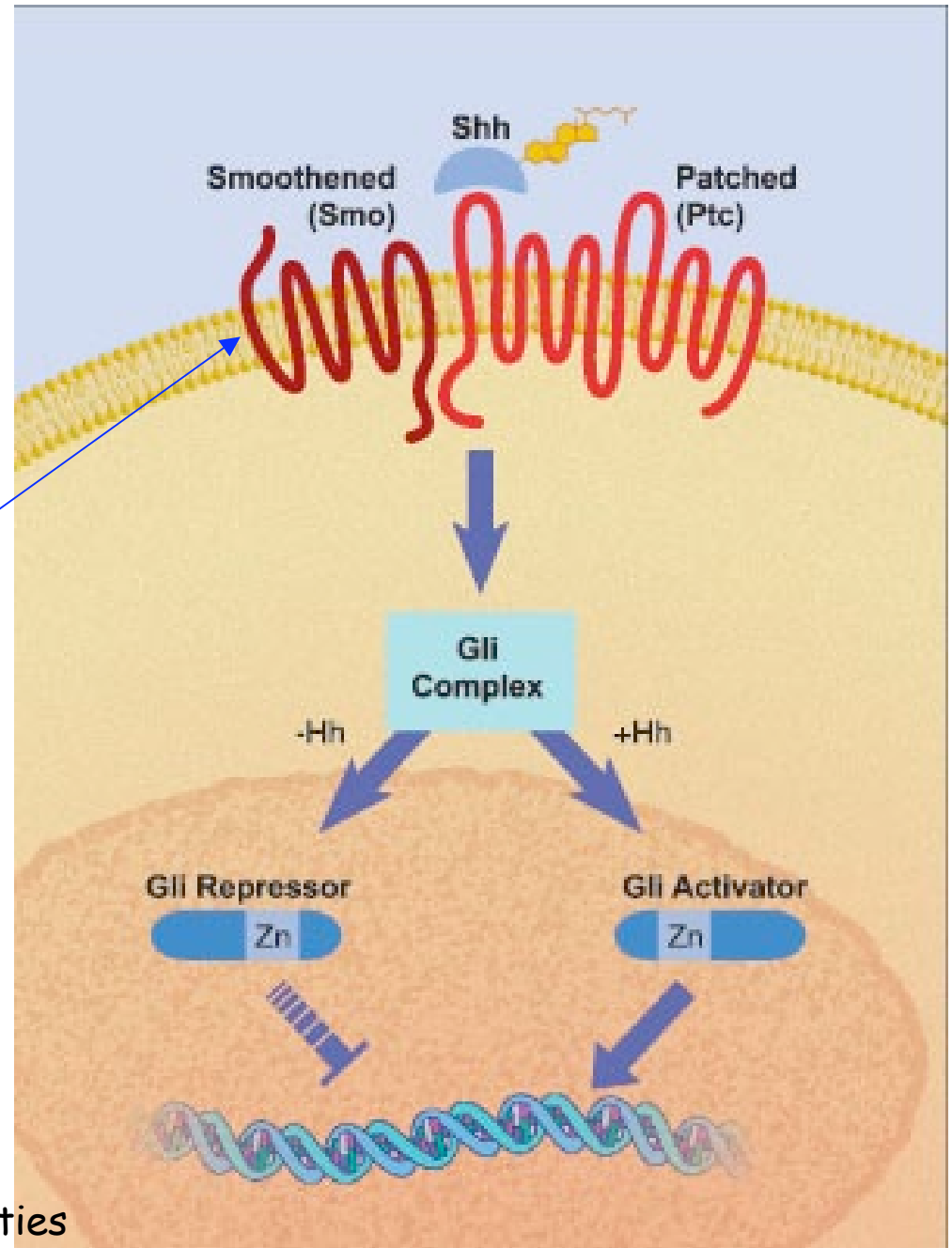
- In 1970, Gunter Blobel showed that proteins have an N-terminal signal sequence which directs proteins to the membrane.
- Proteins have to be transported to other organelles: nucleus, mitochondria,...
- Can we computationally identify the 'signal' which distinguishes the cellular compartment?

Predicting Transmembrane Regions

- For transmembrane proteins, can we predict the transmembrane, outer, and inner regions?



These signals are of varying length and depend more on compositional, rather than position dependent properties



The answer is HMMs

- Certainly, it is not the only answer! Many other formalisms have been proposed, such as Neural Networks, with varying degree of success.

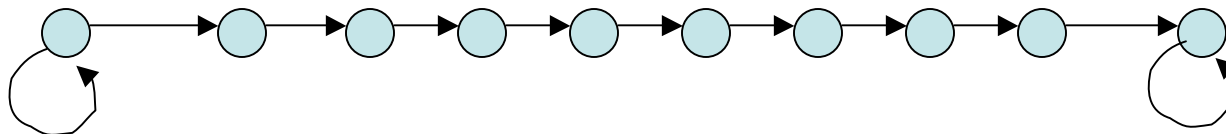
Revisiting automaton

- Recall that Profiles were introduced as a generalization of automata
- The OR operator is replaced by a distribution. However, the $*$ operator (Kleene's closure) has no direct analog.
- Instead we introduce probabilities directly into automata.

Profile HMMs

- Problem: Given Sequence x , Profile P of length l ,
 - compute $\Pr[x|P] = \max_i \Pr[x_i \dots x_{i+l} | P]$
- Construct an automaton with a state (node) for every column
- Extra states at the beginning and end.
- In each step, the automaton emits a symbol, and moves to a new state. Unlike finite automata, the emission and transition are governed by probabilistic rules

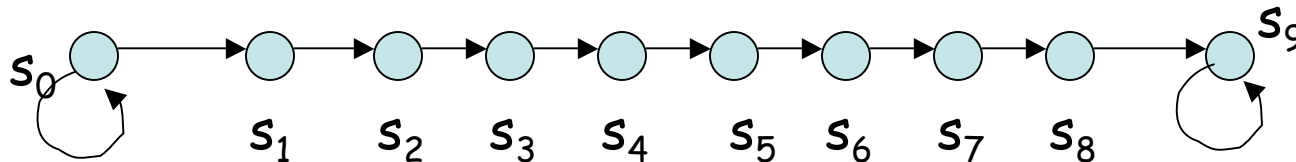
	1	2	3	4	5	6	7	8
A	0.9	0.4	0.3	0.6	0.1	0.0	0.2	1.0
C	0.0	0.2	0.7	0.0	0.3	0.0	0.0	0.0
G	0.1	0.2	0.0	0.0	0.3	1.0	0.3	0.0
T	0.0	0.2	0.0	0.4	0.3	0.0	0.5	0.0



Emission Probability for states

- Each state π emits a base (residue) with a probability (defined by the profile)
- Thus $\Pr(A | s_1) = 0.9$, $\Pr(T | s_4) = 0.4, \dots$

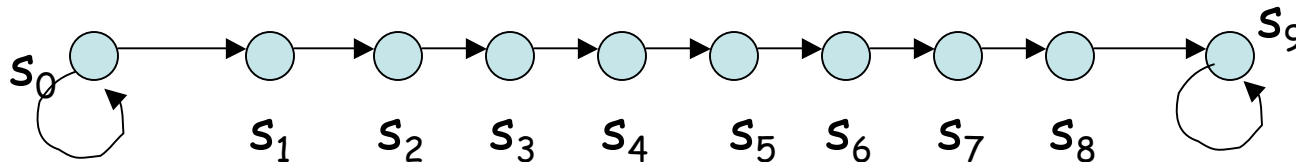
	1	2	3	4	5	6	7	8
A	0.9	0.4	0.3	0.6	0.1	0.0	0.2	1.0
C	0.0	0.2	0.7	0.0	0.3	0.0	0.0	0.0
G	0.1	0.2	0.0	0.0	0.3	1.0	0.3	0.0
T	0.0	0.2	0.0	0.4	0.3	0.0	0.5	0.0



Transition probabilities

- The probability of emitting a base depends upon a state.
- An initial state π_0 is defined. We move to subsequent states using a probabilistic transition
- EX: $\Pr(s_0 \rightarrow s_1) = \Pr(s_0 \rightarrow s_0) = 0.5$, $\Pr(s_1 \rightarrow s_2) = 1.0$

	1	2	3	4	5	6	7	8
A	0.9	0.4	0.3	0.6	0.1	0.0	0.2	1.0
C	0.0	0.2	0.7	0.0	0.3	0.0	0.0	0.0
G	0.1	0.2	0.0	0.0	0.3	1.0	0.3	0.0
T	0.0	0.2	0.0	0.4	0.3	0.0	0.5	0.0



Emission of a sequence

- What is the probability that the automaton emitted the sequence $x_1..x_n$
- The problem becomes easier if we know the sequence of states $\pi=\pi_0..\pi_n$ that the automaton was in

$$\Pr[x | \pi] = P(\pi_0 | \pi_1) \prod_i \Pr(x_i | \pi_i) P(\pi_i | \pi_{i+1})$$

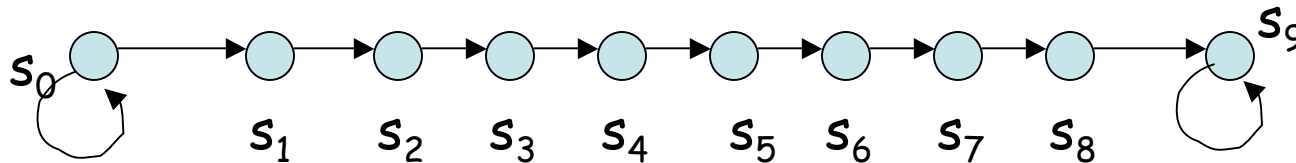
Computing $\Pr(x|\pi)$

- Example: Consider the sequence *GGGA ACTTGTACCC*

— X = G G G A A C T T G T A C C C
 - π = 0 0 0 0 1 2 3 4 5 6 7 8 9 9 9

- $\Pr(x_i | \pi_i)$: .25 .25 .25 .9 .4 .7 .4 .3 1 .5 1 .25 .25 .25
- $\Pr(\pi_i \rightarrow \pi_{i+1}) =$.5 .5 .5 .5 1 1 1 1 1 1 1 1 1 1

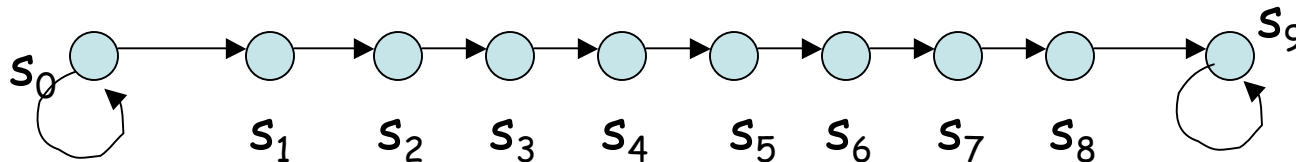
	1	2	3	4	5	6	7	8
A	0.9	0.4	0.3	0.6	0.1	0.0	0.2	1.0
C	0.0	0.2	0.7	0.0	0.3	0.0	0.0	0.0
G	0.1	0.2	0.0	0.0	0.3	1.0	0.3	0.0
T	0.0	0.2	0.0	0.4	0.3	0.0	0.5	0.0



HMM: Formal definition

- $M=(\Sigma,Q,A,E)$, where
- Σ is the alphabet (EX: $\{A,C,G,T\}$)
- Q : set of states, with an initial state q_0 , and perhaps, a final state. (EX: $\{s_0,\dots,s_9\}$)
- A : Transition Probabilities, $A[i,j] = \Pr(s_i \rightarrow s_j)$ (EX: $A[0,1]=0.5$)
- E : Emission probabilities $e_i(b) = \Pr(b|s_i)$ (EX: $e_3(T)=0.7$)

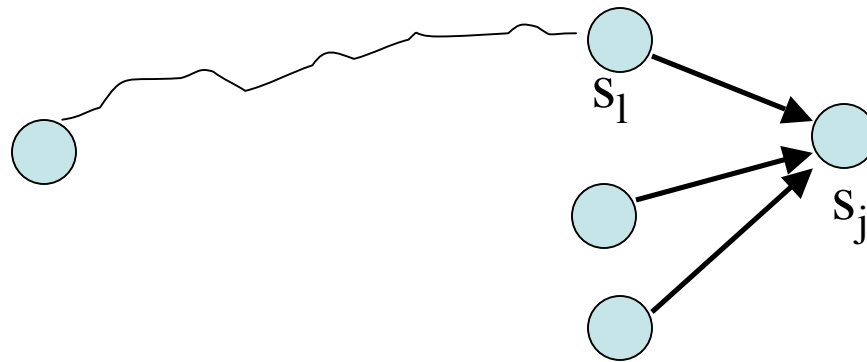
	1	2	3	4	5	6	7	8
A	0.9	0.4	0.3	0.6	0.1	0.0	0.2	1.0
C	0.0	0.2	0.7	0.0	0.3	0.0	0.0	0.0
G	0.1	0.2	0.0	0.0	0.3	1.0	0.3	0.0
T	0.0	0.2	0.0	0.4	0.3	0.0	0.5	0.0



Pr(x|M)

- Given the state sequence π , we know how to compute $\text{Pr}(x|\pi)$.
- We would like to compute
 - $\text{Pr}(x|M) = \max_{\pi} \text{Pr}(x|\pi)$
- Let $|x| = n$, and $|Q|=m$, with s_m being the final state.
- As is common in Dynamic programming, we consider the probability of generating a prefix of x
- Define $v(i,j)$ = Probability of generating $x_1..x_i$, and ending in state s_j
- Is it sufficient to compute $v(i,j)$ for all i,j ?
 - Yes, because $\text{Pr}(x|M) = v(n,m)$

Computing $v(i,j)$



- If the previous state (at time $i-1$) was s_l ,
 - then $v(i,j) = v(i-1,l).A[l,j].e_j(x_i)$
- The previous state must be one of the m states.
- Therefore $v(i,j) = \max_{l \in Q} \{v(i-1,l).A[l,j]\}.e_j(x_i)$

The Viterbi Algorithm

- $v(i,j) = \max_{l \in Q} \{v(i-1,l) \cdot A[l,j]\} \cdot e_j(x_i)$
- We take logarithms and an approximation to maintain precision
 - $S(i,j) = \log(e_j(x_i)) + \max_{l \in Q} \{ S[i-1,l] + \log(A[l,j]) \}$
- The Viterbi Algorithm
- For $i = 1..n$
 - For $j = 1..M$
 - $S(i,j) = \log(e_j(x_i)) + \max_{l \in Q} \{ S[i-1,l] + \log(A[l,j]) \}$

Probability of being in specific states

- What is the probability that we were in state k at step I ?

$$\frac{\text{Pr}[\text{All paths that passed through state } k \text{ at step } I, \text{ and emitted } x]}{\text{Pr}[\text{All paths that emitted } x]}$$

$$= \frac{\text{Pr}[x, \square_i = k]}{\text{Pr}[x]}$$