Applications of HMM - profile HMMs and gene models

Bartek Wilczyński

April 21^{st}, 2020

other materials

- Where to read more on today's topics:
  - *Biol. Sequence Analysis, Durbin et al. Chap. 5*
  - Protein alignment - Hmmer webpage `http://hmmer.janelia.org/`
  - Gene finding - Glimmer webpage `http://ccb.jhu.edu/software/glimmerhmm/`

How proteins work

- Proteins – long chains of aminoacids – are the the building blocks that all living organisms are made of
- Most globular proteins have a *native* 3-dimensional structure, i.e. the structure they *fold* to in natural conditions
- The function of a protein is determined (to a large degree) by its overall 3d-fold and the aminoacids placed in its active sites

Markov Models - refresh

- The model consists of a state space $Q \neq \emptyset$ (for our purposes Q is finite)
- and a transition probability matrix p_{ij} where $i, j \in Q$
- The model has no memory, the probability of moving from state i to j depends only on the state i.
- Higher order MMs can be simulated on a 0-order (memory-less) MM by exponentially increasing the alphabet size

Hidden Markov Model

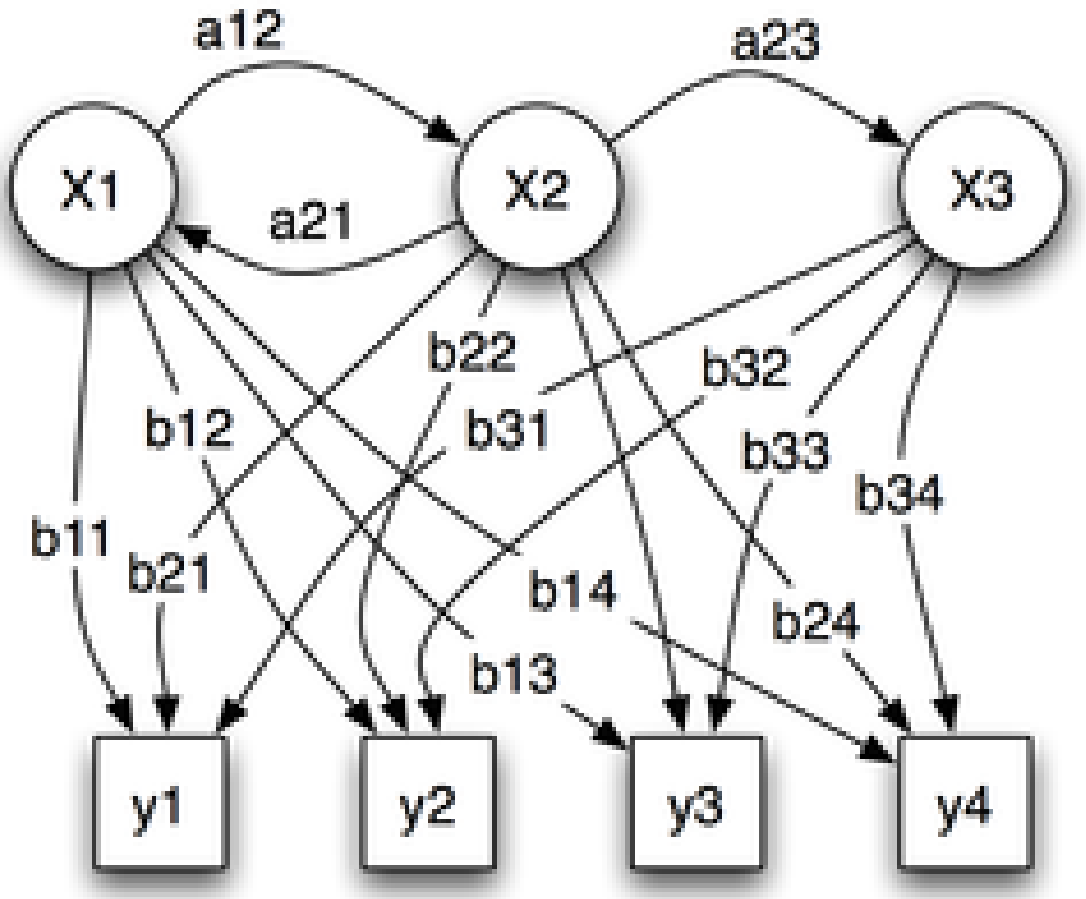


image (c) wikipedia

Hidden Markov Model - summary

- For a given HMM, we can **simulate** its trajectories and calculate the **probability of generating a word** given a trajectory
- If we know the word generated by a known HMM, we can use *Viterbi* algorithm to find out the **most probable trajectory** and *forward-backward algorithms* to calculate **probabilities of all trajectories** resulting in emitting this word
- If we know the number of states and emission symbols we can use a large training body to find (locally) optimal transition and emission matrices by the *Baum-Welch algorithm*.
- While the notion of time is natural for Markov Models for DNA sequence evolution, HMMs very frequently use their "time" to represent generating sequences (e.g. the CpG island model)

Sequence profile as an HMM

If we have a sequence alignment, we can represent it as a chain-like HMM, with

- one state for each position
- 1-off-diagonal transition matrix
- emission matrix reresenting probabilities of "observing" each character at each position

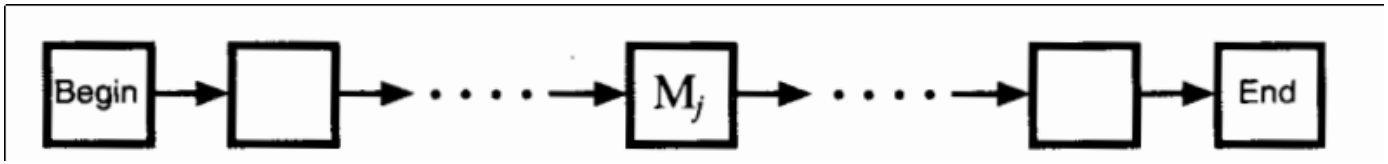


image (c) Durbin et al.

HMM alignment and protein structures

Conserved protein structures have residues with certain preferences for different AAs (HMM states)

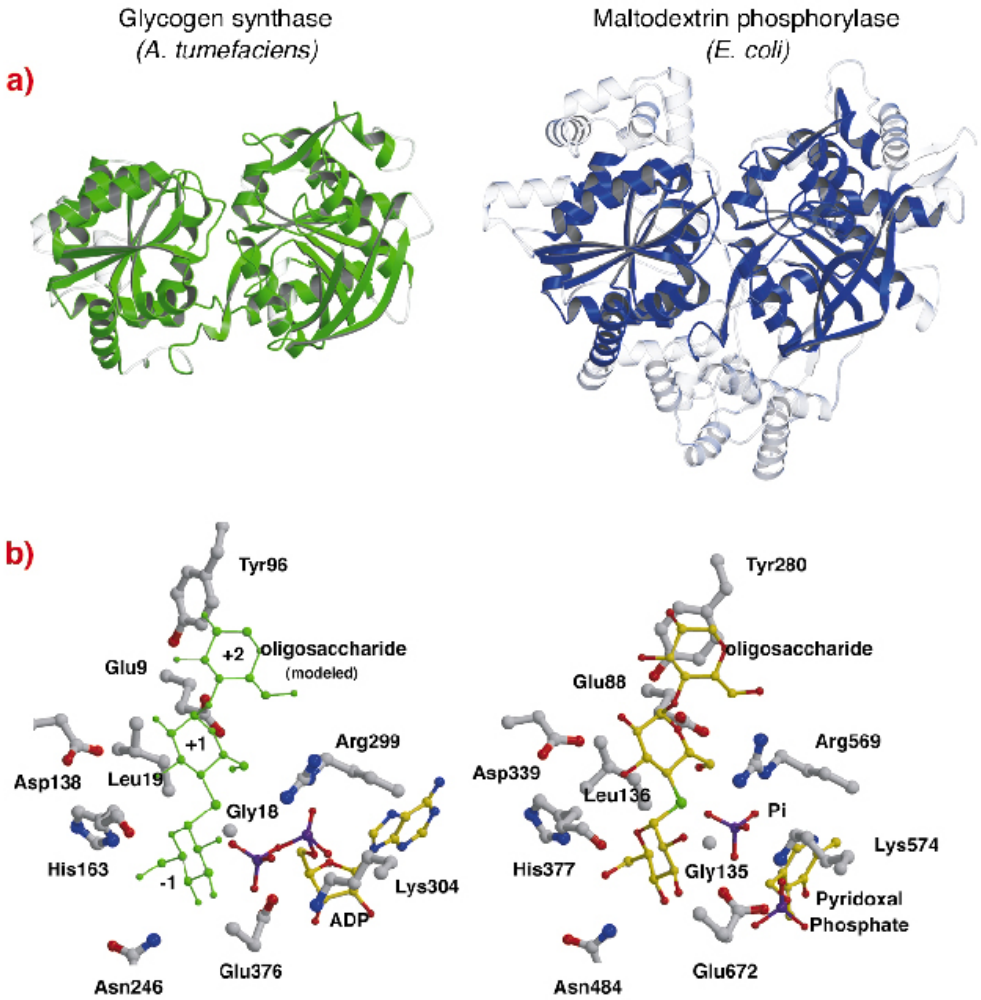


image (c) Buschiazzo et al. 2004 EMBO J.

HMM alignment and protein structures

This can be seen in their structural alignment (a difficult problem itself)

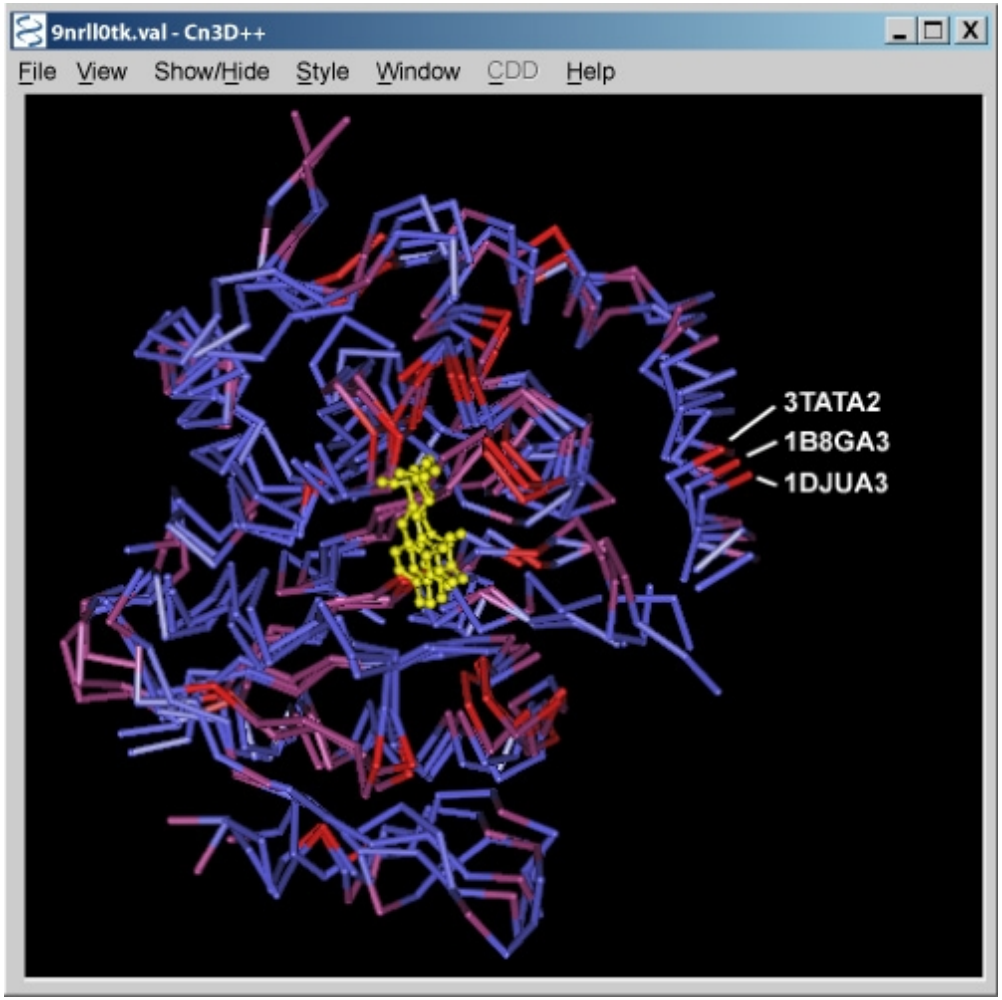


image (c) Sayers and Bryant

HMMs with insertions

HMMs also include additional states for generating sequences with insertions (new residues)

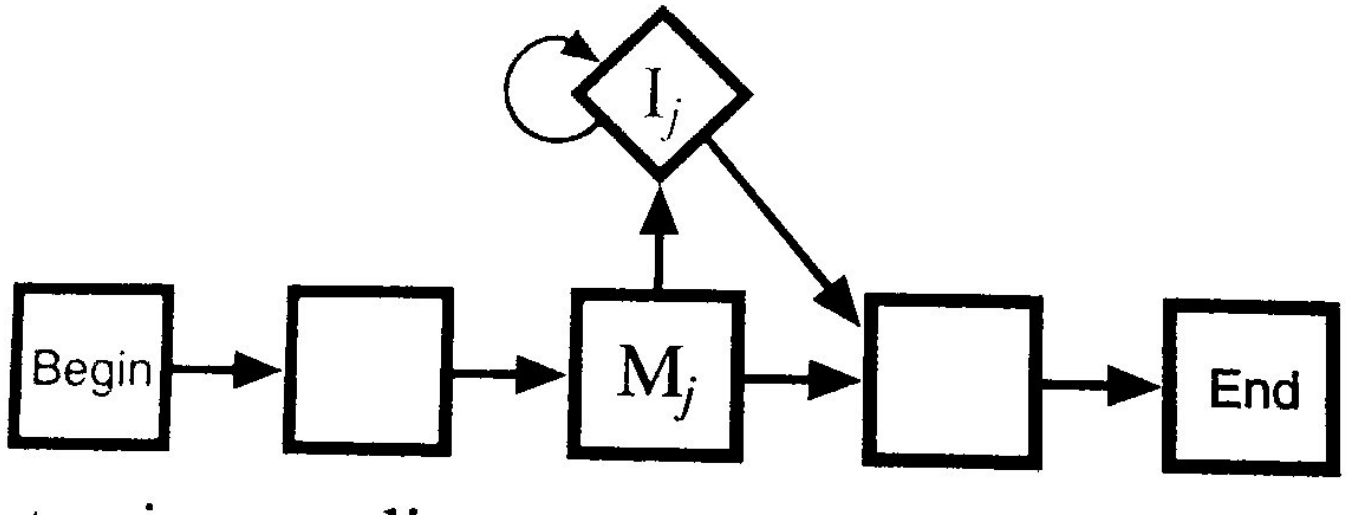


image (c) Durbin et al.

HMMs with deletions

HMMs also include additional states for generating sequences with deletions (lost residues)

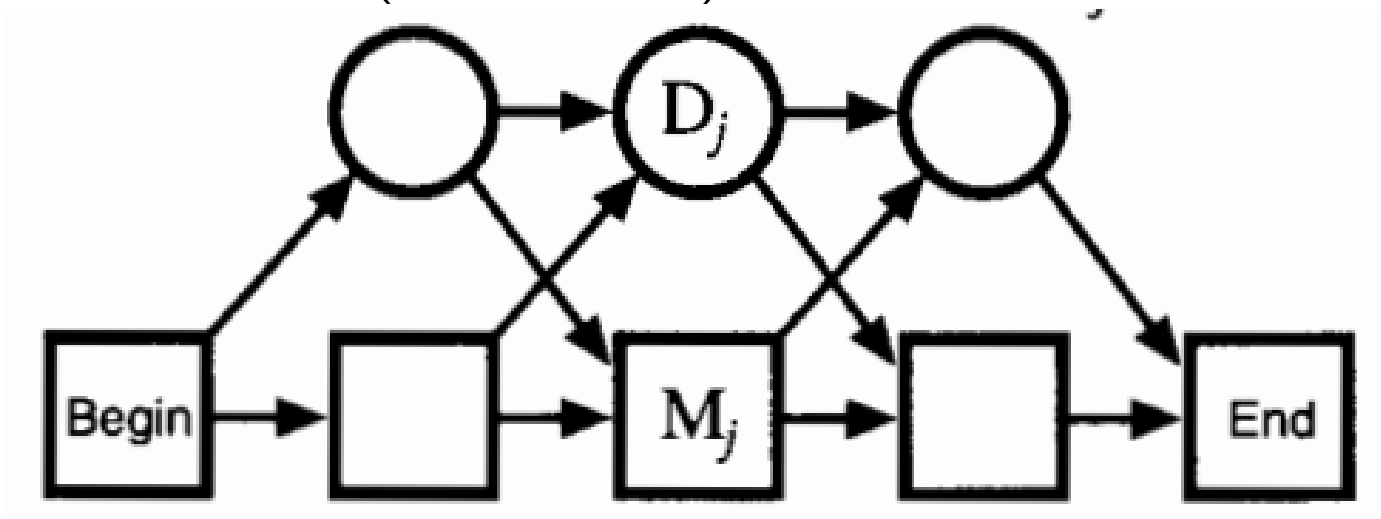


image (c) Durbin et al.

HMMs alignment with all states

(almost) Complete set of states

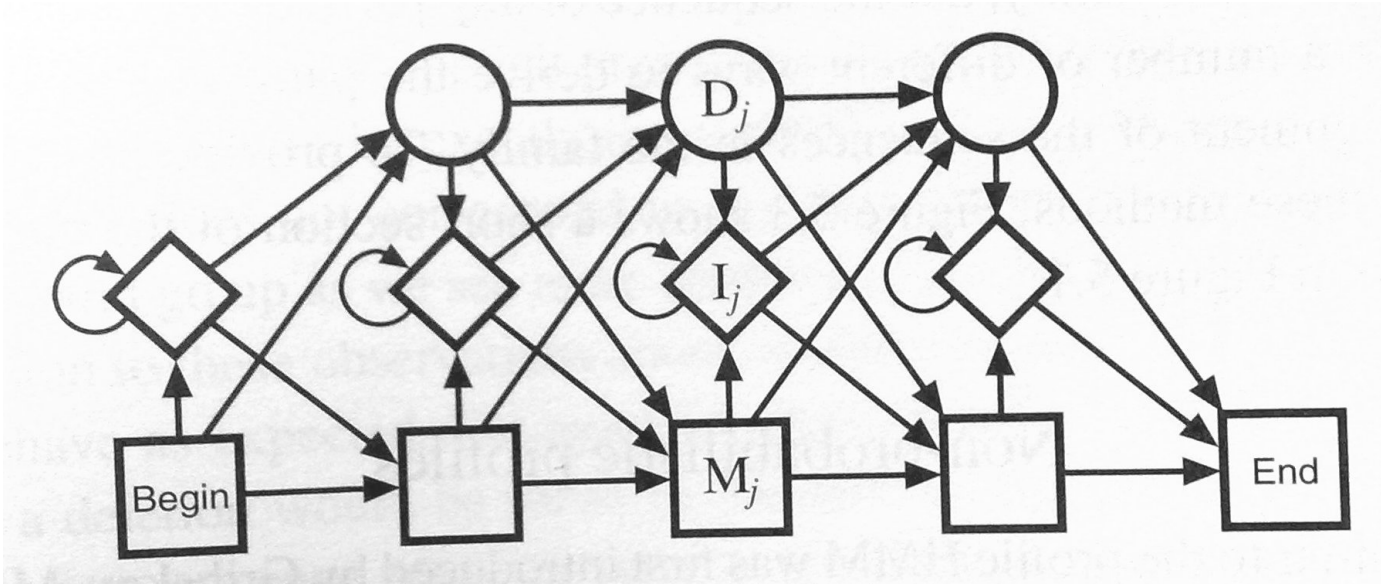


image (c) Durbin et al.

HMMs alignment with all states

And now also with emissions

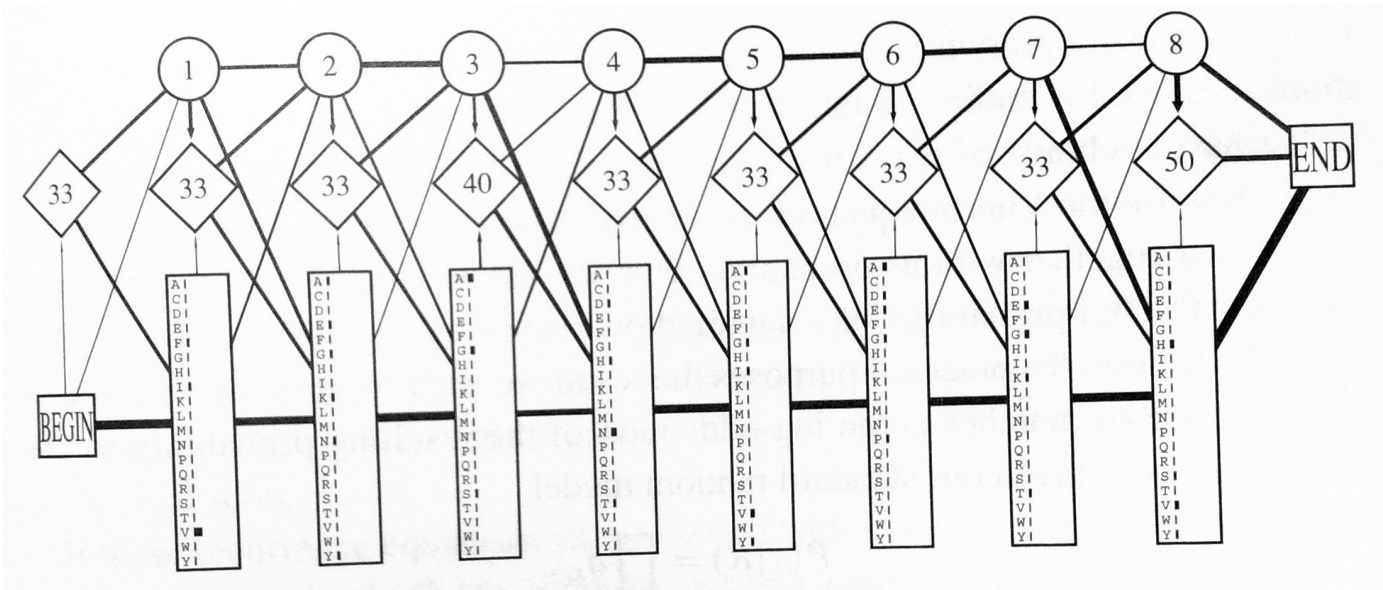


image (c) Durbin et al.

HMM alignments - properties

Using an HMM profile, we can

- sample "random" sequences conforming to the model (by simulating trajectories)
- align a new sequence wih it (using the Viterbi algorithm)
- represent different preferences for insertions/deletions for different parts of a protein
- We can even (with some care) align two different HMMs with dynamic programming
- But can we reconstruct HMMs from data?

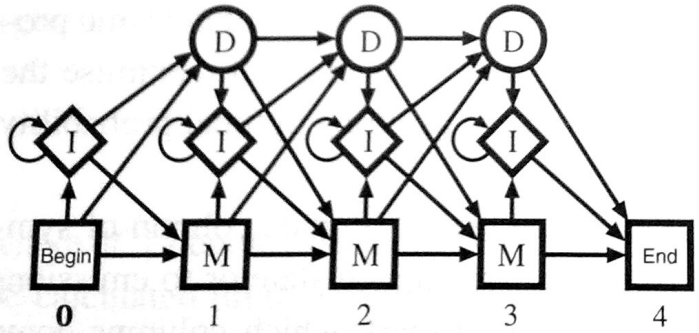
HMM construction from multiple alignment

Assume we know the alignment **and** the "residue" positions

(a) Multiple alignment:

	x	x	.	.	.	x
bat	A	G	-	-	-	C
rat	A	-	A	G	-	C
cat	A	G	-	A	A	-
gnat	-	-	A	A	A	C
goat	A	G	-	-	-	C
	1	2	.	.	.	3

(b) Profile-HMM architecture:



(c) Observed emission/transition counts

		model position 0	1	2	3
match emissions	A	-	4	0	0
	C	-	0	0	4
	G	-	0	3	0
	T	-	0	0	0
insert emissions	A	0	0	6	0
	C	0	0	0	0
	G	0	0	1	0
	T	0	0	0	0
state transitions	M-M	4	3	2	4
	M-D	1	1	0	0
	M-I	0	0	1	0
	I-M	0	0	2	0
	I-D	0	0	1	0
	I-I	0	0	4	0
	D-M	-	0	0	1
	D-D	-	1	0	0
	D-I	-	0	2	0

If we don't know the gap positions, we can use a dynamic programming approach

image (c) Durbin et al.

Local HMM matching

We can also model a local substructure (domain) of a protein by an HMM, then we need an HMM able to perform "local" alignmnent to a query sequence:

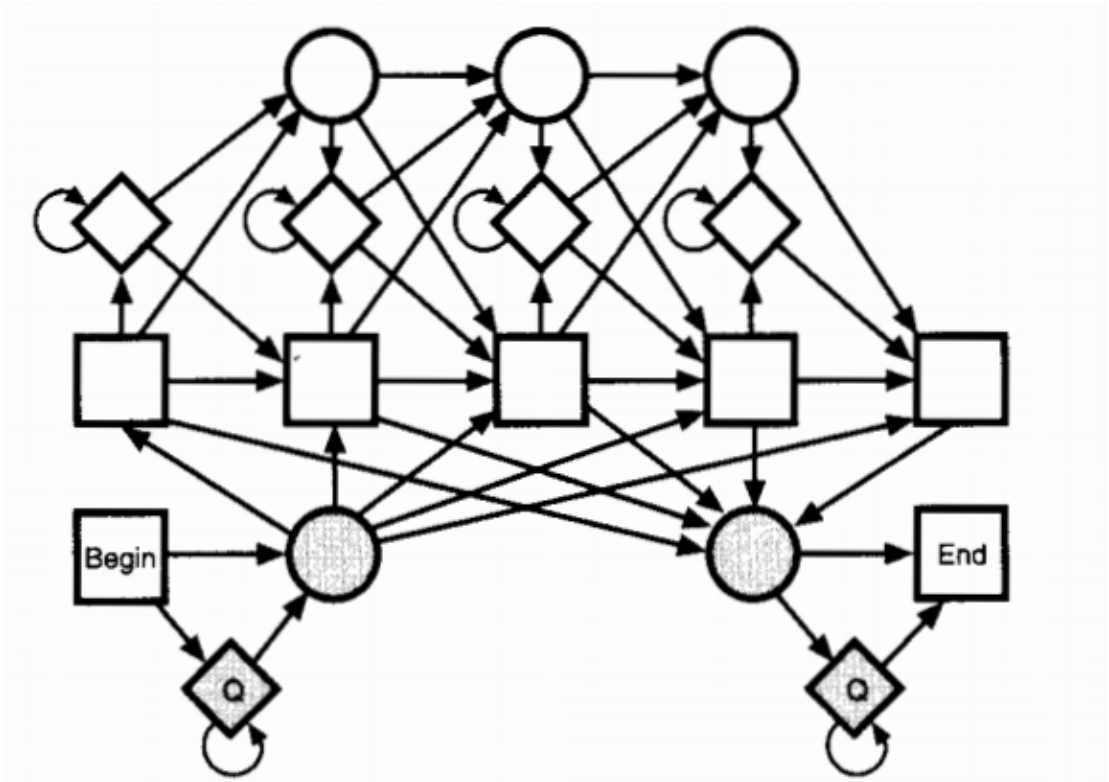


image (c) Durbin et al.

Multiple Local HMM matching

Such local substructure (domain) can occur multiple times in a query sequence:

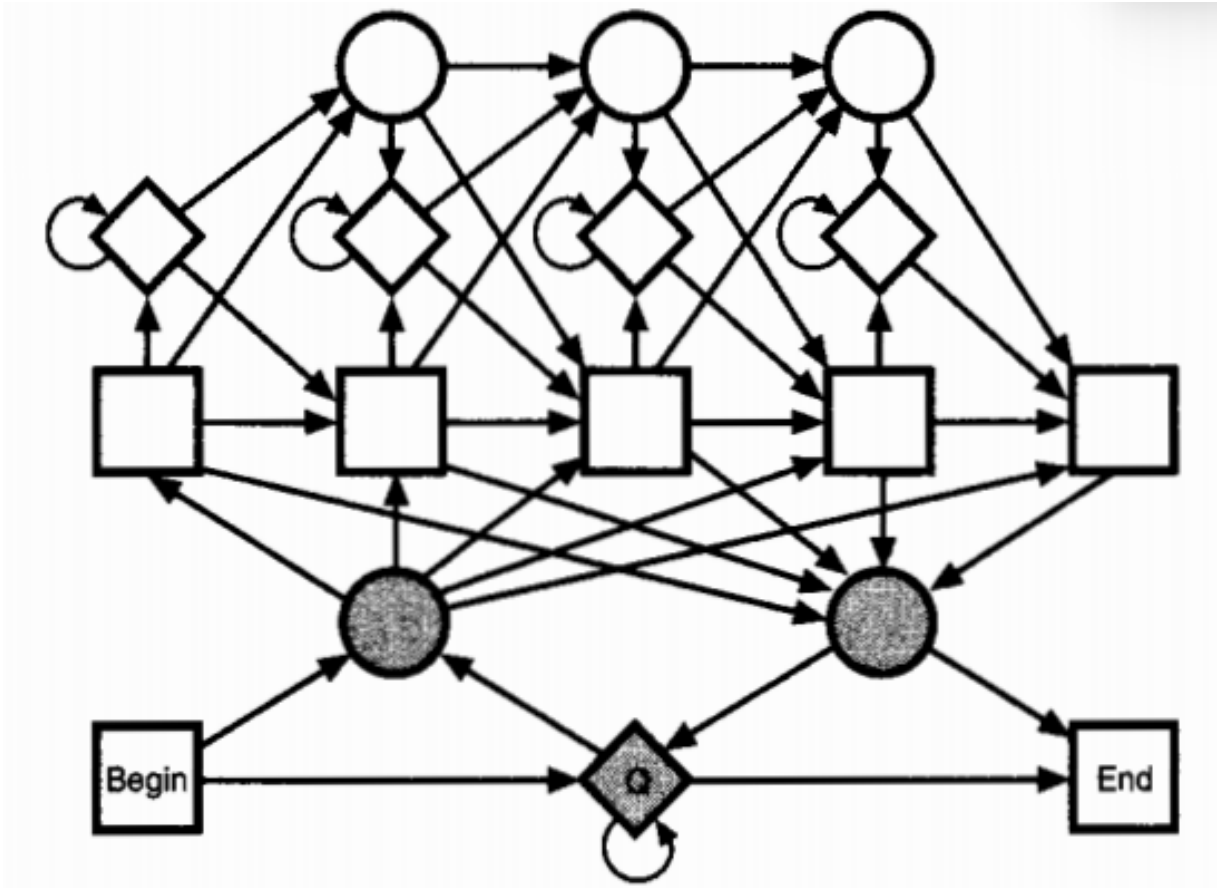


image (c) Durbin et al.

Gene finding - Problem statement

- When sequencing a new genome (done on a daily basis now...) we get only the DNA sequence, no annotation
- We know there are protein coding genes, but we don't know where they are
- For many genes, we can find their transcripts by extracting RNA from the cells and sequencing it (EST libraries)
- This will not work for many genes (e.g. rarely transcribed) and is quite expensive (another round of sequencing...)
- Knowing all the genes is important for most functional studies and we need computational (cheap) ways of doing it

Gene finding - typical approach

- Given the start and stop codons, for any genome sequence, there is only limited number of Open reading frames (ORFs) - subsequences beginning from a start codon and finishing with a stop codon.
- We know the codon-code, so we can find all possible ORFs in linear time (using compressed representation).
- Not all ORFs are genes: there are many short sequence motifs which enable transcription of a given orf, but we don't necessarily know all of them (and they may vary between species).

Transcription initiation

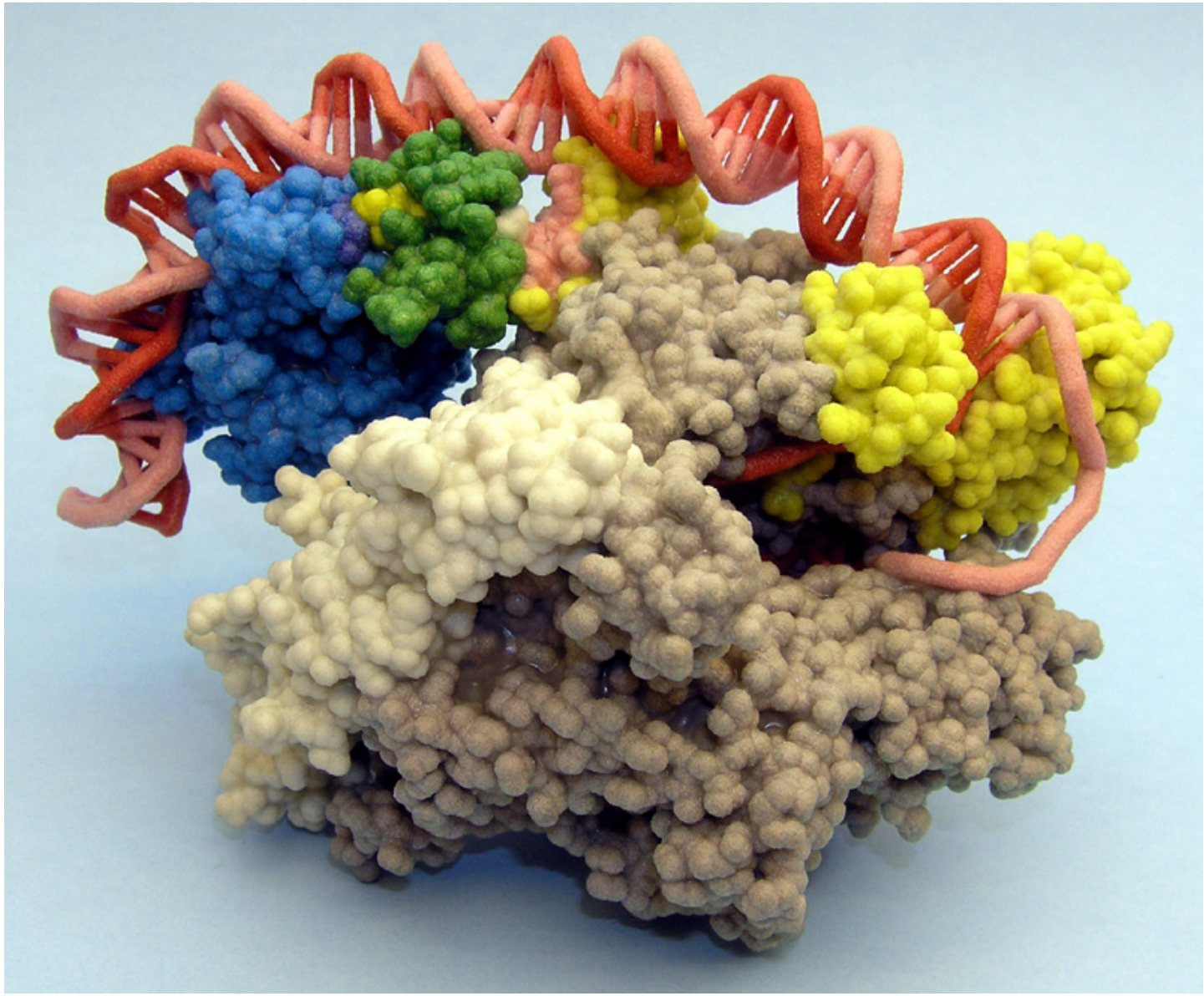
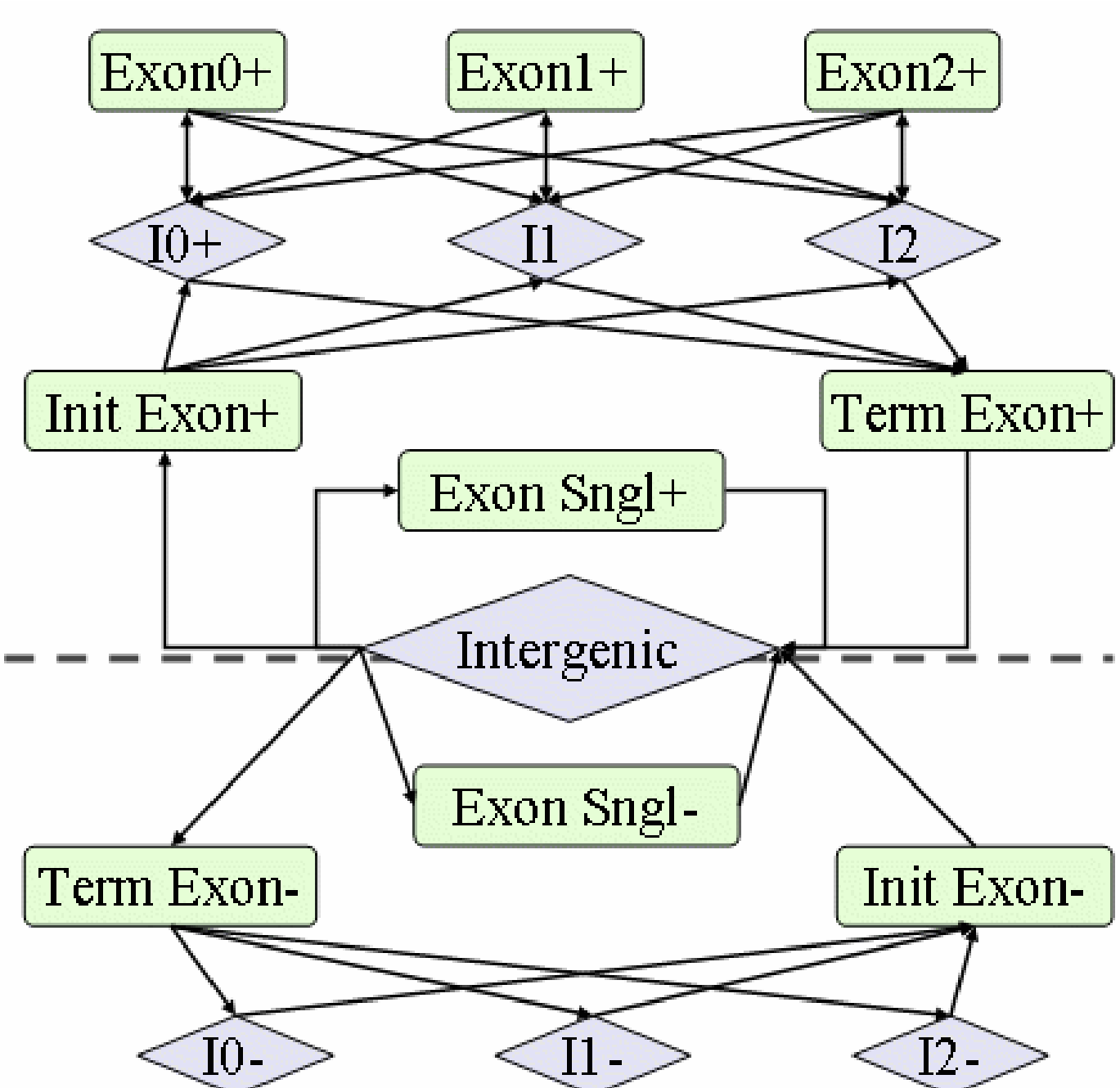


image (c) pingrysmatteam

Representing TSSs by Higher order Markov Models

- We can use a higher-order markov model to represent gene sequences (to account for dependencies between positions)
- We can train them on known genes (from EST libraries, human annotation or high confidence predictions)
- Because some motifs are long, we need a high order MM ($\geqslant 8$), but this requires very many training examples
- This can be solved by using a "variable order MM" (VOM) or interpolated Markov Model (IMM) which uses higher order dependencies only for frequent enugh words

GlimmerHMM


Exon0+
Exon1+
Exon2+
I0+
I1
I2
Init Exon+
Term Exon+
Exon Sngl+
Intergenic
Exon Sngl-
Term Exon-
Init Exon-
I0-
I1-
I2-