Chapter 6

Profiles and Hidden Markov Models

Overview

1. Introduction
2. Introduction to Biological Databases
3. Pairwise Sequence Alignment
4. Database Similarity Searching
5. Multiple Sequence Alignment
6. Profiles and Hidden Markov Models
7. Protein Motifs and Domain Prediction
8. Gene Prediction
9. Promoter and Regulatory Element Prediction
10. Phylogenetics Basics
11. Phylogenetic Tree Construction Methods and Programs
12. Protein Structure Basics
13. Protein Structure Visualization, Comparison and Classification
14. Protein Secondary Structure Prediction
15. Protein Tertiary Structure Prediction
16. RNA Structure Prediction
17. Genome Mapping, Assembly and Comparison
18. Functional Genomics
19. Proteomics

Models representing the information contained in a multiple sequence alignment (MSA)

- Consensus sequence
- Position-specific scoring matrices (PSSMs) and profiles
- Hidden Markov models (HMMs)

Purpose of such models: comparison (alignment) with query sequences so as to detect more distantly related homologs

Consensus sequence

	Amino acid position				
.....	*n*	*n*+1	*n*+2	n+3	.....
	A	W	Q	R	
	A	W	N	K	
	G	Y	Q	R	
	A	W	Q	R	
	A	F	-	R	
	S	W	-	K	
	A	F	Q	R	
Consensus sequence:	**A**	**W**	**Q**	**R**	

Drawback of consensus sequences: detailed frequency information with respect to non-consensus residues is entirely lost

Position-specific scoring matrix (PSS matrix)

- Somewhat resembles substitution matrices (cf. Chapter 3)
- Statistical model reflecting the frequency distribution of amino acid or nucleotide residues in a multiple sequence alignment for each position in the alignment
- The horizontal direction represents residue positions in the multiple sequence alignment
- Each column contains the likelihoods of finding each of the 20 residues at that position
- Values in the table are log odds scores (as in substitution matrices)

Position-specific scoring matrix (PSS matrix)

- The PSS matrix is a representation of the entire set of related sequences in the alignment and preserves frequency information for every position in the sequence
- A probabilistic model like this can be used for database searching and alignment (much like a single sequence, *e.g.* a consensus, but with better sensitivity)

Position-specific scoring matrices: an example

Position	1	2	3	4	5	6
Sequence 1	A	T	G	T	C	G
Sequence 2	A	A	G	A	C	T
Sequence 3	T	A	C	T	C	A
Sequence 4	C	G	G	A	G	G
Sequence 5	A	A	C	C	T	G

Convert multiple alignment to a raw frequency table

Pos.	1	2	3	4	5	6	Overall freq.
A	0.6	0.6	—	0.4	—	0.2	0.30
T	0.2	0.2	—	0.4	0.2	0.2	0.20
G	—	0.2	0.6	—	0.2	0.6	0.27
C	0.2	—	0.4	0.2	0.6	—	0.23

Normalize the values by dividing them by overall freq.

Pos.	1	2	3	4	5	6	Overall freq.
A	2.0	2.0	—	1.33	—	0.67	0.30
T	1.0	1.0	—	2.0	1.0	1.0	0.20
G	—	0.74	2.22	—	0.74	2.22	0.27
C	0.87	—	1.74	0.87	2.61	—	0.23

Convert the values to log to base of 2

Pos.	1	2	3	4	5	6
A	1.0	1.0	—	0.41	—	-0.58
T	0.0	0.0	—	1.0	0.0	0.0
G	—	-0.43	1.15	—	-0.43	1.15
C	-0.2	—	0.8	-0.2	1.38	—

Matching a given sequence to a PSS matrix

Match **AACTCG** in the matrix

Find nucleotides at respective pos. of the matrix

Pos.	1	2	3	4	5	6
A	1.0	1.0	—	0.41	—	-0.58
T	0.0	0.0	—	1.0	0.0	0.0
G	—	-0.43	1.15	—	-0.43	1.15
C	-0.2	—	0.8	-0.2	1.38	—

Calculate the sum of log odds scores

$$1.0 + 1.0 + 0.8 + 1.0 + 1.38 + 1.15 = 6.33$$

Matching a given sequence to a PSS matrix

- Total match score of $2^{6.33} = 80$ implies that probability of the sequence fitting the matrix is 80 times more likely than random chance

Profiles

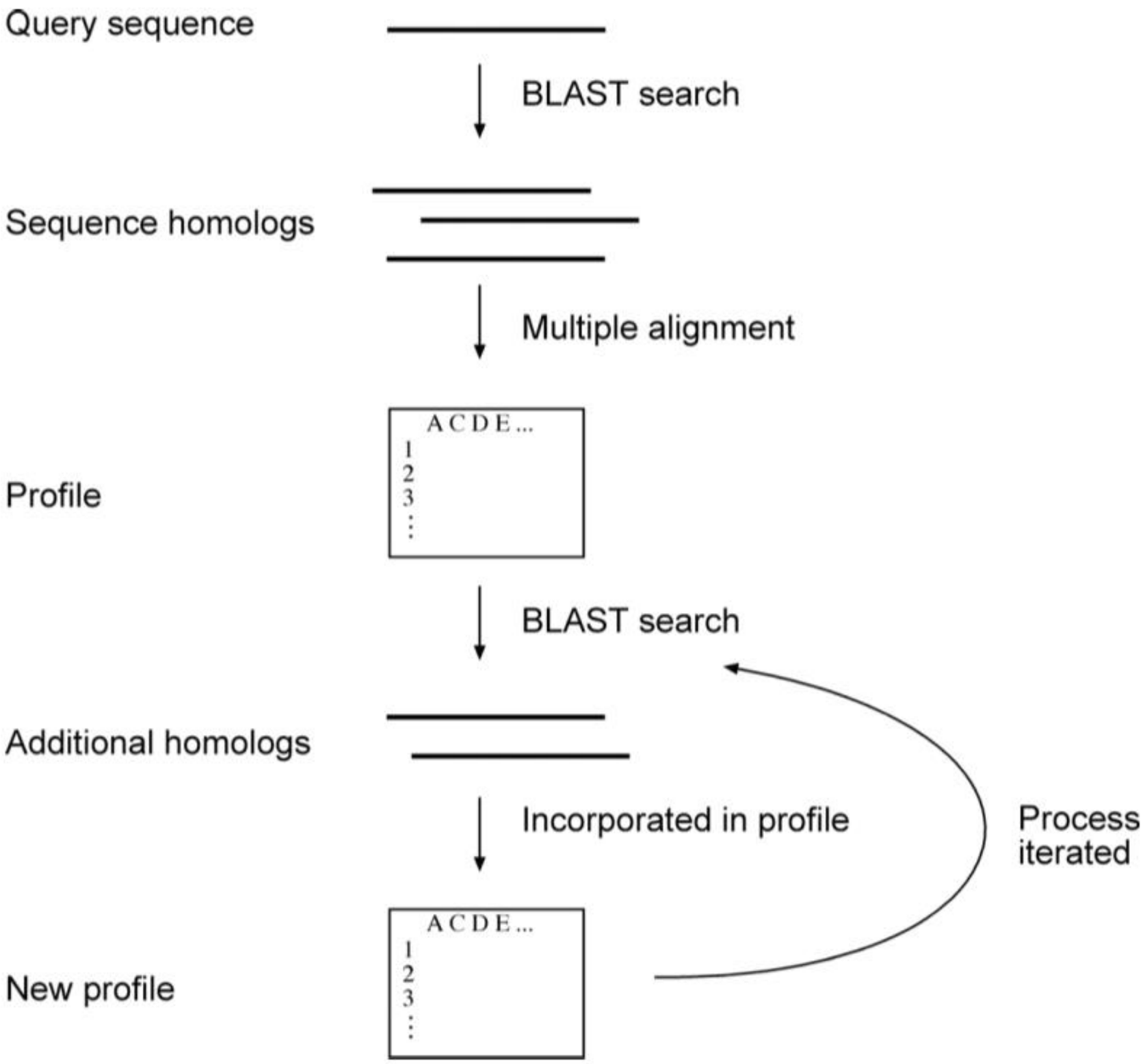
- Profiles are position-specific scoring matrices with additional penalty information regarding insertions and deletions for a sequence family
- Gap penalty scores are often set arbitrarily; therefore, a series of gap parameters have to be tested to find the ones that leads to optimal alignments between query sequences and profiles

An application of profiles: PSI-BLAST

Position-specific iterated BLAST (= PSI-BLAST) builds profiles and performs database searches in an iterative fashion:

- First, a normal BLAST search finds similar sequences
- These are used to build a multiple sequence alignment, from which a profile is created
- Profile is used in a next round of searching to identify more distantly related members of the same family
- New hits are combined with previous ones to form a new alignment and, from that, an updated profile
- This cycle is repeated (iteration) until no new hits are found

PSI-BLAST


Query sequence
BLAST search
Sequence homologs
Multiple alignment
A C D E ...
1
2
3
Profile
BLAST search
Additional homologs
Incorporated in profile
Process iterated
A C D E ...
1
2
3
New profile

Properties of PSI-BLAST

- Profiles are constructed automatically and fine-tuned in each successive cycle, enabling identification of distantly related sequences
- PSI-BLAST is able to identify about three times more homologs than regular BLAST, which mainly fall within the range of less than 30% sequence identity
- Very efficient for identifying conserved core domains within otherwise poorly conserved sequences
- High sensitivity implies (somewhat) lower selectivity

Potential problem with PSI-BLAST: profile drift

- Erroneous inclusions of unrelated sequences (false positives) results in biased profiles
- As a consequence further errors are incorporated in subsequent cycles
- Solutions:
  - ✓ Careful inspection by the user, at each iteration
  - ✓ Only limited number of cycles instead of reaching full convergence (typically three to five iterations are enough)

Markov models

- Markov model (= Markov chain): sequence of events (“states”) that occur one after another in a chain
- Each state determines the probability of the next state
- Markov chain moves from one state to the next with a certain probability (= transition probability)

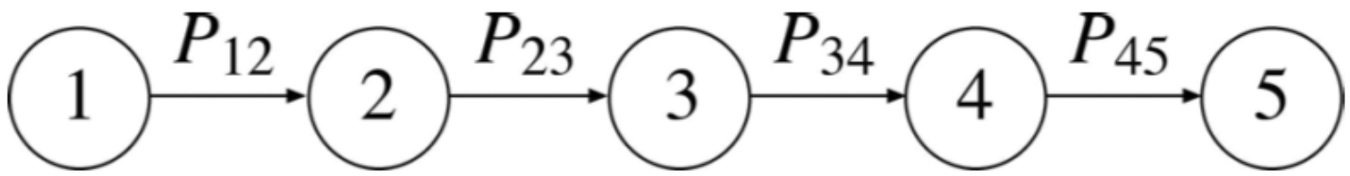


Figure 6.4: A simple representation of a Markov chain, which consists of a linear chain of events or states (numbered) linked by transition probability values between events (states).

The order of Markov models

The complexity of Markov models can vary:

- Zero-order Markov model: describes the probability of the current state, independent of the previous state(s)
- First-order Markov model: describes the probability of the current state as being determined by the previous state
- Second-order Markov model: probability of the current state is determined by the previous two states

Markov models and biological sequences

- For biological sequences, Markov chains might (for instance) be used to model the probability of a given residue, taking into account the identity of the previous residue (first-order Markov model)
- Likewise, a second-order Markov model could be used to model the unique trimer frequencies in biological sequences, *i.e.* three linked residues occurring simultaneously as in the case of a codon
- Example: a second-order Markov model can be used to distinguish coding from non-coding random sequences, in which the nucleotides within a triplet are entirely independent

Hidden Markov Models

- A Hidden Markov Model (HMM) combines one Markov chain with observed states with other chains of unobserved (or "hidden") states
- Hidden states influence the likelihood of the observed states
- *E.g.* gaps may influence what comes next in the sequence, but do not correspond to actual residues and are treated as unobservable states

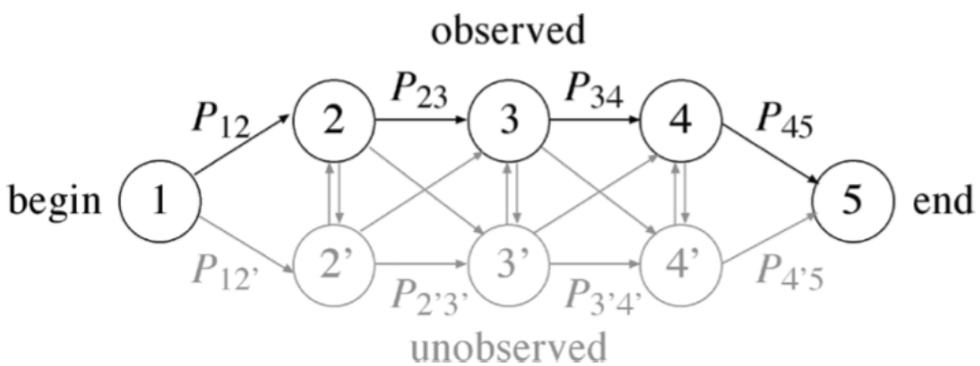
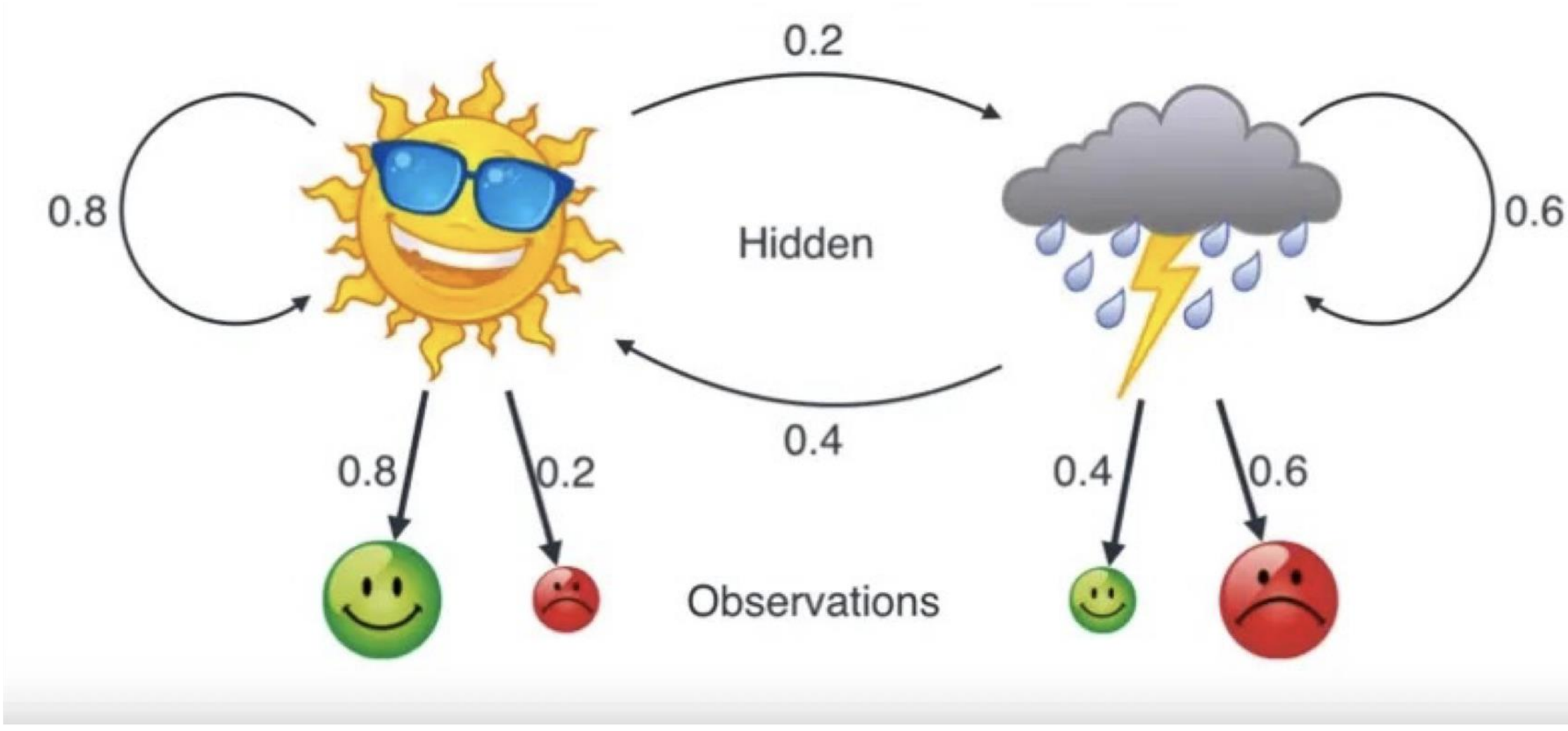


Figure 6.5: A simplified HMM involving two interconnected Markov chains with observed states and common "begin" and "end" states. The observed states are colored in black and unobserved states in grey. The transition probability values between observed states or between unobserved states are labeled. The probability values between the observed and hidden states are unlabeled.

Hidden Markov Models: a simple example



Hidden Markov Models and MSAs

- Each state in the HMM comprises a number of elements or symbols: the four nucleotides or the twenty amino acids
- A probability value is associated with each of the symbols within a state; this is called an "emission probability“, *e.g.* in the case of a nucleotide: the probability of finding that nucleotide at that position
- For the calculation of the total probability of a chain of events (*i.e.* a succession of certain states), both transition and emission probabilities need to be taken into account (*i.e.* multiplied)

Calculating total probability in a Markov model

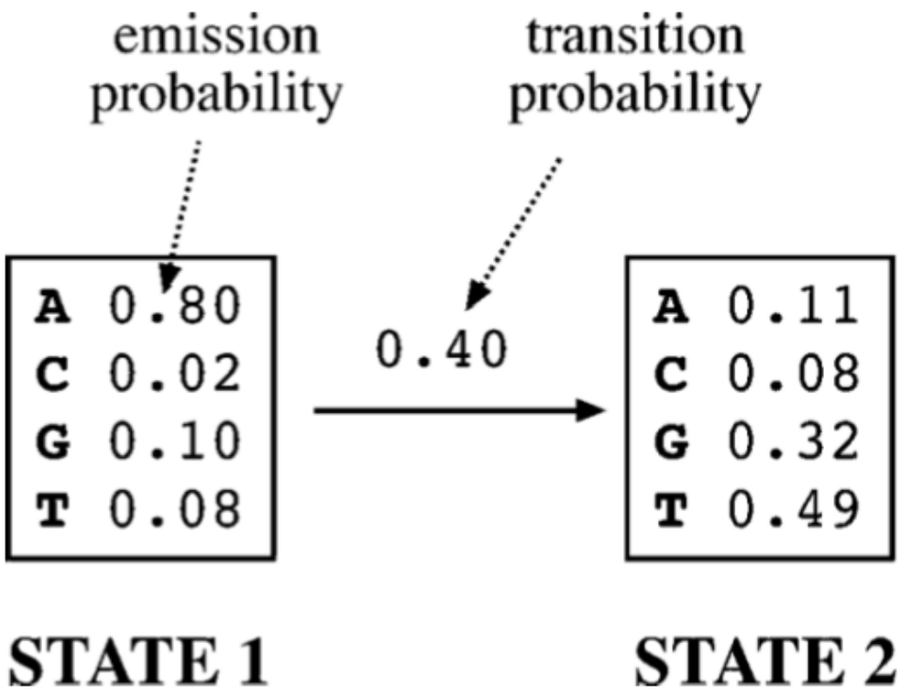


Figure 6.6: Graphic illustration of a simplified partial HMM for DNA sequences with emission and transition probability values. Both probability values are used to calculate the total probability of a particular path of the model. For example, to generate the sequence AG, the model has to progress from A from STATE 1 to G in STATE 2, the probability of this path is $0.80 \times 0.40 \times 0.32 = 0.102$. Obviously, there are $4 \times 4 = 16$ different sequences this simple model can generate. The one that has the highest probability is AT.

Representing MSAs with gaps as HMMs

Three kinds of states: match (there is a residue at that position, described by a frequency distribution that we can actually measure in our MSA), insertion and deletion

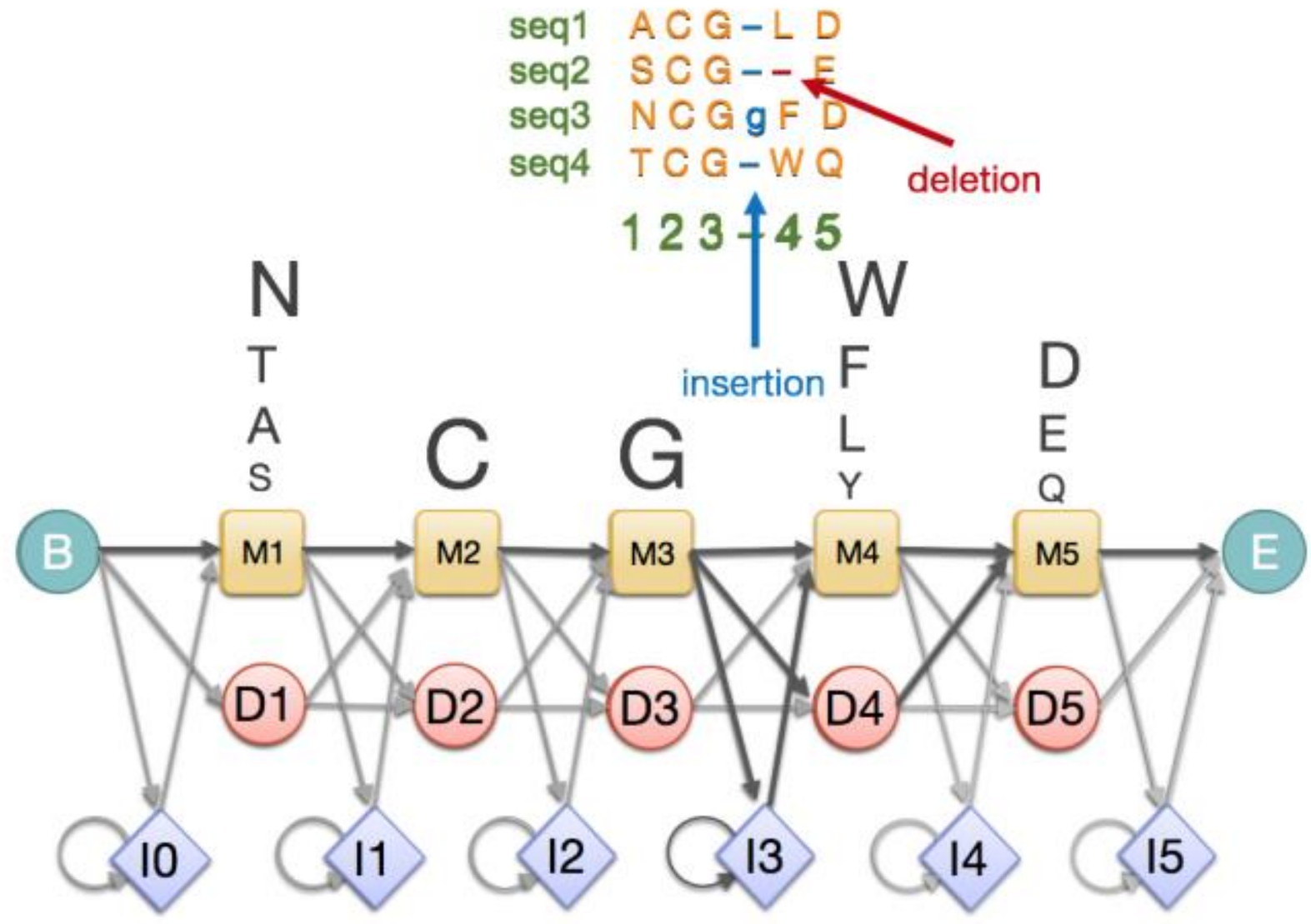
- Match states are considered observed states (as these states comprise frequency information for each possible residue at a particular position)
- Insertions and deletions are considered hidden states (these do not comprise any frequency information, and only play a role in the HMM *via* the transition probabilities to and from them)
- A special graphical representation can be used to more clearly represent the HMM

HMM representing an alignment with 5 amino acids

Actual amino acids: observed states (M, yellow)

Deletions (D, red) and insertions (I, blue): hidden states

Circles indicate self-looping (allows insertion of any number of residues)



Each path beginning from B and ending in E generates a unique sequence with a probability value

Training of an HMM

The process of obtaining the optimal parameters (*i.e.* all of the emission and transition probabilities) for the HMM is called "training":

- **Emission probabilities** for a state (*i.e.* a residue in the sequence) are based on the residue frequencies for a given column of the multiple sequence alignment
- **Transition probabilities** come from the observed frequencies of gaps and insertions (which are considered hidden states) at given positions in the multiple sequence alignment

Once a HMM is established, it can be used to determine how well an unknown sequence matches the model, very much like profiles can

Aligning a sequence to a Hidden Markov Model

- The path that generates the highest probability represents the optimal alignment to the HMM model of the sequence family

- To find the optimal path within a HMM, a matrix of probability values for every state at every residue position needs to be constructed

- *E.g.* the *Viterbi algorithm* can be used to determine the most probable path

Viterbi algorithm

	M_0	I_1	D_1	M_1	I_2	D_2	M_2	I_3	D_3	M_3
S_0	0.26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
S_1	0.00	0.00	0.00	0.19	0.03	0.00	0.00	0.00	0.00	0.00
S_2	0.00	0.00	0.00	0.00	0.00	0.00	0.37	0.01	0.00	0.00
S_3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.42

Figure 6.8: Score matrix constructed from a simple HMM with the optimal score path chosen by the Vertibi algorithm. M, I, and D represent match, insert, and delete states. State 0 (S_0) is the beginning state; S_3 is the end state. The probability value for each state (S) is the maximum emission probability of each state multiplied by the transition probability to that state. The Viterbi algorithm works in a trace-back procedure by traveling from the lower right corner to the upper left corner to find the highest scored path.

Regularisation

- HMM construction often suffers from limited sampling size causing overrepresentation of observed characters while ignoring unobserved characters (= "overfitting")
- "Smoothing" (= "regularization") is needed to ensure that the model is representative for other members of the family and not only for the training set of sequences
  - *E.g.* pseudocount: addition of an extra amino acid not observed in the training set

Regularisation

Automatisation of regularisation by simulating amino acid distribution in a sequence alignment:

- *E.g. Dirichlet* mixture: amino acid distribution derived from prior distribution of amino acids found in a large number of conserved protein domains
- Weighting scheme that gives pseudocounts to amino acids and makes the distribution more reasonable
- Smoothing should not be too strong, so as to not distort the observed sequence patterns in the training set

Applications of HMMs

- HMMs are able to accurately model insertions and deletions (in contrast to profile calculation)
- Handling of insertions and deletions is a major problem in recognising highly divergent sequences
- Hence HMMs have more predictive power and are more robust in describing subtle patterns of a sequence similarity in a family than probability modeling by profiles
- HMM package available at: http://hmmer.wustl.edu/