Hidden Markov Modeling

Michael Wasserman
Quantitative Understanding in Biology
June 3, 2009

Markov Modeling

- Stochastic process
- State machine assumption
- Transitions probabilistic
- Markov assumption
  - Current state depends only on a finite history of previous states
  - 1st order: current state depends only on previous state ➔ $P(X_t | X_{0:t-1}) = P(X_t | X_{t-1})$

Markov Process

- To define 1st order process need
  - States
    - sunny, cloudy, rainy
  - Initial probabilities (π vector)
    - [sunny cloudy rainy] = [0.8 0.1 0.1]
  - Transition probabilities (A matrix)

$$
\begin{array}{ccc} & & Today \\ Yesterday & \begin{array}{c} \\ \text{sun} \\ \text{cloud} \\ \text{rain} \end{array} & \begin{bmatrix} \text{sun} & \text{cloud} & \text{rain} \\ 0.50 & 0.375 & 0.125 \\ 0.25 & 0.125 & 0.625 \\ 0.25 & 0.375 & 0.375 \end{bmatrix} \end{array}
$$

What is a HMM?

- Markov process with unobserved states
- Do observe output dependent on state
  - Hermit forecast weather
- HMM definition
  - Initial probabilities (π vector)
  - Transition probabilities (A matrix)
  - Emission probabilities (B matrix)

		Seaweed			
		Dry	Dryish	Damp	Soggy
weather	Sun	0.60	0.20	0.15	0.05
	Cloud	0.25	0.25	0.25	0.25
	Rain	0.05	0.10	0.35	0.50

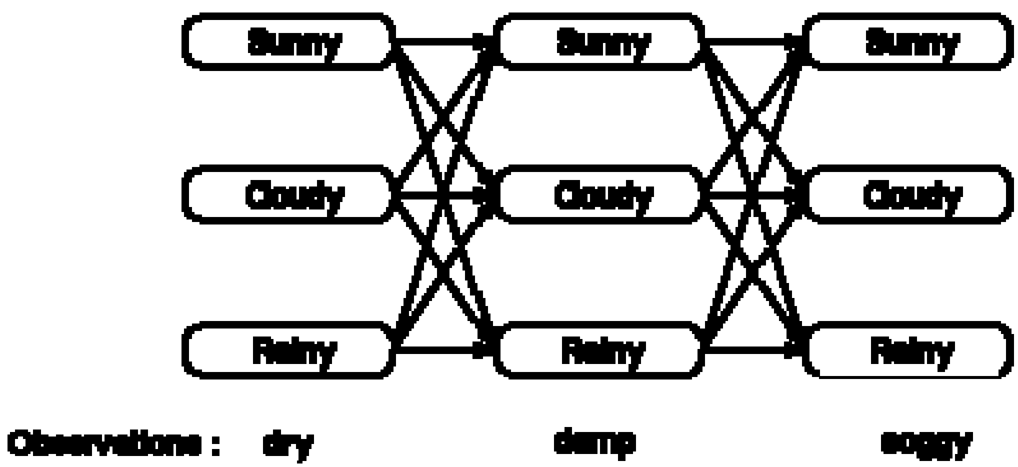
http://www.comp.leeds.ac.uk/roger/HiddenMarkovModels/html_dev/main.html

HMM Uses

- Evaluation
  - Match most likely system to observations
  - Forward algorithm
- Decoding
  - Find most probable sequence of hidden states
  - Viterbi algorithm
- Learning
  - Define HMM parameters to a given data set
  - Forward-backward algorithm

Evaluation

- Probability of observed sequence given HMM
- Number of paths needed increases exponentially with time
- Solution: Use time invariance of probabilities



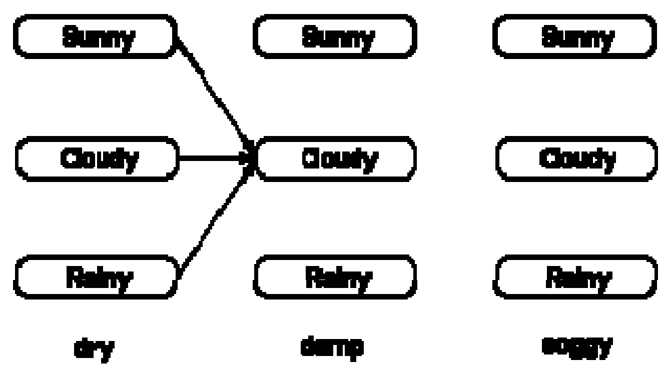
Forward Algorithm

- Reduce complexity with recursion
  - N^T vs N^2T
- Partial probability α = sum of all possible paths to a state
  - $\alpha_{t+1}(j)$= Pr(observation | hidden state is j) x Pr(all paths to state j at time t)

$$\alpha_{t+1}(j) = b_{jk_{t+1}} \sum_{i=1}^{n} \alpha_t(i) a_{ij}$$

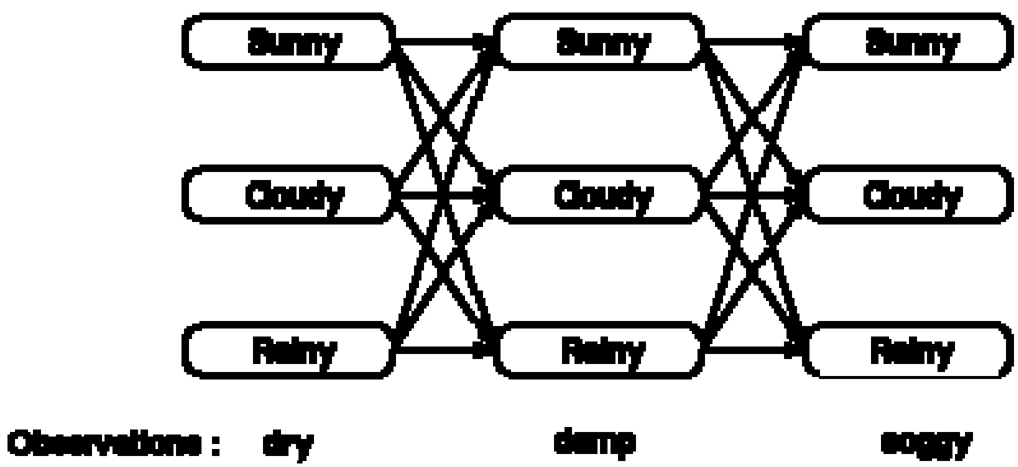
  - Special case: t = 1

$$\alpha_1(j) = \pi(j).b_{jk_1}$$



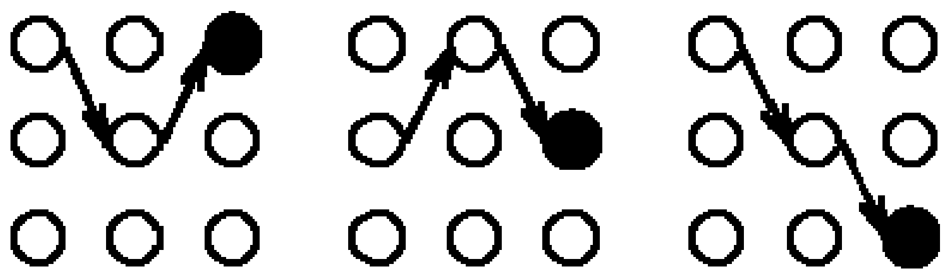
Decoding

- Find most probable sequence of hidden states
  - Maximize Pr(observed sequence | hidden state combination)
  - Again, reduce complexity with recursion!



Viterbi Algorithm

- Partial probability δ = most likely path to a state
- Overall best path = state with the max δ and its partial best path



Partial best paths, each with a δ

Viterbi Algorithm

- Probability of most probable path to state X
  - Pr(X at time t) = $\max_{i=A,B,C}$ Pr(i_{t-1}) x Pr(X|i) x Pr (obs. at time t|X)

    $$\delta_t(i) = \max_j(\delta_{t-1}(j) a_{ji} b_{ik_t})$$

  - Special case: t = 1

    $$\delta_1(i) = \pi(i) b_{ik_1}$$

- If I am here, by what route is it most likely I arrived?
  - "Remember" past best states with backpointers

    $$\phi_t(i) = argmax_j(\delta_{t-1}(j) a_{ji})$$

Viterbi Algorithm

- Termination: determine state at final t

  $$i_t = argmax(\delta_T(i))$$

- Back-track
  - For t = t-1 ➔1

    $$i_t = \phi_{t+1}(i_{t+1})$$

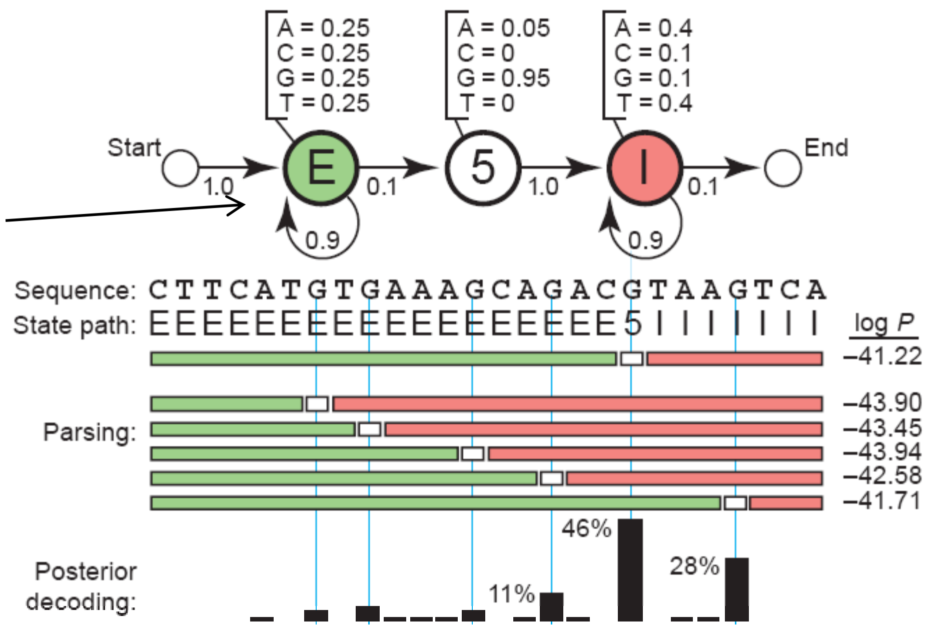
- Advantages
  - Reduce complexity (N^T vs N^2T)
  - Robust to noise
    - Looks at whole sequence before deciding on most likely final state ➔ back-track to best path

HMM Applications

- Speech recognition
- Bioinformatics

Emission probabilities (B)

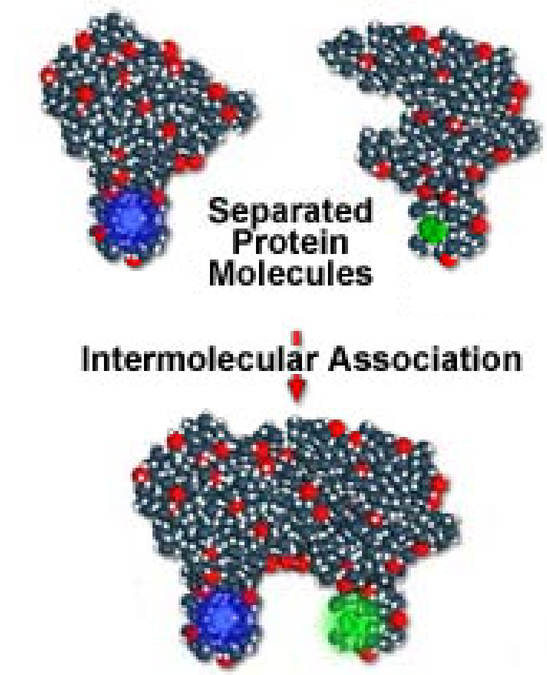
Transition Probabilities (A)



Eddy SR. *Nature Biotechnology* **22**:1315(2004)

HMM Applications

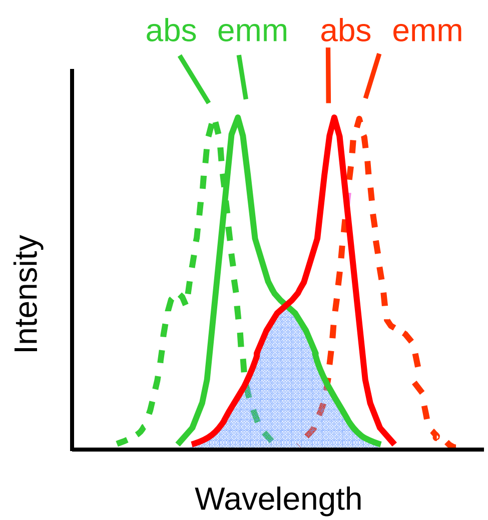
- Single-molecule imaging
  - Molecular movements diffraction-limited (hidden)
  - Infer conformational changes from observed changes in fluorescence

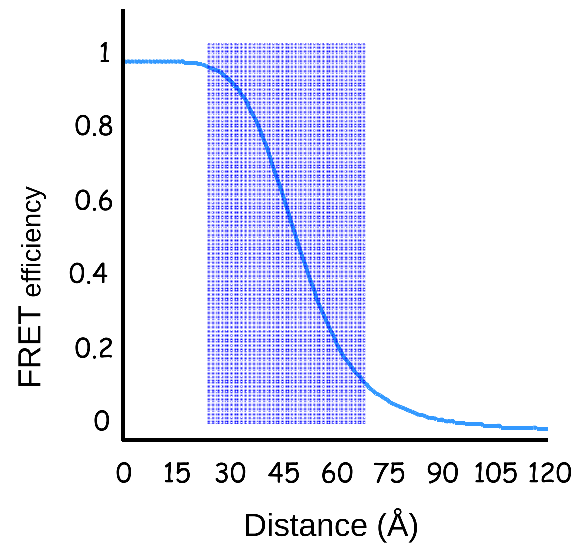


http://www.olympusfluoview.com

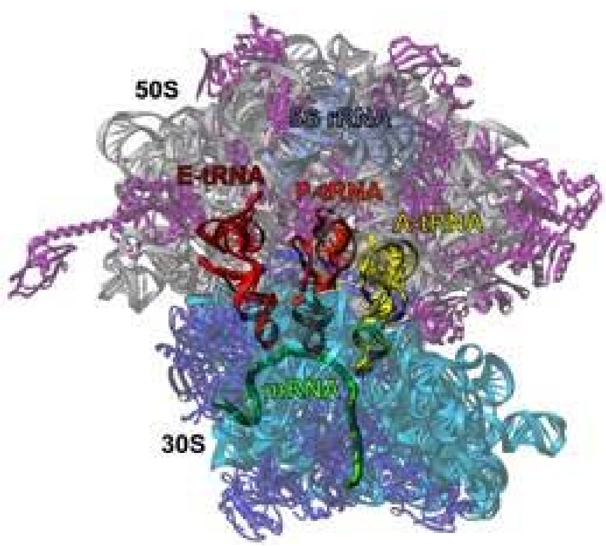
FRET

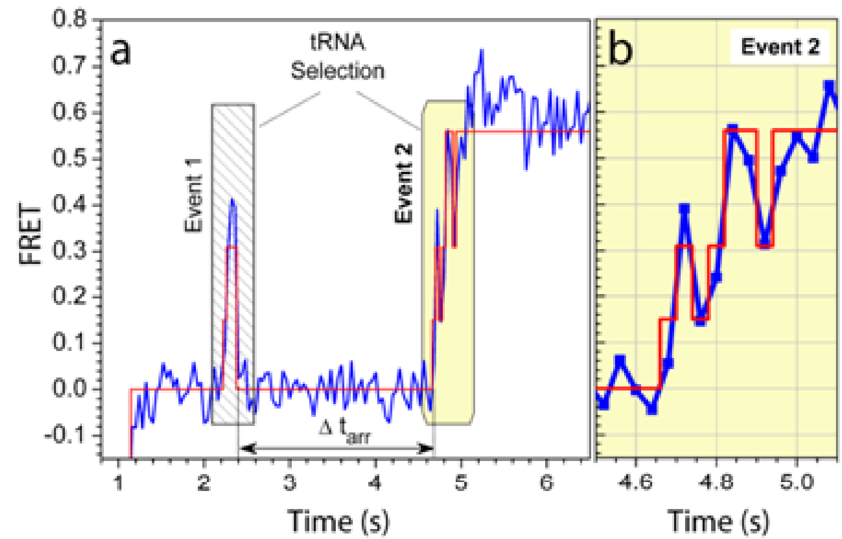
- Fluorescence Resonance Energy Transfer
- Used as a spectroscopic ruler
  - Measure real-time changes in molecular conformations

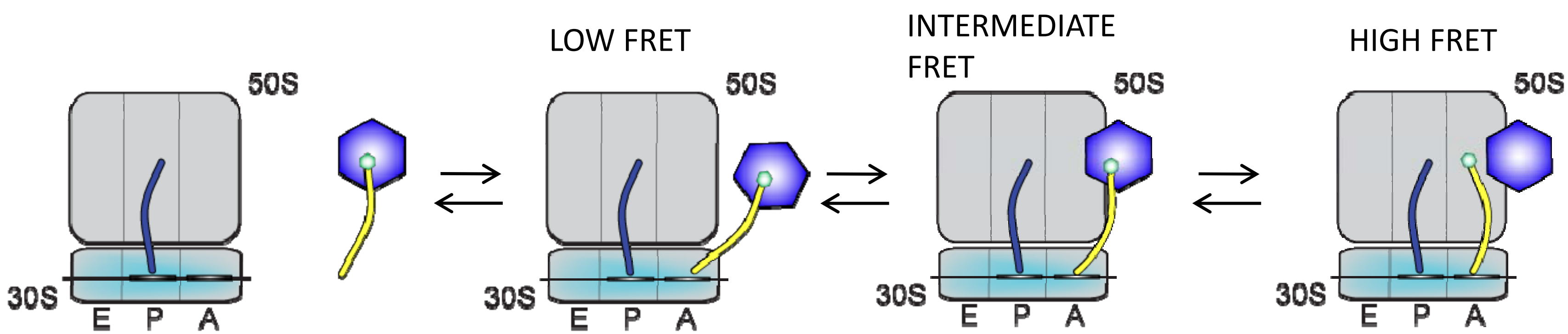




FRET Example - Ribosome

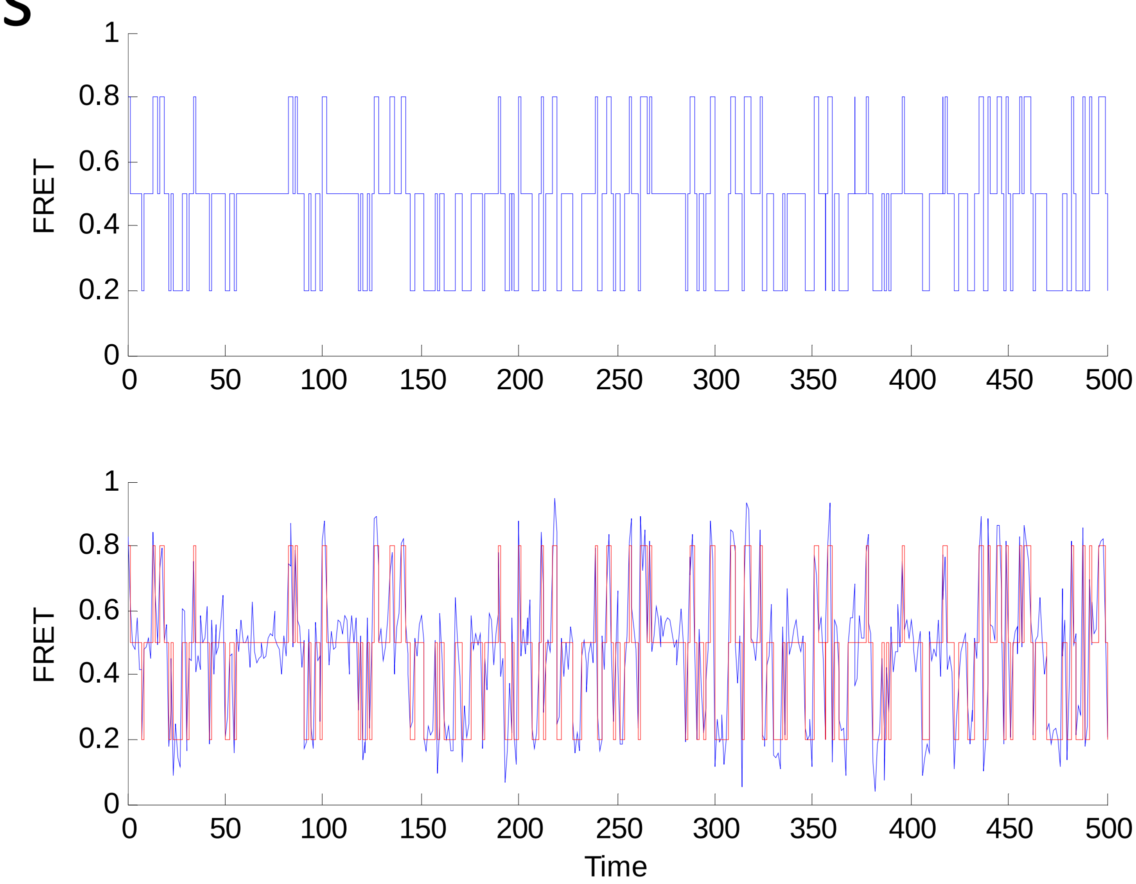






Simulation

- See how well model captures underlying states



Summary

- HMMs are useful to infer underlying Markov states of a system
- Many applications (speech to science)
- Advantage: computationally friendly
- Disadvantage: assumes time invariance in parameters

Additional Questions?