# Hidden Markov Modeling in single-molecule FRET data analysis

Yang Liu, Karin A. Dahmen, Yann Chemla, and Taekjip Ha

Center for the Physics of Living Cell
Department of Physics
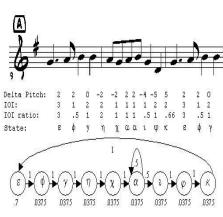University of Illinois at Urbana-Champaign

April 14, 2009

# Applications of HMM

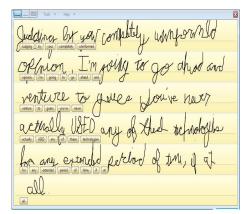Part I: Hidden Markov Modeling  Part II: smFRET data analysis
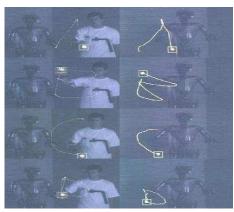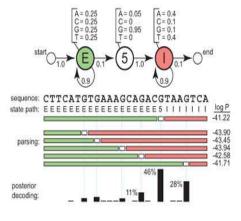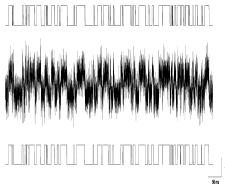


speech recognition



score following



handwriting recognition
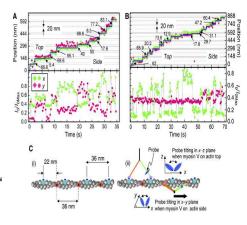


gesture recognition
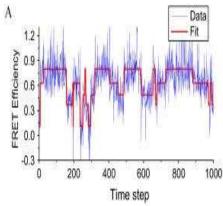


DNA sequence,



ion channel recording,



molecular motor,



smFRET data analysis

# Outline
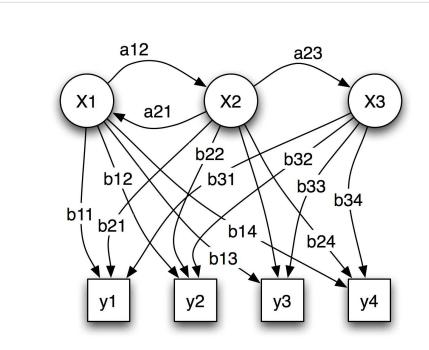
1. **Introduction : HMM**

   - Simple Example: Urn and Ball
   - Key Assumptions of HMM
   - Basic Elements of HMM
   - Basic Problems and Solutions

2. **Application: smFRET data analysis**

   - Goal
   - univariate HMM (UHMM)
   - multivariate HMM (MHMM)
   - reliability tests
   - conclusions
   - outlook



Probabilistic parameters of a HMM.
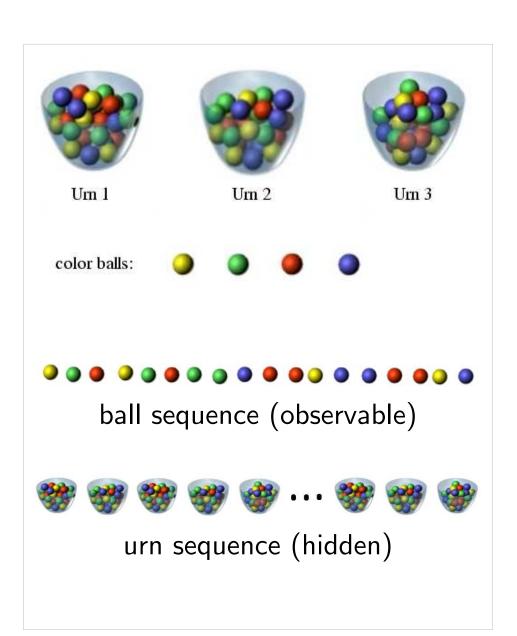
# Part I: Hidden Markov Modeling

# Urn and Ball Problem

ball sequence (observable)

urn sequence (hidden)
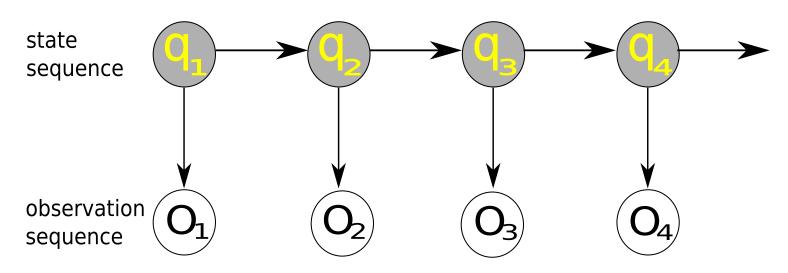
- $N = 3$ urns containing colored balls
- $M = 4$ distinct colors of balls
- Each urn has a possibly different distribution of colors
- ball sequence generation rule:

  1. pick initial urn according to some random process
  2. randomly pick a ball from the urn and then replace it
  3. select another urn according to a random selection process associated with the urn
  4. repeat steps 2 and 3

Given a ball sequence, what's the most probable urn sequence?

# Key Assumptions of HMM

Part I: Hidden Markov Modeling  Part II: smFRET data analysis



A standard hidden Markov model.

# Key Assumptions of HMM

Part I: Hidden Markov Modeling   Part II: smFRET data analysis



A standard hidden Markov model.

# Key Assumptions of HMM

A standard hidden Markov model.

- **state sequence: Markov process (with $N$ hidden states $\{1, \cdots, N\}$)**

  - first-order Markov chain : $P(q_t|q_{t-1}, o_{t-1}; \cdots; q_1, o_1) = P(q_t|q_{t-1})$
  - time-independent transition matrix : $P(q_t = j|q_{t-1} = i) = a_{ij}$
  - initial state distribution : $\pi_i = P(q_1 = i)$

# Key Assumptions of HMM

A standard hidden Markov model.

- **state sequence: Markov process (with $N$ hidden states $\{1, \cdots, N\}$)**

    - first-order Markov chain : $P(q_t | q_{t-1}, o_{t-1}; \cdots; q_1, o_1) = P(q_t | q_{t-1})$
    - time-independent transition matrix : $P(q_t = j | q_{t-1} = i) = a_{ij}$
    - initial state distribution : $\pi_i = P(q_1 = i)$

- **observation sequence**

    - $o_t$ depends on $q_t$ only: $P(o_t | q_t; q_{t-1}, o_{t-1}; \cdots; q_1, o_1) = P(o_t | q_t)$
    - observation probability distribution: $b_j(o) = P(o_t = o | q_t = j)$

**parameter set :** $\lambda = (A, B, \pi)$

Part I: Hidden Markov Modeling  Part II: smFRET data analysis



probabilistic parameters of a discrete HMM

- $N$ : number of (hidden) states.
- $M$ : number of symbols.
- $\pi$ : initial state distribution ($1 \le i \le N$):

$$\pi_i = P(q_1 = i)$$

- $A$ : state-transition probability matrix ($1 \le i, j \le N$):

$$a_{ij} = P(q_{t+1} = j | q_t = i)$$
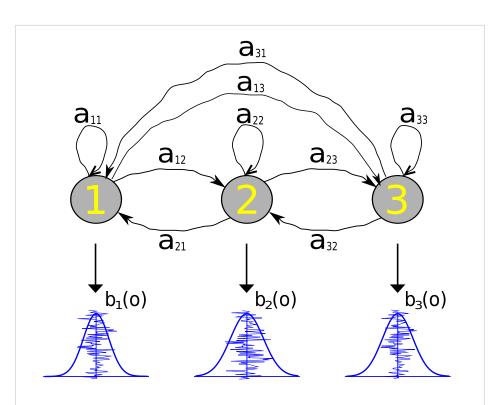
- $B$ : observation probability distribution ($1 \le j \le N$, $1 \le k \le M$):

$$b_j(k) = P(O_t = k | q_t = j)$$

**parameter set :** $\lambda = (A, B, \pi)$

Part I: Hidden Markov Modeling  Part II: smFRET data analysis



probabilistic parameters of a
continuous HMM

- $N$ : number of (hidden) states.
- $\pi$ : initial state distribution ($1 \leq i \leq N$):

$$\pi_i = P(q_1 = i)$$

- $A$ : state-transition probability matrix ($1 \leq i, j \leq N$):

$$a_{ij} = P(q_{t+1} = j | q_t = i)$$

- $B$ : observation probability density function ($1 \leq j \leq N$):

$$b_j(o) = \mathcal{N}(o, \mu_j, U_j)$$

i.e. Gaussian distribution with mean $\mu_j$ and variance $U_j$.

**Notations**:

complete parameter set of HMM : $\lambda = (A, B, \pi)$;

state sequence : $Q = (q_1, q_2, \cdots, q_T)$;

observation sequence : $O = (o_1, o_2, \cdots, o_T)$.

**Notations**:

complete parameter set of HMM : $\lambda = (A, B, \pi)$;

state sequence : $Q = (q_1, q_2, \cdots, q_T)$;

observation sequence : $O = (o_1, o_2, \cdots, o_T)$.

**Three basic problems and solutions**:

1. *"Evaluation"*: Given the observation sequence $O$ and a model $\lambda$, how do we efficiently compute $P(O|\lambda)$, the probability of the observation sequence, given the model?

   *Forward-backward algorithm* : calculate $P(O|\lambda)$.

**Notations**:

complete parameter set of HMM : $\lambda = (A, B, \pi)$;

state sequence : $Q = (q_1, q_2, \cdots, q_T)$;

observation sequence : $O = (o_1, o_2, \cdots, o_T)$.

**Three basic problems and solutions**:

1. *"Evaluation"*: Given the observation sequence $O$ and a model $\lambda$, how do we efficiently compute $P(O|\lambda)$, the probability of the observation sequence, given the model?

   *Forward-backward algorithm* : calculate $P(O|\lambda)$.

2. *"Decoding"*: Given the observation sequence $O$ and a model $\lambda$, how do we choose a corresponding state sequence $Q$ which is optimal in some meaningful sense (i.e. best "explains" the observations)?

   *Viterbi algorithm* : find the state sequence which maximizes $P(Q, O|\lambda)$. This state sequence is called the **Viterbi path**.

# Basic Problems and Solutions

**Notations**:

complete parameter set of HMM : $\lambda = (A, B, \pi)$;

state sequence : $Q = (q_1, q_2, \cdots, q_T)$;

observation sequence : $O = (o_1, o_2, \cdots, o_T)$.

**Three basic problems and solutions**:

1. *"Evaluation"*: Given the observation sequence $O$ and a model $\lambda$, how do we efficiently compute $P(O|\lambda)$, the probability of the observation sequence, given the model?
   *Forward-backward algorithm* : calculate $P(O|\lambda)$.

2. *"Decoding"*: Given the observation sequence $O$ and a model $\lambda$, how do we choose a corresponding state sequence $Q$ which is optimal in some meaningful sense (i.e. best "explains" the observations)?
   *Viterbi algorithm* : find the state sequence which maximizes $P(Q, O|\lambda)$. This state sequence is called the **Viterbi path**.

3. *"Learning"*: Given the observation sequence $O$, how do we adjust the model parameters $\lambda$ to maximize $P(O|\lambda)$?
   *Baum-Welch algorithm* : iteratively reestimate the hidden states by their expected values $\Rightarrow$ locally maximize $P(O|\lambda)$.

**brute force solution : enumerate paths and sum probabilities ?**

$$P(O|\lambda) = \sum_{q \in \mathcal{Q}} P(O|q, \lambda) \, P(q|\lambda)$$

$\mathcal{Q}$ : the space of all state sequences of length $T$

**brute force solution : enumerate paths and sum probabilities ?**

$$P(O|\lambda) = \sum_{q \in \mathcal{Q}} P(O|q, \lambda) \; P(q|\lambda)$$

$\mathcal{Q}$ : the space of all state sequences of length $T$

$$P(O|q, \lambda) = b_{q_1}(o_1) b_{q_2}(o_2) \cdots b_{q_T}(o_T)$$
$$P(q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \cdots a_{q_{T-1} q_T}$$

**brute force solution : enumerate paths and sum probabilities ?**

$$P(O|\lambda) = \sum_{q \in \mathcal{Q}} P(O|q, \lambda) \; P(q|\lambda)$$

$\mathcal{Q}$ : the space of all state sequences of length $T$

$$P(O|q, \lambda) = b_{q_1}(o_1) b_{q_2}(o_2) \cdots b_{q_T}(o_T)$$
$$P(q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \cdots a_{q_{T-1} q_T}$$

$$P(O|\lambda) = \sum_{q \in \mathcal{Q}} \pi_{q_1} b_{q_1}(o_1) a_{q_1 q_2} b_{q_2}(o_2) \cdots a_{q_{T-1} q_T} b_{q_T}(o_T)$$

# Problem 1 : Evaluation $P(O|\lambda)$

Part I: Hidden Markov Modeling  Part II: smFRET data analysis

**brute force solution : enumerate paths and sum probabilities ?**

$$P(O|\lambda) = \sum_{q \in \mathcal{Q}} P(O|q, \lambda) \, P(q|\lambda)$$

$\mathcal{Q}$ : the space of all state sequences of length $T$

$$P(O|q, \lambda) = b_{q_1}(o_1) b_{q_2}(o_2) \cdots b_{q_T}(o_T)$$
$$P(q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \cdots a_{q_{T-1} q_T}$$

$$P(O|\lambda) = \sum_{q \in \mathcal{Q}} \pi_{q_1} b_{q_1}(o_1) a_{q_1 q_2} b_{q_2}(o_2) \cdots a_{q_{T-1} q_T} b_{q_T}(o_T)$$
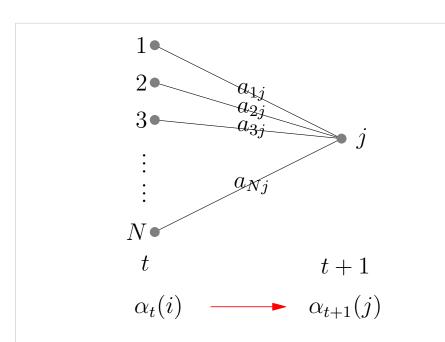
$N^T$ state sequences of length $T \implies$ time complexity : $\mathcal{O}(N^T T)$
Typical values: $N = 2 \sim 10$, $T = 100 \sim 5000$.

HMM                                                                 Yang Liu – 11 / 54

Part I: Hidden Markov Modeling  Part II: smFRET data analysis

## Forward Algorithm



$$\alpha_t(i) \xrightarrow{\hspace{2cm}} \alpha_{t+1}(j)$$

**forward variable:**

the probability of observing the partial sequence $(o_1, o_2, \cdots, o_t)$ such that state $q_t$ is $i$ :

$$\alpha_t(i) = P(o_1, o_2, \cdots, o_t, q_t = i|\lambda)$$

1.  Initialization:

$$\alpha_1(i) = P(o_1, q_1 = i|\lambda) = \pi_i b_i(o_1)$$

2.  Induction:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(o_{t+1})$$

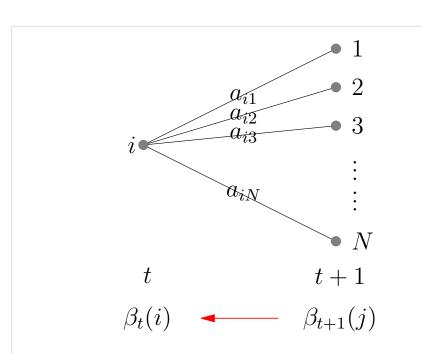3.  Termination:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

Time complexity : $\mathcal{O}(N^2 T)$

## Backward Algorithm



**backward variable**:
the probability of observing the partial sequence $(o_{t+1}, o_{t+2}, \cdots, o_T)$ such that state $q_t$ is $i$ :

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \cdots, o_T | q_t = i, \lambda)$$

1.  Initialization:

$$\beta_T(i) = 1$$

2.  Induction:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)$$

3.  Termination:

$$P(O|\lambda) = \sum_{i=1}^{N} \pi_i b_i(o_1) \beta_1(i)$$

Time complexity : $\mathcal{O}(N^2 T)$

Important relation:

$$P(O|\lambda) = \sum_{i=1}^{N} P(O, q_t = i|\lambda) = \sum_{i=1}^{N} \alpha_t(i)\ \beta_t(i)$$

For example:

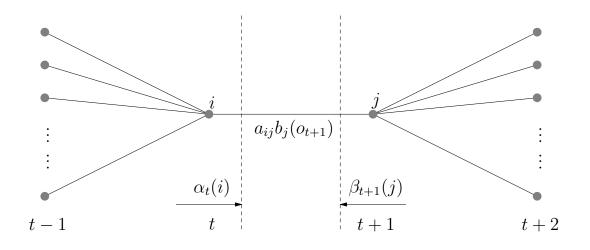$$t = T \qquad\qquad P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \cdot 1$$

$$t = 1 \qquad\qquad P(O|\lambda) = \sum_{i=1}^{N} \alpha_1(i) \cdot \beta_1(i)$$

# $\xi_t(i,j)$ and $\gamma_t(j)$

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j | O, \lambda)$$

$$= \frac{P(q_t = i, q_{t+1} = j, O | \lambda)}{P(O | \lambda)}$$

$$= \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \alpha_t(i) \beta_t(i)}$$

is the probability of being in state $i$ at time $t$ and state $j$ at time $t+1$, given $O$ and $\lambda$.

$$\gamma_t(i) = P(q_t = i | O, \lambda)$$
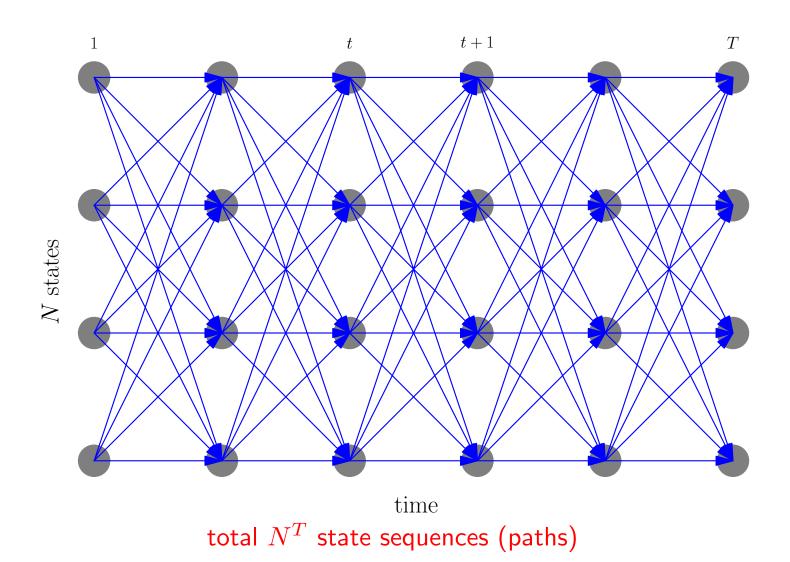
$$= \sum_{j=1}^{N} \xi_t(i,j)$$

$$= \frac{\alpha_t(i) \beta(i)}{\sum_{i=1}^{N} \alpha_t(i) \beta_t(i)}$$

is the probability of being in state $i$ at time $t$, given $O$ and $\lambda$.

# find state sequence $Q$ to maximize $P(Q, O | \lambda)$

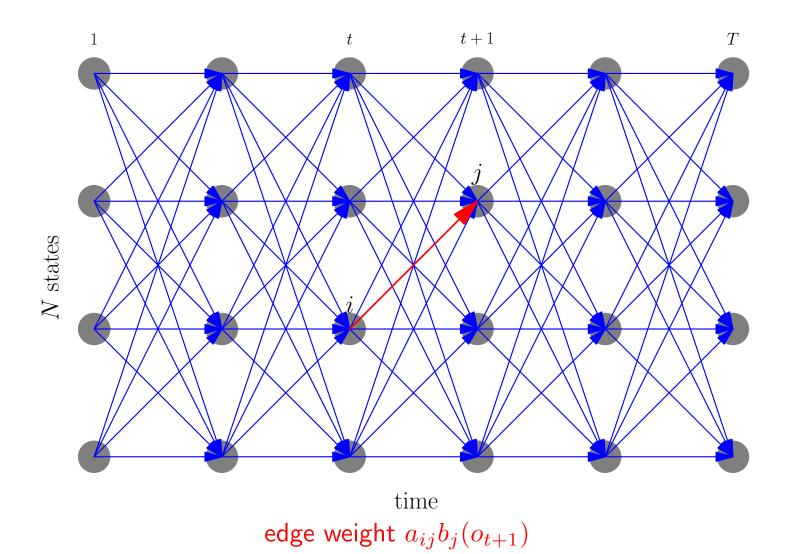Part I: Hidden Markov Modeling   Part II: smFRET data analysis



total $N^T$ state sequences (paths)

## find state sequence $Q$ to maximize $P(Q, O|\lambda)$

Part I: Hidden Markov Modeling   Part II: smFRET data analysis
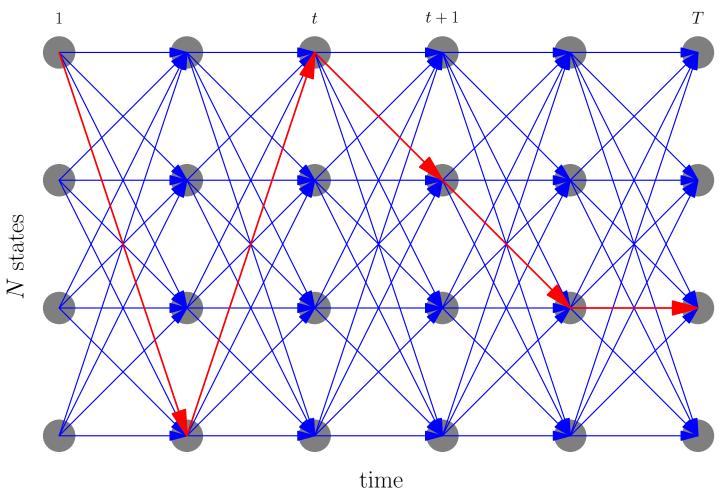


edge weight $a_{ij}b_j(o_{t+1})$

**find state sequence $Q$ to maximize $P(Q, O | \lambda)$**
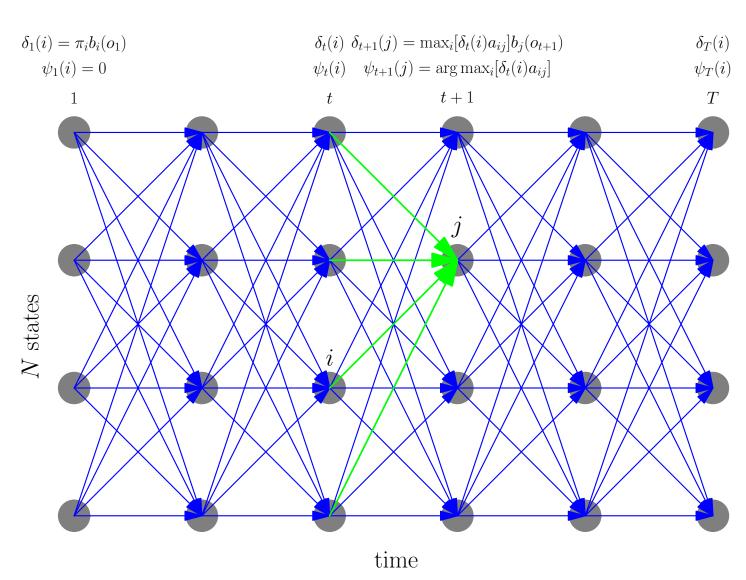Part I: Hidden Markov Modeling   Part II: smFRET data analysis



Find the "longest" path with "length" defined as products of its edges' weights

## find state sequence $Q$ to maximize $P(Q, O|\lambda)$

### Part I: Hidden Markov Modeling  Part II: smFRET data analysis

$$\delta_1(i) = \pi_i b_i(o_1)$$
$$\psi_1(i) = 0$$

$$\delta_t(i) \quad \delta_{t+1}(j) = \max_i [\delta_t(i) a_{ij}] b_j(o_{t+1})$$
$$\psi_t(i) \quad \psi_{t+1}(j) = \arg\max_i [\delta_t(i) a_{ij}]$$

$$\delta_T(i)$$
$$\psi_T(i)$$



Viterbi algorithm : dynamic programming

## Viterbi algorithm

Define :

$$\delta_t(i) = \max_{q_1, q_2, \cdots, q_{t-1}} P(q_1, q_2, \cdots, q_t = i, o_1, o_2, \cdots, o_t | \lambda)$$

$\delta_t(i)$ is the highest probability along a single path at time t, which accounts for the first $t$ observations and ends in state $q_t = i$.

Induction:

$$\delta_{t+1}(j) = \max_i [\delta_t(i) a_{ij}] \ b_j(o_{t+1})$$

To retrieve the state sequence, we need to keep track of the argument which maximized $\delta_{t+1}(j)$ for each $t$ and $j$. We do this via the array $\psi_t(j)$.

# Problem 2 : Decoding
## find state sequence $Q$ to maximize $P(Q, O | \lambda)$
Part I: Hidden Markov Modeling   Part II: smFRET data analysis

1.   Initialization:

$$\delta_1(i) = \pi_i b_i(o_1)$$
$$\psi_1(i) = 0$$

2.   Recursion:

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(o_t)$$
$$\psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}]$$

3.   Termination:

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)]$$
$$q_T^* = \arg \max_{1 \leq i \leq N} [\delta_T(i)]$$

4.   Path backtracking: $q_t^* = \psi_{t+1}(q_{t+1}^*)$   with   $t = T-1, T-2, \cdots, 1$

Unfortunately, no analytic or exact method because of complexity. $\implies$
Iterative solution: **Baum-Welch algorithm**

1. Initial estimation (guess) : $\lambda$
2. Reestimation: compute new $\tilde{\lambda}$ based on $\lambda$ and observation $O$
3. If $\log P(O|\tilde{\lambda}) - \log P(O|\lambda) < \text{DELTA}$ stop
4. Else set $\lambda = \tilde{\lambda}$ and goto step 2.

Unfortunately, no analytic or exact method because of complexity. $\implies$
Iterative solution: **Baum-Welch algorithm**

1. Initial estimation (guess) : $\lambda$
2. Reestimation: compute new $\tilde{\lambda}$ based on $\lambda$ and observation $O$
3. If $\log P(O|\tilde{\lambda}) - \log P(O|\lambda) < \text{DELTA}$ stop
4. Else set $\lambda = \tilde{\lambda}$ and goto step 2.

Consider **Baum's auxiliary function** :

$$Q(\lambda, \tilde{\lambda}) \equiv \sum_{q \in \mathcal{Q}} P(O, q|\lambda) \log P(O, q|\tilde{\lambda})$$

- An elegant proof using Jensen's inequality shows that if $Q(\lambda, \tilde{\lambda}) > Q(\lambda, \lambda)$ then $P(O|\tilde{\lambda}) > P(O|\lambda)$, i.e. the maximization of $Q(\lambda, \tilde{\lambda})$ leads to increased likelihood.
- Maximization of $Q(\lambda, \tilde{\lambda})$ can be exactly derived $\implies$ reestimation formulas.
- Reestimation procedure guarantees that $P(O|\tilde{\lambda}) > P(O|\lambda)$ unless $\tilde{\lambda} = \lambda$.
- But it leads to local maxima only!

# Problem 3 : "Learning"

**maximize** $Q(\lambda, \tilde{\lambda})$

$$Q(\lambda, \tilde{\lambda}) \equiv \sum_{q \in \mathcal{Q}} P(O, q | \lambda) \log P(O, q | \tilde{\lambda})$$

Use

$$P(O, q | \tilde{\lambda}) = \tilde{\pi}_{q_1} \tilde{b}_{q_1}(o_1) \prod_{t=1}^{T-1} \tilde{a}_{q_t \, q_{t+1}} \tilde{b}_{q_{t+1}}(o_{t+1})$$

then

$$Q(\lambda, \tilde{\lambda}) = \sum_{q \in \mathcal{Q}} \log \tilde{\pi}_{q_1} \, P(O, q | \lambda) + \sum_{q \in \mathcal{Q}} \sum_{t=1}^{T-1} \log \tilde{a}_{q_t \, q_{t+1}} \, P(O, q | \lambda) + \sum_{q \in \mathcal{Q}} \sum_{t=1}^{T} \log \tilde{b}_{q_t}(o_t) \, P(O, q | \lambda)$$

We can optimize each term individually, standard constrained optimization techniques, i.e. the Lagrange multiplier method.

Consider the first term

$$\sum_{q \in \mathcal{Q}} \log \tilde{\pi}_{q_1} \; P(O, q | \lambda) = \sum_{i=1}^{N} \log \tilde{\pi}_i \; P(O, q_1 = i | \lambda)$$

Adding the Lagrange multiplier $\theta$, using the constraint that $\sum_{i=1}^{N} \tilde{\pi}_i = 1$, and setting the derivative with respect to $\tilde{\pi}_i$ to be zero, we have

$$\frac{\partial}{\partial \tilde{\pi}_i} \left[ \sum_{i=1}^{N} \log \tilde{\pi}_i \; P(O, q_1 = i | \lambda) + \theta \left( \sum_{i=1}^{N} \tilde{\pi}_i - 1 \right) \right] = 0$$

Taking the derivative yields $\frac{P(O, q_1 = i | \lambda)}{\tilde{\pi}_i} + \theta = 0$. Rearranging the terms and summing over $i$ yields $\theta = -\sum_{i=1}^{N} P(O, q_1 = i | \lambda) = -P(O | \lambda)$. One then solves for $\tilde{\pi}_i$:

$$\tilde{\pi}_i = \frac{P(O, q_1 = i | \lambda)}{P(O | \lambda)} = P(q_1 = i | O, \lambda)$$

## maximize $Q(\lambda, \tilde{\lambda})$
### Part I: Hidden Markov Modeling  Part II: smFRET data analysis

Reestimation formulas (for Gaussian observations):

$$\tilde{\pi}_i = \gamma_1(i)$$

$$\tilde{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$\tilde{\mu}_i = \frac{\sum_{t=1}^{T} \gamma_t(i) \, o_t}{\sum_{t=1}^{T} \gamma_t(i)}$$

$$\tilde{\sigma}_i^2 = \frac{\sum_{t=1}^{T} \gamma_t(i) \, (o_t - \tilde{\mu}_i)^2}{\sum_{t=1}^{T} \gamma_t(i)}$$

with

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j | O, \lambda)$$
$$\gamma_t(i) = P(q_t = i | O, \lambda)$$

Note that

$$\xi_t(i,j) = \frac{P(q_t = i, q_{t+1} = j, O|\lambda)}{P(O|\lambda)}$$

$$= \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \alpha_t(i) \beta_t(i)}$$

is the probability of being in state $i$ at time $t$ and state $j$ at time $t+1$, given $O$ and $\lambda$. And

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j)$$

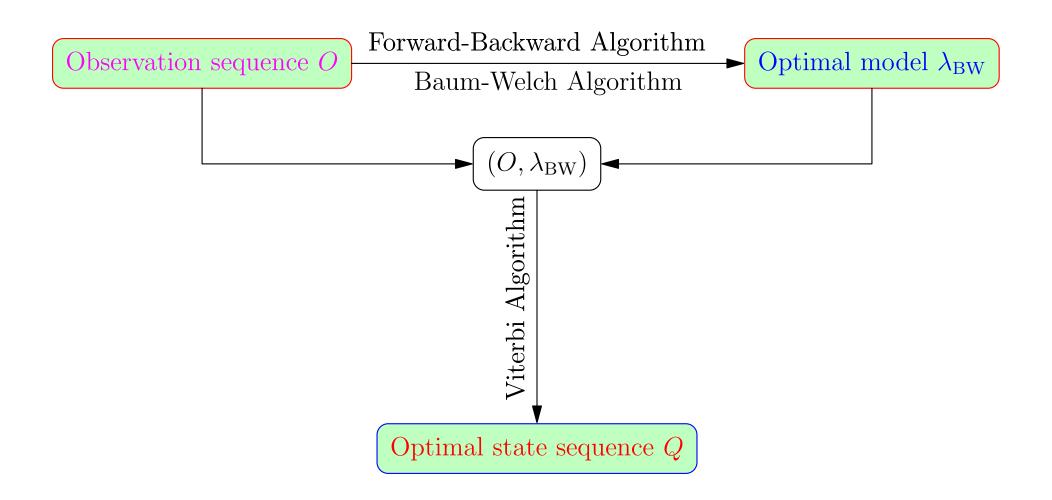is the probability of being in state $i$ at time $t$, given $O$ and $\lambda$.

*"Expectation-Maximization Algorithm"*

# Flowchart

# Part II: smFRET data analysis
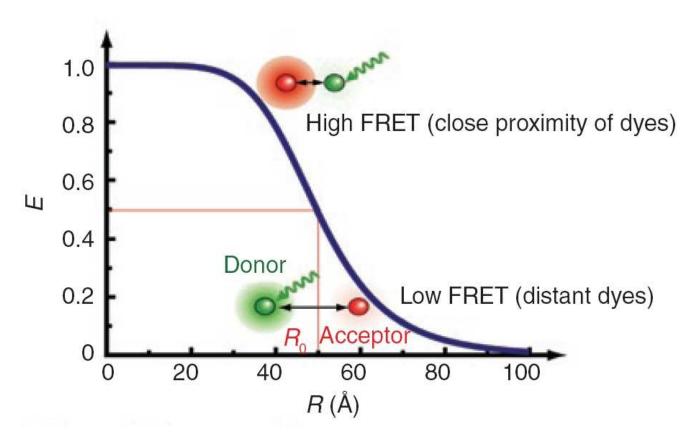
# single-molecule FRET

FRET efficiency $E$, as a function of inter-dye distance $R$:

$$E = [1 + (R/R_0)^6]^{-1}$$
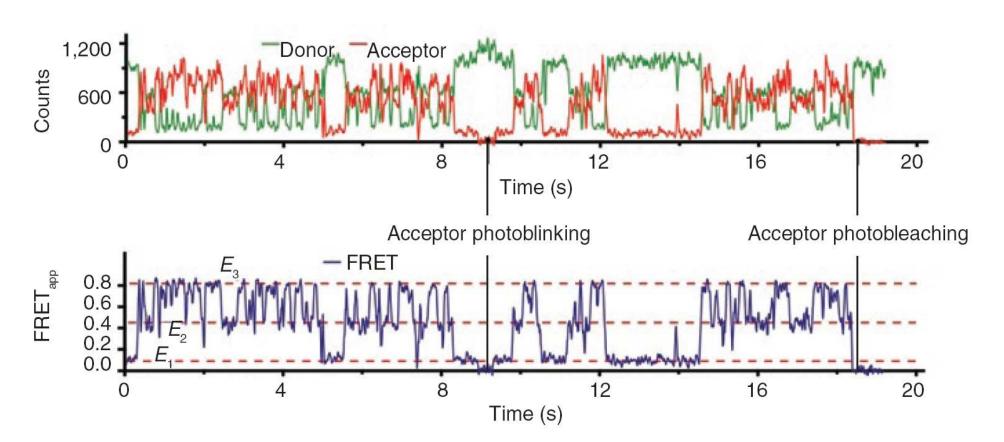
(R. Roy, S. Hohng & T.Ha, Nature Methods 2008)

Two-channel signal: $I_A$ and $I_D \Rightarrow$ FRET efficiency:
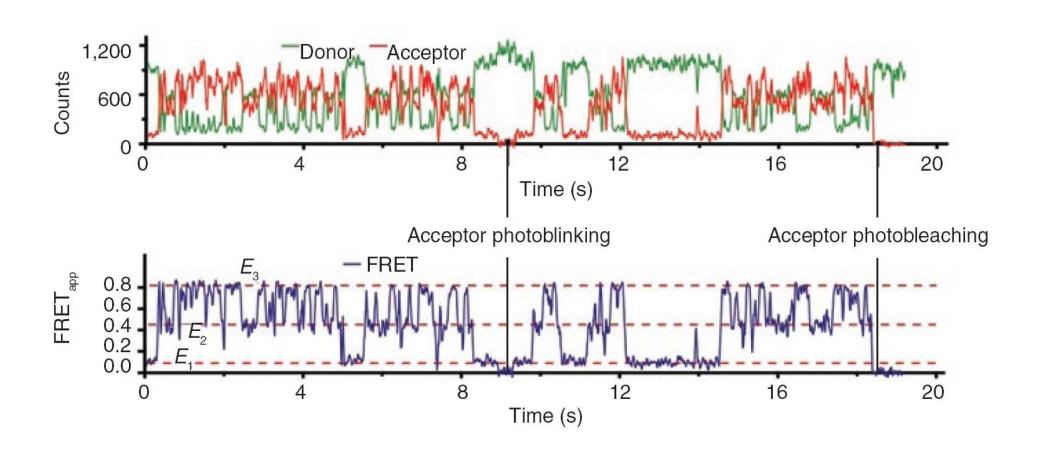
$$E_{\text{app}} = \frac{I_A}{I_A + I_D}$$

# single-molecule FRET

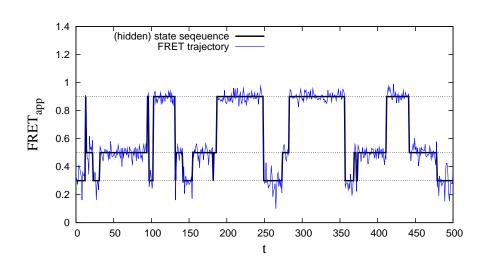**Goal :**   Analyze the noisy data in a <span style="color:red">probabilistic</span> and <span style="color:red">less user-dependent</span> way!

⇓
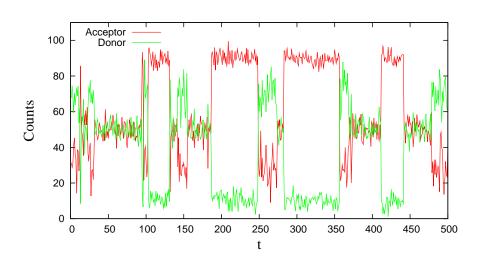
## Hidden Markov Modeling

# Status

**Previous: Analyze the FRET trajectory using univariate HMM.**

S.A. McKinney, C. Joo and T. Ha, *Analysis of single molecule FRET trajectories using hidden Markov modeling*, Biophysical Journal (2006)

"HaMMy": FRET analysis package, written by Sean McKinney

**Now: Analyze the original two channels using multivariate HMM.**

Yang Liu, Karin A. Dahmen, Yann Chemla, and Taekjip Ha, *Multivariate Hidden Markov Modeling in analysis of Single-molecule FRET*, to be submitted to J. Phys. Chem. B

"EHaMMy": extended FRET analysis package, written by Yang Liu

# Probabilistic Parameters

## Univariate HMM



## Multivariate HMM

| number of hidden states | $N$ | |
|---|---|---|
| state-transition probability | $a_{ij}$ | |
| observation probability distribution | $b_i(o)$ | $b_i(\vec{o})$ |
| initial state distribution | $\pi_i$ | |

# Probabilistic Parameters

## Univariate HMM

$$b_i(o) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{(o - \mu_i)^2}{2\sigma_i^2}\right)$$

1. more realistic one: $\beta$-distribution, but simple Gaussian works very well
2. $o$ : the observed FRET value.
3. mean value : the idealized FRET value for state $i$

$$\mu_i = \text{FRET}_i$$

## Multivariate HMM

$$b_i(\vec{o}) = \prod_{k=1}^{d} \frac{e^{-\mu_{i,k}} \mu_{i,k}^{o_{t,k}}}{o_{t,k}!}$$

with $d = 2$ and constraints

$$\mu_{i,1} + \mu_{i,2} = \langle I_A \rangle_i + \langle I_D \rangle_i = I_{\text{tot}i}$$

1. instrument noise mainly due to shot noise
2. noises of two channels are uncorrelated

3. $\vec{o}(t) = (o_{t,1}, o_{t,2}) = (I_A(t), I_D(t))$
4. $\vec{\mu}_i = (\mu_{i,1}, \mu_{i,2}) = (\langle I_A \rangle_i, \langle I_D \rangle_i)$
5. constraints due to anti-correlated feature of $\langle I_A \rangle_i$ and $\langle I_D \rangle_i$

# Reestimation Formulas

## Univariate HMM

$$b_i(o) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{(o-\mu_i)^2}{2\sigma_i^2}\right)$$

$$\tilde{\mu}_i = \frac{\sum_{t=1}^{T} \gamma_t(i)\, o_t}{\sum_{t=1}^{T} \gamma_t(i)}$$

$$\tilde{\sigma}_i^2 = \frac{\sum_{t=1}^{T} \gamma_t(i)\, (o_t - \tilde{\mu}_i)^2}{\sum_{t=1}^{T} \gamma_t(i)}$$

## Multivariate HMM

$$b_i(\vec{o}) = \prod_{k=1}^{d} \frac{e^{-\mu_{i,k}} \mu_{i,k}^{o_{t,k}}}{o_{t,k}!}$$

with constraints $\mu_{i,1} + \mu_{i,2} = I_{\text{tot}\,i}$

$$\tilde{\mu}_{i,k} = \frac{I_{\text{tot}\,i} \sum_{t=1}^{T} o_{t,k}\, \gamma_t(i)}{\sum_{t=1}^{T} \left(\sum_{k=1}^{d} o_{t,k}\right) \gamma_t(i)}$$

$$\tilde{\pi}_i = \gamma_1(i)$$

$$\tilde{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

# Reestimation Formulas

## multiple independent sequences $\{O^{(1)}, O^{(2)}, \cdots, O^{(M)}\}$

Part I: Hidden Markov Modeling   Part II: smFRET data analysis

### Univariate HMM

$$b_i(o) = \frac{1}{\sqrt{2\pi}\sigma_i} \, \exp\left(-\frac{(o - \mu_i)^2}{2\sigma_i^2}\right)$$

$$\tilde{\mu}_i = \frac{\sum_{m=1}^{M} \sum_{t=1}^{T_m} \gamma_t^{(m)}(i) \, o_t^{(m)}}{\sum_{m=1}^{M} \sum_{t=1}^{T_m} \gamma_t^{(m)}(i)}$$

$$\tilde{\sigma}_i^2 = \frac{\sum_{m=1}^{M} \sum_{t=1}^{T_m} \gamma_t^{(m)}(i) \, (o_t^{(m)} - \tilde{\mu}_i)^2}{\sum_{m=1}^{M} \sum_{t=1}^{T_m} \gamma_t^{(m)}(i)}$$

### Multivariate HMM

$$b_i(\vec{o}) = \prod_{k=1}^{d} \frac{e^{-\mu_{i,k}} \mu_{i,k}^{o_{t,k}}}{o_{t,k}!}$$

with constraints $\mu_{i,1} + \mu_{i,2} = I_{\text{tot}\,i}$

$$\tilde{\mu}_{i,k} = \frac{I_{\text{tot}\,i} \sum_{m=1}^{M} \sum_{t=1}^{T_m} o_{t,k}^{(m)} \, \gamma_t^{(m)}(i)}{\sum_{m=1}^{M} \sum_{t=1}^{T_m} \left(\sum_{k=1}^{d} o_{t,k}^{(m)}\right) \gamma_t^{(m)}(i)}$$

$$\tilde{\pi}_i = \frac{1}{M} \sum_{m=1}^{M} \gamma_1^{(m)}(i)$$

$$\tilde{a}_{ij} = \frac{\sum_{m=1}^{M} \sum_{t=1}^{T_m - 1} \xi_t^{(m)}(i, j)}{\sum_{m=1}^{M} \sum_{t=1}^{T_m - 1} \gamma_t^{(m)}(i)}$$

# Example: a five-state system ($N = 5$)

1. Initial state distribution $\vec{\pi} = (0.2, 0.2, 0.2, 0.2, 0.2)$
2. Transition matrix

$$A = \begin{pmatrix} 0.9 & 0.025 & 0.025 & 0.025 & 0.025 \\ 0.025 & 0.9 & 0.025 & 0.025 & 0.025 \\ 0.025 & 0.025 & 0.9 & 0.025 & 0.025 \\ 0.025 & 0.025 & 0.025 & 0.9 & 0.025 \\ 0.025 & 0.025 & 0.025 & 0.025 & 0.9 \end{pmatrix}$$

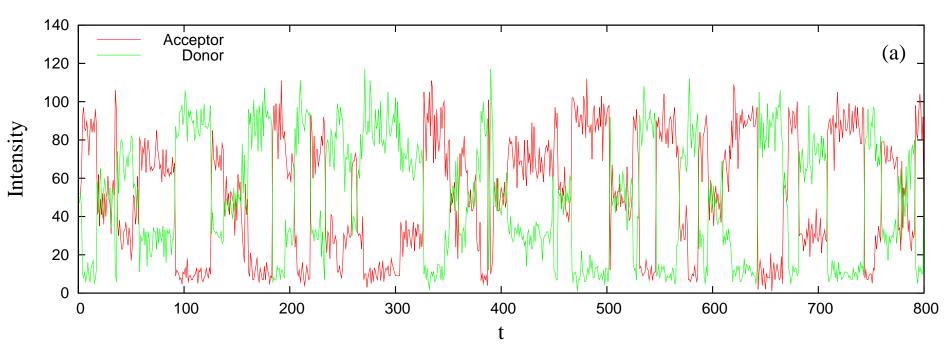3. Observation probability distribution (2D Poisson) with mean vectors

$$\vec{\mu}_1 = (10, 90) \qquad \Rightarrow \text{FRET}_1 = 0.1$$
$$\vec{\mu}_2 = (30, 70) \qquad \Rightarrow \text{FRET}_2 = 0.3$$
$$\vec{\mu}_3 = (50, 50) \qquad \Rightarrow \text{FRET}_3 = 0.5$$
$$\vec{\mu}_4 = (70, 30) \qquad \Rightarrow \text{FRET}_4 = 0.7$$
$$\vec{\mu}_5 = (90, 10) \qquad \Rightarrow \text{FRET}_5 = 0.9$$

Part I: Hidden Markov Modeling  Part II: smFRET data analysis
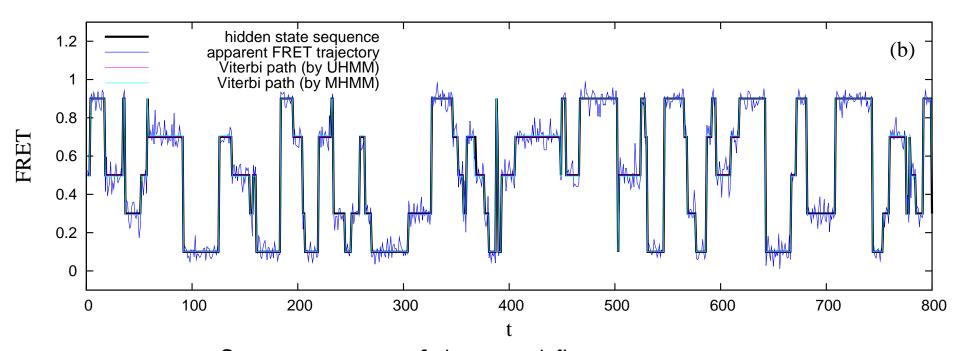


Observation sequence of the tested five-state system.

# Example: a five-state system ($N = 5$)

State sequences of the tested five-state system.

- FRET trajectory : calculated from $\mathrm{FRET} = I_{\mathrm{A}}/(I_{\mathrm{A}} + I_{\mathrm{D}})$
- The "hidden" state sequence is buried under the noisy FRET trajectory.
- Using either UHMM or MHMM, we can extract the model parameter and calculate the Viterbi paths which fit the original state sequence very well.

# Information Criterions
## determine the number of hidden states

**Goal : to best explain the data with a minimum of free parameters.**

■ Adding parameters to any model will always improve the fit but not necessarily the statistical significance.

■ Information criterions (penalized log-likelihood scores) take into account the tradeoff between bias and variance in model selection.

■ three most popular ones:

$$\text{Akaike} \quad \text{AIC} = -2 \, \log L + 2 \, k$$

$$\text{Bayesian} \quad \text{BIC} = -2 \, \log L + k \log n$$

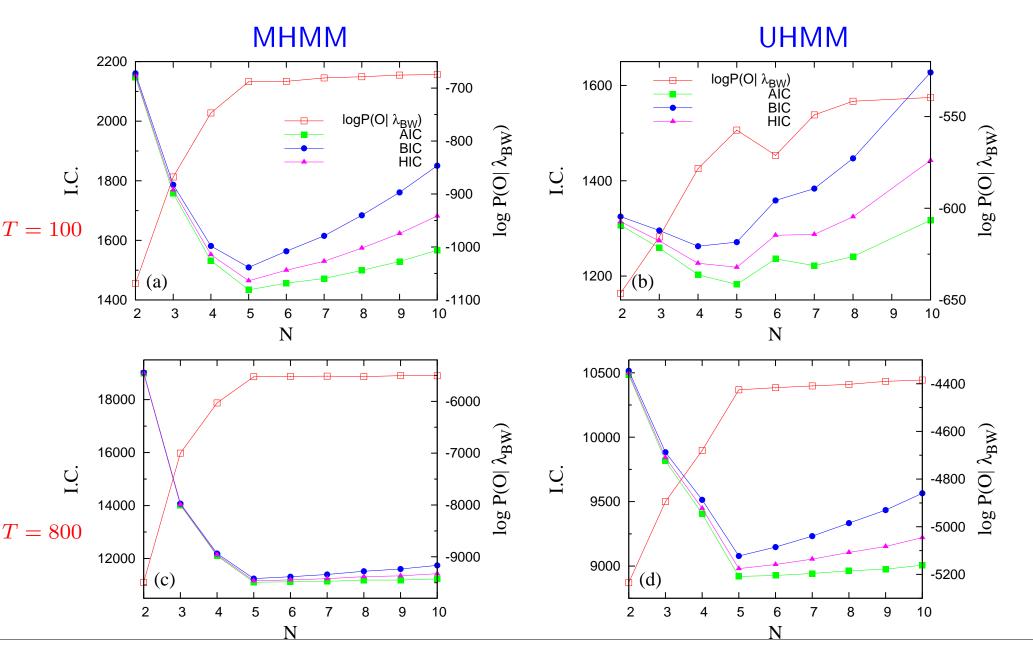$$\text{Hannan-Quinn} \quad \text{HIC} = -2 \, \log L + 2 \, k \log(\log n)$$

◆ $L$ : maximized value of the likelihood function for the estimated model: $\log L = \max[\log P(O|\lambda)] \approx \log P(O|\lambda_{\text{BW}})$

◆ $k$ : number of free parameters. For UHMM, $k = N^2 + 2N - 1$.

◆ $n$ : number of data points. For UHMM, $n = T$ the trace length.

# Reliability

## determine the model parameters
Part I: Hidden Markov Modeling  Part II: smFRET data analysis

To compare the performance of MHMM and UHMM further, we test their reliability for a simple two-state system.

1.  Initial probability $\vec{\pi} = (0.5, 0.5)$

2.  Transition matrix $A = \begin{pmatrix} 0.95 & 0.05 \\ 0.02 & 0.98 \end{pmatrix}$

3.  Observation probability distribution (2D Poisson) with mean vectors

$$\vec{\mu}_1 = (300, 700) \qquad \Rightarrow \text{FRET}_1 = 0.3$$
$$\vec{\mu}_2 = (700, 300) \qquad \Rightarrow \text{FRET}_2 = 0.7$$

## To be fair

■  UHMM and MHMM start from the **same** initial guess.

   ◆  $A$ and $\pi$: randomly generated
   ◆  $B$ : obtained by analyzing the histograms of signals
   ◆  $B$ parameters for UHMM and MHMM are related : $\text{FRET} = I_A/(I_A + I_D)$

■  UHMM and MHMM stop according to the **same** criterion : $\Delta \log P = 10^{-4}$.

# Reliability

## determine the model parameters
Part I: Hidden Markov Modeling  Part II: smFRET data analysis

We test algorithm responses to changes in model and trace parameters for MHMM and UHMM with $100$ traces (with standard trace length $T = 10,000$).
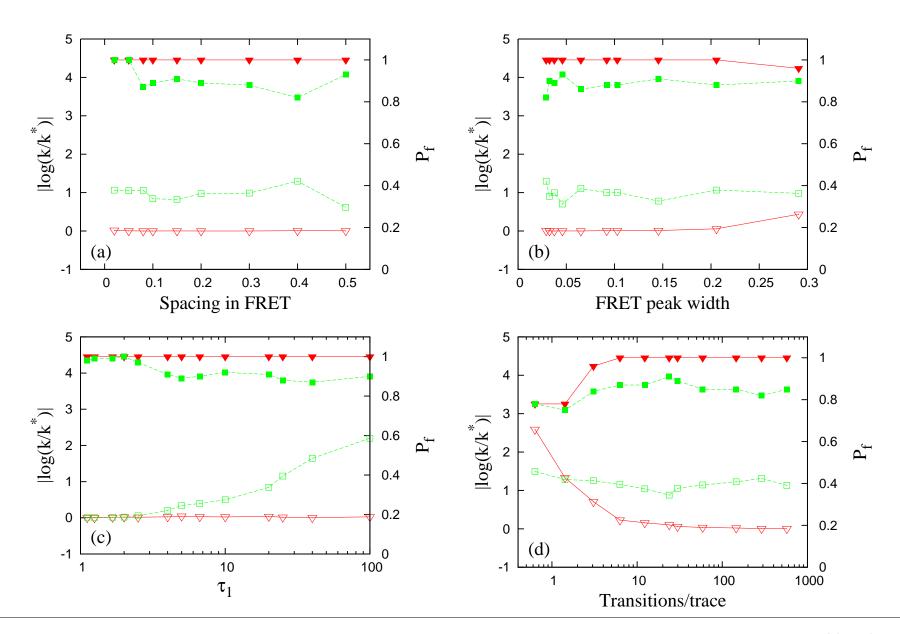
■ **reliability measures**

1. $P_f$ : the fraction of the 100 traces returned the true FRET values : $\mathrm{FRET}_1 \pm 0.05$ and $\mathrm{FRET}_2 \pm 0.05$
2. $|\log(k/k^*)|$ : the systematic error of transition rate with the obtained $k = a_{12}$ and the input value $k^* = a_{12}^*$

■ if input parameters are perfectly recovered, $P_f = 1$ and $|\log(k/k^*)| = 0$.

■ **test different impacts**

1. $\Delta\mathrm{FRET}$ : spacing between the two FRET states.
2. $\delta$ : FRET peak width, by changing the total intensity ($\langle I_A \rangle + \langle I_D \rangle$).
3. $\tau_1$: dwell time of state 1, by tuning $a_{11}^*$ but fixing $N_{tr}$
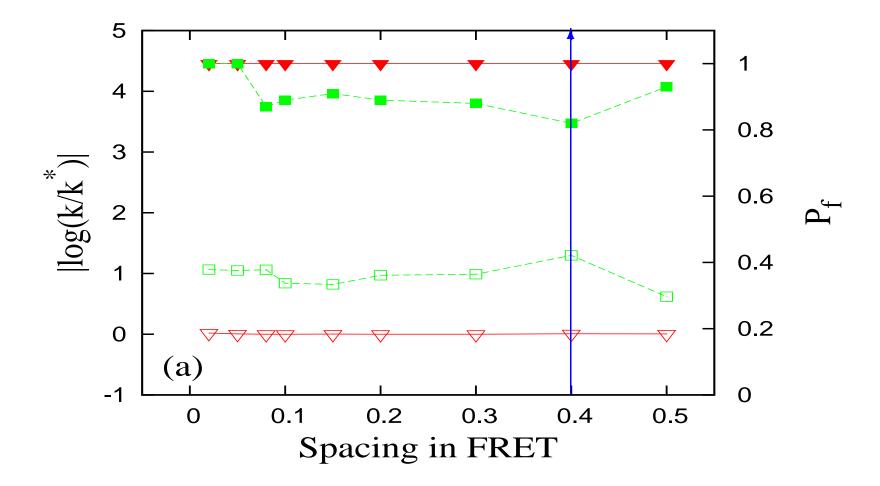4. $N_{tr}$ : number of state transitions, by simply changing $T$.

# Reliability
## determine the model parameters
Part I: Hidden Markov Modeling  Part II: smFRET data analysis

For the standard input model parameters, consider the increasing likelihood during the Baum-Welch iterations for each trace analysis.
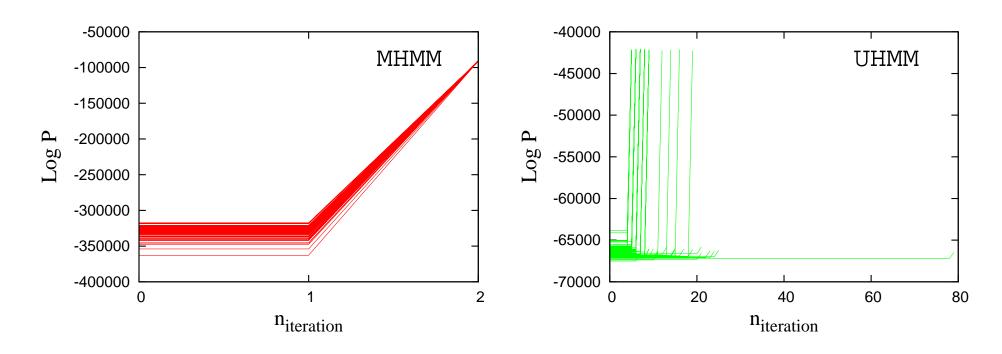
# Reliability

## determine the model parameters
Part I: Hidden Markov Modeling   Part II: smFRET data analysis

For the standard input model parameters, consider the increasing likelihood during the Baum-Welch iterations for each trace analysis.



- ■ With MHMM, all trace analyses return the true model parameters.
- ■ With UHMM, many trace analyses get trapped in local maxima $\Longrightarrow$ Unreliable !
- ■ $n_{\text{iteration}}$ of UHMM is much bigger than that of MHMM $\Longrightarrow$ Inefficient !

# Analyzing Multiple Sequences

**Two methods:**

1. Method I: analyze one by one and then average in some way

   ■ transition rates (which are found distributed asymmetrically) : one can average over logarithms and mean values are obtained by exponentiation.
   ■ mean FRET values: simple average is good enough.

2. Method II: analyze them simultaneously

   ■ those traces must be independent from each other
   ■ no extra average is needed

**Test the simple two-state system** (with $100$ traces of length $T = 100$):

1. Initial probability $\vec{\pi} = (0.5, 0.5)$

2. Transition matrix $A = \begin{pmatrix} 0.95 & 0.05 \\ 0.02 & 0.98 \end{pmatrix}$

3. Observation probability distribution (2D Poisson) with mean vectors
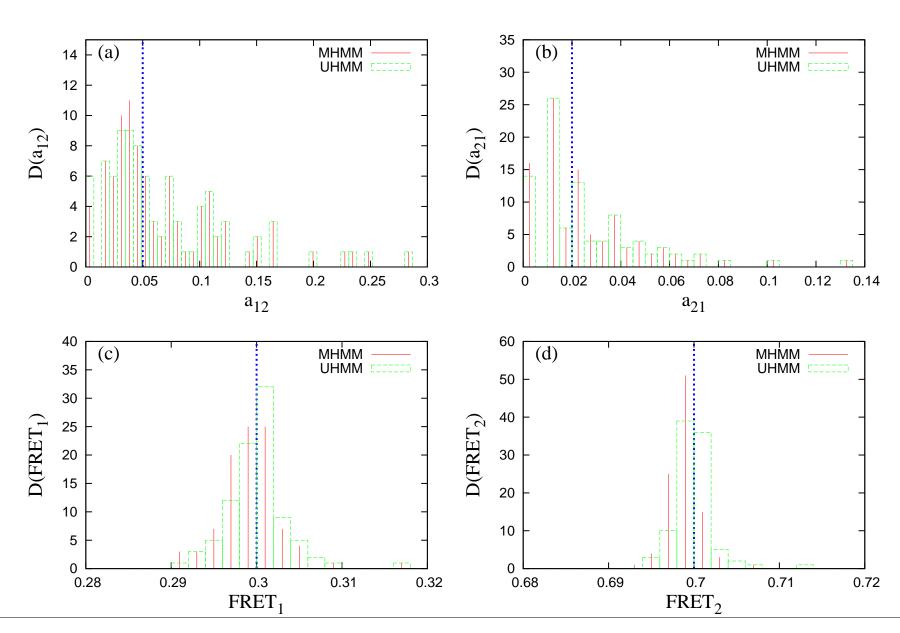
$$\vec{\mu}_1 = (300, 700) \qquad\qquad \Rightarrow \mathrm{FRET}_1 = 0.3$$
$$\vec{\mu}_2 = (700, 300) \qquad\qquad \Rightarrow \mathrm{FRET}_2 = 0.7$$

# Analyzing Multiple Sequences

| input | result of Method I | | result of Method II | |
|---|---|---|---|---|
| | MHMM | UHMM | MHMM | UHMM |
| $a_{12} = 0.05$ | 0.0684 (36.8%) | 0.0709 (41.9%) | 0.0526 (5.23%) | 0.0526 (5.23%) |
| $a_{21} = 0.02$ | 0.0250 (25.1%) | 0.0307 (53.5%) | 0.0210 (4.97%) | 0.0210 (4.97%) |
| $\mathrm{FRET}_1 = 0.3$ | 0.3147 (4.91%) | 0.3216 (7.19%) | 0.3003 (0.100%) | 0.3001 (0.041%) |
| $\mathrm{FRET}_2 = 0.7$ | 0.6993 (0.097%) | 0.6936 (0.915%) | 0.6997 (0.043%) | 0.6999 (0.015%) |

■ Method I: mean values deviates from input values with significant errors.

◆ transition rates are averaged in the special way. If averaged directly, deviations are even bigger.

◆ MHMM gives better result than UHMM.

■ Method II: obtained parameters are much closer to input ones.

◆ 100 times larger data set is taken into account.

◆ it is hard for both UHMM and MHMM to get trapped in local maxima $\implies$ the same high-quality result.

# Conclusions

**We develop a MHMM for smFRET data analysis.**

1.  Numerical tests in (1) determining number of states; (2) reliability in response to varying model parameters show that MHMM is much better than UHMM.
2.  With multiple traces, analyzing them simultaneously gives much better result than averaging over individual analysis result.
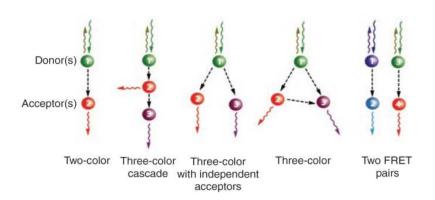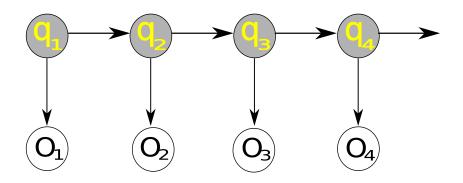
# Outlook I:

## more colors $\Rightarrow$ more channels

multi-color FRET scheme $\implies$ multivariate HMM



Two-color  Three-color cascade  Three-color with independent acceptors  Three-color  Two FRET pairs

Donor(s)  Acceptor(s)

(*R. Roy, S. Hohng & T.Ha, Nature Methods 2008*)
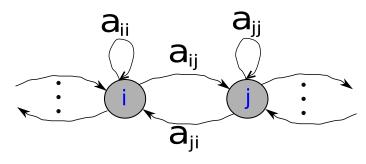
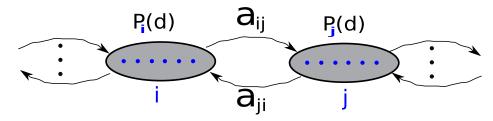$$O_t \longrightarrow \vec{O}_t = (O_t^1, O_t^2, O_t^3)$$

# Outlook II:
## inclusion of explicit state duration distribution
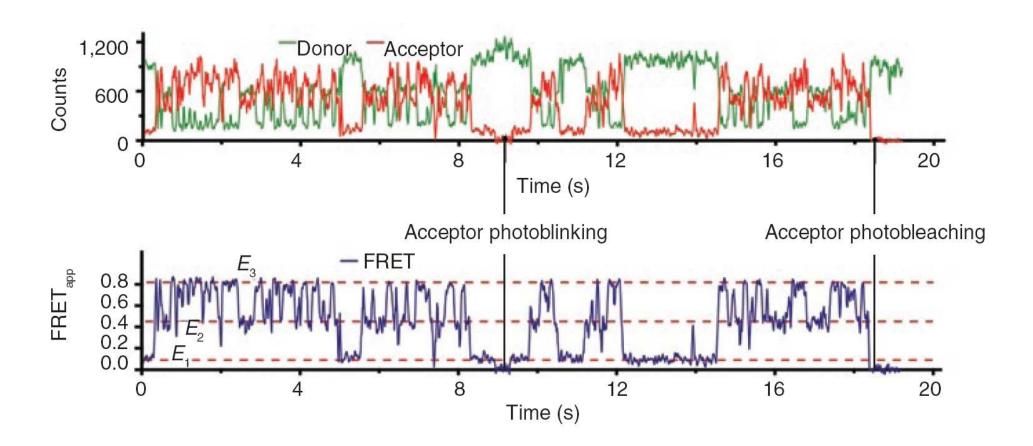Part I: Hidden Markov Modeling   Part II: smFRET data analysis



- In HMM, $p_i(d) = a_{ii}^{d-1}(1 - a_{ii})$. For most physical signals, this exponetial state duration distribution is inappropriate.
- We would prefer to explicitly model duration distribution $\Rightarrow$ Hidden semi-Markov Model (HSMM).

- Data missed due to photoblinking.
- Aapproximated by adding an artifical state $\Rightarrow$ systematic error in determining $a_{ij}$.
- Deal with missing data with *Expectation-Maximization algorithm*?

# Outlook IV:
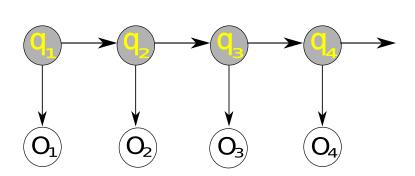## more state variables
Part I: Hidden Markov Modeling   Part II: smFRET data analysis

Combine sm-FRET and other sm-techniques, e.g. optical traps?
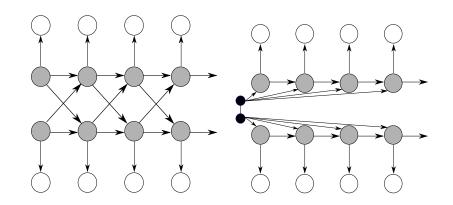
**Model I: single chain**

**Model II: coupled chains**



standard coupled HMM;   event-coupled HMM
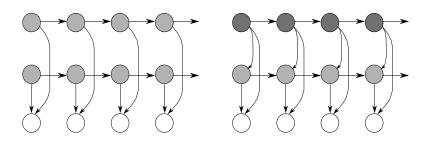
$$q_t \longrightarrow \vec{q}_t = (q_t^1, q_t^2, \cdots)$$

**(composite state)**

factorial HMM;   input-output HMM

# References

1. *A tutorial on hidden Markov models and selected applications in speech recognition*, L. R. Rabiner, *Proc. IEEE*, **77**, 257-286 (1989).

2. *A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains*, Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss, *Ann. Math. Stat.* **41**, 164-171 (1970).

3. *Analysis of Single-Molecule FRET Trajectories Using Hidden Markov Modeling*, Sean A. McKinney and Chirlmin Joo and Taekjip Ha, *Biophys. J.*, **91**, 1941-1951 (2006).

4. *Multivariate Hidden Markov Modeling in analysis of Single-molecule FRET*, Yang Liu, Karin A. Dahmen, Yann Chemla, and Taekjip Ha, to be submitted to *J. Phys. Chem. B*

# References

1. *A tutorial on hidden Markov models and selected applications in speech recognition*,
   L. R. Rabiner, *Proc. IEEE*, **77**, 257-286 (1989).
2. *A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains*,
   Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss, *Ann. Math. Stat.* **41**, 164-171 (1970).
3. *Analysis of Single-Molecule FRET Trajectories Using Hidden Markov Modeling*,
   Sean A. McKinney and Chirlmin Joo and Taekjip Ha, *Biophys. J.*, **91**, 1941-1951 (2006).
4. *Multivariate Hidden Markov Modeling in analysis of Single-molecule FRET*,
   Yang Liu, Karin A. Dahmen, Yann Chemla, and Taekjip Ha, to be submitted to *J. Phys. Chem. B*

# Thank you!