Lecture 8
Learning Sequence Motif Models Using Expectation Maximization (EM)

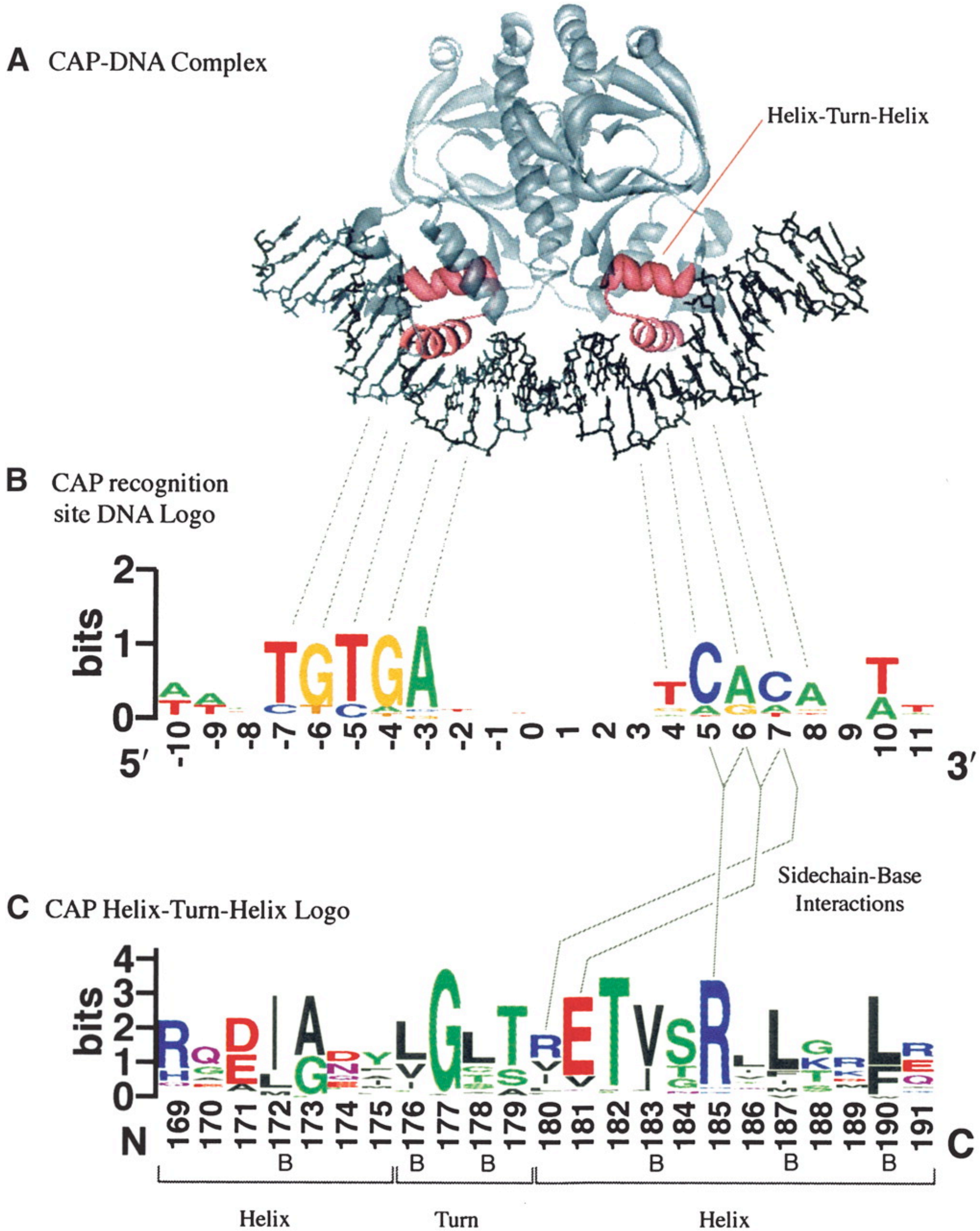
Colin Dewey

February 14, 2008

Sequence Motifs

- what is a sequence *motif* ?
  - a sequence *pattern* of biological significance
  - typically repeated several times in the genome
- examples
  - protein binding sites in DNA
  - protein sequences corresponding to common functions or conserved pieces of structure

Sequence Motifs Example

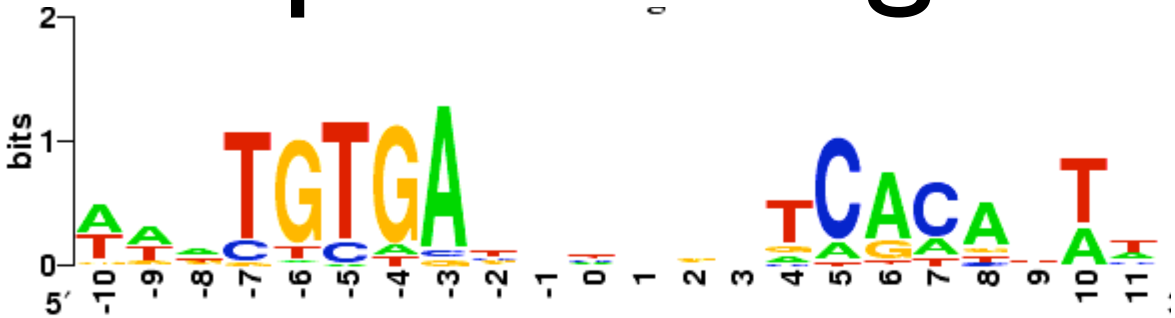


CAP-binding motif model based on 59 binding sites in E.coli

helix-turn-helix motif model based on 100 aligned protein sequences

Figure from Crooks et al., *Genome Research* 14:1188-90, 2004.

Sequence Logos



- Based on *entropy* (H) of random variable (X) representing distribution of character states at each position

$$H(X) = -\sum_x p(x) \log_2 p(x)$$

- Height of logo at given position determined by decrease in entropy (from maximum)

$$H_{max} - H(X) = \log_2 N - \left(-\sum_x p(x) \log_2 p(x) \right)$$

The Motif Model Learning Task

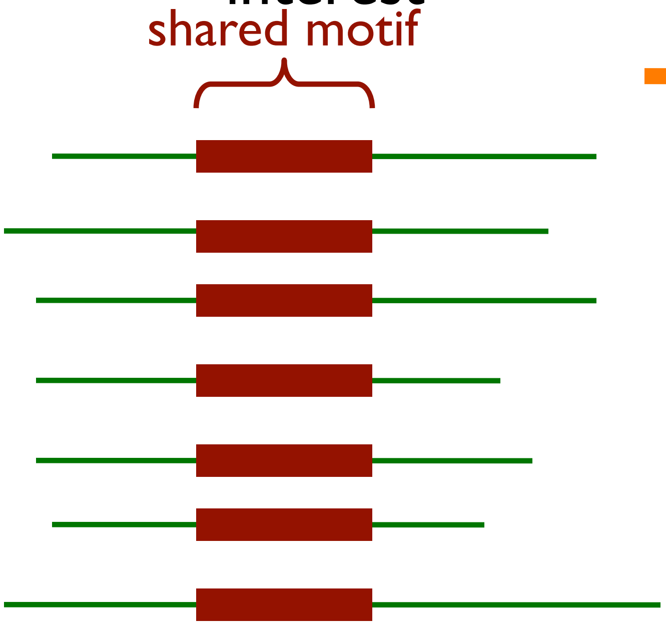
given: a set of sequences that are thought to contain an unknown motif of interest

do:

- infer a model of the motif
- predict the locations of the motif in the given sequences

Motifs and *Profile Matrices* (a.k.a. *Position Weight Matrices*)

- given a set of aligned sequences, it is straightforward to construct a profile matrix characterizing a motif of interest



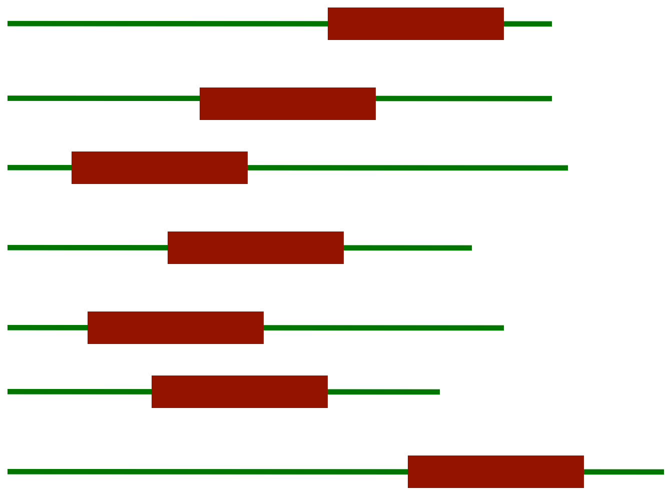
sequence positions

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

- each element represents the probability of given character at a specified position

Motifs and Profile Matrices

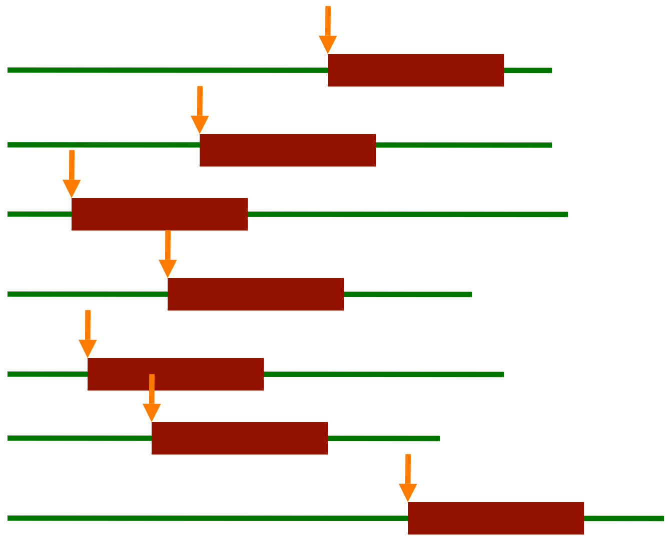
- how can we construct the profile if the sequences aren't aligned?
  - in the typical case we don't know what the motif looks like



- use an Expectation Maximization (EM) algorithm

The EM Approach

- EM is a family of algorithms for learning probabilistic models in problems that involve *hidden state*
- in our problem, the hidden state is where the motif starts in each training sequence



The MEME Algorithm

- Bailey & Elkan, 1993, 1994, 1995
- uses EM algorithm to find multiple motifs in a set of sequences
- first EM approach to motif discovery: Lawrence & Reilly 1990

Representing Motifs in MEME

- a motif is
  - assumed to have a fixed width, W
  - represented by a matrix of probabilities: $p_{c,k}$ represents the probability of character c in column k
- also represent the "background" (i.e., outside the motif) probability of each character: $p_{c,0}$ represents the probability of character c in the background

MEME Motif example

- example: a motif model of length 3

$$p = \begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ A & 0.25 & 0.1 & 0.5 & 0.2 \\ C & 0.25 & 0.4 & 0.2 & 0.1 \\ G & 0.25 & 0.3 & 0.1 & 0.6 \\ T & 0.25 & 0.2 & 0.2 & 0.1 \end{array}$$

background

Basic EM Approach

- the element $Z_{i,j}$ of the matrix Z represents the probability that the motif starts in position j in sequence i
- example: given 4 DNA sequences of length 6, where W=3

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

$$Z = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline \texttt{seq1} & 0.1 & 0.1 & 0.2 & 0.6 \\ \texttt{seq2} & 0.4 & 0.2 & 0.1 & 0.3 \\ \texttt{seq3} & 0.3 & 0.1 & 0.5 & 0.1 \\ \texttt{seq4} & 0.1 & 0.5 & 0.1 & 0.3 \end{array}$$

Basic EM Approach

given: length parameter W, training set of sequences

set initial values for p

do

re-estimate Z from p (E –step)

re-estimate p from Z (M-step)

until change in $p < \varepsilon$

return: p, Z

Calculating the Probability of a Sequence Given a Hypothesized Starting Position

$$\Pr(X_i \mid Z_{i,j}=1, p) = \underbrace{\prod_{k=1}^{j-1} p_{c_k,0}}_{\text{before motif}} \underbrace{\prod_{k=j}^{j+W-1} p_{c_k,k-j+1}}_{\text{motif}} \underbrace{\prod_{k=j+W}^{L} p_{c_k,0}}_{\text{after motif}}$$

X_i is the ith sequence

$Z_{i,j}$ is 1 if motif starts at position j in sequence i

c_k is the character at position k in sequence i

Example

$$X_i = \texttt{G C } \boxed{\texttt{T G T}} \texttt{ A G}$$

$$p = \begin{array}{ccccc} & 0 & 1 & 2 & 3 \\ \texttt{A} & 0.25 & 0.1 & 0.5 & 0.2 \\ \texttt{C} & 0.25 & 0.4 & 0.2 & 0.1 \\ \texttt{G} & 0.25 & 0.3 & 0.1 & 0.6 \\ \texttt{T} & 0.25 & 0.2 & 0.2 & 0.1 \end{array}$$

$$\Pr(X_i \mid Z_{i3} = 1, p) =$$

$$p_{\mathrm{G},0} \times p_{\mathrm{C},0} \times p_{\mathrm{T},1} \times p_{\mathrm{G},2} \times p_{\mathrm{T},3} \times p_{\mathrm{A},0} \times p_{\mathrm{G},0} =$$

$$0.25 \times 0.25 \times 0.2 \times 0.1 \times 0.1 \times 0.25 \times 0.25$$

The E-step: Estimating Z

- to estimate the starting positions in Z at step t

$$Z_{i,j}^{(t)} = \frac{\Pr(X_i \mid Z_{i,j}=1, p^{(t)}) \Pr(Z_{i,j}=1)}{\sum_{k=1}^{L-W+1} \Pr(X_i \mid Z_{i,k}=1, p^{(t)}) \Pr(Z_{i,k}=1)}$$

- this comes from Bayes' rule applied to

$$\Pr(Z_{i,j}=1 \mid X_i, p^{(t)})$$

The E-step: Estimating Z

- assume that, *a priori*, it is equally likely that the motif will start in any position

$$Z_{i,j}^{(t)} = \frac{\Pr(X_i \mid Z_{i,j} = 1, p^{(t)}) \cancel{\Pr(Z_{i,j} = 1)}}{\sum_{k=1}^{L-W+1} \Pr(X_i \mid Z_{i,k} = 1, p^{(t)}) \cancel{\Pr(Z_{i,k} = 1)}}$$

Example: Estimating Z

X_i = **G C T G T A G**

$p =$

	0	1	2	3
A	0.25	0.1	0.5	0.2
C	0.25	0.4	0.2	0.1
G	0.25	0.3	0.1	0.6
T	0.25	0.2	0.2	0.1

$$Z_{i,1} = 0.3 \times 0.2 \times 0.1 \times 0.25 \times 0.25 \times 0.25 \times 0.25$$

$$Z_{i,2} = 0.25 \times 0.4 \times 0.2 \times 0.6 \times 0.25 \times 0.25 \times 0.25$$

$$\vdots$$

- then normalize so that $\sum_{j=1}^{L-W+1} Z_{i,j} = 1$

The M-step: Estimating p

- recall $p_{c,k}$ represents the probability of character c in position k ; values for $k=0$ represent the background

$$p_{c,k}^{(t+1)} = \frac{n_{c,k} + d_{c,k}}{\sum_b (n_{b,k} + d_{b,k})}$$

pseudo-counts

expected # of c's at position k of motif

$$n_{c,k} = \begin{cases} \sum_i \sum_{\{j|X_{i,j+k-1}=c\}} Z_{i,j} & k > 0 \\ n_c - \sum_{j=1}^{W} n_{c,j} & k = 0 \end{cases}$$

total # of c's in data set

Example: Estimating p

A C A G C A

$$Z_{1,1} = 0.1, Z_{1,2} = 0.7, Z_{1,3} = 0.1, Z_{1,4} = 0.1$$

A G G C A G

$$Z_{2,1} = 0.4, Z_{2,2} = 0.1, Z_{2,3} = 0.1, Z_{2,4} = 0.4$$

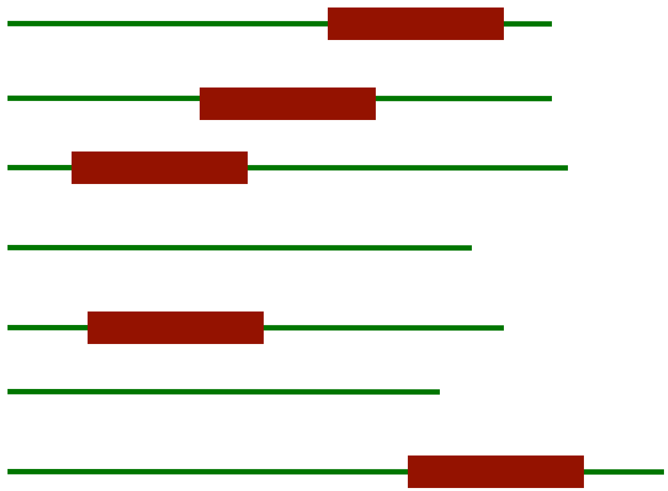
T C A G T C

$$Z_{3,1} = 0.2, Z_{3,2} = 0.6, Z_{3,3} = 0.1, Z_{3,4} = 0.1$$

$$p_{\mathrm{A},1} = \frac{Z_{1,1} + Z_{1,3} + Z_{2,1} + Z_{3,3} + 1}{Z_{1,1} + Z_{1,2} \ldots + Z_{3,3} + Z_{3,4} + 4}$$

The ZOOPS Model

- the approach as we've outlined it, assumes that each sequence has exactly one motif occurrence per sequence; this is the OOPS model
- the ZOOPS model assumes *zero or one* occurrences per sequence



E-step in the ZOOPS Model

- we need to consider another alternative: the ith sequence doesn't contain the motif
- we add another parameter (and its relative)

$$\lambda$$

- prior probability that any position in a sequence is the start of a motif

$$\gamma = (L - W + 1)\lambda$$

- prior probability of a sequence containing a motif

E-step in the ZOOPS Model

$$Z_{i,j}^{(t)} = \frac{\Pr(X_i \mid Z_{i,j} = 1, p^{(t)})\lambda^{(t)}}{\Pr(X_i \mid Q_i = 0, p^{(t)})(1-\gamma^{(t)}) + \sum_{k=1}^{L-W+1} \Pr(X_i \mid Z_{i,k} = 1, p^{(t)})\lambda^{(t)}}$$

- Q_i is a random variable for which $Q_i = 1$ if sequence X_i contains a motif, $Q_i = 0$ otherwise

$$\Pr(Q_i = 1) = \sum_{j=1}^{L-W+1} Z_{i,j}$$

$$\Pr\left(X_i \mid Q_i = 0, p\right) = \prod_{j=1}^{L} p_{c_j,0}$$

M-step in the ZOOPS Model

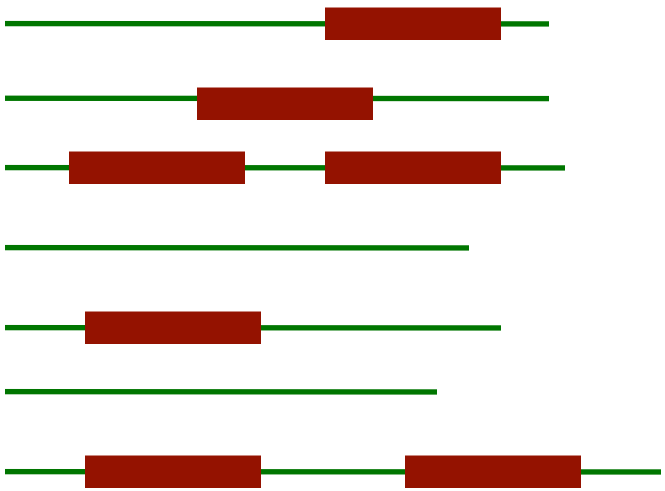
- update p same as before
- update γ as follows:

$$\lambda^{(t+1)} = \frac{1}{n(L-W+1)} \sum_{i=1}^{n} \sum_{j=1}^{L-W+1} Z_{i,j}^{(t)}$$

$$\gamma^{(t+1)} \triangleq \lambda^{(t+1)}(L-W+1)$$

The TCM Model

- the TCM (t̲wo-c̲omponent m̲ixture model) assumes *zero or more* motif occurrences per sequence



Likelihood in the TCM Model

- the TCM model treats each length W subsequence *independently*
- to determine the likelihood of such a subsequence:

$$\Pr(X_{i,j} \mid Z_{i,j} = 1, p) = \prod_{k=j}^{j+W-1} p_{c_k, k-j+1}$$

assuming a motif starts there

$$\Pr(X_{i,j} \mid Z_{i,j} = 0, p) = \prod_{k=j}^{j+W-1} p_{c_k, 0}$$

assuming a motif doesn't start there

E-step in the TCM Model

$$Z_{i,j}^{(t)} = \frac{\Pr(X_{i,j} \mid Z_{i,j} = 1, p^{(t)})\lambda^{(t)}}{\underbrace{\Pr(X_{i,j} \mid Z_{i,j} = 0, p^{(t)})(1-\lambda^{(t)})}_{\text{subsequence isn't a motif}} + \underbrace{\Pr(X_{i,j} \mid Z_{i,j} = 1, p^{(t)})\lambda^{(t)}}_{\text{subsequence is a motif}}}$$

- M-step same as before

Extending the Basic EM Approach in MEME

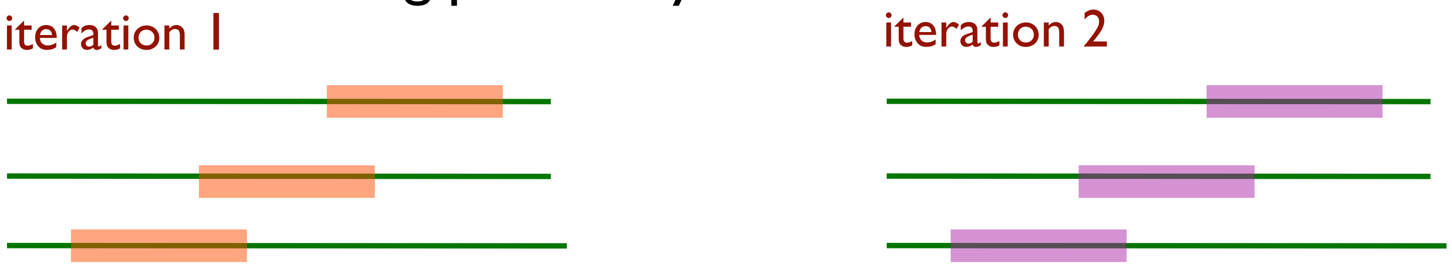
- How to choose the width of the motif?
- How to find multiple motifs in a group of sequences?
- How to choose good starting points for the parameters?
- How to use background knowledge to bias the parameters?

Choosing the Width of the Motif

- try various widths
  - estimate the parameters each time
  - apply a likelihood ratio test based on
    - probability of data under motif model
    - probability of data under null model
  - penalized by # of parameters in the model

Finding Multiple Motifs

- we might want to find multiple motifs in a given set of sequences
- how can we do this without
  - rediscovering previously learned motifs



  - discovering a motif that substantially overlaps with previously learned motifs

Finding Multiple Motifs

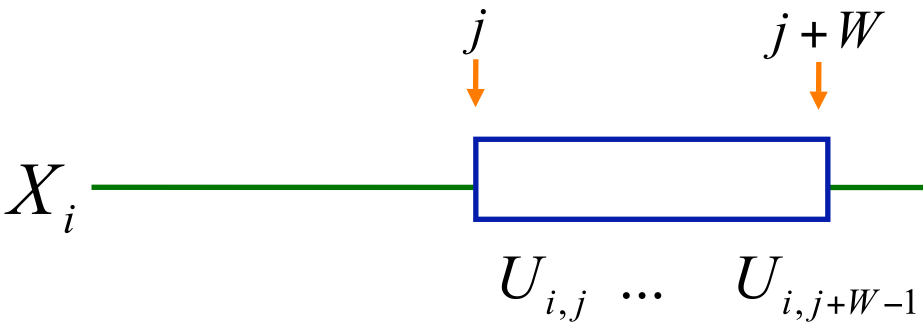
- basic idea: discount the likelihood that a new motif starts in a given position if this motif would overlap with a previously learned one
- when re-estimating $Z_{i,j}$, multiply by $\Pr(V_{i,j} = 1)$

$$V_{i,j} = \begin{cases} 1, & \text{no previous motifs in } [X_{i,j}, \ldots, X_{i,j+w-1}] \\ 0, & \text{otherwise} \end{cases}$$

Finding Multiple Motifs

- To determine $\Pr(V_{i,j} = 1)$ need to take into account individual positions in the window
- Use $U_{i,j}$ random variables to encode positions that are *not* part of a motif

$$U_{i,j} = \begin{cases} 1, & \text{if } X_{i,j} \notin \text{previous motif occurrence} \\ 0, & \text{otherwise} \end{cases}$$



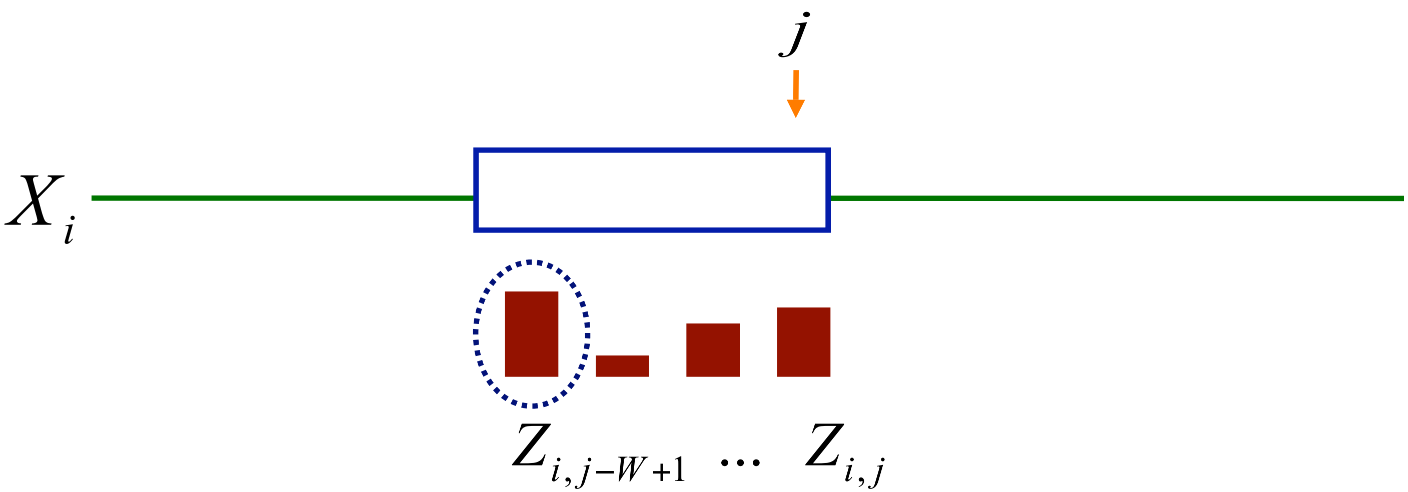
Finding Multiple Motifs

$$U_{i,j} = \begin{cases} 1, & \text{if } X_{i,j} \notin \text{previous motif occurrence} \\ 0, & \text{otherwise} \end{cases}$$

“pass” p

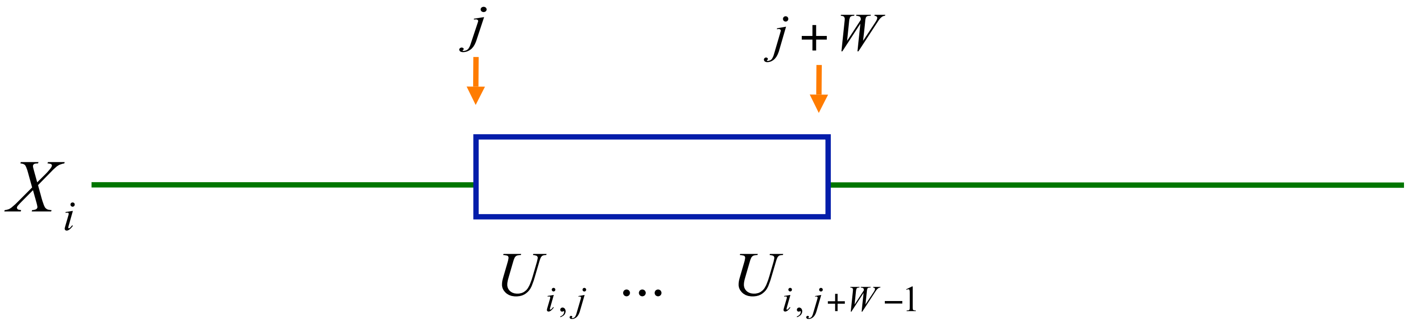
Z values at end of pass p

$$U_{i,j}^{(p)} = U_{i,j}^{(p-1)} \left(1 - \max_{k=j-W+1 \ldots j} Z_{i,k}^{(p)} \right)$$



Finding Multiple Motifs

$$\Pr(V_{i,j}=1)=\min\left(\Pr(U_{i,j}=1),\ldots,\Pr(U_{i,j+W-1}=1)\right)$$



Starting Points in MEME

- EM is susceptible to local maxima
- for every distinct subsequence of length W in the training set
  - derive an initial p matrix from this subsequence
  - run EM for one iteration
- choose motif model (i.e., p matrix) with highest likelihood
- run EM to convergence

Using Subsequences as Starting Points for EM

- set values corresponding to letters in the subsequence to some value π
- set other values to $(1-\pi)/(N-1)$ where N is the size of the alphabet
- example: for the subsequence TAT with $\pi = 0.5$

$$p = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline A & 0.17 & 0.5 & 0.17 \\ C & 0.17 & 0.17 & 0.17 \\ G & 0.17 & 0.17 & 0.17 \\ T & 0.5 & 0.17 & 0.5 \end{array}$$

Final MEME algorithm

procedure MEME $\left(X\text{:dataset of sequences}\right)$

  for $p = 1$ to $p_{\max}$ do

    for $W = W_{\min}$ to $W_{\max}$ by $\sqrt{2}$ do

      for $\lambda^0 = \lambda_{\min}$ to $\lambda_{\max}$ by 2 do

        Choose good θ^0 given W and λ^0

        Run EM to convergence

        Remove outer columns of motif (finetune W)

      end

    end

    Output best model, adjusted for overfit

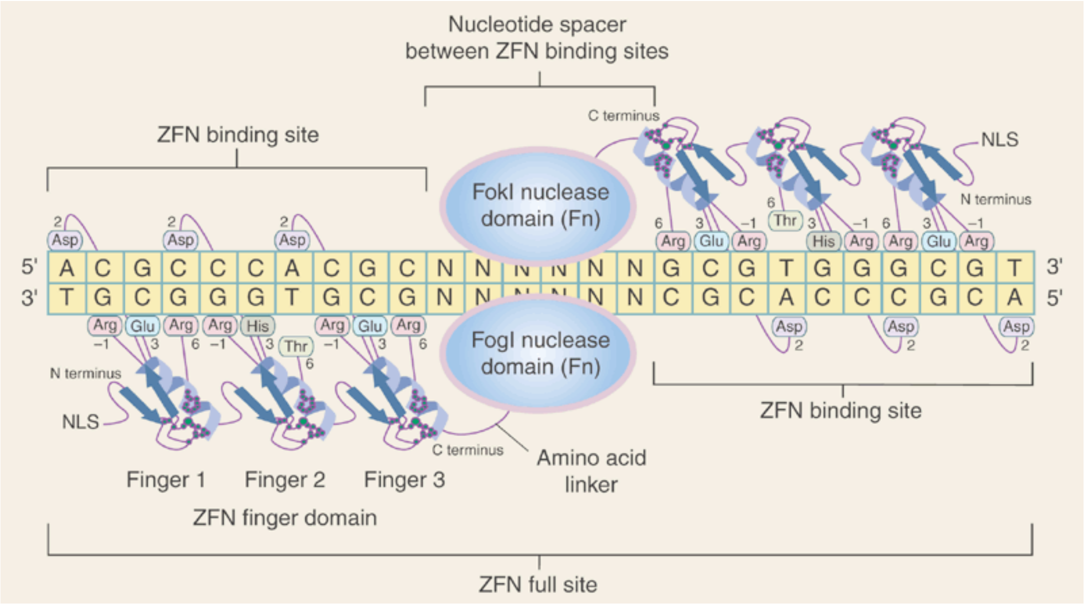
    Update priors U_{ij} for multiple motifs

  end

end

Using Background Knowledge to Bias the Parameters

- accounting for palindromes that are common in DNA binding sites



- using Dirichlet mixture priors to account for biochemical similarity of amino acids

Representing Palindromes

- parameters in probabilistic models can be "tied" or "shared"

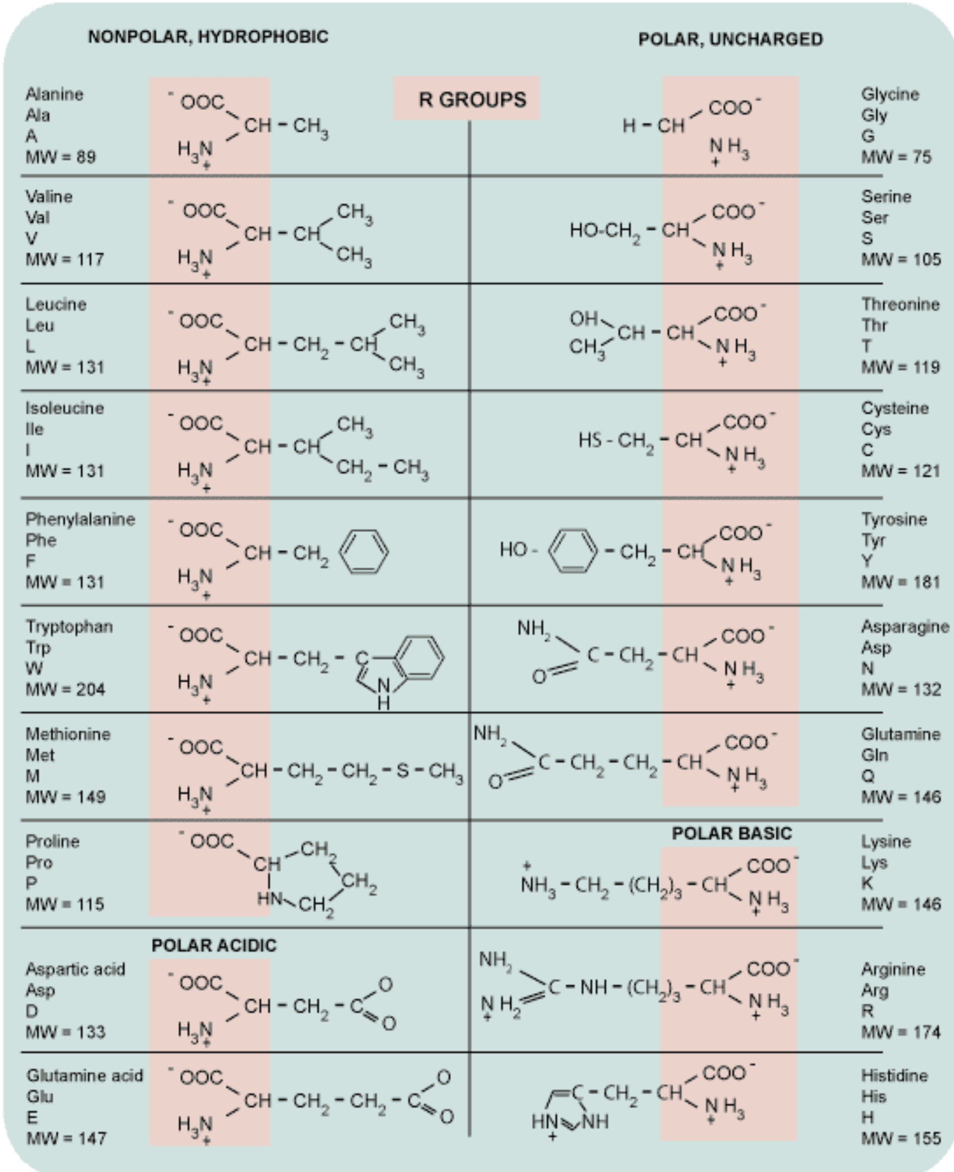
$$\begin{bmatrix} p_{a,0} & p_{a,1} & \cdots & p_{a,W} \\ p_{c,0} & p_{c,1} & \cdots & p_{c,W} \\ p_{g,0} & p_{g,1} & \cdots & p_{g,W} \\ p_{t,0} & p_{t,1} & \cdots & p_{t,W} \end{bmatrix}$$

- during motif search, try tying parameters according to palindromic constraint; accept if it increases likelihood test (half as many parameters)

Dirichlet mixtures in MEME

- Useful for protein motifs
- Amino acids can be grouped into classes based on certain properties (size, charge, etc.)
- Idea: use different pseudocounts in columns depending on most likely classes

Amino acids


NONPOLAR, HYDROPHOBIC
POLAR, UNCHARGED
R GROUPS
Alanine
Ala
A
MW = 89
Glycine
Gly
G
MW = 75
Valine
Val
V
MW = 117
Serine
Ser
S
MW = 105
Leucine
Leu
L
MW = 131
Threonine
Thr
T
MW = 119
Isoleucine
Ile
I
MW = 131
Cysteine
Cys
C
MW = 121
Phenylalanine
Phe
F
MW = 131
Tyrosine
Tyr
Y
MW = 181
Tryptophan
Trp
W
MW = 204
Asparagine
Asp
N
MW = 132
Methionine
Met
M
MW = 149
Glutamine
Gln
Q
MW = 146
Proline
Pro
P
MW = 115
POLAR BASIC
Lysine
Lys
K
MW = 146
POLAR ACIDIC
Aspartic acid
Asp
D
MW = 133
Arginine
Arg
R
MW = 174
Glutamine acid
Glu
E
MW = 147
Histidine
His
H
MW = 155

Dirichlet distribution

- Each motif column corresponds to a multinomial distribution $\mathbf{p}_k$
- Using pseudocounts for estimating $\mathbf{p}_k$ is essentially equivalent to using Dirichlet prior
- Dirichlet distribution is the *conjugate prior* to the multinomial

Multinomial
$$\mathbb{P}[n|\theta] = M^{-1}(n) \prod_{i=1}^{K} \theta_i^{n_i}$$

Dirichlet
$$\mathbb{P}[\theta|\beta] = Z^{-1}(\beta) \prod_{i=1}^{K} \theta_i^{\beta_i - 1} \delta\left(\sum_{i=1}^{K} \theta_i - 1\right)$$

Mixtures of Dirichlets

$$\mathbb{P}[\mathbf{p}_k|\beta^1,\ldots,\beta^R] = \sum_{i=1}^{R} q_i \mathcal{D}(\mathbf{p}_k|\beta^i)$$

Think of q_i as prior probability of ith Dirichlet distribution.

Given $\mathbf{c}_k$ (expected counts of characters in column k),

$$\mathbb{P}[\mathbf{p}_k|\mathbf{c}_k] = \sum_i \mathbb{P}[\mathbf{p}_k|\beta^i,\mathbf{c}_k]\mathbb{P}[\beta^i|\mathbf{c}_k] = \sum_i \mathcal{D}[\mathbf{p}_k|\beta^i+\mathbf{c}_k]\mathbb{P}[\beta^i|\mathbf{c}_k]$$

$$\mathbb{P}[\beta^i|\mathbf{c}_k] = \frac{q_i\mathbb{P}[\mathbf{c}_k|\beta^i]}{\sum_j q_j\mathbb{P}[\mathbf{c}_k|\beta^j]}$$

PME: *posterior mean estimate* (different than maximum likelihood estimate)

$$\mathbf{p}_k^{PME} = \frac{\mathbf{c}_k+\mathbf{d}_k}{|\mathbf{c}_k+\mathbf{d}_k|} \quad \text{where} \quad d_{k,\pi} = \sum_{i=1}^{R} \mathbb{P}[\beta^i|\mathbf{c}_k]\beta^i_\pi$$

See chapter 11 of textbook for more details

Maximum Likelihood Estimation

- During Maximization (M) step of the EM algorithm, we have to find the parameters that maximize the expected log likelihood
- How to do this in general?
  - Take derivative of expected log likelihood with respect to parameter to be optimized
  - Set derivative equal to zero and solve for parameter
  - Use constrained optimization techniques for dependent parameters

Likelihood in MEME

- OOPS model

$$\begin{aligned}\ell &= \log \mathbb{P}[X, Z|\theta] = \sum_{i=1}^{n}\sum_{j=1}^{m} Z_{i,j} \log \mathbb{P}[X_i|Z_{i,j}=1,\theta] + n \log \frac{1}{m} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m} Z_{i,j} (\sum_{k=1}^{W} \mathbf{I}(i, j+k-1)^{\mathrm{T}} \log \mathbf{p}_k + \sum_{k=\Delta_{i,j}} \mathbf{I}(i,k)^{\mathrm{T}} \log \mathbf{p}_0) + n \log \frac{1}{m}\end{aligned}$$

$$\begin{aligned}X_i &: \quad \text{sequence } i \text{ in data set } (i = 1, \ldots, n) \\ Z_{i,j} &: \quad \text{binary indicator random variable, 1 if motif starts at position } j \text{ in sequence } i \\ \mathbf{I}(i,j) &: \quad \text{vector-valued indicator variable, } \mathbf{I}(i,j)_\pi = 1 \text{ if } X_{i,j} = \pi, \text{ or 0 otherwise.} \\ \Delta_{i,j} &= \quad \{1, 2, \ldots, j-1, j+w, \ldots, L\}\end{aligned}$$

Constrained optimization

- Use Lagrange multipliers for parameter constraint

$$\tilde{\ell} = \sum_{i=1}^{n}\sum_{j=1}^{m} Z_{i,j}\left(\sum_{k=1}^{W}\mathbf{I}(i,j+k-1)^{\mathrm{T}}\log\mathbf{p}_k + \sum_{k=\Delta_{i,j}}\mathbf{I}(i,k)^{\mathrm{T}}\log\mathbf{p}_0\right) + n\log\frac{1}{m} + \sum_{k=0}^{W}\lambda_k\left(1-\sum_{\pi}p_{k,\pi}\right)$$

$$\frac{\partial\tilde{\ell}}{\partial p_{k,\pi}} = \sum_{i=1}^{n}\sum_{j=1}^{m} Z_{i,j}\mathbf{I}(i,j+k-1)_{\pi}\frac{1}{p_{k,\pi}} - \lambda_k, \text{ for } k>0$$

- Maximizing parameters of this function are maximizing parameters of original, subject to constraints

Constrained optimization

- Set derivative to zero, solve for λ_k

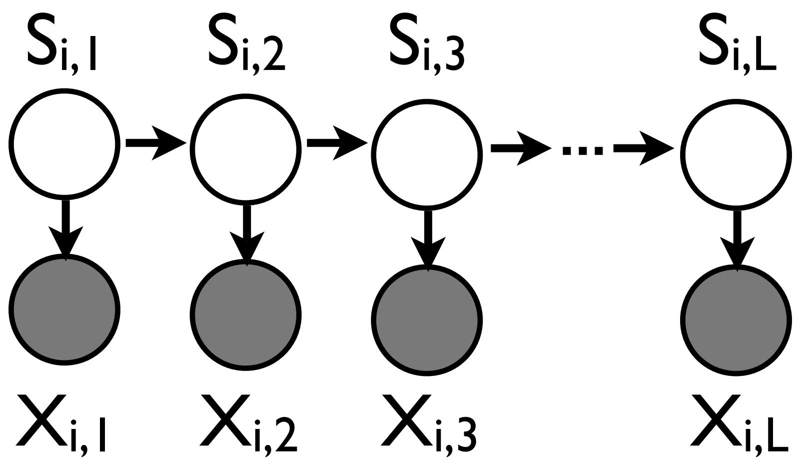
$$
\begin{aligned}
0 &= \sum_{i=1}^{n}\sum_{j=1}^{m} Z_{i,j}\mathbf{I}(i,j+k-1)_\pi \frac{1}{\hat{p}_{k,\pi}} - \lambda_k \\
\lambda_k &= \frac{1}{\hat{p}_{k,\pi}} \sum_{i=1}^{n}\sum_{j=1}^{m} Z_{i,j}\mathbf{I}(i,j+k-1)_\pi \\
\lambda_k &= \frac{c_{k,\pi}}{\hat{p}_{k,\pi}} \\
\hat{p}_{k,\pi}\lambda_k &= c_{k,\pi} \\
\sum_\pi \hat{p}_{k,\pi}\lambda_k &= \sum_\pi c_{k,\pi} \\
\lambda_k &= |\mathbf{c}_k|
\end{aligned}
$$

- Solve for $\hat{p}_{k,\pi}$

$$\hat{p}_{k,\pi} = \frac{c_{k,\pi}}{\lambda_k} = \frac{c_{k,\pi}}{|\mathbf{c}_k|}$$

Hidden Markov model representation

For each sequence X_i



$X_{i,j}$: observed character j of sequence i

$S_{i,j}$: hidden state j for sequence i

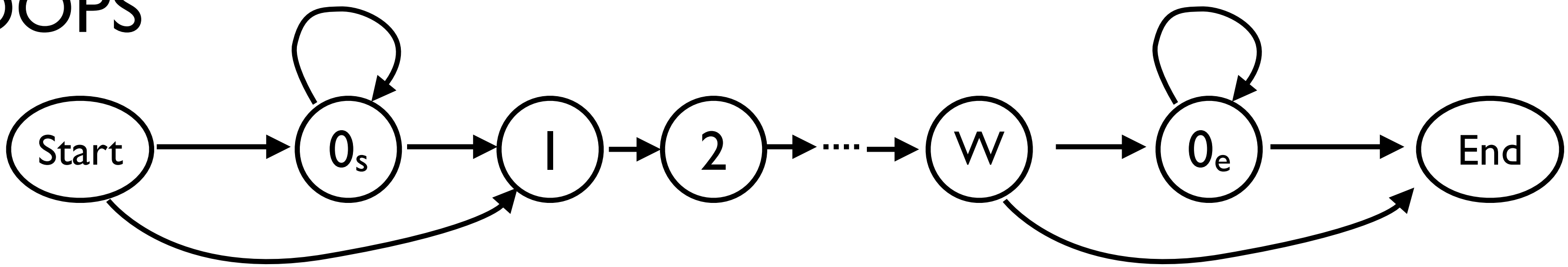
$S_{i,j}$ ε {0, 1, 2, ..., W}

Emission probabilities = profile matrix probabilities

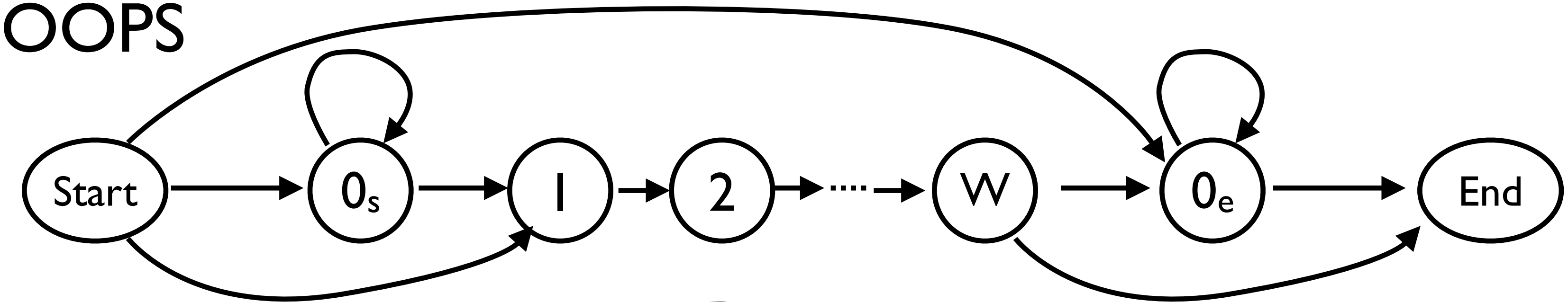
(state = column in profile)

HMM state transitions

OOPS



ZOOPS



???