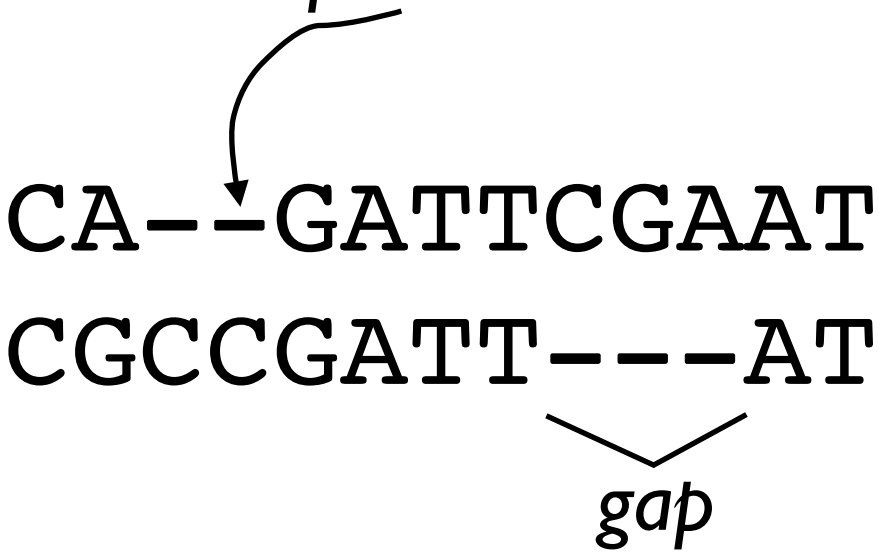
BMI/CS 776 Lecture #13: Statistical sequence alignment

Colin Dewey

March 4, 2008

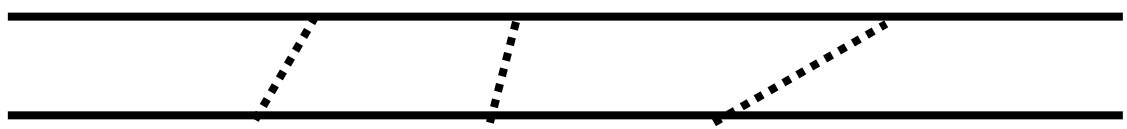
What is a pairwise alignment?

- matching of *homologous* positions in two sequences
- positions with no homologous pair are aligned with a *space* '–'



Global alignment properties

- **colinearity**: homologous positions must be in same order and orientation



- **complete**: all sequence positions are aligned

```
CA--GATTCGAAT
CGCCGATT---AT
```

Global (complete)

```
  ..GATTC....
....GATT-..
```

Local (incomplete)

Classical alignment

- Define features of interest in alignments
  - # matches, mismatches, spaces, gaps, etc.
- Assign weights (the parameters) to each feature
- Optimal alignment = alignment with maximum weight
- Exponentially-many possible alignments -- we can not score each one separately
- Dynamic programming to find optimal alignment

Picking Alignments

Alignment 1

```
mel          TGTTGTGTGATGTTGATTTCTTTACGACTCCTATCAAACTAAACCCATAAAGCATTCAATTCAAAGCATATA------------
pse          T----TTTGATGTTGATTTCTTTACGAGTTTGATAGAACTAAACCCATAAAGCATTCAATTCGTAGCATATAGCTCTCCTCTGC
             *    * ******************** *   **  *************************  ********

mel          -------CATGTGAAAATCCCAGCGAGAACTCCTTATTAATCCAG--------------------CGCAGTCGGCGGCGGCGGC
pse          CATTCGGCATGTGAAAA------------TCCTTATTAATCCAGAACGTGTGCGCCAGCGTCAGCGCCAGCGCCGGCAGCAGC
                    **********             ***************                    ***   ** **** ** **

mel          GCGCAGTCAGC----------GGTGGCAGCGCAGTATATAAATAAAGTCTTATAAGAAACTCGTGAGCG---------------
pse          -CGCAG-CAGCAAAACGGCACGCTGGCAGCGGAGTATATAAATAA--TCTTATAAGAAACTCGTGTGAGCGCAACCGGGCAGCG
              ***** ****          * ******** *************  ****************** * *

mel          ---AAAGAGAGCG-TTTTATTTATGTGCGTCAGCGTCGGCCGCAACAGCGCCGTCAGCACTGGCAGCGACTGCGAC
pse          GCCAAAGAGAGCGATTTTATTTATGTG-----------------ACTGCGCTGCCTG----------GTCCTCGGC
                ********** *************                 ** **** * * *          * *  ** *
```

Alignment summary: 27 mismatches, 12 gaps, 116 spaces

Alignment 2

```
mel          TGTTGTGTGATGTTGATTTCTTTACGACTCCTATCAAACTAAACCCATAAAGCATTCAATTCAAAGCATATACATGTGAAAATC
pse          T----TTTGATGTTGATTTCTTTACGAGTTTGATAGAACTAAACCCATAAAGCATTCAATTCGTAGCATATAGCTCTCCTCTGC
             *    * ******************** *   **  *************************  ********  * *      *

mel          CCAGCGAGA------ACTCCTTATTAATCCAGCGCAGTCGGCGGCGGCGGCGCGCAGTCAGCGGTGGCAGCGCAGTATATAAAT
pse          CATTCGGCATGTGAAAATCCTTATTAATCCAGAAC---------------------------------------------
             *   **  *      * ***************  *

mel          AAAGTCTTATAAGAAACTCGTGAGCGAAAGAGAGCGTTTTATTTATGTGCGTCAGCGTCGGCCGCAACAGCGCCGTCAGCACTG
pse          ---------------------------------------GTGTGCGCCAGCGTCAGCGCCAGCGCCGGCAGCAGCCGCA
                                                      ****** ******* **  ** *  ** *  ****

mel          GCAGCGA-------------------------------------------------------------------------
pse          GCAGCAAAACGGCACGCTGGCAGCGGAGTATATAAATAATCTTATAAGAAACTCGTGTGAGCGCAACCGGGCAGCGGCCAAAGA
             ***** *

mel          --------------------------------CTGCGAC
pse          GAGCGATTTTATTTATGTGACTGCGCTGCCTGGTCCTCGGC
                                            *  ** *
```

Alignment summary: 45 mismatches, 4 gaps, 214 spaces

Breaking into subproblems

- Consider optimal alignment of first i characters of sequence x and first j characters of sequence y
- Three possibilities for last column of optimal alignment:

1. x_i and y_j aligned to each other
2. x_i aligned to space
3. y_j aligned to space

3 Cases

$F_{i,j}$: score of best alignment between first i characters of x and first j characters of y

1. x_i and y_j aligned to each other

$$F_{i,j} = F_{i-1,j-1} + s(x_i, y_j)$$

score of aligning x_i to y_j (substitution matrix)

2. x_i aligned to space

$$F_{i,j} = F_{i-1,j} + e$$

score of a space

3. y_j aligned to space

$$F_{i,j} = F_{i,j-1} + e$$

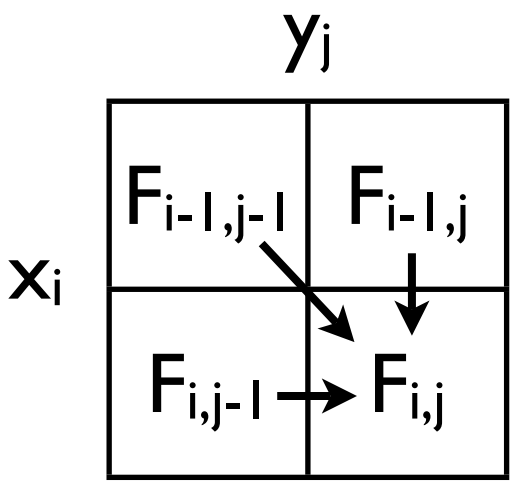
Needleman-Wunsch

- Parameters = Substitution matrix (s) & space score (e)
- F(i,j) = Score of optimal alignment of length i prefix of x and length j prefix of y

$$F(i,j) = \max \begin{cases} F(i-1,j-1) + s(x_i, y_j), \\ F(i-1,j) + e, \\ F(i,j-1) + e \end{cases}$$

Alignment matrix

		C	G
	$F_{0,0}$	$F_{0,1}$	$F_{0,2}$
C	$F_{1,0}$	$F_{1,1}$	$F_{1,2}$
A	$F_{2,0}$	$F_{2,1}$	$F_{2,2}$
G	$F_{3,0}$	$F_{3,1}$	$F_{3,2}$



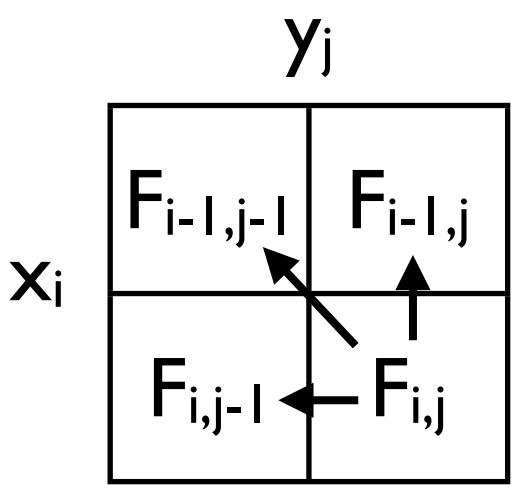
$$F_{i,j} = \max \begin{cases} F_{i-1,j-1} + s(x_i, y_j) \\ F_{i-1,j} + e \\ F_{i,j-1} + e \end{cases}$$

Needleman-Wunsch Algorithm

- Starting with $F_{0,0,}$ fill in alignment matrix
- Score of optimal alignment will be $F_{n,m}$ (bottom-right corner), where |x| = n and |y| = m
- *Traceback* to obtain an optimal alignment

Traceback

- When computing alignment matrix entry, keep track of which term(s) gave the maximum (i.e., the argmax)
- Store pointer from each cell to best previous cell
- Alignment = path from $F_{n,m}$ to $F_{0,0}$
  - diagonal edge: align x_i to y_j
  - horizontal edge: gap y_j
  - vertical edge: gap x_i



Affine gap scores

- Additional score for "gaps": d
- Maintain three matrices, for optimal alignments ending in a...
- match/mismatch: H(i,j)
- insertion: I(i,j)
- deletion: D(i,j)

Affine gap scores

$$H(i,j) = \max \begin{cases} H(i-1,j-1) + s(x_i, y_j), \\ I(i-1,j-1) + s(x_i, y_j), \\ D(i-1,j-1) + s(x_i, y_j), \end{cases}$$

$$I(i,j) = \max \begin{cases} H(i,j-1) + d + e, \\ I(i,j-1) + e, \\ D(i,j-1) + d + e \end{cases}$$

$$D(i,j) = \max \begin{cases} H(i-1,j) + d + e, \\ I(i-1,j) + d + e, \\ D(i-1,j) + e \end{cases}$$

- Termination: take max of H(m,n), I(m,n), D(m,n)

Choosing parameters

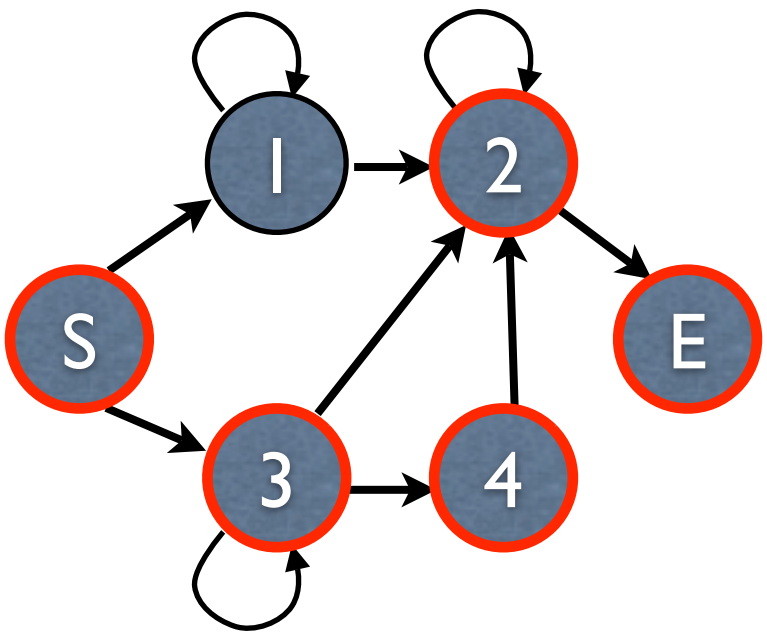
- No standard way of choosing weights for classical alignment
- Usually very subjective
  - Repeat alignment with different weights until alignment "looks good"
- We rarely have training data
  - We don't know the evolutionary "truth"

Statistical alignment

- Treat alignment probabilistically
- Parameters represent probabilities of substitutions, insertions, deletions occurring
- Every alignment assigned a probability
  - classical alignment corresponds to finding maximum likelihood alignment
- *Parameter estimation* is now possible

Hidden Markov models

- Each state emits a single character

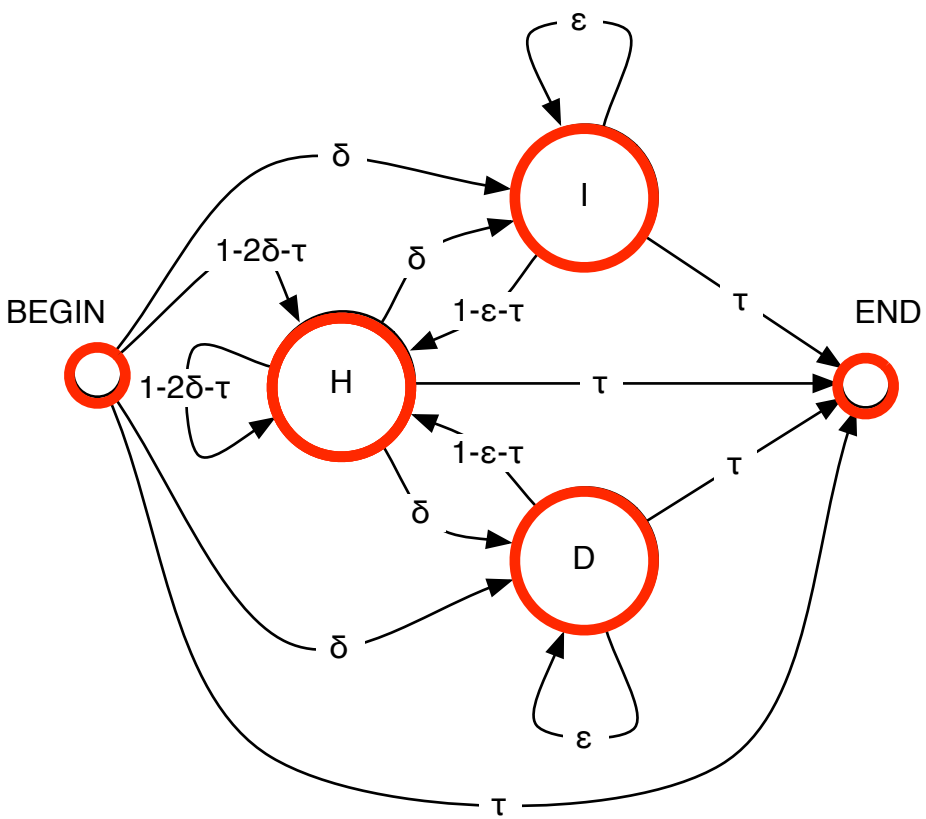


hidden: S 3 3 3 4 2 2 E

observed: A T A G G C

Pair Hidden Markov Models

- Each non-silent state emits one or a pair of characters



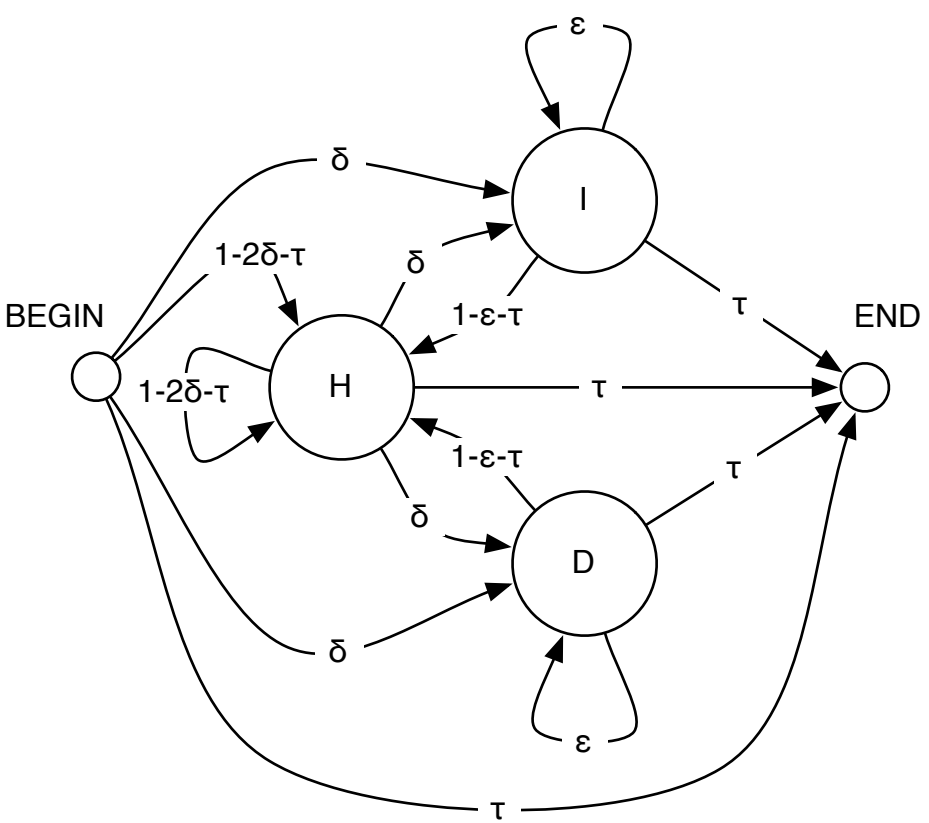
hidden: B H H I I H D H E

observed: A A G C G C
A T G T C

sequence 1: AAGCGC
sequence 2: ATGTC

Transition probabilities

- Probabilities of moving between states at each step



state i \ state i+1	B	H	I	D	E
B		1-2δ-τ	δ	δ	τ
H		1-2δ-τ	δ	δ	τ
I		1-ε-τ	ε		τ
D		1-ε-τ		ε	τ
E					

Emission probabilities

- Begin (B), and End (E) states silent
- Possible emission probabilities for H, I, D:

Deletion (D)

A	0.3
C	0.2
G	0.3
T	0.2

single character

Insertion (I)

A	0.1
C	0.4
G	0.4
T	0.1

single character

Homology (H)

	A	C	G	T
A	0.13	0.03	0.06	0.03
C	0.03	0.13	0.03	0.06
G	0.06	0.03	0.13	0.03
T	0.03	0.06	0.03	0.13

pairs of characters

PHMM Paths = Alignments

Observed sequences

x: AAGCGC

y: ATGTC

Possible path

B H H I I H D H E

A A G C G - C

A T - - G T C

Computing alignments with PHMMs

- "optimal alignment" ⇔ "most likely alignment"
- most likely alignment = $\underset{A}{\text{argmax}}\ P(A \mid x,y)$
- most likely alignment = most likely path
- most likely path is given by Viterbi algorithm

PHMM Viterbi

- Probability of most likely sequence of hidden states generating length i prefix of x and length j prefix of y, with the last state being:

H
$$v^H(i,j) = e_H(x_i, y_j) \max \begin{cases} v^H(i-1,j-1)t_{HH}, \\ v^I(i-1,j-1)t_{IH}, \\ v^D(i-1,j-1)t_{DH} \end{cases}$$

I
$$v^I(i,j) = e_I(y_j) \max \begin{cases} v^H(i,j-1)t_{HI}, \\ v^I(i,j-1)t_{II}, \\ v^D(i,j-1)t_{DI} \end{cases}$$

D
$$v^D(i,j) = e_D(x_i) \max \begin{cases} v^H(i-1,j)t_{HD}, \\ v^I(i-1,j)t_{ID}, \\ v^D(i-1,j)t_{DD} \end{cases}$$

for $i > 0$ and $j > 0$, formulas are slightly different for $i = 0$ and/or $j = 0$

PHMM alignment

- Calculate probability of most likely alignment

$$v^E(m,n) = max(v^M(m,n)t_{HE}, v^I(m,n)t_{IE}, v^D(m,n)t_{DE})$$

- Traceback, as in Needleman-Wunsch, to obtain sequence of state states giving highest probability

  HIDHHDDIIHH...

Correspondence with NW

- NW values ≈ logarithms of PHMM Viterbi values

$$\log v^H(i,j) = \log e_H(x_i, y_j) + \max \begin{cases} \log v^H(i-1, j-1) + \log t_{HH}, \\ \log v^I(i-1, j-1) + \log t_{IH}, \\ \log v^D(i-1, j-1) + \log t_{DH} \end{cases}$$

$$\log v^I(i,j) = \log e_I(y_j) + \max \begin{cases} \log v^H(i, j-1) + \log t_{HI}, \\ \log v^I(i, j-1) + \log t_{II}, \\ \log v^D(i, j-1) + \log t_{DI} \end{cases}$$

$$\log v^D(i,j) = \log e_D(x_i) + \max \begin{cases} \log v^H(i-1, j) + \log t_{HD}, \\ \log v^I(i-1, j) + \log t_{ID}, \\ \log v^D(i-1, j) + \log t_{DD} \end{cases}$$

PHMM Forward

- Probability of all sequences of hidden states generating length i prefix of x and length j prefix of y, with the last state being:

H $$f^H(i,j) = e_H(x_i, y_j)(f^H(i-1,j-1)t_{HH} + f^I(i-1,j-1)t_{IH} + f^D(i-1,j-1)t_{DH})$$

I $$f^I(i,j) = e_I(y_j)(f^H(i,j-1)t_{HI} + f^I(i,j-1)t_{II} + f^D(i,j-1)t_{DI})$$

D $$f^D(i,j) = e_D(x_i)(f^H(i-1,j)t_{HD} + f^I(i-1,j)t_{ID} + f^D(i-1,j)t_{DD})$$

$$\mathbb{P}(x,y) = f^E(m,n) = f^H(m,n)t_{HE} + f^I(m,n)t_{IE} + f^D(m,n)t_{DE}$$

for $i > 0$ and $j > 0$, formulas are slightly different for $i = 0$ and/or $j = 0$

Posterior probabilities

- There are similar recurrences for the *backward* values (probabilities of suffixes, given a start state)
- From the *forward* and *backward* values, we can calculate the posterior probability of the event ($event_{i,j,S}$) that the path (alignment) passed through a certain state S, after generating length i and j prefixes

$$P(x,y, event_{i,j,S}) = f^S(i,j)b^S(i,j)$$

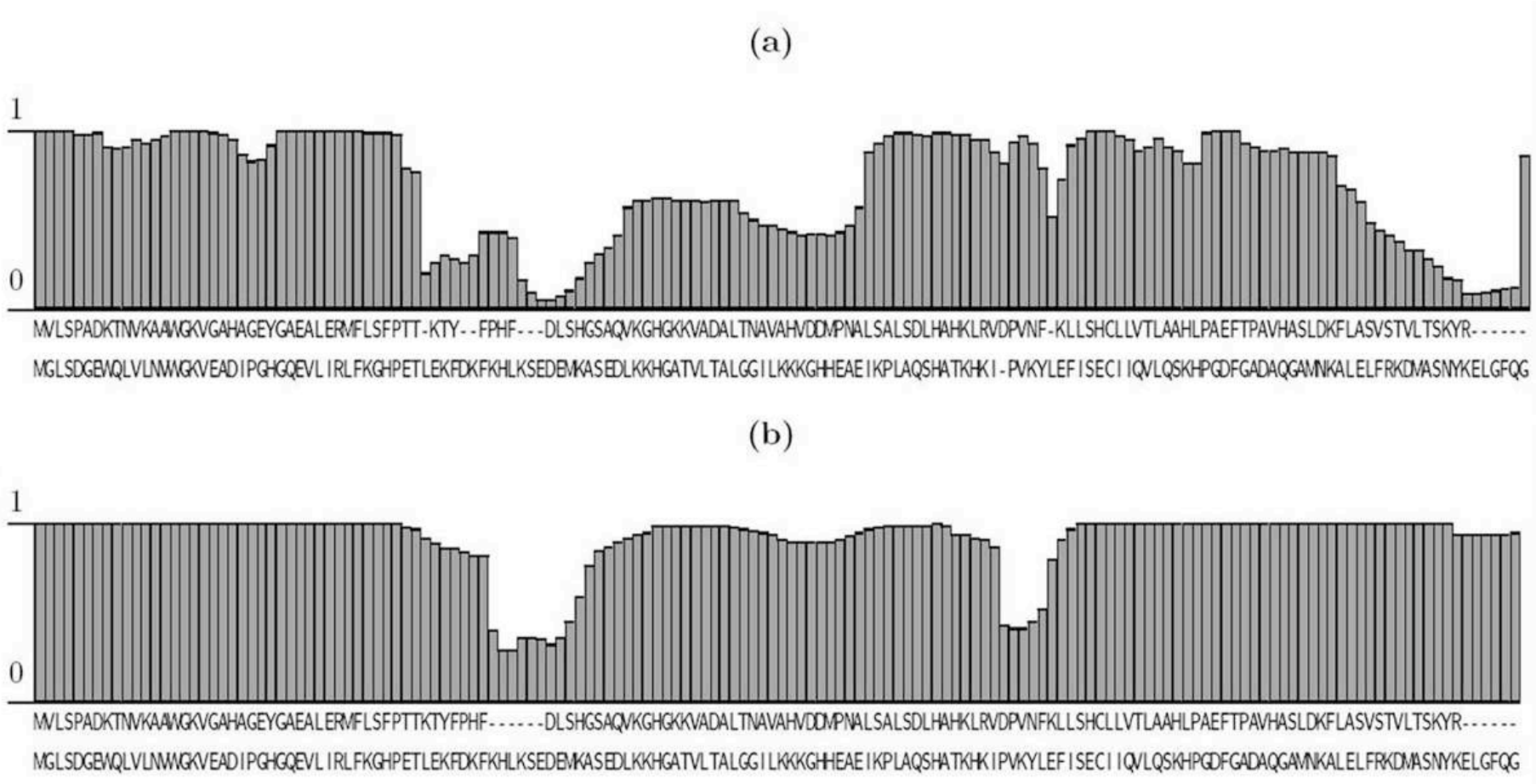
$$P(x,y) = P(x,y, event_{n,m,E}) = f^E(n,m)b^E(n,m) = f^E(n,m)$$

$$P(event_{i,j,S} \mid x,y) = f^S(i,j)b^S(i,j)/P(x,y) = f^S(i,j)b^S(i,j)/f^E(n,m)$$

Uses for posterior probabilities

- Suboptimal sampling of alignments
- Posterior probability of pairs of residues being homologous (aligned to each other)
- Posterior probability of a residue being gapped
- Used for training model parameters (EM)

Posterior probabilities

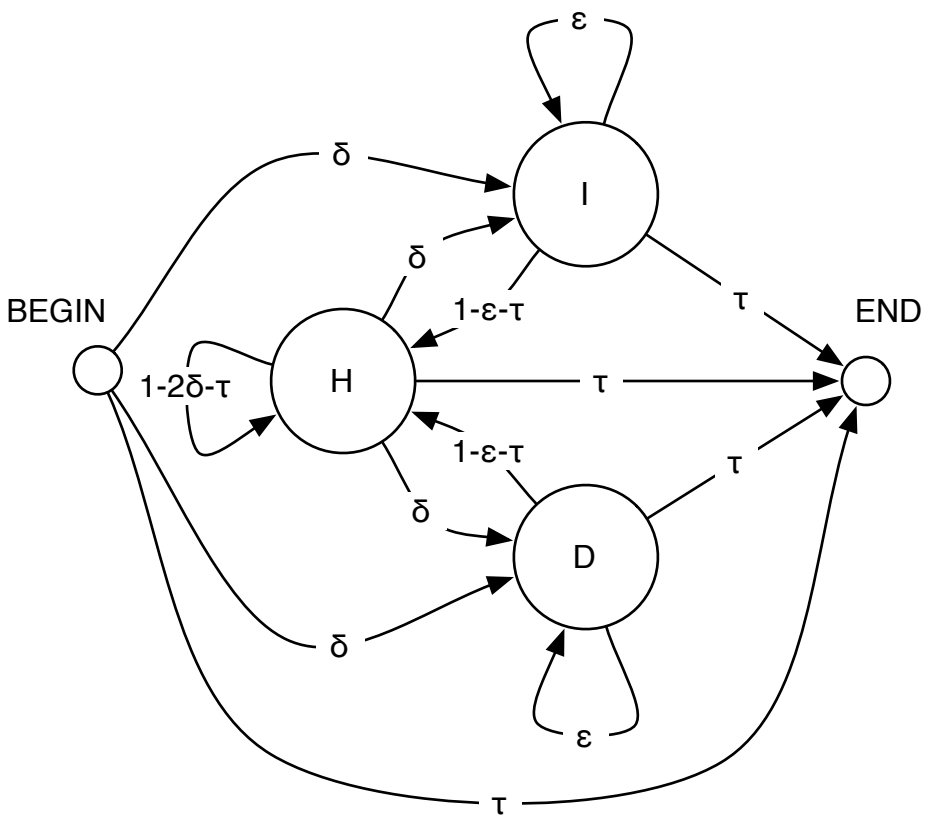


Plot posterior probability of each alignment column

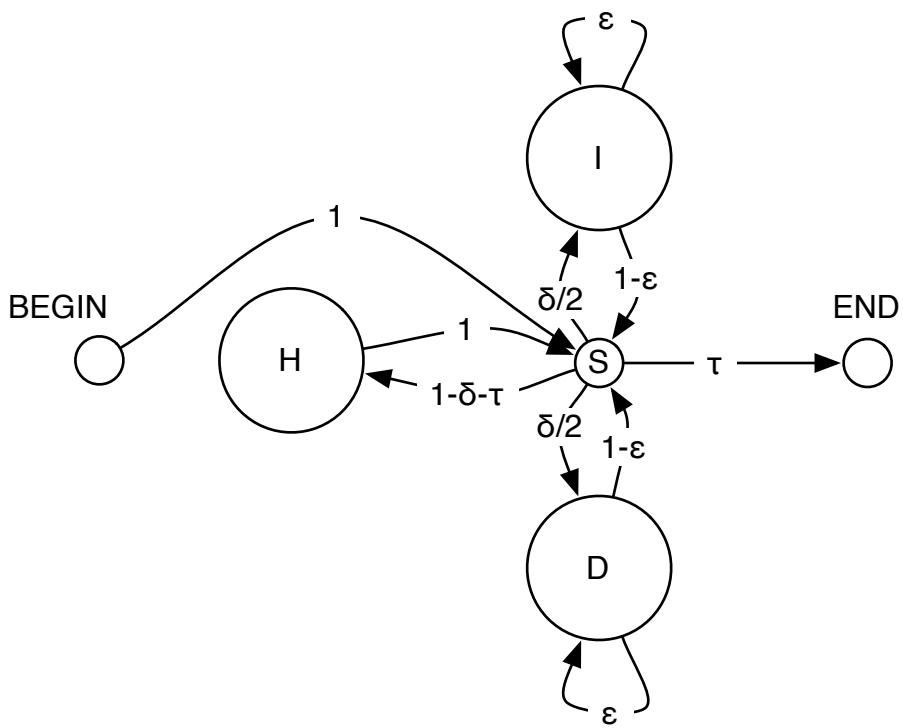
Parameter training

- supervised training
  - given: sequences and correct alignments
  - do: calculate parameter values that maximize joint likelihood of sequences and alignments
- unsupervised training
  - given: sequence pairs, but *no* alignments
  - do: calculate parameter values that maximize marginal likelihood of sequences (sum over all possible alignments)

A better PHMM

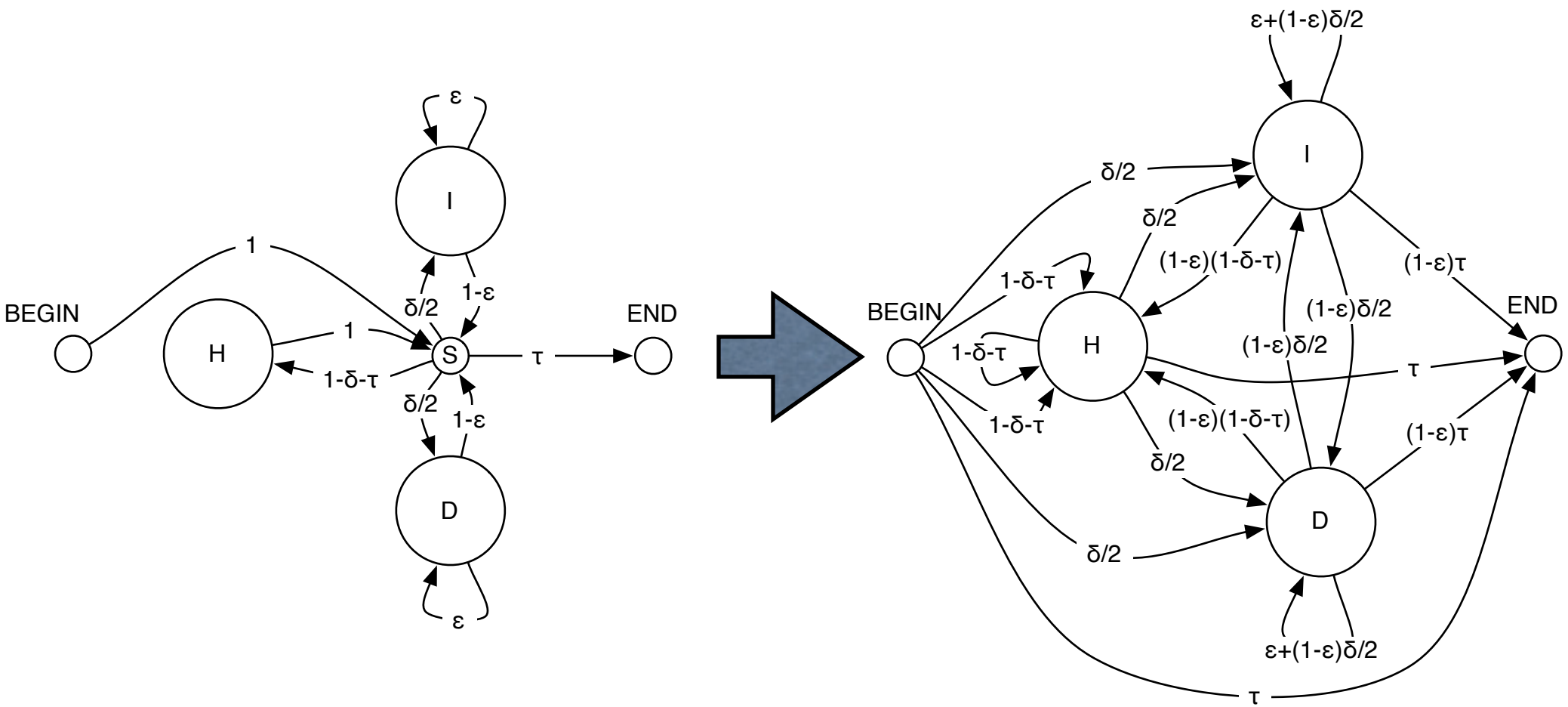


Durbin et al., 1998



Dewey, 2006

Silent state elimination



Alignment summaries and PHMMs

Probability of alignment with *m* matches, *x* mismatches, *g* gaps and e spaces:

$$P(m,x,g,s) = \tau \left(\frac{(1-\mu)(1-\delta-\tau)}{|\Sigma|}\right)^m \left(\frac{\mu(1-\delta-\tau)}{|\Sigma|(|\Sigma|-1)}\right)^x \left(\frac{\delta(1-\epsilon)}{2\epsilon}\right)^g \left(\frac{\epsilon}{|\Sigma|}\right)^s$$

μ : probability of mismatch $\qquad$ $|\Sigma|$: size of alphabet

Transform to log space:

$$M = \log\left(\frac{(1-\mu)(1-\delta-\tau)}{|\Sigma|}\right) \qquad X = \log\left(\frac{\mu(1-\delta-\tau)}{|\Sigma|(|\Sigma|-1)}\right)$$

$$G = \log\left(\frac{\delta(1-\epsilon)}{2\epsilon}\right) \qquad S = \log\left(\frac{\epsilon}{|\Sigma|}\right)$$

$$\log P(m,x,g,s) = M \cdot m + X \cdot x + G \cdot g + S \cdot s + \log\tau$$

Conclusions

- Statistical alignment with PHMM is essentially equivalent to affine gap NW alignment
- Main advantages of statistical approach:
  - Parameter learning (supervised & unsupervised)
  - Posterior probabilities on alignments and alignment features
- Slight disadvantage: slower due to floating point arithmetic (on some machines)