# Comparative Gene Finding

BMI/CS 776

www.biostat.wisc.edu/bmi776/

Spring 2015

Colin Dewey

cdewey@biostat.wisc.edu

# Goals for Lecture

the key concepts to understand are the following:

- using related genomes as an additional source of evidence for gene finding
- the TWINSCAN approach: use a pre-computed conservation sequence that is aligned to the given DNA sequence
- pair HMMs
- the correspondence between Viterbi in a pair HMM and standard dynamic programming for sequence alignment
- the SLAM approach: use a pair HMM to simultaneously align and parse sequences

# Why use comparative methods?

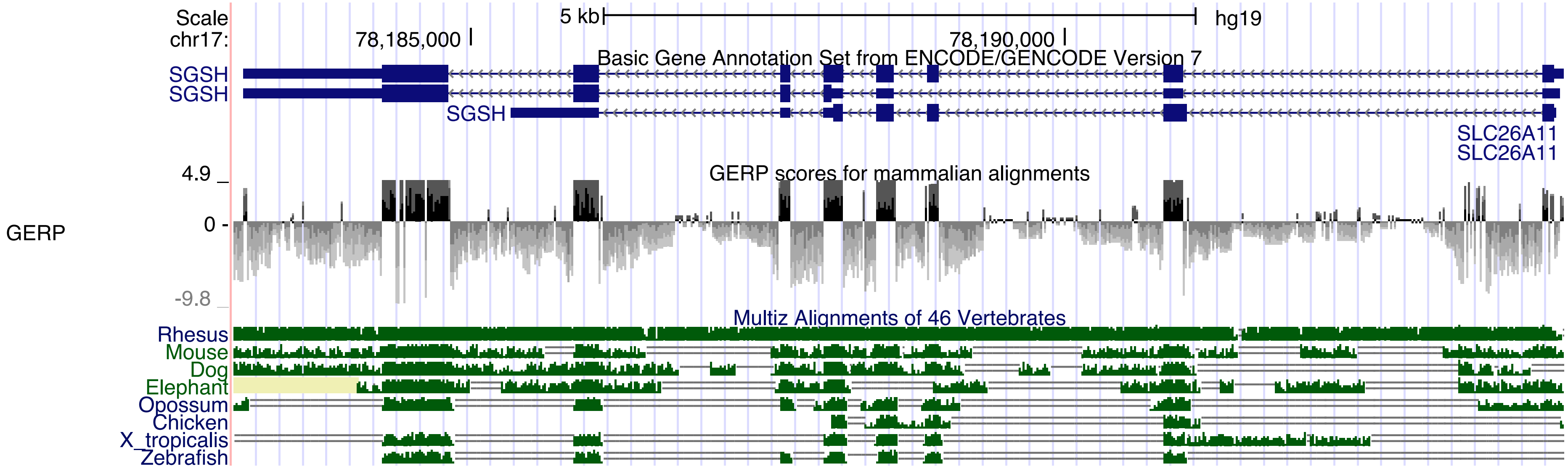
- genes are among the most conserved elements in the genome

  ⇒use conservation to help infer locations of genes

- some signals associated with genes are short and occur frequently

  ⇒use conservation to eliminate from consideration false candidate sites

# Conservation as powerful information source



# TWINSCAN

Korf et al., *Bioinformatics* 2001

- prediction with TWINSCAN

  given: a sequence to be parsed, $x$

  using BLAST, construct a conservation sequence, $c$

  have HMM simultaneously parse (using Viterbi) $x$ and $c$

- training with TWINSCAN

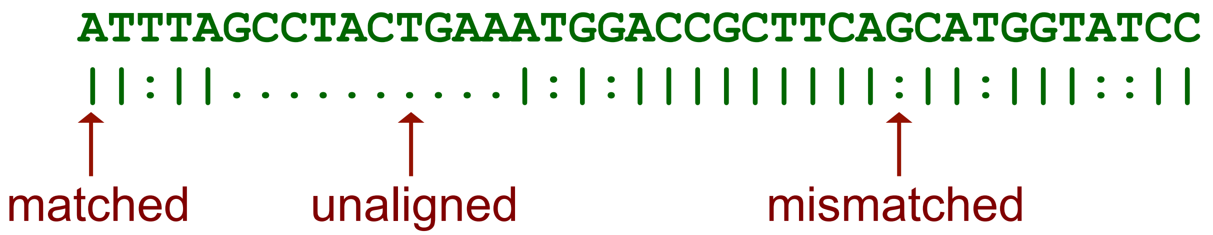
  given: set of training sequences $X$ with known gene structure annotations

  for each $x$ in $X$

  - construct a conservation sequence $c$ for $x$
  - infer emission parameters for both $x$ and $c$

# Conservation Sequences in TWINSCAN

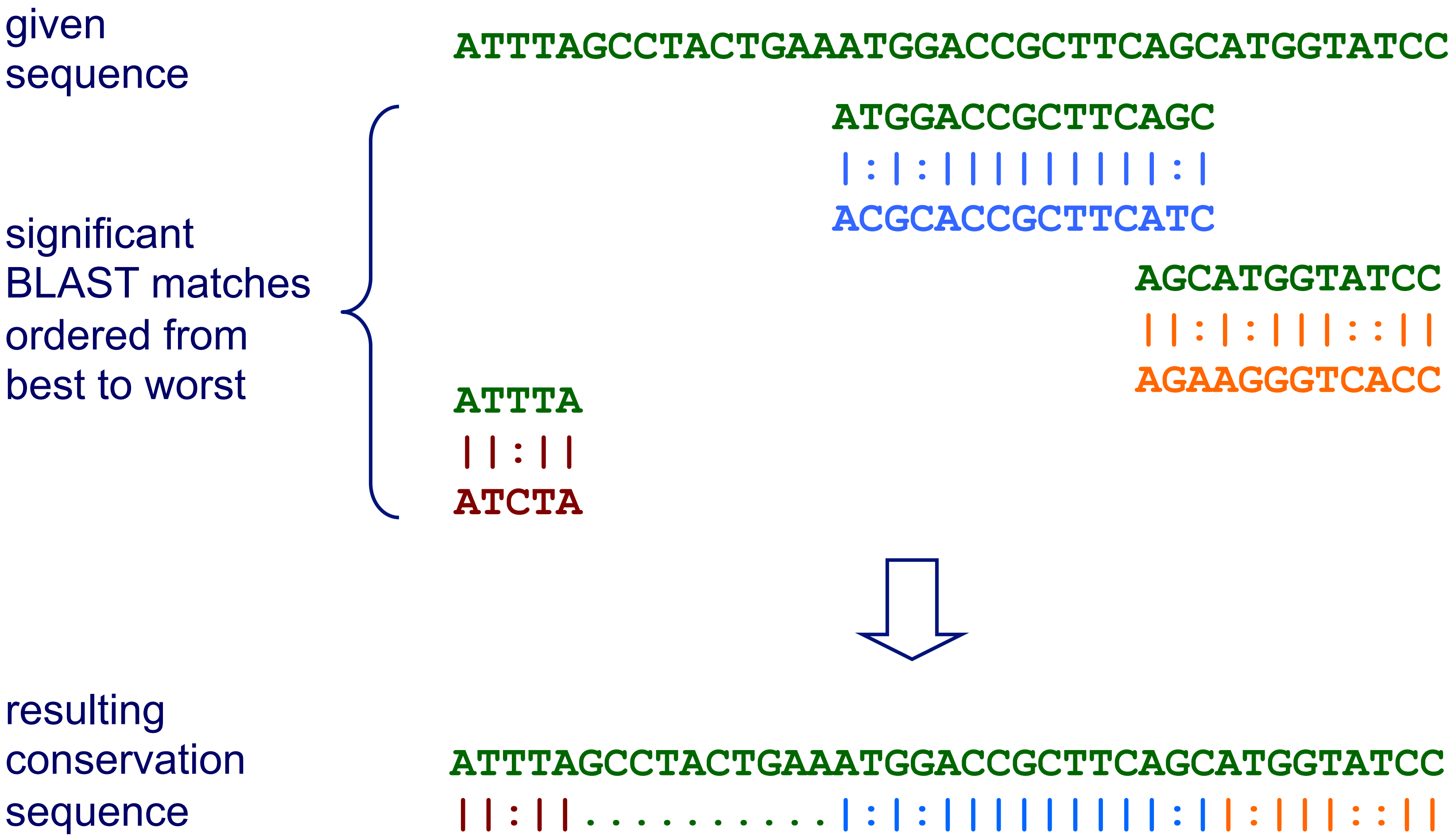
- before processing a given sequence, TWINSCAN first computes a corresponding *conservation sequence*



```
ATTTAGCCTACTGAAATGGACCGCTTCAGCATGGTATCC
||:||..........|:|:|||||||||:||:|||::||
```

```
Given: a sequence of length $n$, a set of aligned BLAST matches
$c[1...n]$ = unaligned
sort BLAST matches by alignment score
for each BLAST match $h$ (from best to worst)
    for each position $i$ covered by $h$
        if $c[i]$ == unaligned
            $c[i] = h[i]$
```

# Conservation Sequence Example



given sequence

```
ATTTAGCCTACTGAAATGGACCGCTTCAGCATGGTATCC
```

significant BLAST matches ordered from best to worst

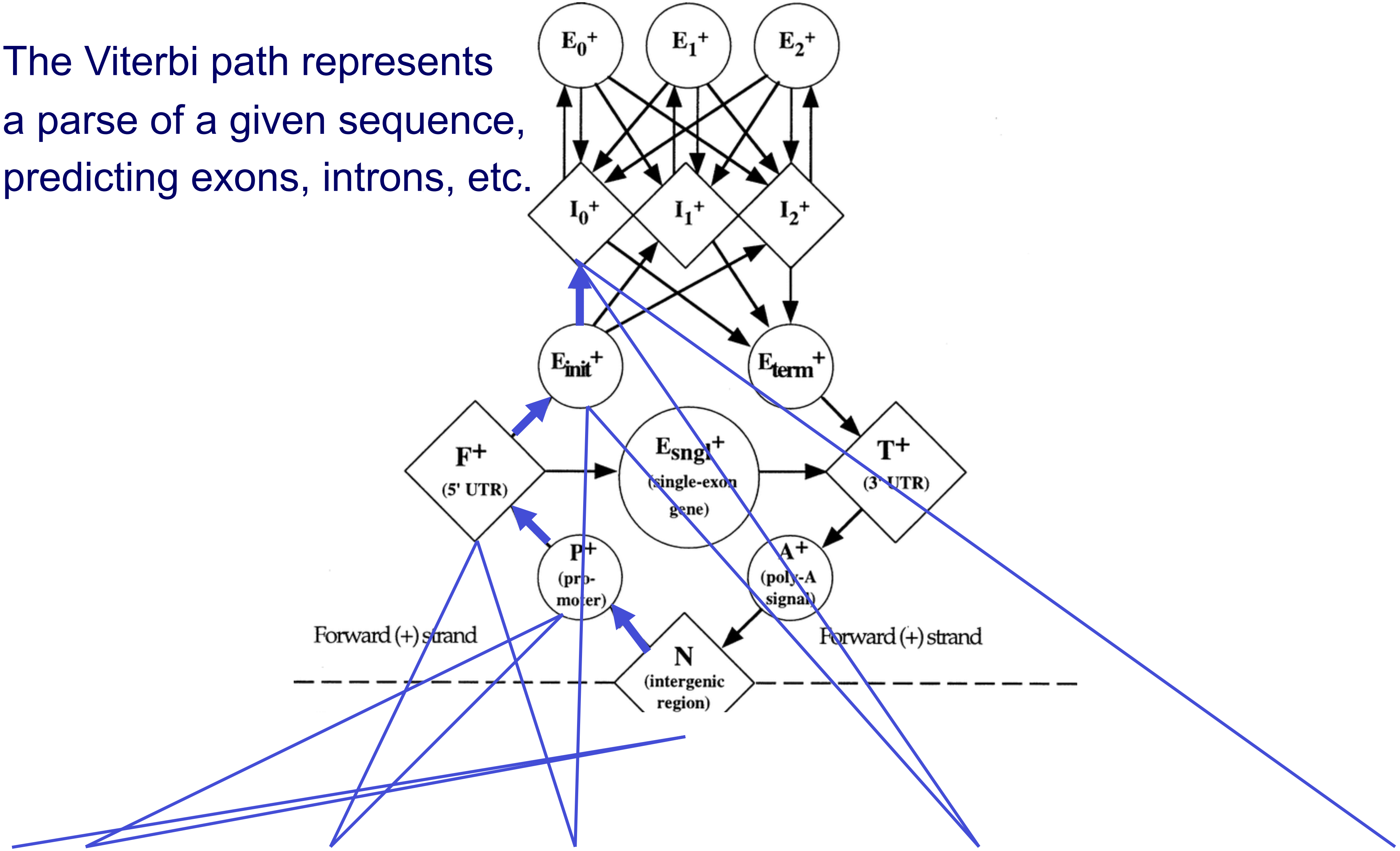
```
               ATGGACCGCTTCAGC
               |:|:|||||||||:|
               ACGCACCGCTTCATC
                           AGCATGGTATCC
                           ||:|:|||::||
                           AGAAGGGTCACC
ATTTA
||:||
ATCTA
```

resulting conservation sequence

```
ATTTAGCCTACTGAAATGGACCGCTTCAGCATGGTATCC
||:||..........|:|:|||||||||:||:|||::||
```

# Parsing a DNA Sequence

The Viterbi path represents a parse of a given sequence, predicting exons, introns, etc.

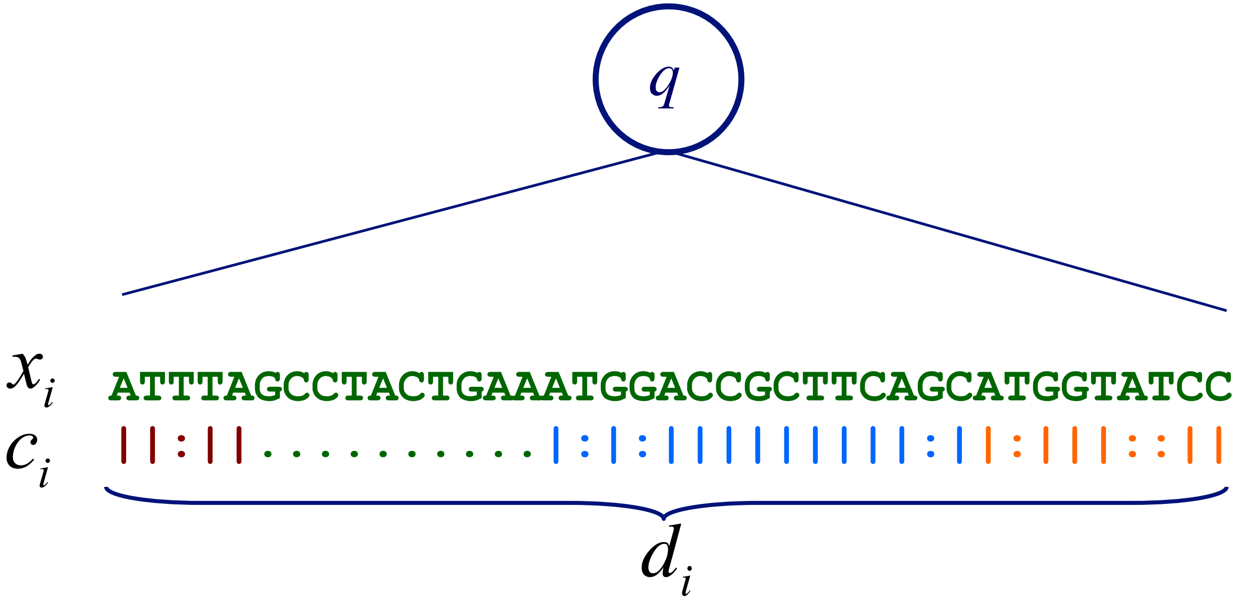


ACCGTTACGTGTCATTCTACGTGATCATCGGATCCTAGAATCATCGATCCGTGCGATCGATCGGATTAGCTAGCTTAGCTAGGAGAGCATCGATCGGATCGAGGAGGAGCCTATATAAATCAA

# Modeling Sequences in TWINSCAN

- each state “emits” two sequences
  - the given DNA sequence, $x$
  - the conservation sequence, $c$
- it treats them as conditionally independent given the state

$$\Pr(x_i, c_i \mid q) = \Pr(d_i \mid q)\ \Pr(x_i \mid q, d_i)\ \Pr(c_i \mid q, d_i)$$



# Modeling Sequences in TWINSCAN

- conservation sequence is treated just as a string over a 3-character alphabet ( | , : , . )

- conservation sequence emissions modeled by
  - inhomogeneous 2nd-order chains for splice sites
  - homogeneous 5th-order Markov chains for other states

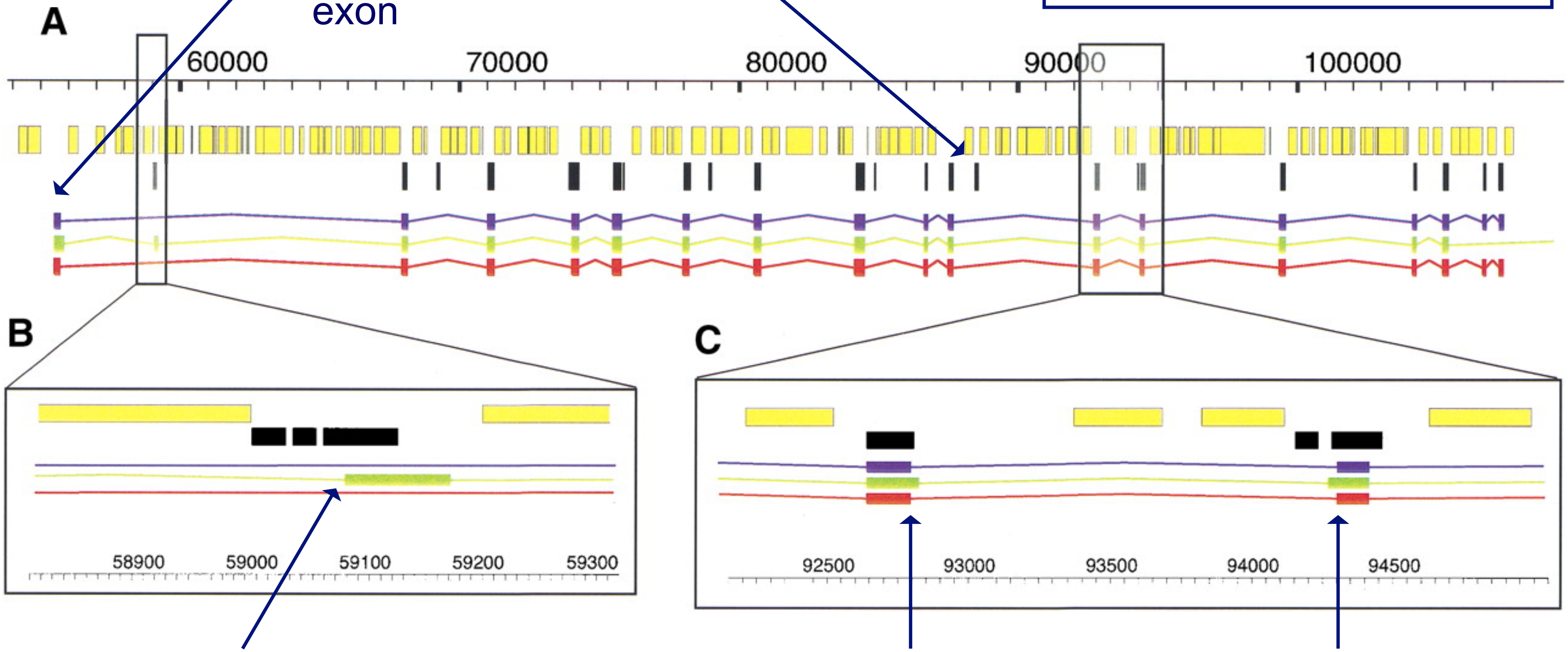
- like GENSCAN, based on hidden semi-Markov models

- algorithms for learning, inference same as GENSCAN

# TWINSCAN vs. GENSCAN

■ mouse alignments
🟦 RefSeq (gold standard)
🟩 GENSCAN prediction
🟥 TWINSCAN prediction

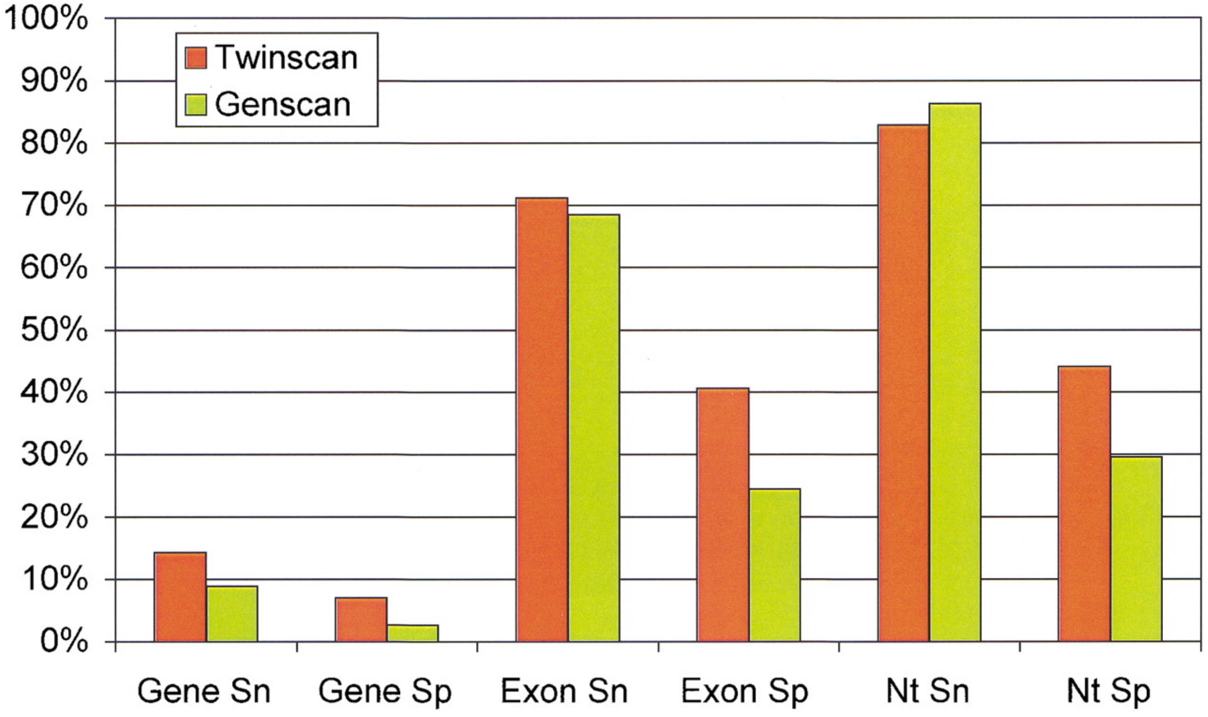
conservation is neither necessary nor sufficient to predict an exon



TWINSCAN correctly omits this exon prediction because conserved region ends within it

TWINSCAN correctly predicts both splice sites because they are within the aligned regions

# GENSCAN vs. TWINSCAN: Empirical Comparison



$$\text{sensitivity (Sn)} = \frac{TP}{TP + FN}$$

$$\text{specificity (Sp)} = \frac{TP}{TP + FP}$$

note: the definition of *specificity* here is somewhat nonstandard; it’s the same as *precision*

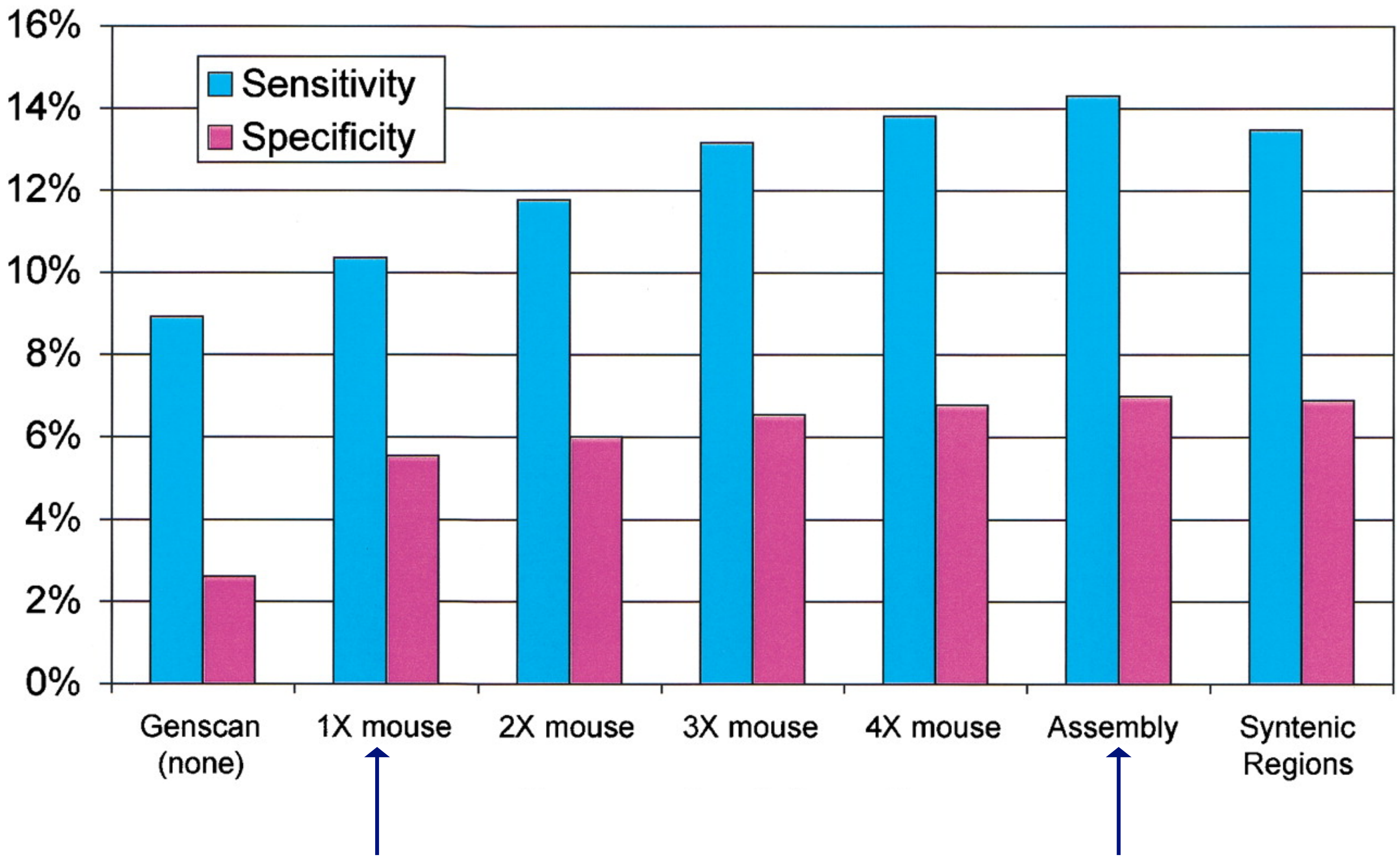
genes exactly correct?

exons exactly correct?

nucleotides correct?

Figure from Flicek et al., *Genome Research*, 2003

# Accuracy of TWINSCAN as a Function of Sequence Coverage



very crude mouse genome sequence

good mouse genome sequence

# SLAM

Pachter et al., *RECOMB* 2001

- prediction with SLAM

  given: a pair of sequences to be parsed, $x$ and $y$

  find approximate alignment of $x$ and $y$

  run constrained Viterbi to have HMM simultaneously parse and align $x$ and $y$

- training with SLAM

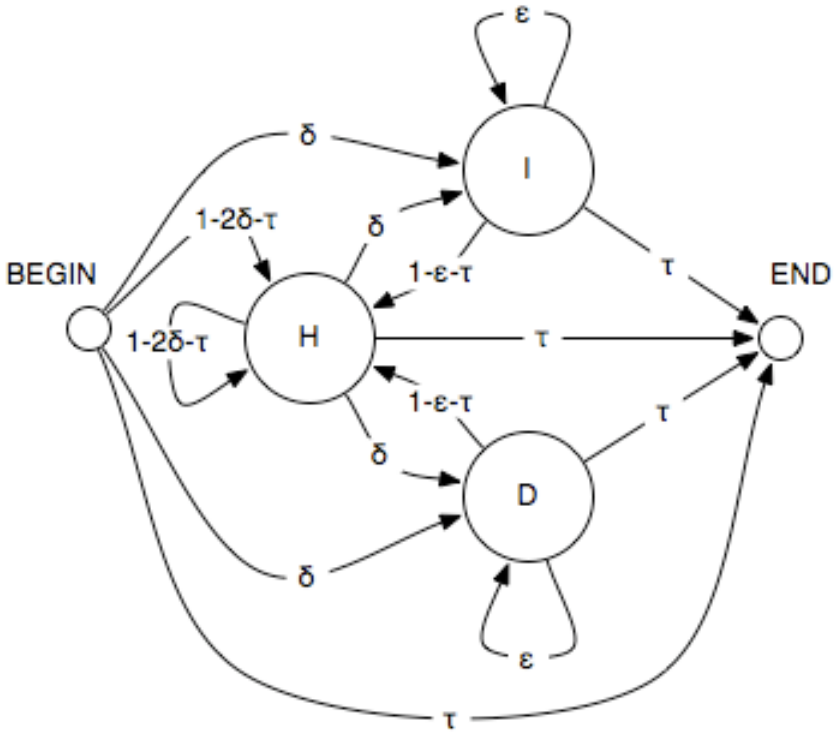
  given: a set of aligned pairs of training sequences $X$

  for each $x$, $y$ in $X$

  infer emission/alignment parameters for both $x$ and $y$

# Pair Hidden Markov Models

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

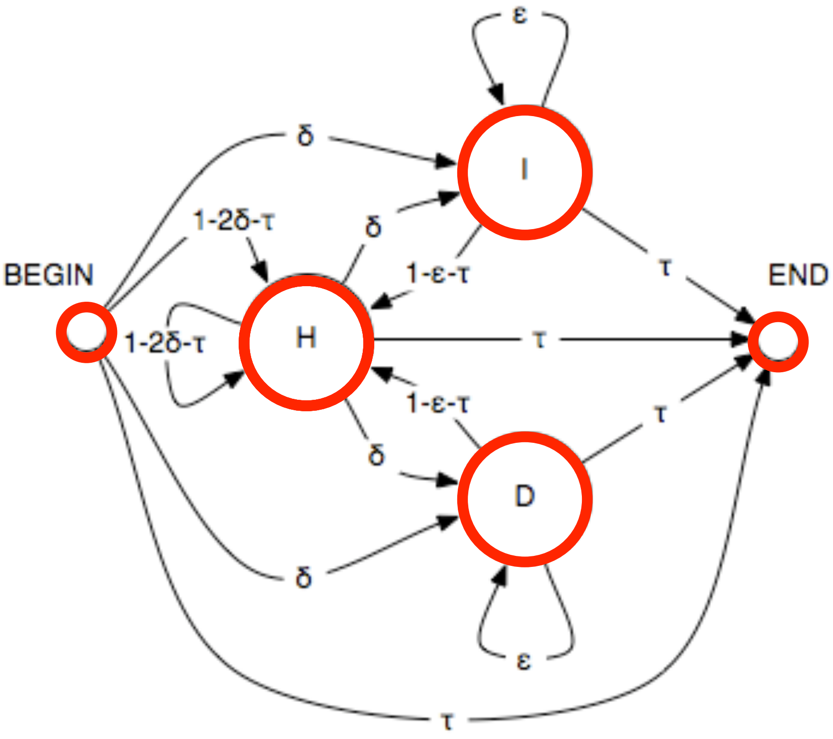


H: homology (match) state

I: insert state

D: delete state

# PHMM Paths = Alignments



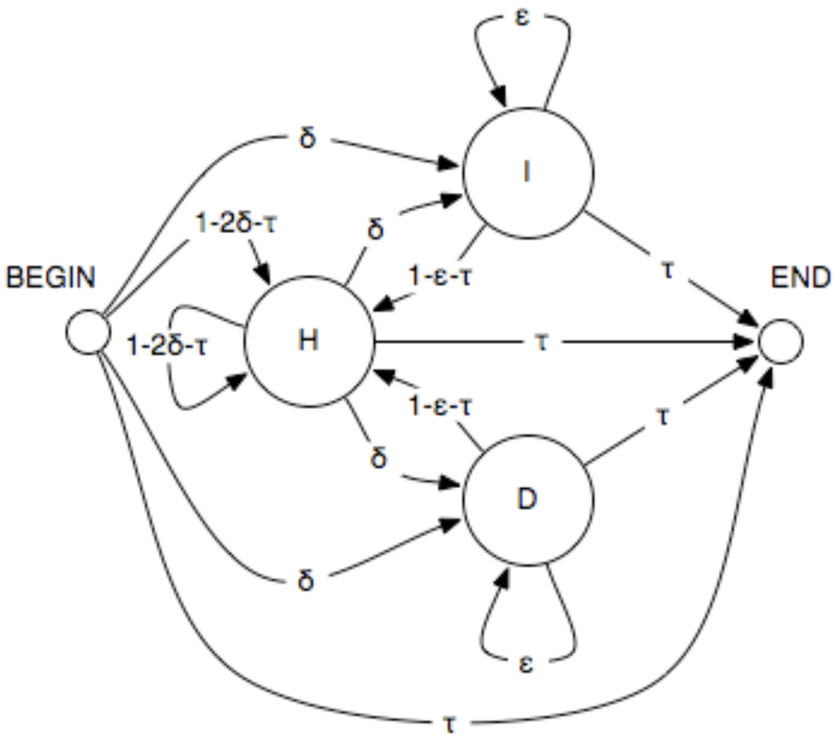
sequence 1: AAGCGC

sequence 2: ATGTC

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| hidden: | B | H | H | I | I | H | D | H | E |
| observed: | | A | A | G | C | G | | C | |
| | | A | T | | | G | T | C | |

# Transition Probabilities

- probabilities of moving between states at each step



state i+1

state i

|   | 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

## Deletion (D)

$$e_D(x_i)$$

| | |
|---|---|
| A | 0.3 |
| C | 0.2 |
| G | 0.3 |
| T | 0.2 |

single character

## Insertion (I)

$$e_I(y_j)$$

| | |
|---|---|
| A | 0.1 |
| C | 0.4 |
| G | 0.4 |
| T | 0.1 |

single character

## Homology (H)

$$e_H(x_i, y_j)$$

| | 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 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

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

- note that the recurrence relations here allow $I \rightarrow D$ and $D \rightarrow I$ transitions

# 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 (NW), to obtain sequence of state states giving highest probability

HIDHHDDIIHH...

# Correspondence with NW

- NW values ≈ logarithms of PHMM Viterbi values

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

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

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

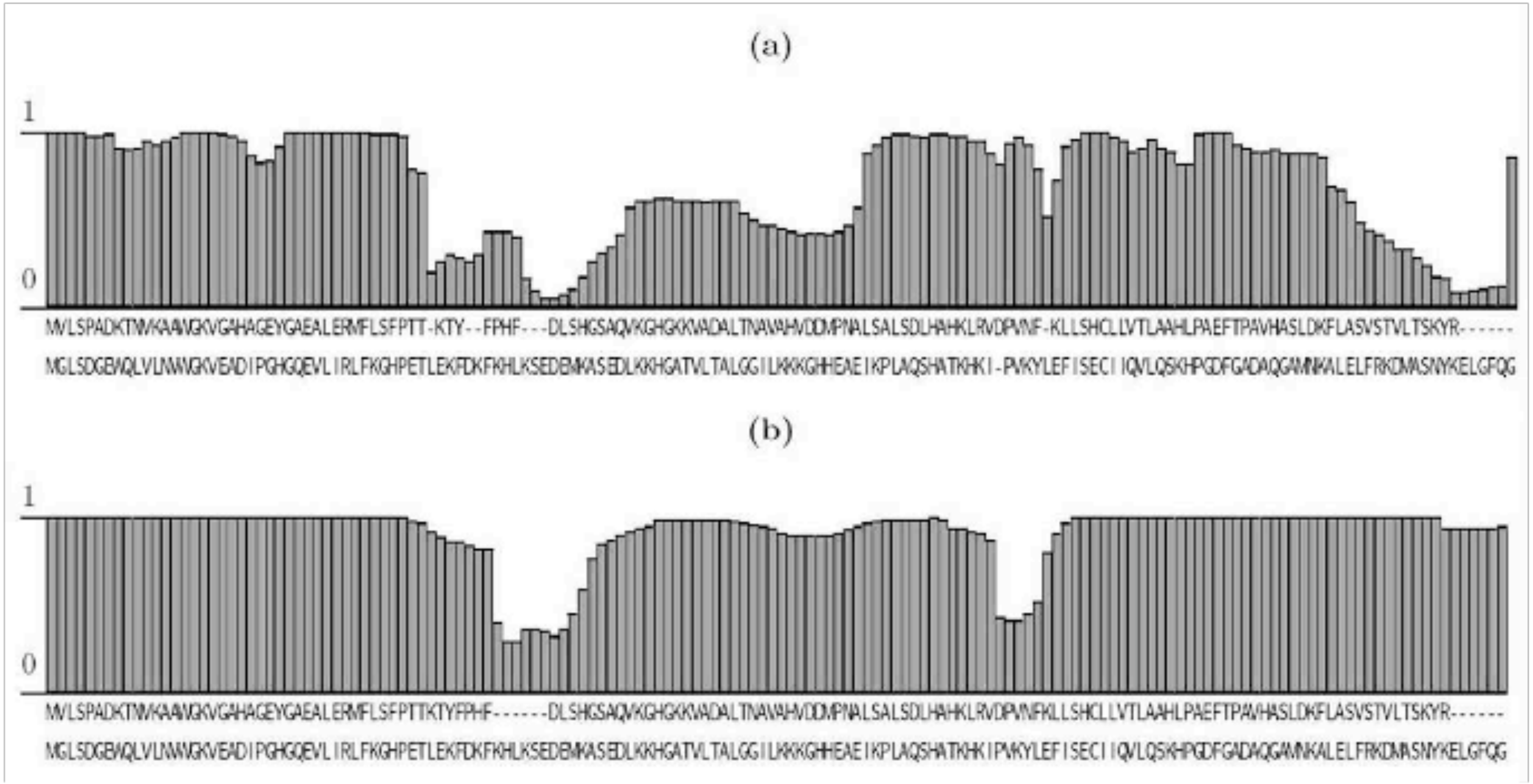
# Posterior Probabilities

- there are similar recurrences for the *Forward* and *Backward* values
- from the *Forward* and *Backward* values, we can calculate the posterior probability of the event that the path passes through a certain state $S$, after generating length $i$ and $j$ prefixes

# Uses for Posterior Probabilities

- sampling of suboptimal alignments
- posterior probability of pairs of residues being homologous (aligned to each other)
- posterior probability of a residue being gapped
- training model parameters (EM)

# Posterior Probabilities



plot of 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)

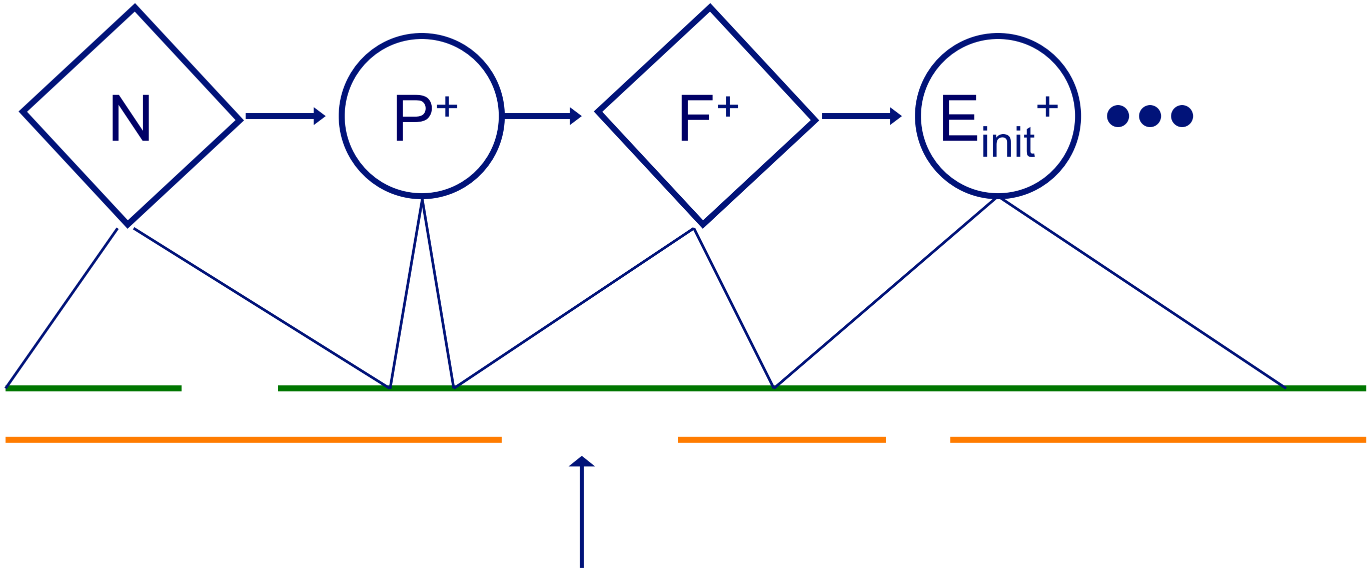
# *Generalized* Pair HMMs

- represent a parse π, as a sequence of states and a sequence of associated lengths for <u>each</u> input sequence

$$\vec{q} = \{q_1, q_2, \ldots, q_n\}$$

$$\vec{d} = \{d_1, d_2, \ldots, d_n\}$$

$$\vec{e} = \{e_1, e_2, \ldots, e_n\}$$



may be gaps
in the sequences

# Generalized Pair HMMs

- representing a parse π, as a sequence of states and associated lengths (durations)

$$\vec{q} = \{q_1, q_2, \ldots, q_n\}$$

$$\vec{d} = \{d_1, d_2, \ldots, d_n\} \qquad \vec{e} = \{e_1, e_2, \ldots, e_n\}$$

- the joint probability of generating parse π and sequences $x$ and $y$

$$P(x,y,\pi) = a_{start,1} P(d_1, e_1 \mid q_1) P(x_1, y_1 \mid q_1, d_1, e_1) \times$$

$$\prod_{k=2}^{n} a_{k-1,k} P(d_k, e_k \mid q_k) P(x_k, y_k \mid q_k, d_k, e_k)$$

# Generalized Pair HMM Algorithms

- Generalized HMM Forward Algorithm

$$f_l(i) = \sum_k \sum_{d=1}^{D} \left[ f_k(i-d) \;\; a_{kl} \;\; P(d \mid q_l) \; P(x_{i-d+1}^{i} \mid q_l, d) \right]$$

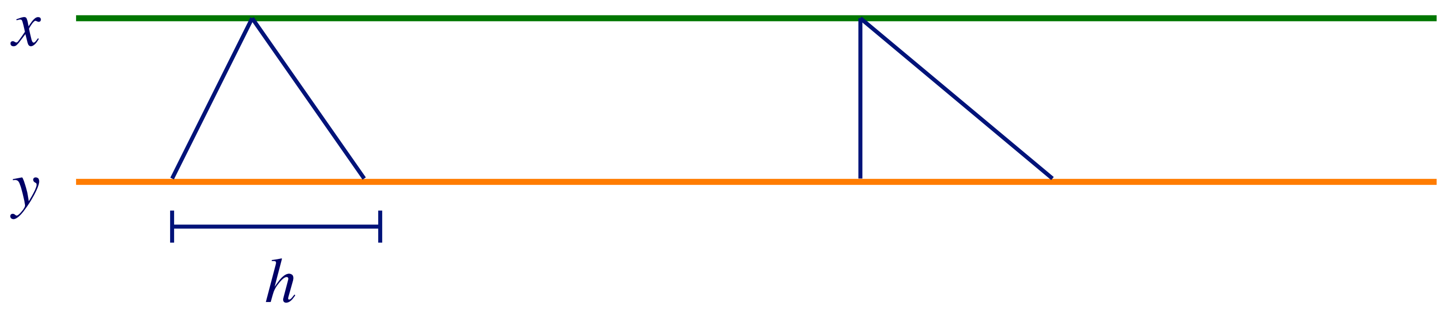
- Generalized Pair HMM Algorithm

$$f_l(i,j) = \sum_k \sum_{d=1}^{D} \sum_{e=1}^{D} \left[ f_k(i-d, j-e) \; a_{kl} P(d,e \mid q_l) \; P(x_{i-d+1}^{i} y_{j-e+1}^{j} \mid q_l, d, e) \right]$$

- Viterbi: replace sum with max

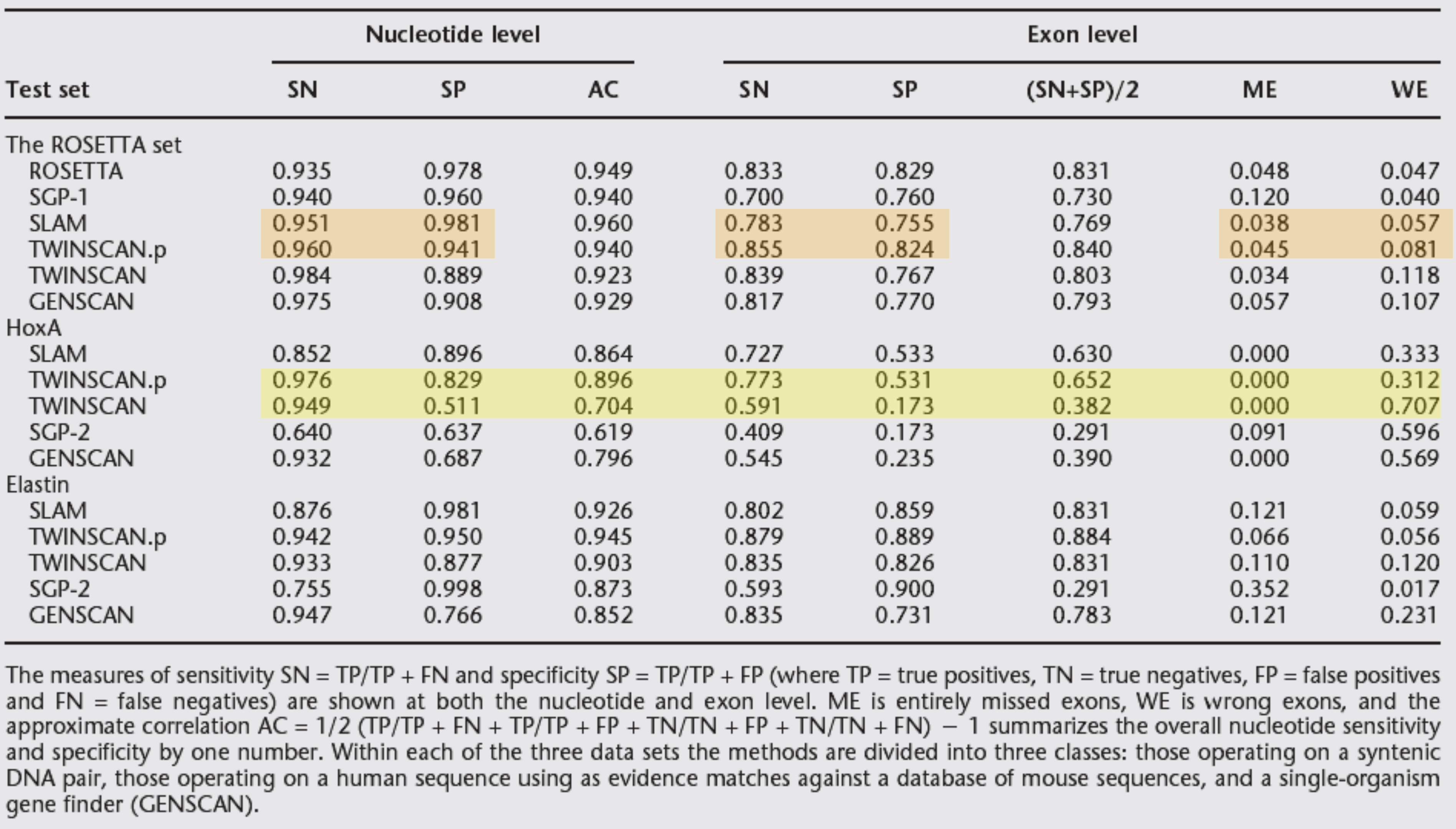
# Prediction in SLAM

- could find alignment and gene predictions by running Viterbi
- to make it more efficient
  - find an approximate alignment (using a fast anchor-based approach)
  - each base in $x$ constrained to align to a window of size $h$ in $\mathrm{y}$



- analogous to banded alignment methods

# GENSCAN, TWINSCAN, & SLAM



| | Nucleotide level | | | Exon level | | | | |
|---|---|---|---|---|---|---|---|---|
| Test set | SN | SP | AC | SN | SP | (SN+SP)/2 | ME | WE |
| The ROSETTA set | | | | | | | | |
| ROSETTA | 0.935 | 0.978 | 0.949 | 0.833 | 0.829 | 0.831 | 0.048 | 0.047 |
| SGP-1 | 0.940 | 0.960 | 0.940 | 0.700 | 0.760 | 0.730 | 0.120 | 0.040 |
| SLAM | 0.951 | 0.981 | 0.960 | 0.783 | 0.755 | 0.769 | 0.038 | 0.057 |
| TWINSCAN.p | 0.960 | 0.941 | 0.940 | 0.855 | 0.824 | 0.840 | 0.045 | 0.081 |
| TWINSCAN | 0.984 | 0.889 | 0.923 | 0.839 | 0.767 | 0.803 | 0.034 | 0.118 |
| GENSCAN | 0.975 | 0.908 | 0.929 | 0.817 | 0.770 | 0.793 | 0.057 | 0.107 |
| HoxA | | | | | | | | |
| SLAM | 0.852 | 0.896 | 0.864 | 0.727 | 0.533 | 0.630 | 0.000 | 0.333 |
| TWINSCAN.p | 0.976 | 0.829 | 0.896 | 0.773 | 0.531 | 0.652 | 0.000 | 0.312 |
| TWINSCAN | 0.949 | 0.511 | 0.704 | 0.591 | 0.173 | 0.382 | 0.000 | 0.707 |
| SGP-2 | 0.640 | 0.637 | 0.619 | 0.409 | 0.173 | 0.291 | 0.091 | 0.596 |
| GENSCAN | 0.932 | 0.687 | 0.796 | 0.545 | 0.235 | 0.390 | 0.000 | 0.569 |
| Elastin | | | | | | | | |
| SLAM | 0.876 | 0.981 | 0.926 | 0.802 | 0.859 | 0.831 | 0.121 | 0.059 |
| TWINSCAN.p | 0.942 | 0.950 | 0.945 | 0.879 | 0.889 | 0.884 | 0.066 | 0.056 |
| TWINSCAN | 0.933 | 0.877 | 0.903 | 0.835 | 0.826 | 0.831 | 0.110 | 0.120 |
| SGP-2 | 0.755 | 0.998 | 0.873 | 0.593 | 0.900 | 0.291 | 0.352 | 0.017 |
| GENSCAN | 0.947 | 0.766 | 0.852 | 0.835 | 0.731 | 0.783 | 0.121 | 0.231 |

The measures of sensitivity SN = TP/TP + FN and specificity SP = TP/TP + FP (where TP = true positives, TN = true negatives, FP = false positives and FN = false negatives) are shown at both the nucleotide and exon level. ME is entirely missed exons, WE is wrong exons, and the approximate correlation AC = 1/2 (TP/TP + FN + TP/TP + FP + TN/TN + FP + TN/TN + FN) − 1 summarizes the overall nucleotide sensitivity and specificity by one number. Within each of the three data sets the methods are divided into three classes: those operating on a syntenic DNA pair, those operating on a human sequence using as evidence matches against a database of mouse sequences, and a single-organism gene finder (GENSCAN).

# TWINSCAN vs. SLAM

- both use multiple genomes to predict genes
- both use generalized HMMs
- TWINSCAN
  - takes as an input a genomic sequence, and a conservation sequence computed from an informant genome
  - models probability of both sequences; assumes they’re conditionally independent given the state
  - predicts genes only in the genomic sequence
- SLAM
  - takes as input two genomic sequences
  - models joint probability of pairs of aligned sequences
  - can simultaneously predict genes and compute alignments