

Markov Models for Gene Finding

BMI/CS 776

www.biostat.wisc.edu/bmi776/

Spring 2020

Daifeng Wang

daifeng.wang@wisc.edu

Outline for Gene Finding

- Interpolated Markov Model
 - Finding bacterial genes
- Generalized Hidden Markov Model
 - Finding eukaryotic genes
 - Comparative information

Interpolated Markov Models for Gene Finding

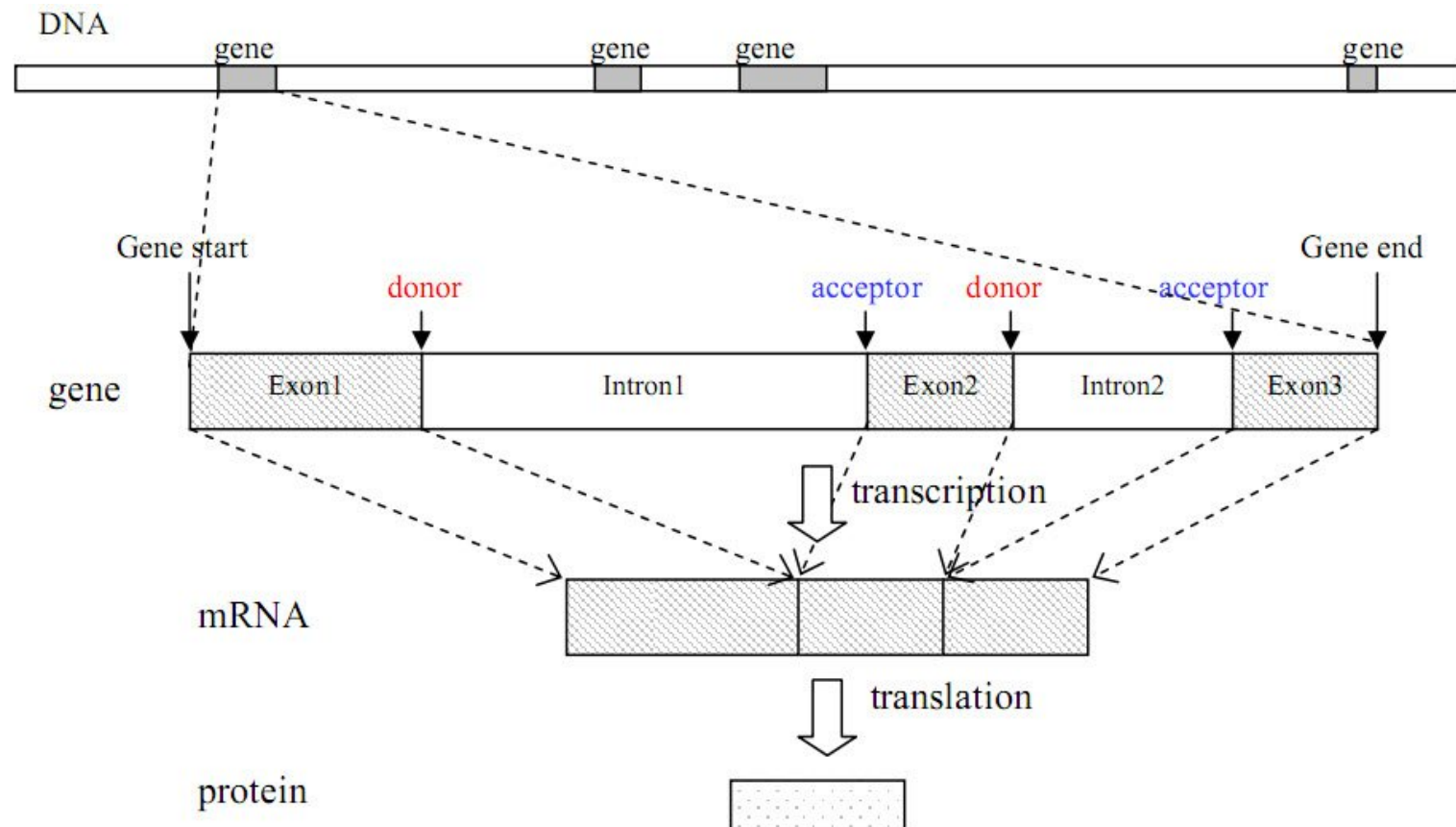
Key concepts

- the gene-finding task
- the trade-off between potential predictive value and parameter uncertainty in choosing the order of a Markov model
- interpolated Markov models

The Gene Finding Task

Given: an uncharacterized DNA sequence

Do: locate the genes in the sequence, including the coordinates of individual *exons* and *introns*



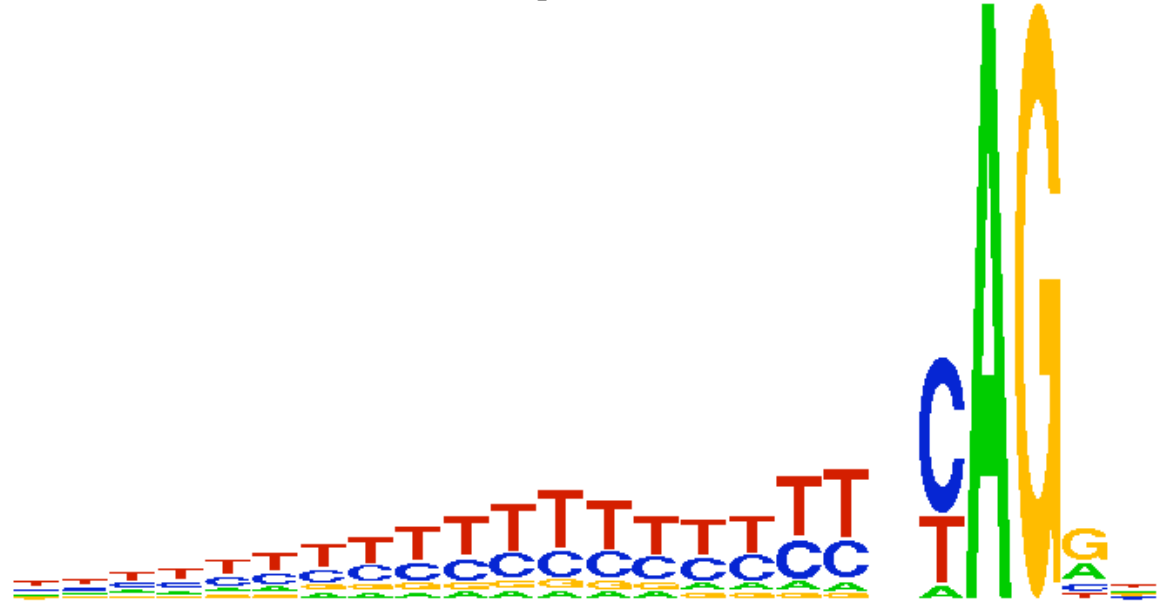
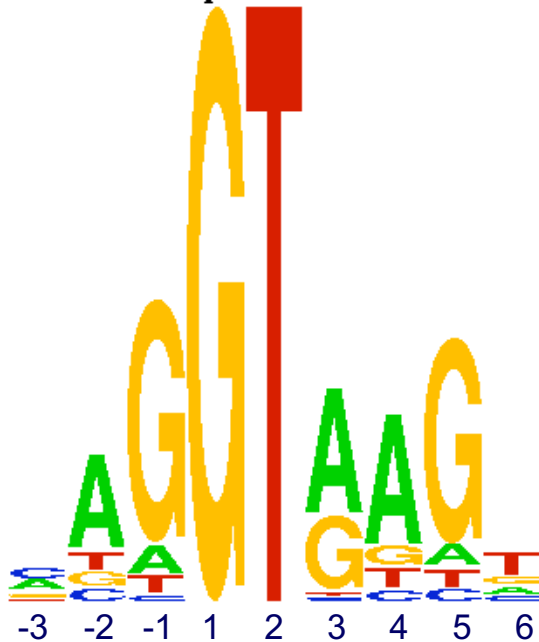
Splice Signals Example

donor sites

acceptor sites

5' splice site

3' splice site



exon

exon

Figures from Yi Xing

- There are significant dependencies among non-adjacent positions in donor splice signals
- Informative for inferring hidden state of HMM

Sources of Evidence for Gene Finding

- **Signals:** the sequence *signals* (e.g. splice junctions) involved in gene expression (e.g., RNA-seq reads)
- **Content:** statistical properties that distinguish protein-coding DNA from non-coding DNA (**focus in this lecture**)
- **Conservation:** signal and content properties that are conserved across related sequences (e.g. orthologous regions of the mouse and human genome)

Gene Finding: Search by Content

- Encoding a protein affects the statistical properties of a DNA sequence
 - some amino acids are used more frequently than others (Leu more prevalent than Trp)
 - different numbers of codons for different amino acids (Leu has 6, Trp has 1)
 - for a given amino acid, usually one codon is used more frequently than others
 - this is termed *codon preference*
 - these preferences vary by species

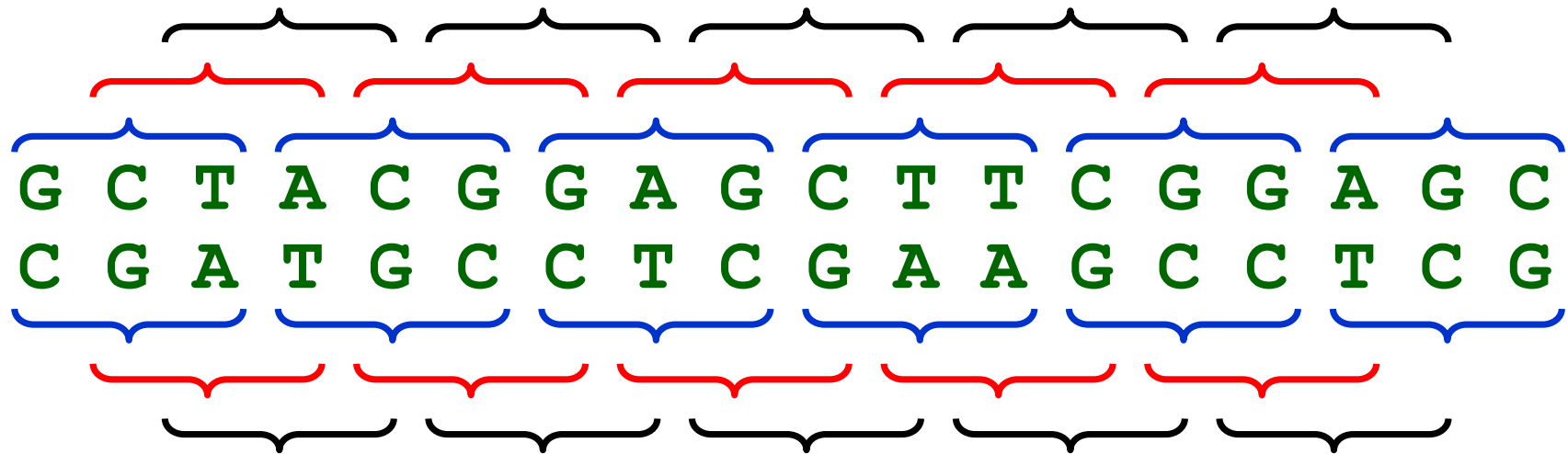
Codon Preference in E. Coli

AA	codon	/1000

Gly	GGG	1.89
Gly	GGA	0.44
Gly	GGU	52.99
Gly	GGC	34.55
Glu	GAG	15.68
Glu	GAA	57.20
Asp	GAU	21.63
Asp	GAC	43.26

Reading Frames

- A given sequence may encode a protein in any of the six reading frames



Open Reading Frames (ORFs)

- An ORF is a sequence that
 - starts with a potential start codon (e.g., ATG)
 - ends with a potential stop codon, *in the same reading frame* (e.g., TAG, TAA, TGA)
 - doesn't contain another stop codon in-frame
 - and is sufficiently long (say > 100 bases)

G T T A T G G C T ... T C G T G A T T

- An ORF meets the minimal requirements to be a protein-coding gene in an organism without introns

Markov Models & Reading Frames

- Consider modeling a given coding sequence
- For each “word” we evaluate, we’ll want to consider its position with respect to the reading frame we’re assuming

reading frame

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

G C T A C G

G is in 3rd codon position

C T A C G G

G is in 1st position

T A C G G A

A is in 2nd position



- Can do this using an inhomogeneous model

Inhomogeneous Markov Model

- **Homogenous Markov model:** transition probability matrix does not change over time or position
- **Inhomogenous Markov model:** transition probability matrix depends on the time or position

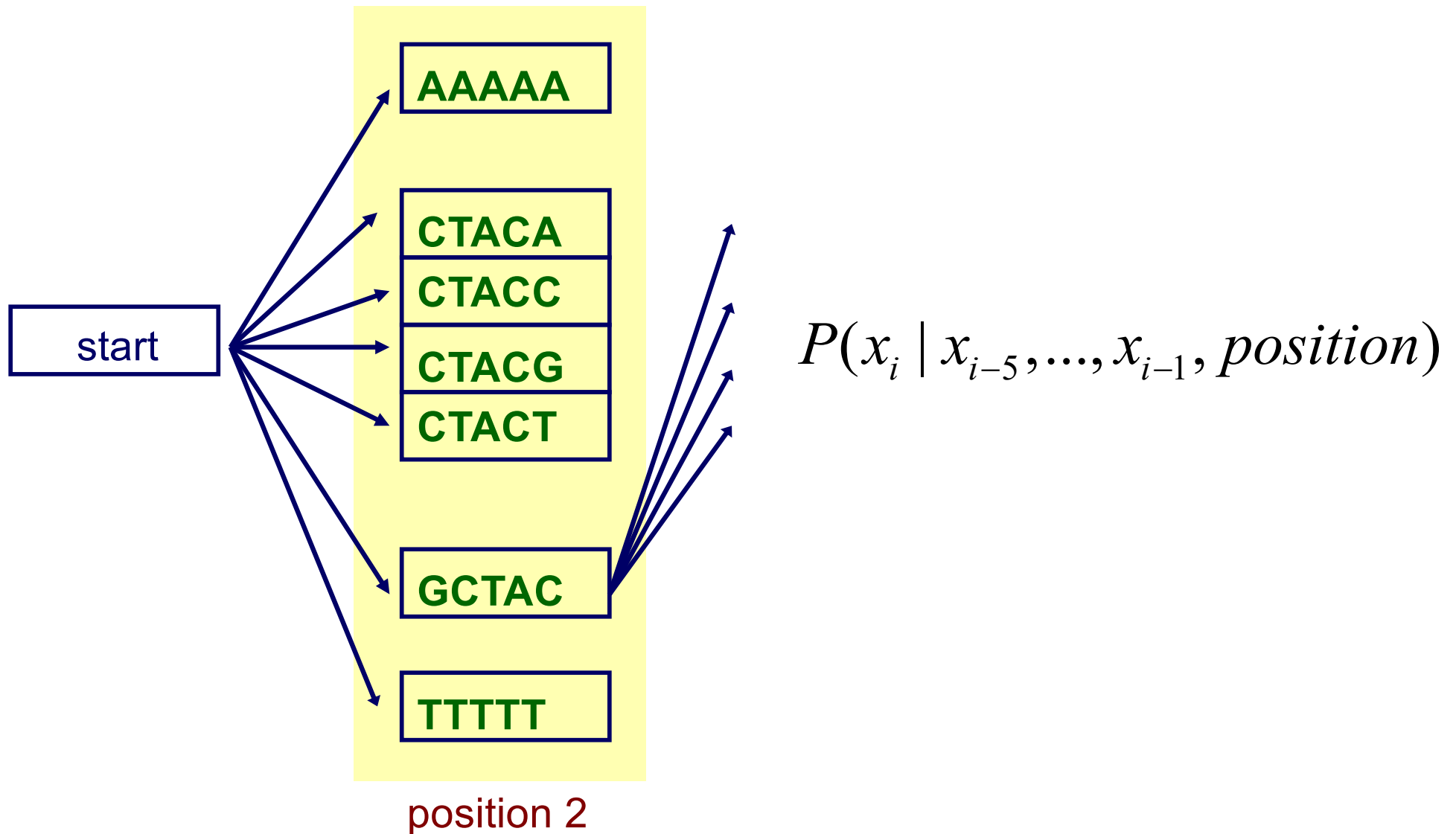
Higher Order Markov Models

- Higher order models remember more “history”
 - n -order $P(x_i | x_{i-1}, x_{i-2}, \dots, x_1) = P(x_i | x_{i-1}, \dots, x_{i-n})$
- Additional history can have predictive value
- Example:
 - predict the next word in this sentence fragment
“...you__” (are, give, passed, say, see, too, ...?)
 - now predict it given more history
“...can you__”
“...say can you__”
“...oh say can you__”

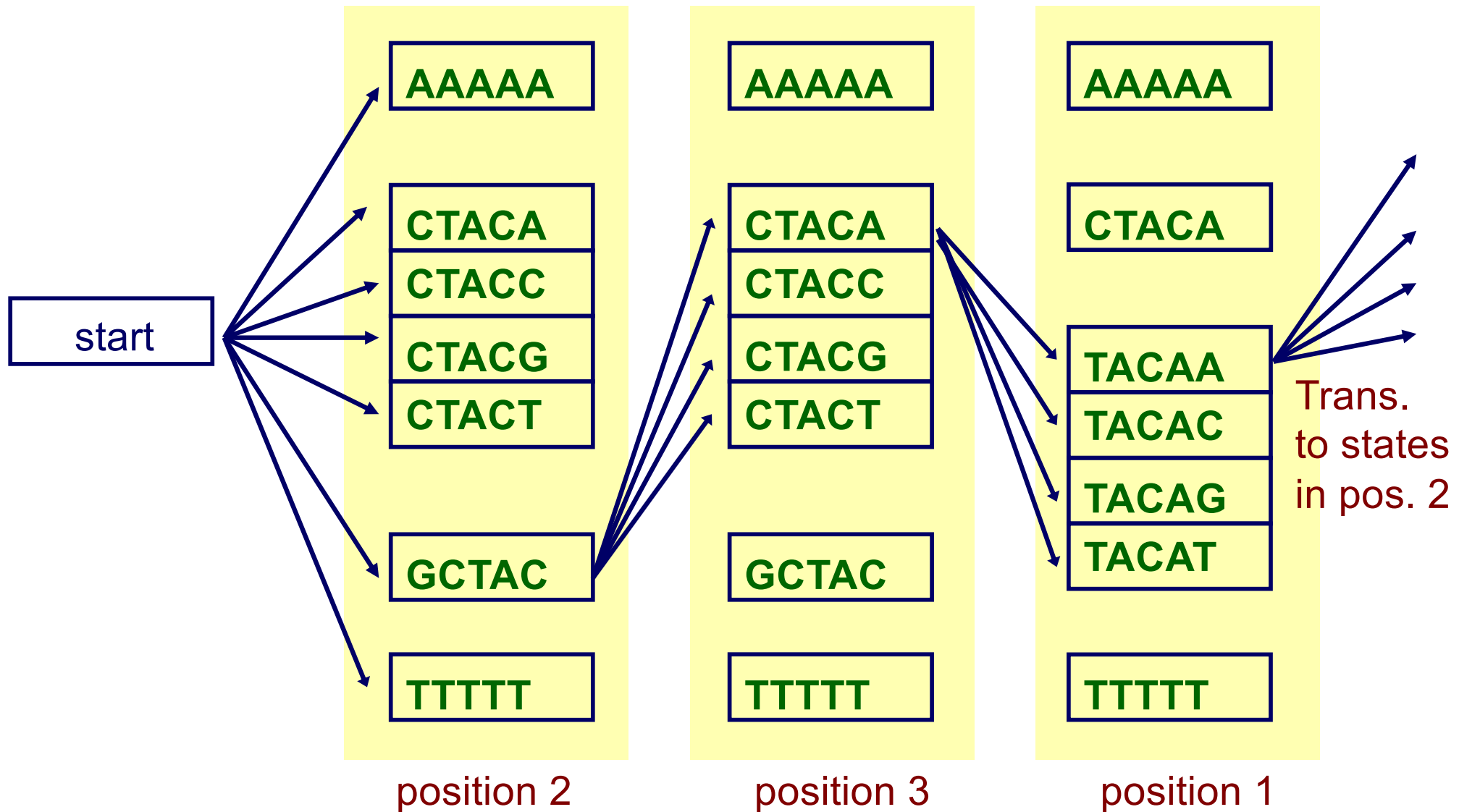


YouTube

A Fifth Order Inhomogeneous Markov Model



A Fifth Order Inhomogeneous Markov Model



Selecting the Order of a Markov Model

- But the number of parameters we need to estimate grows exponentially with the order
 - for modeling DNA we need $O(4^{n+1})$ parameters for an n th order model
- The higher the order, the less reliable we can expect our parameter estimates to be
- Suppose we have 100k bases of sequence to estimate parameters of a model
 - for a 2nd order homogeneous Markov chain, we'd see each history 6250 times on average
 - for an 8th order chain, we'd see each history ~ 1.5 times on average

Interpolated Markov Models

- The IMM idea: manage this trade-off by interpolating among models of various orders
- *Simple* linear interpolation:

$$\begin{aligned} P_{\text{IMM}}(x_i \mid x_{i-n}, \dots, x_{i-1}) &= \lambda_0 P(x_i) \\ &\quad + \lambda_1 P(x_i \mid x_{i-1}) \\ &\quad \dots \\ &\quad + \lambda_n P(x_i \mid x_{i-n}, \dots, x_{i-1}) \end{aligned}$$

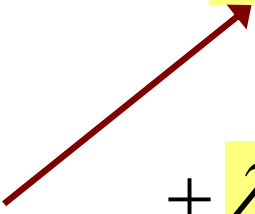
- where $\sum_i \lambda_i = 1$

Interpolated Markov Models

- We can make the weights depend on the history
 - for a given order, we may have significantly more data to estimate some words than others
- *General* linear interpolation

$$P_{\text{IMM}}(x_i \mid x_{i-n}, \dots, x_{i-1}) = \lambda_0 P(x_i) \\ + \lambda_1(x_{i-1}) P(x_i \mid x_{i-1}) \\ \dots \\ + \lambda_n(x_{i-n}, \dots, x_{i-1}) P(x_i \mid x_{i-n}, \dots, x_{i-1})$$

λ is a function of the given history



The GLIMMER System

[Salzberg et al., Nucleic Acids Research, 1998]

- System for identifying genes in bacterial genomes
- Uses 8th order, inhomogeneous, interpolated Markov models



Matt MacManes

@macmanes

Follow



Did people really stop developing ab initio gene predictors in like 2009?

9:40 AM - 29 Dec 2017



Titus Brown @ctitusbrown · 29 Dec 2017



Replying to @macmanes

I think so. From what I recall, bacterial gene prediction is 99% accurate/sensitive, and euk gene prediction is horrendously inaccurate so => mRNAseq and homology methods took over.

IMMs in GLIMMER

- How does GLIMMER determine the λ values?
- First, let's express the IMM probability calculation recursively

$$\begin{aligned} P_{\text{IMM},n}(x_i \mid x_{i-n}, \dots, x_{i-1}) = \\ \lambda_n(x_{i-n}, \dots, x_{i-1})P(x_i \mid x_{i-n}, \dots, x_{i-1}) + \\ [1 - \lambda_n(x_{i-n}, \dots, x_{i-1})]P_{\text{IMM},n-1}(x_i \mid x_{i-n+1}, \dots, x_{i-1}) \end{aligned}$$

- Let $c(x_{i-n}, \dots, x_{i-1})$ be the number of times we see the history x_{i-n}, \dots, x_{i-1} in our training set

$$\lambda_n(x_{i-n}, \dots, x_{i-1}) = 1 \quad \text{if} \quad c(x_{i-n}, \dots, x_{i-1}) > 400$$

IMMs in GLIMMER

- If we haven't seen x_{i-n}, \dots, x_{i-1} more than 400 times, then compare the counts for the following:

n th order history + base

$x_{i-n}, \dots, x_{i-1}, a$

$x_{i-n}, \dots, x_{i-1}, c$

$x_{i-n}, \dots, x_{i-1}, g$

$x_{i-n}, \dots, x_{i-1}, t$

$(n-1)$ th order history + base

$x_{i-n+1}, \dots, x_{i-1}, a$

$x_{i-n+1}, \dots, x_{i-1}, c$

$x_{i-n+1}, \dots, x_{i-1}, g$

$x_{i-n+1}, \dots, x_{i-1}, t$

- Use a statistical test to assess whether the distributions of x_i depend on the order

IMMs in GLIMMER

n th order history + base

$$x_{i-n}, \dots, x_{i-1}, a$$

$$x_{i-n}, \dots, x_{i-1}, c$$

$$x_{i-n}, \dots, x_{i-1}, g$$

$$x_{i-n}, \dots, x_{i-1}, t$$

$(n-1)$ th order history + base

$$x_{i-n+1}, \dots, x_{i-1}, a$$

$$x_{i-n+1}, \dots, x_{i-1}, c$$

$$x_{i-n+1}, \dots, x_{i-1}, g$$

$$x_{i-n+1}, \dots, x_{i-1}, t$$

- Null hypothesis in χ^2 test: x_i distribution is independent of order
- Define $d = 1 - pvalue$
- If d is small we don't need the higher order history

IMMs in GLIMMER

- Putting it all together

$$\lambda_n(x_{i-n}, \dots, x_{i-1}) = \begin{cases} 1 & \text{if } c(x_{i-n}, \dots, x_{i-1}) > 400 \\ d \times \frac{c(x_{i-n}, \dots, x_{i-1})}{400} & \text{else if } d \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

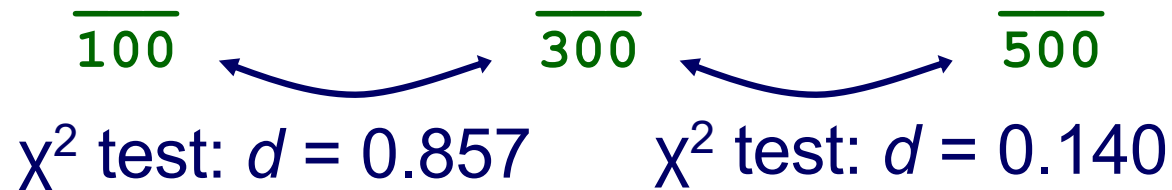
where $d \in (0,1)$

- why 400?
 - “gives ~95% confidence that the sample probabilities are within ± 0.05 of the true probabilities from which the sample was taken”

IMM Example

- Suppose we have the following counts from our training set

ACGA	25	CGA	100	GA	175
ACGC	40	CGC	90	GC	140
ACGG	15	CGG	35	GG	65
ACGT	20	CGT	75	GT	120



$$\lambda_3(\text{ACG}) = 0.857 \times 100/400 = 0.214$$

$$\lambda_2(\text{CG}) = 0 \quad (d < 0.5, \quad c(\text{CG}) < 400)$$

$$\lambda_1(\text{G}) = 1 \quad (c(\text{G}) > 400)$$

IMM Example (Continued)

- Now suppose we want to calculate $P_{\text{IMM},3}(T \mid ACG)$

$$\begin{aligned}P_{\text{IMM},1}(T \mid G) &= \lambda_1(G)P(T \mid G) + (1 - \lambda_1(G))P_{\text{IMM},0}(T) \\ &= P(T \mid G)\end{aligned}$$

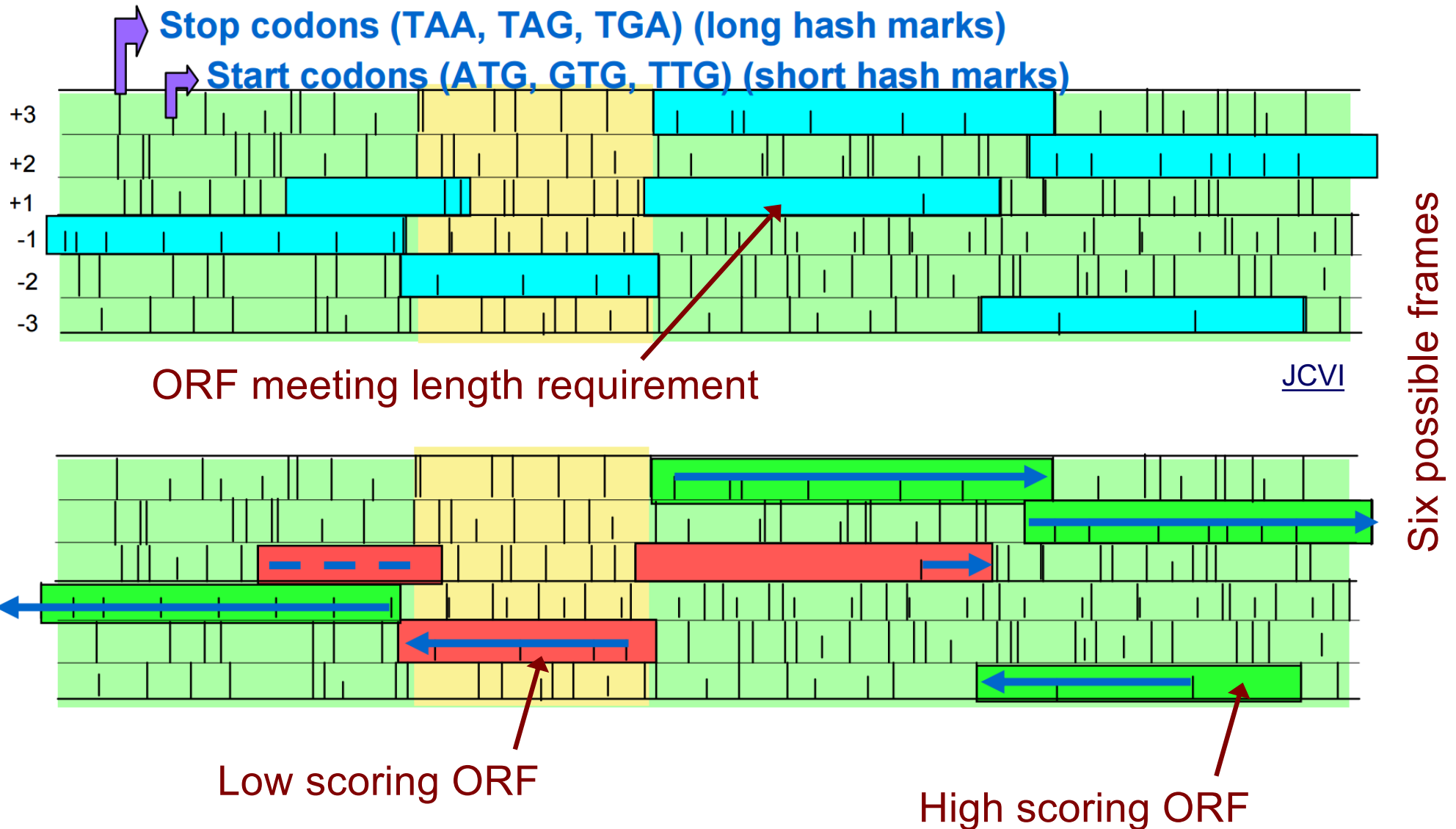
$$\begin{aligned}P_{\text{IMM},2}(T \mid CG) &= \lambda_2(CG)P(T \mid CG) + (1 - \lambda_2(CG))P_{\text{IMM},1}(T \mid G) \\ &= P(T \mid G)\end{aligned}$$

$$\begin{aligned}P_{\text{IMM},3}(T \mid ACG) &= \lambda_3(ACG)P(T \mid ACG) + (1 - \lambda_3(ACG))P_{\text{IMM},2}(T \mid CG) \\ &= 0.214 \times P(T \mid ACG) + (1 - 0.214) \times P(T \mid G) \\ &= 0.214 \times 0.2 + (1 - 0.214) \times 0.24\end{aligned}$$

Gene Recognition in GLIMMER

- Essentially ORF classification
 - Train and estimate IMMs
- For each ORF
 - calculate the probability of the ORF sequence in each of the 6 possible reading frames
 - if the highest scoring frame corresponds to the reading frame of the ORF, mark the ORF as a gene
- For overlapping ORFs that look like genes
 - score overlapping region separately
 - predict only one of the ORFs as a gene

Gene Recognition in GLIMMER



GLIMMER Experiment

- 8th order IMM vs. 5th order Markov model
- Trained on 1168 genes (ORFs really)
- Tested on 1717 annotated (more or less known) genes

GLIMMER Results

	TP	FN	FP & TP?
Model	Genes found	Genes missed	Additional genes
GLIMMER IMM	1680 (97.8%)	37	209
5 th -Order Markov	1574 (91.7%)	143	104

The first column indicates how many of the 1717 annotated genes in *H.influenzae* were found by each algorithm. The ‘additional genes’ column shows how many extra genes, not included in the 1717 annotated entries, were called genes by each method.

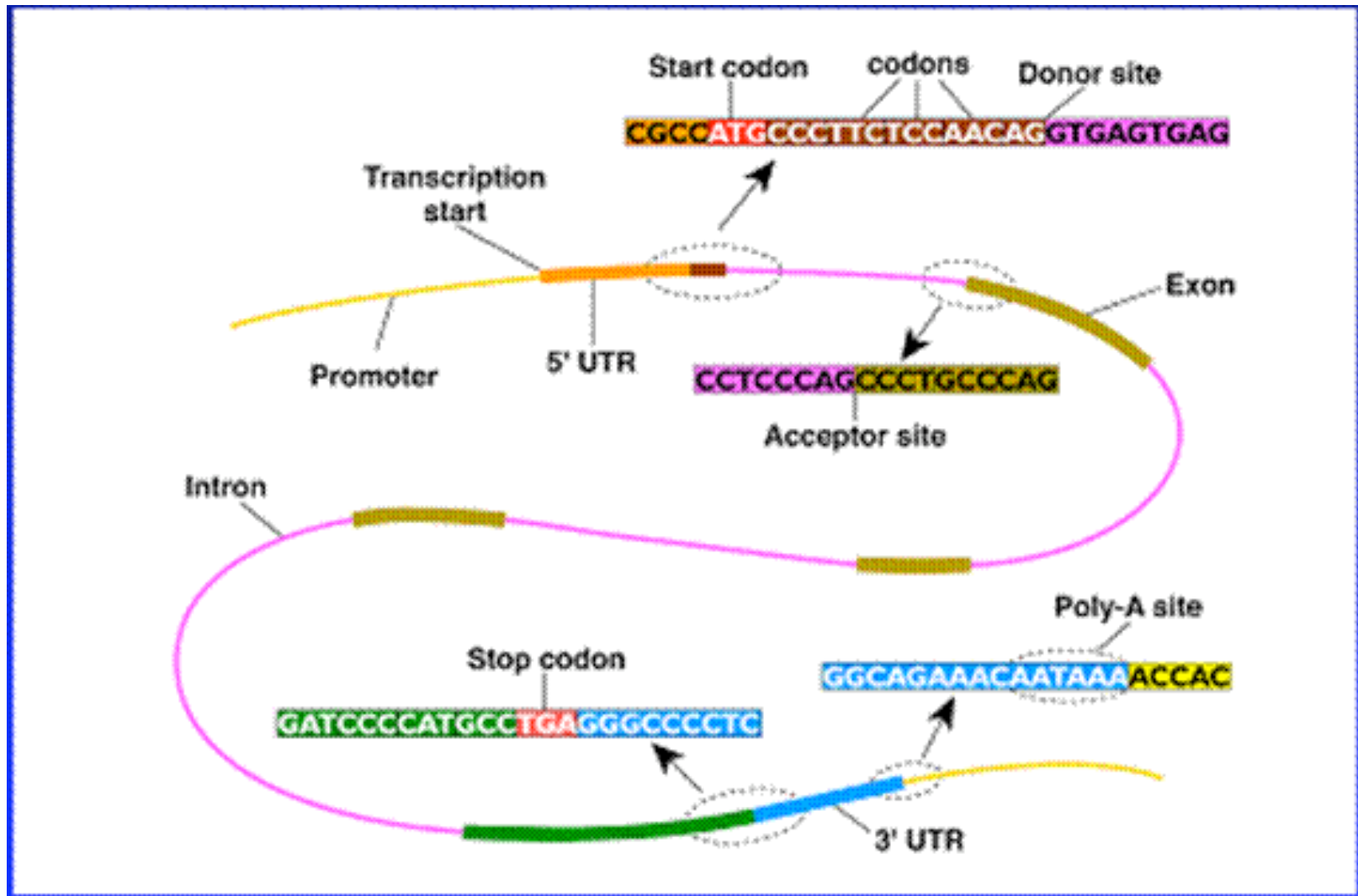
- GLIMMER has greater sensitivity than the baseline
- It's not clear whether its precision/specificity is better

Eukaryotic and Comparative Gene Finding

Key concepts

- Incorporating sequence signals into gene finding with HMMs
- Modeling durations with generalized HMMs (GENSCAN)
- Modeling conversation with pair HMMs
- Related genomes as an additional source of evidence for gene finding
- Extending GENSCAN to emits pairs of observed variables
- Modern gene finding and genome annotation

Eukaryotic Gene Structure



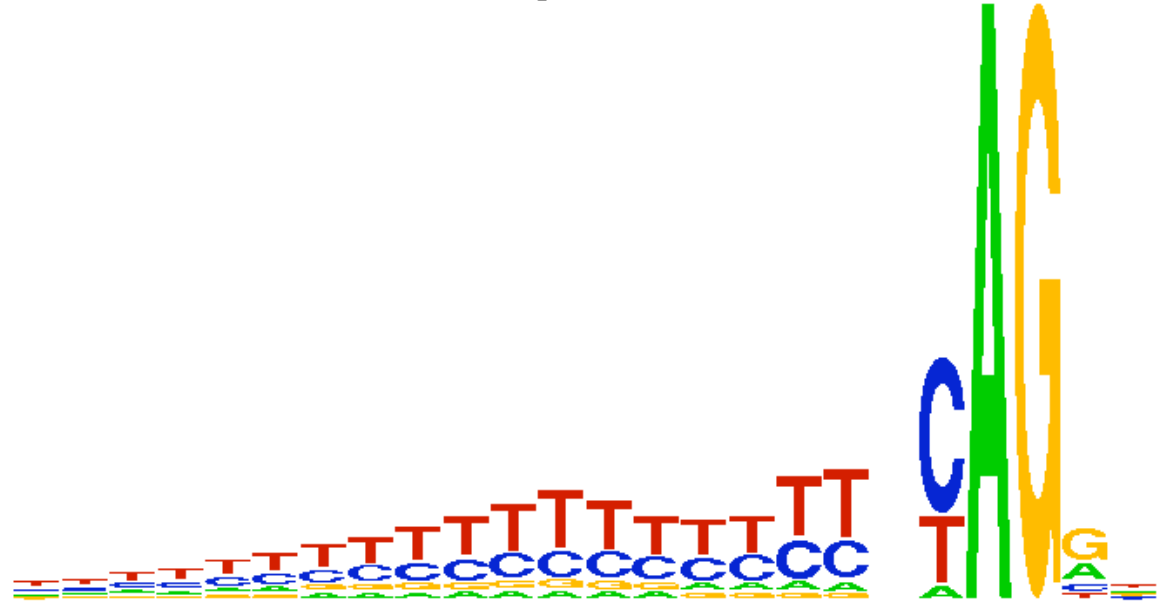
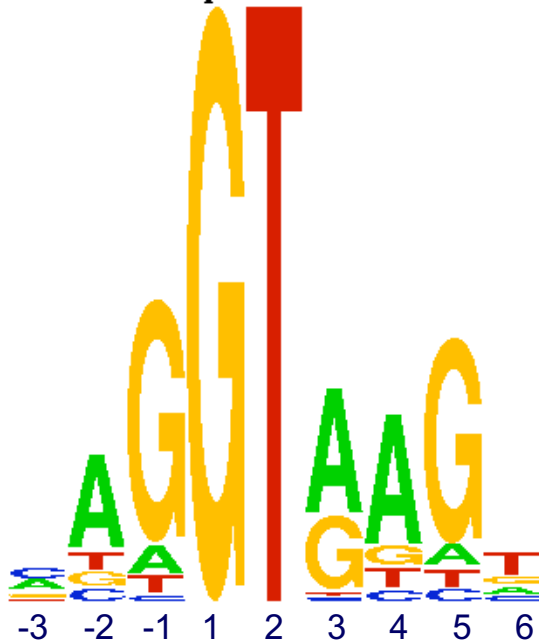
Splice Signals Example

donor sites

acceptor sites

5' splice site

3' splice site



Figures from Yi Xing

exon

exon

- There are significant dependencies among non-adjacent positions in donor splice signals
- Informative for inferring hidden state of HMM

Hidden Markov Model (HMM)

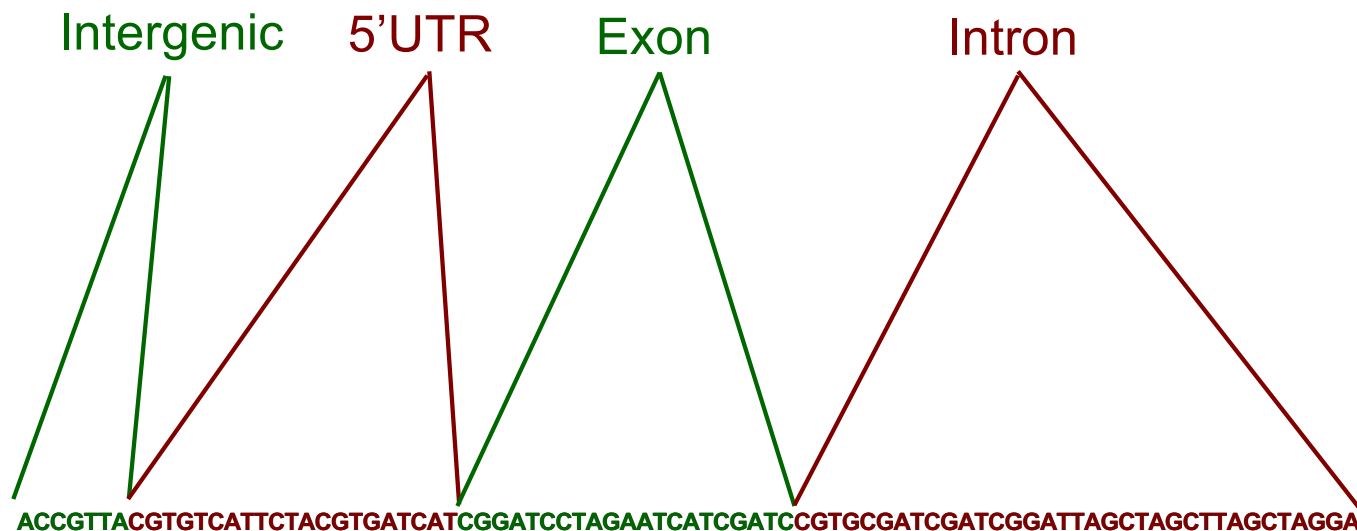
- Hidden states
 - {fair, biased} for coin tossing
 - {Exon, Intron, Intergenic} for Eukaryotic gene
- Emission symbols
 - {H, T} for coin tossing
 - {A, T, C, G} for DNA sequence
- Emission probability from state to symbol
 - $P(A \mid \text{exon}) = 0.85$, $P(A \mid \text{intron}) = 0.05$
- Transition probability among states
 - $P(\text{Exon} \mid \text{Intergenic})$

Parsing a DNA Sequence

- The HMM Viterbi path represents a parse of a given sequence, predicts exons, acceptor sites, introns, etc.

Hidden
state

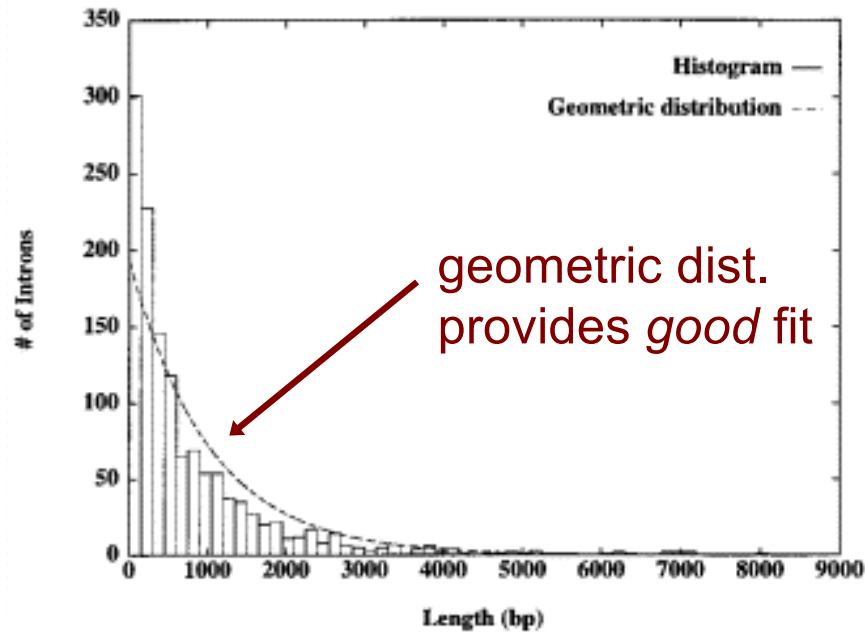
Observed
sequence



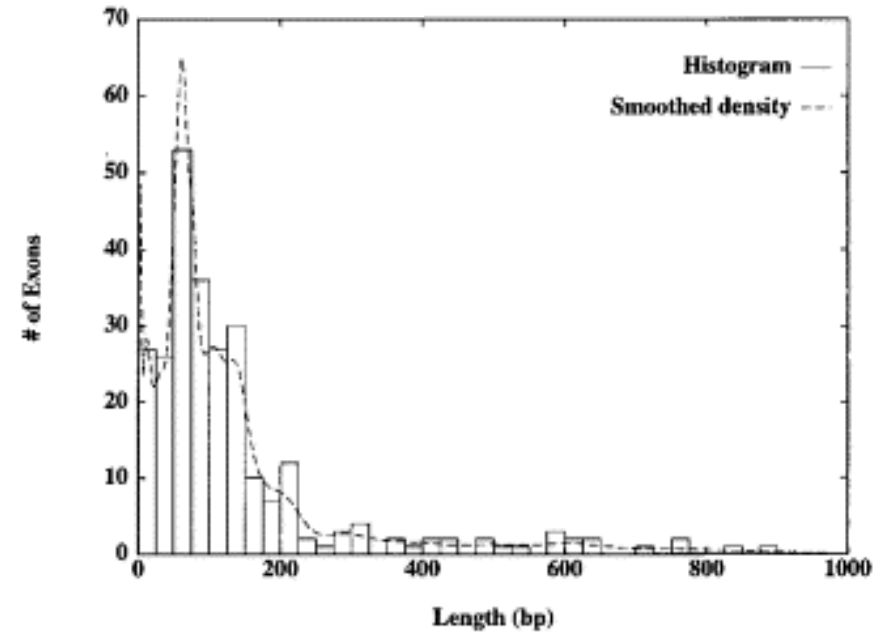
- How can we properly model the transitions from one state to another and the emissions of sequences?

Length Distributions of Introns/Exons

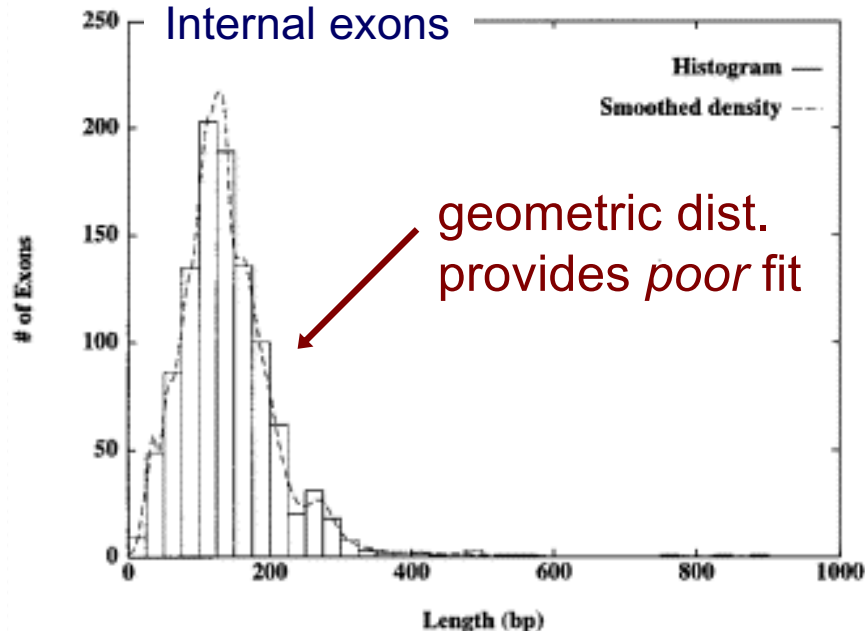
Introns



Initial exons



Internal exons



Terminal exons

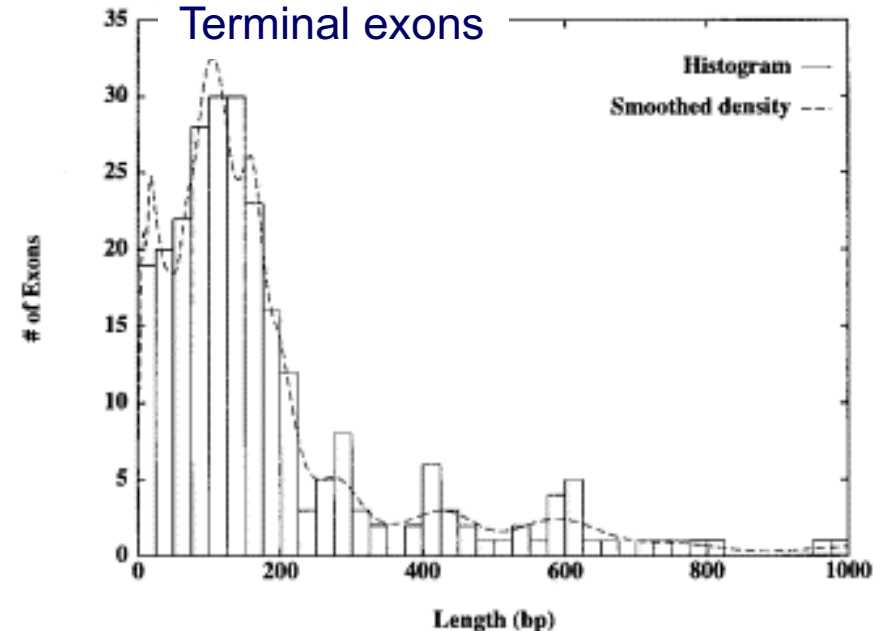


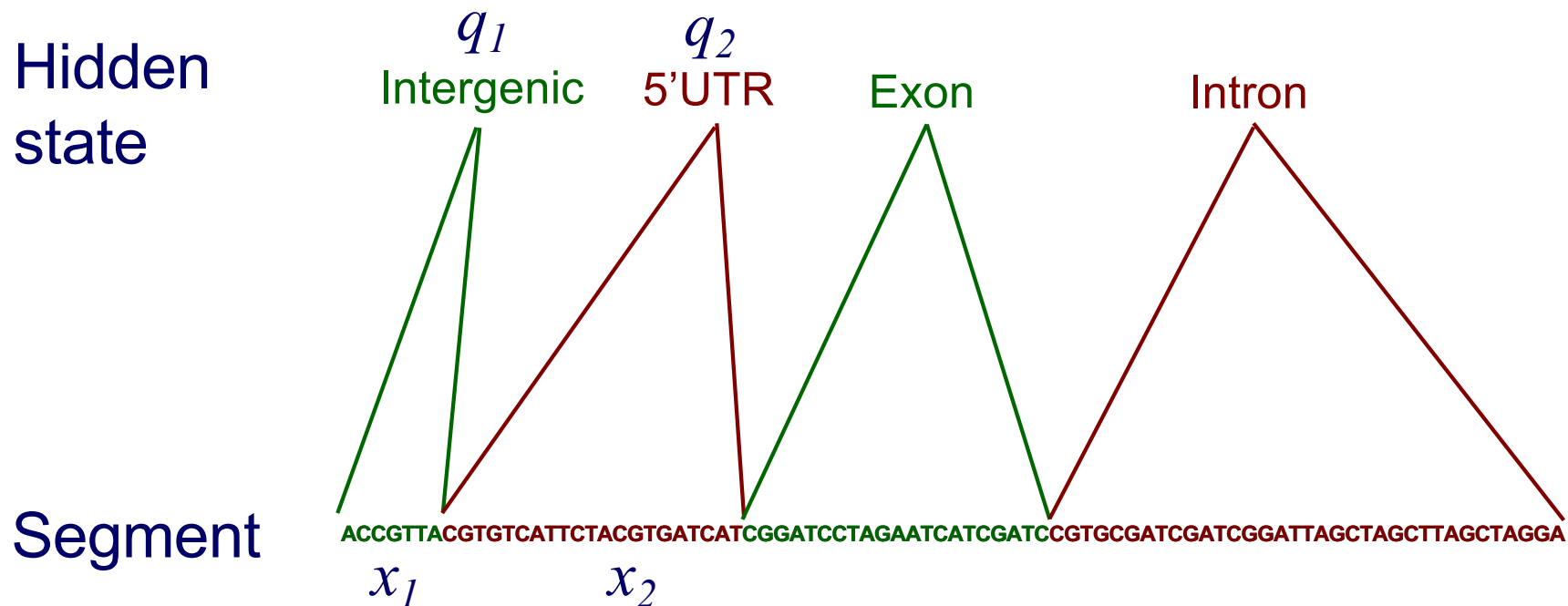
Figure from Burge & Karlin, *Journal of Molecular Biology*, 1997

Duration Modeling in HMMs

- Semi-Markov models are well-motivated for some sequence elements (e.g. exons)
 - **Semi-Markov**: explicitly model length duration of hidden states
 - Also called generalized hidden Markov model (GHMM)
- HMM emits single bases
- Semi-Markov or GHMM emits sequences
 - Duration is sequent length

GHMM models DNA Sequences

- Given a parse π with the hidden states $\{q_1, q_2, \dots\}$ and sequence segments $\{x_1, x_2, \dots, x_n\}$ with lengths $\{d_1, d_2, \dots, d_n\}$ for a sequence X



- Joint probability $P(\pi, X) =$

$$a_{0,1} f_{q_1}(d_1) P(x_1 | q_1, d_1) \prod_{k=2}^n a_{k-1,k} f_{q_k}(d_k) P(x_k | d_k, q_k)$$

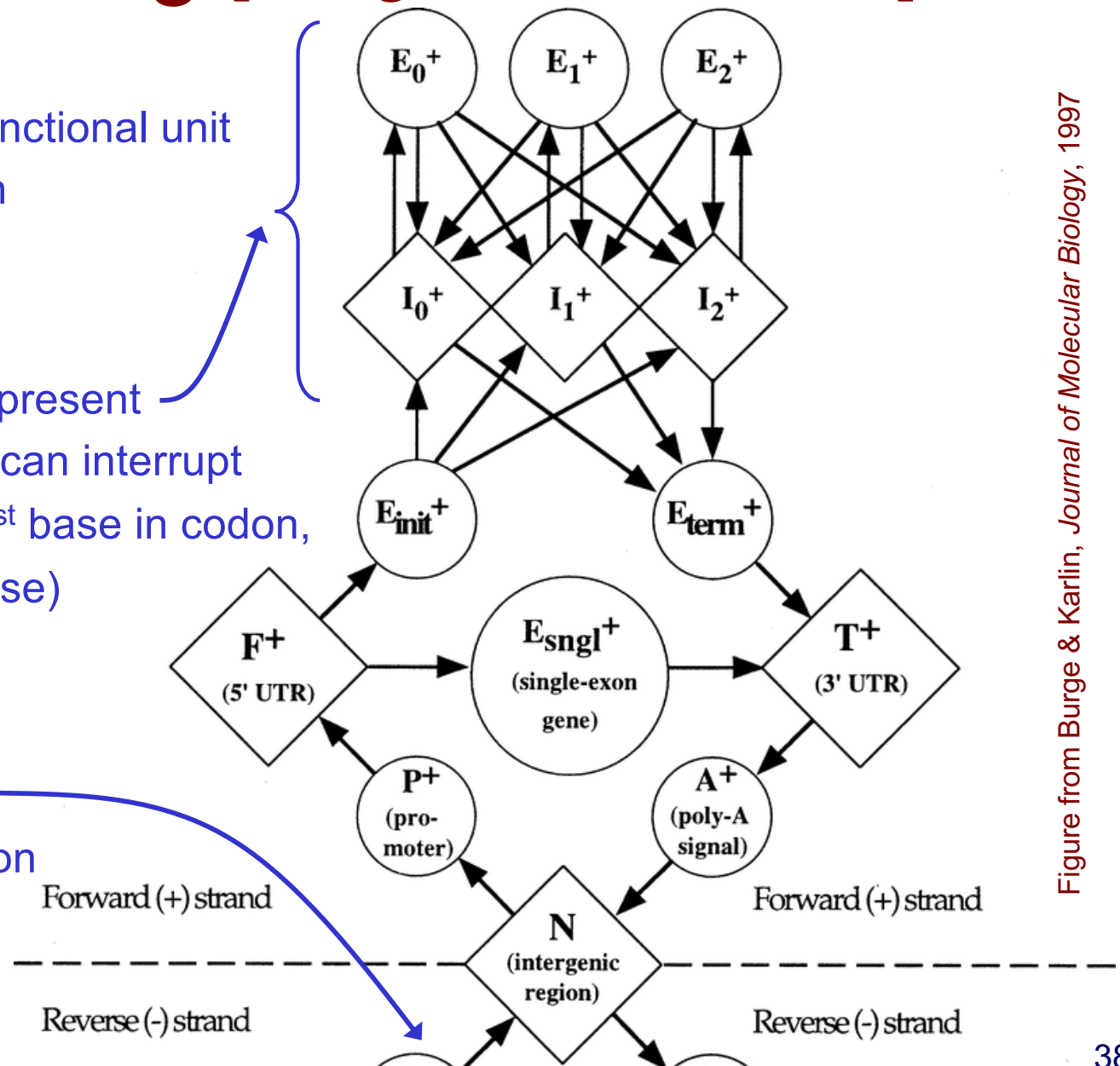
Length probability from previous distributions

The GENSCAN HMM for Eukaryotic Gene Finding [Burge & Karlin '97]

Each shape represents a functional unit of a gene or genomic region

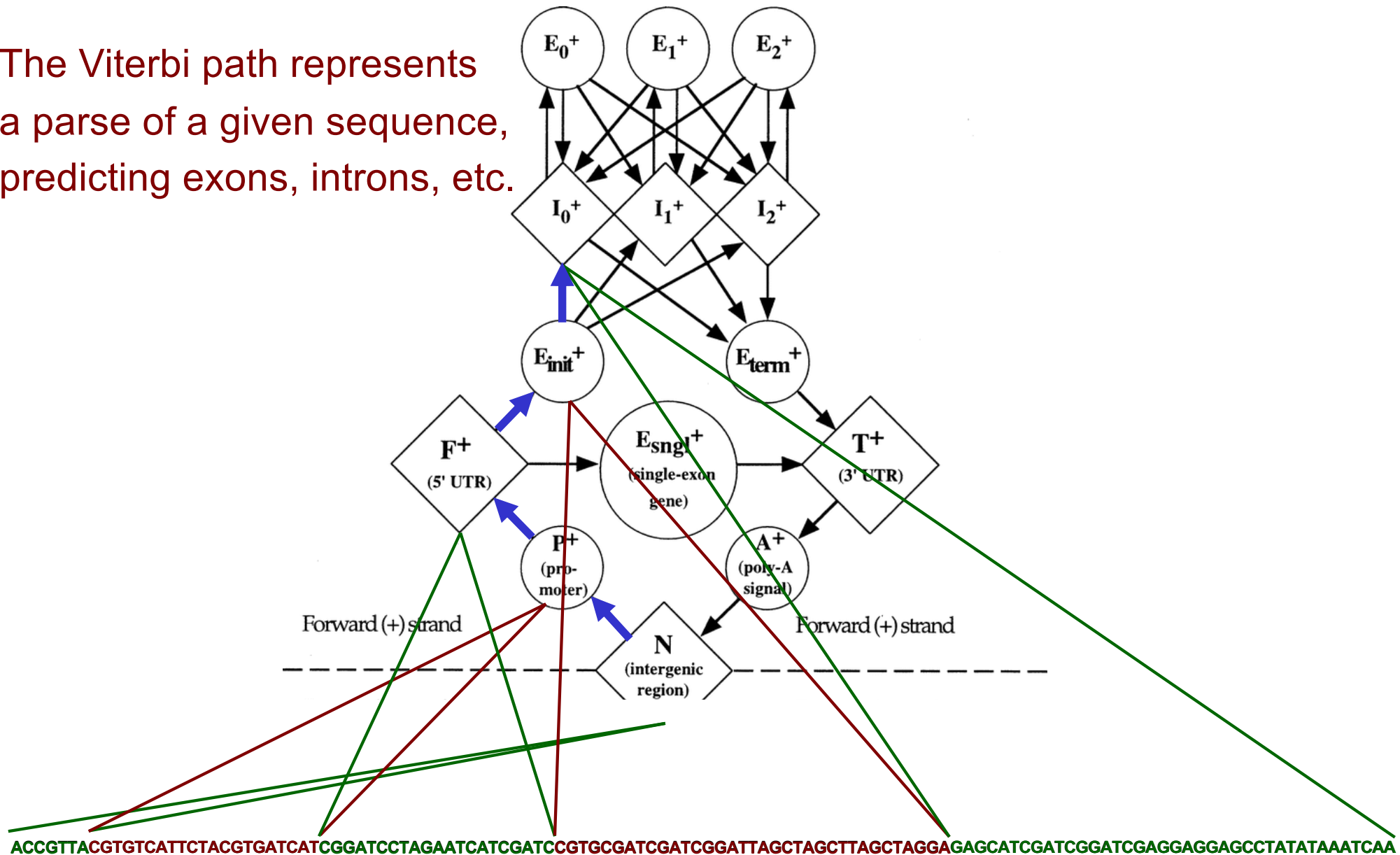
Pairs of intron/exon units represent the different ways an intron can interrupt a coding sequence (after 1st base in codon, after 2nd base or after 3rd base)

Complementary submodel (not shown) detects genes on opposite DNA strand



Parsing a DNA Sequence

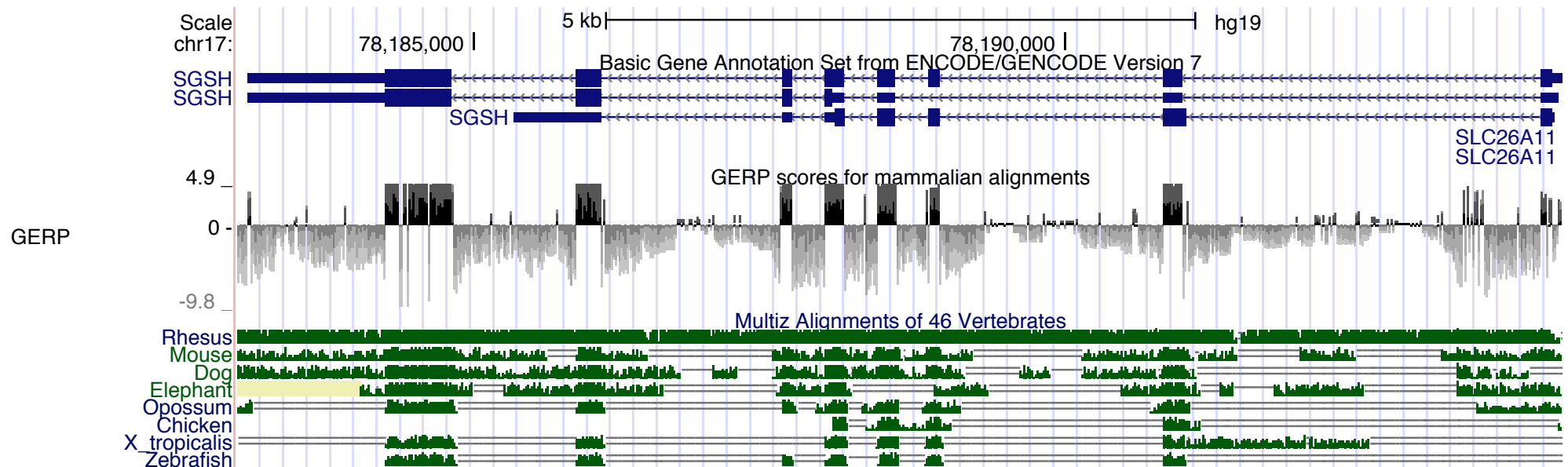
The Viterbi path represents a parse of a given sequence, predicting exons, introns, etc.



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



TWINSKAN

Korf et al., *Bioinformatics* 2001

- prediction with TWINSKAN
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 TWINSKAN
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 TWINSKAN

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

ATTAGCCTACTGAAATGGACCGCTTCAGCATGGTATCC

| | : | | | : | : | | | | | | | | : | | : | | | : : | |

↑ ↑ ↑

matched unaligned mismatched

Given: a sequence of length n , a set of aligned BLAST matches

$c[1 \dots n]$ = unaligned

sort BLAST matches by alignment score

for each BLAST match h (from best to worst)

for each position i covered by h

```
if  $c[i] == \text{unaligned}$ 
```

$$c[i] = h[i]$$

Conservation Sequence Example

given
sequence

ATTTAGCCTACTGAAATGGACCGCTTCAGCATGGTATCC

significant
BLAST
matches
ordered from
best to worst



ATTTA

||: ||

ATCTA

ATGGACCGCTTCAGC

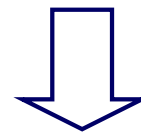
|:|:|||||||:|

ACGCACCGCTTCATC

AGCATGGTATCC

||:|:||||::||

AGAAGGGTCACC



resulting
conservation
sequence

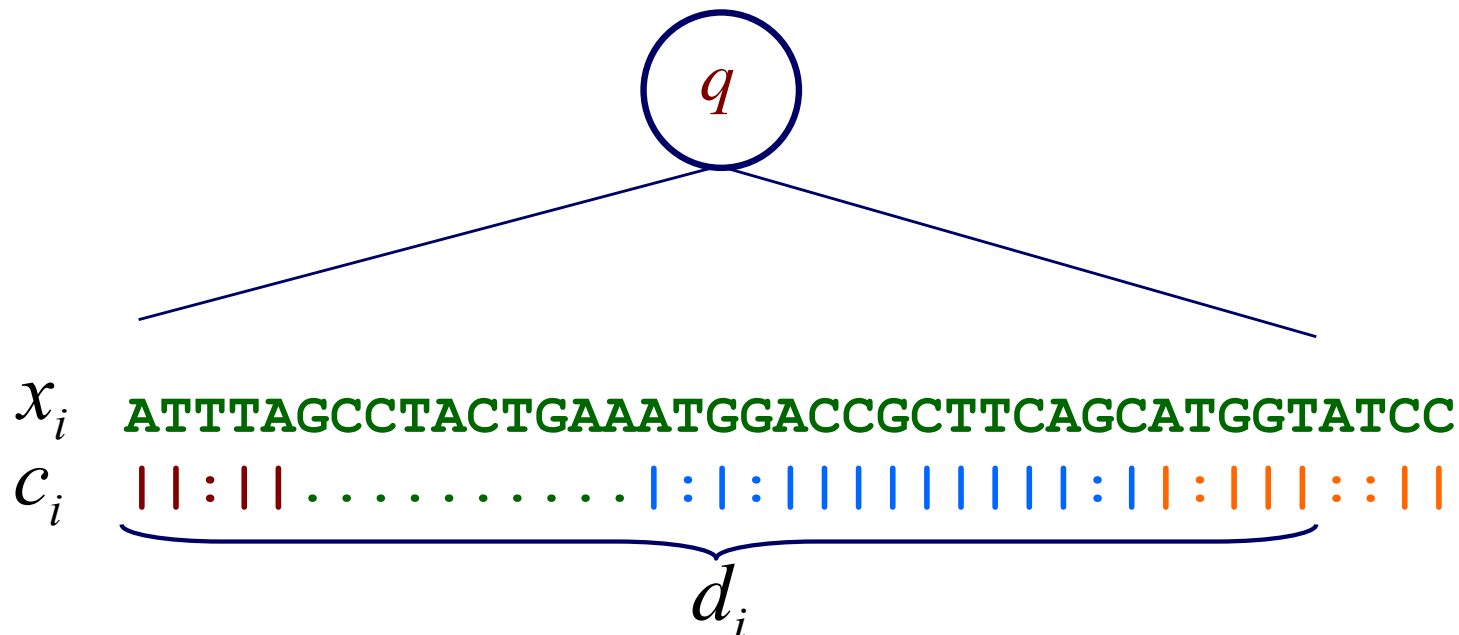
ATTTAGCCTACTGAAATGGACCGCTTCAGCATGGTATCC

||: || |:|:|||||||:| |:|:|:|:|

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

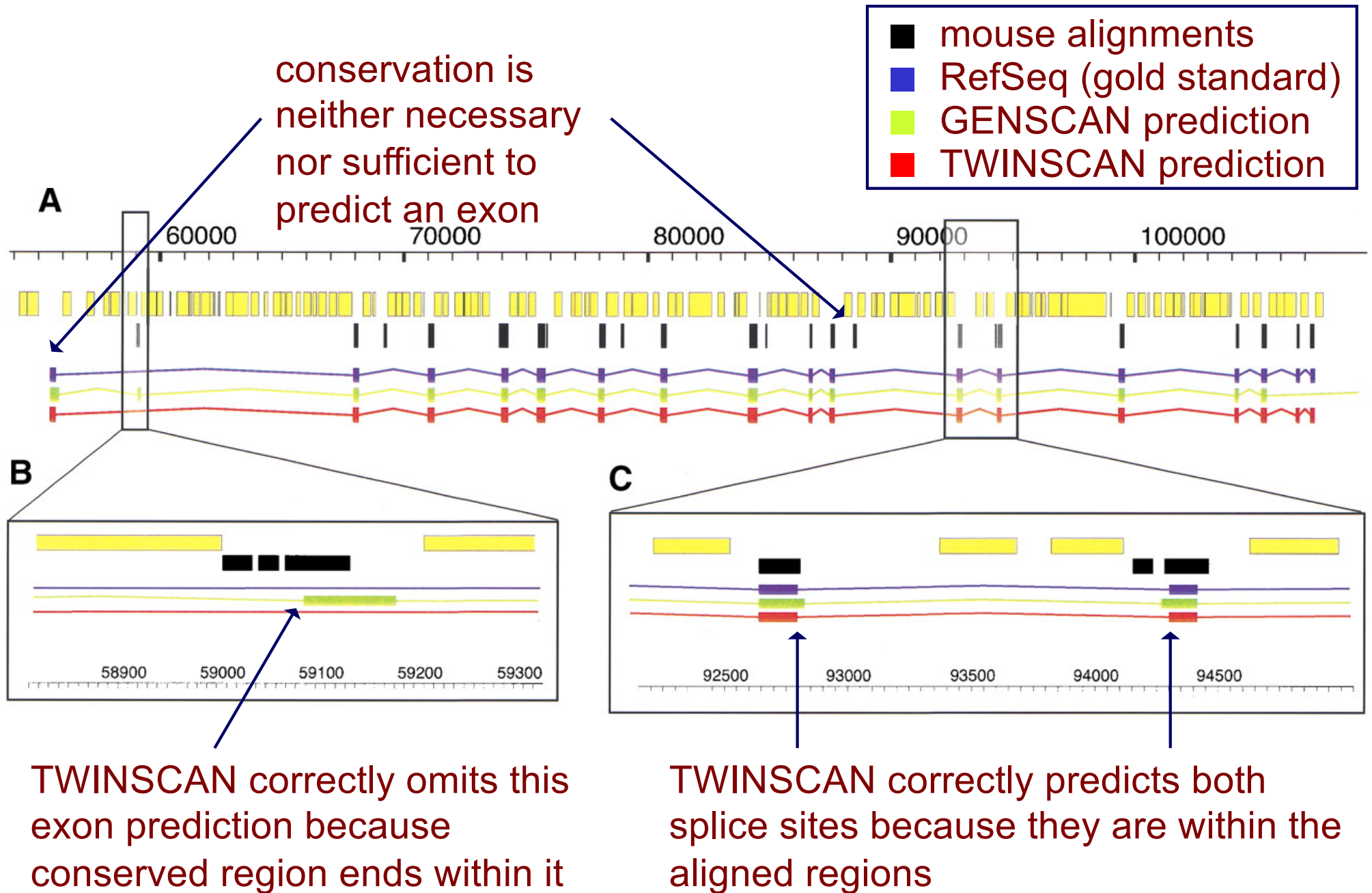
$$\Pr(x_i, c_i \mid q) = \Pr(d_i \mid q) \Pr(x_i \mid q, d_i) \Pr(c_i \mid q, d_i)$$



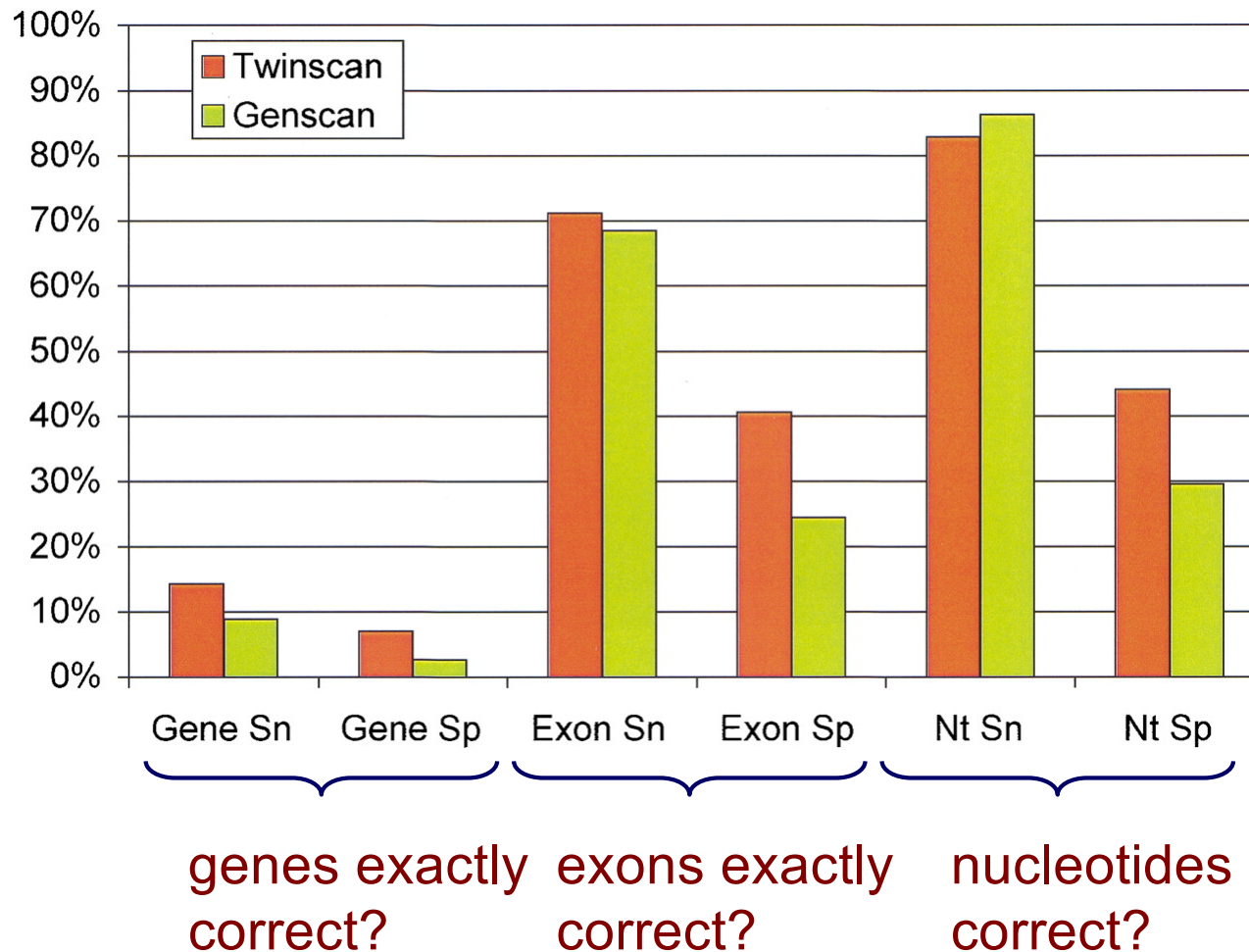
Modeling Sequences in TWINSCAN

- 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



GENSCAN vs. TWINSCAN: Empirical Comparison

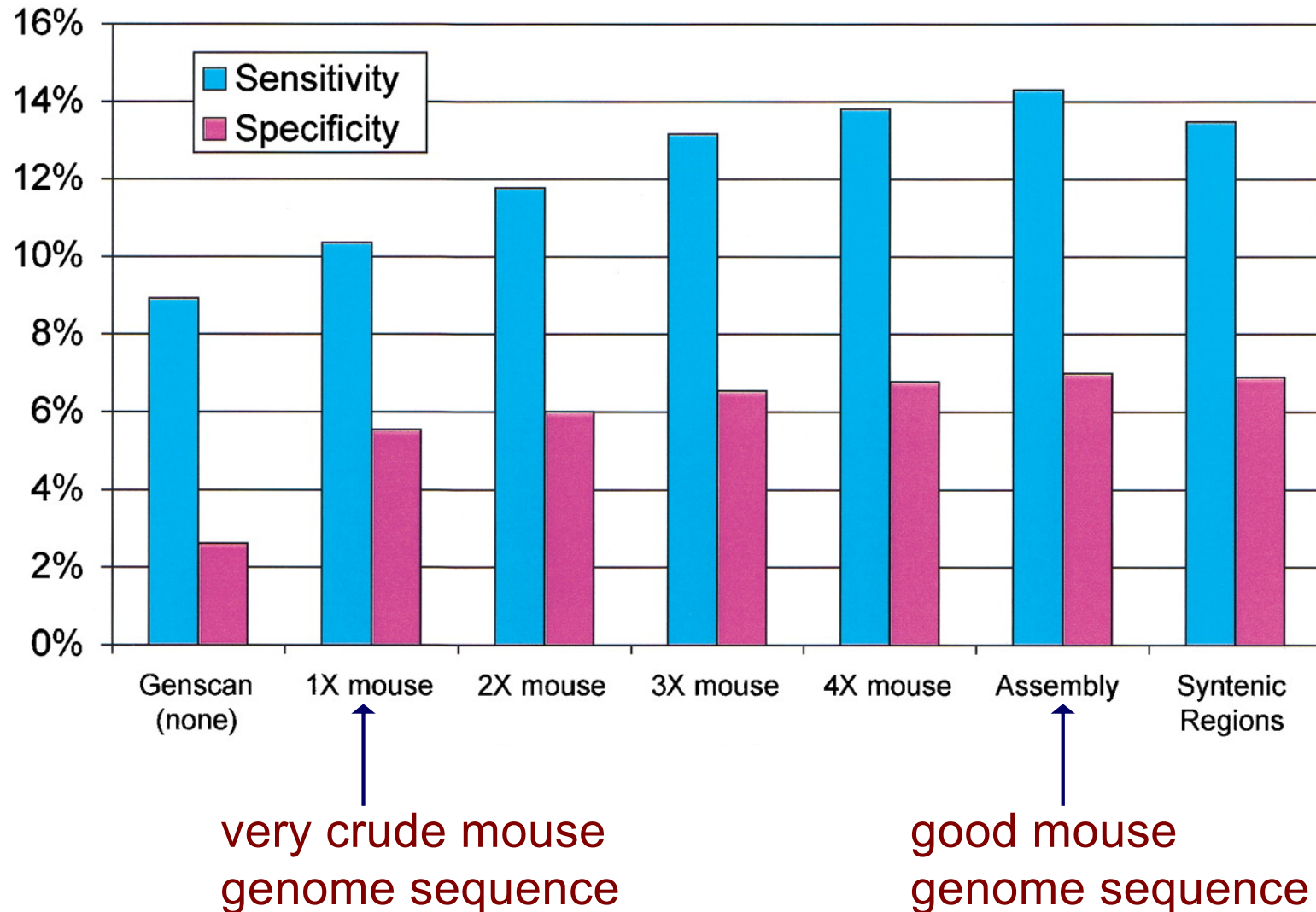


$$\text{sensitivity (Sn)} = \frac{TP}{TP + FN}$$

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

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

Accuracy of TWINSCAN as a Function of Sequence Coverage



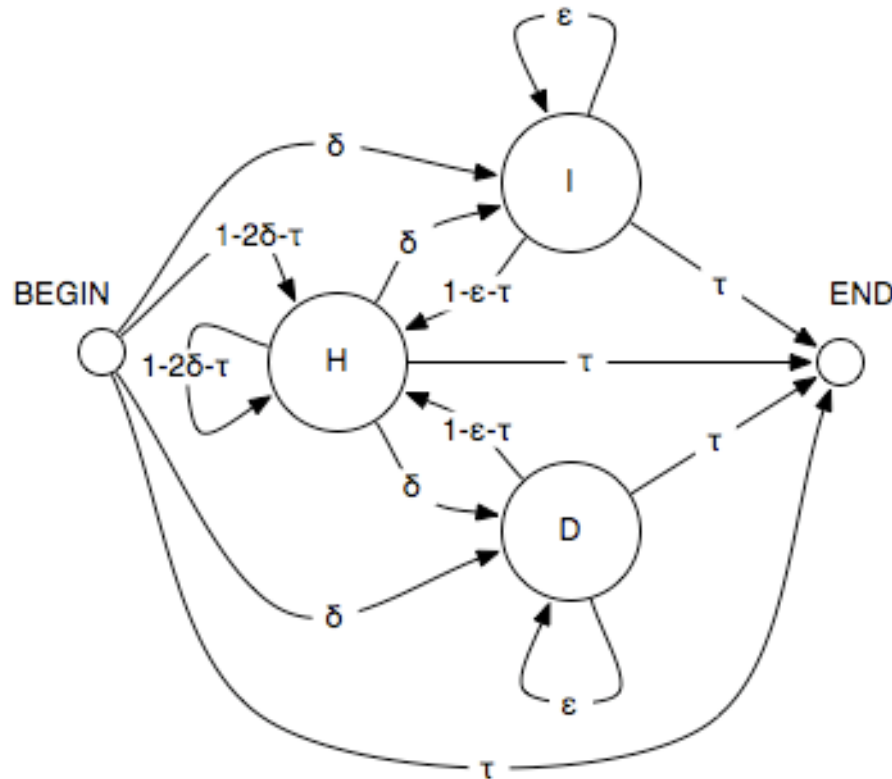
SLAM

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

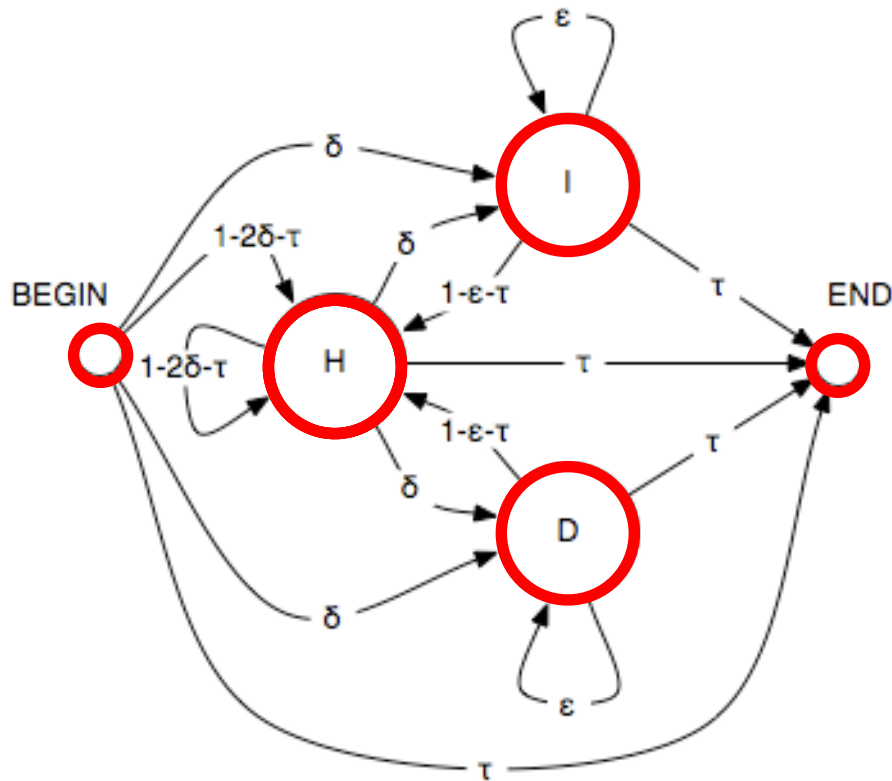


H: homology (match) state

I: insert state

D: delete state

PHMM Paths = Alignments



sequence 1: AAGCGC

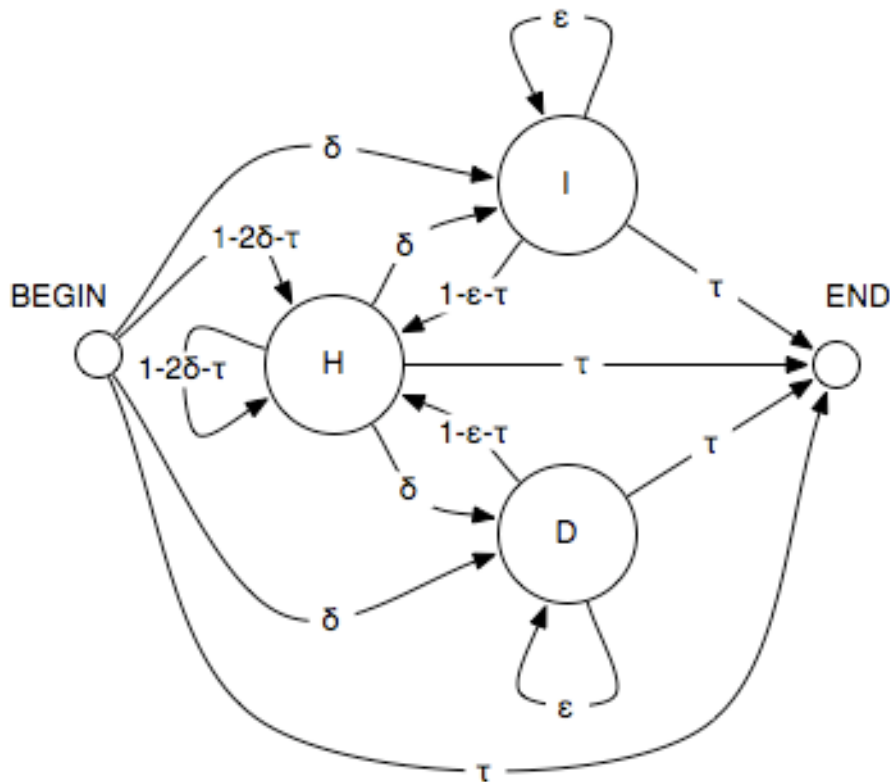
sequence 2: ATGTC

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				
		B	H	I	D	E
state i	B		$1-2\delta-\tau$	δ	δ	τ
	H		$1-2\delta-\tau$	δ	δ	τ
	I		$1-\epsilon-\tau$	ϵ		τ
	D		$1-\epsilon-\tau$		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:

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

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

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

- 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

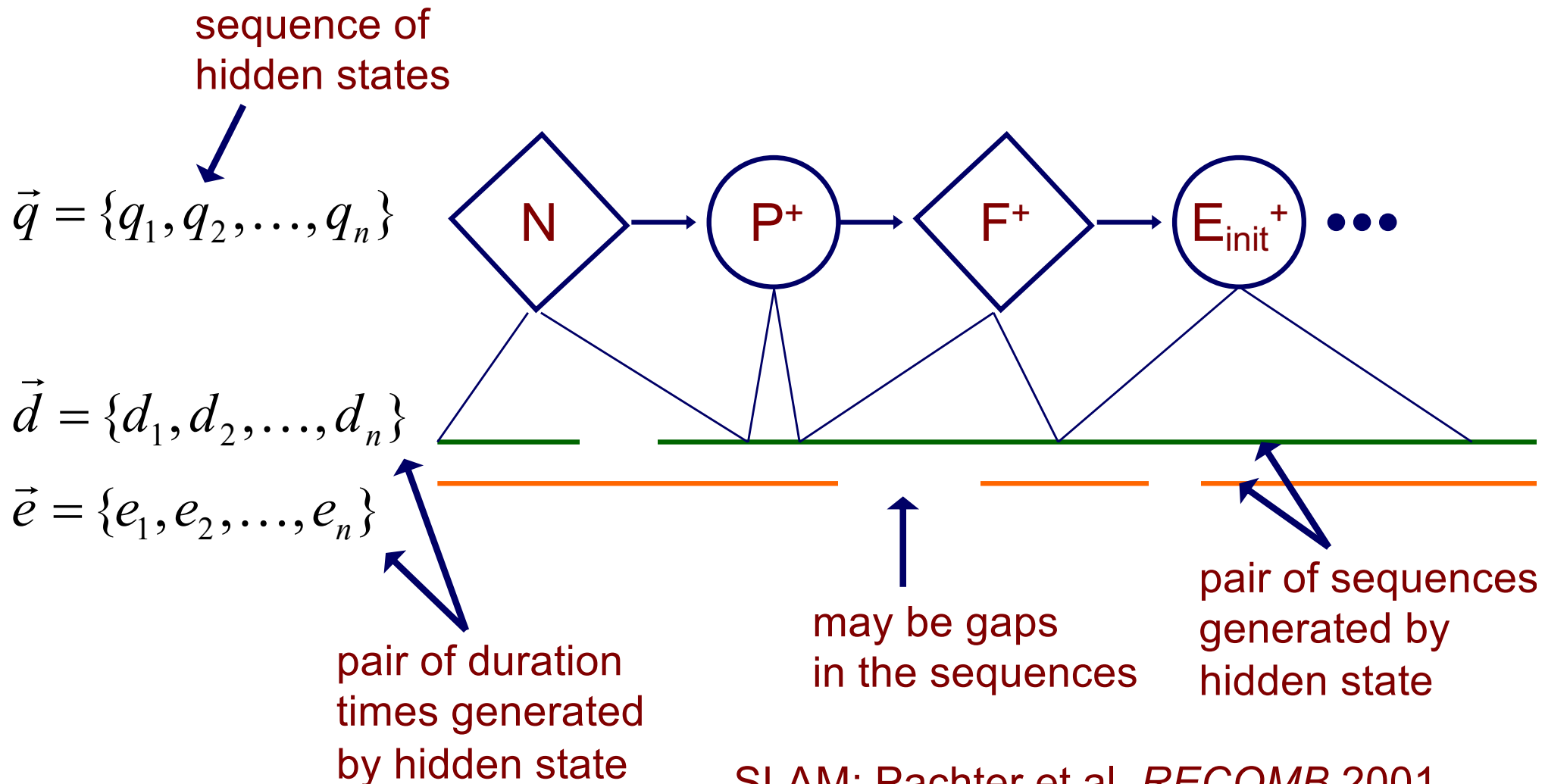
HIDHHDDIIHH...

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 each input sequence



Generalized Pair HMMs

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

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

$$\vec{d} = \{d_1, d_2, \dots, d_n\} \quad \vec{e} = \{e_1, e_2, \dots, e_n\}$$

- 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

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

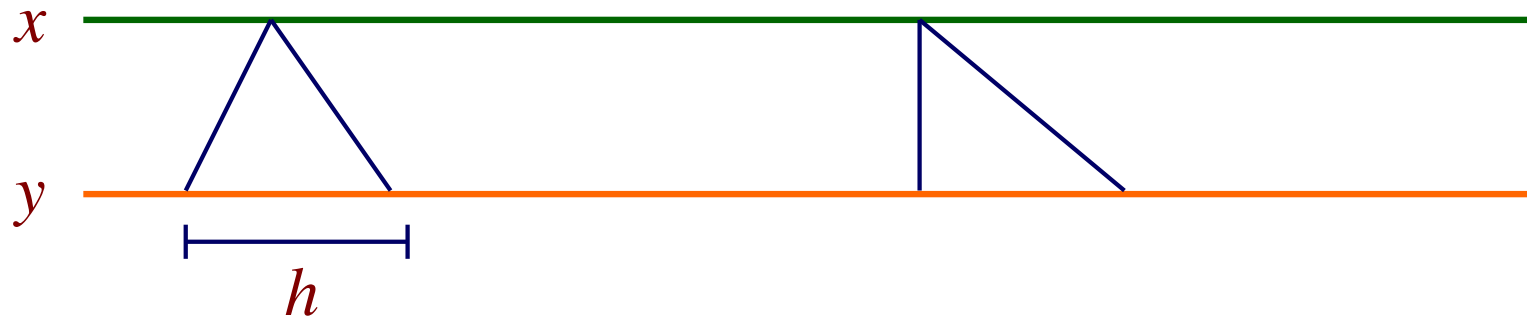
- Generalized Pair HMM Algorithm

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

- 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 y



- analogous to banded alignment methods

GENSCAN, TWINSCAN, & SLAM

Test set	Nucleotide level			Exon level				
	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 $SP = TN / (TN + FP)$ (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).

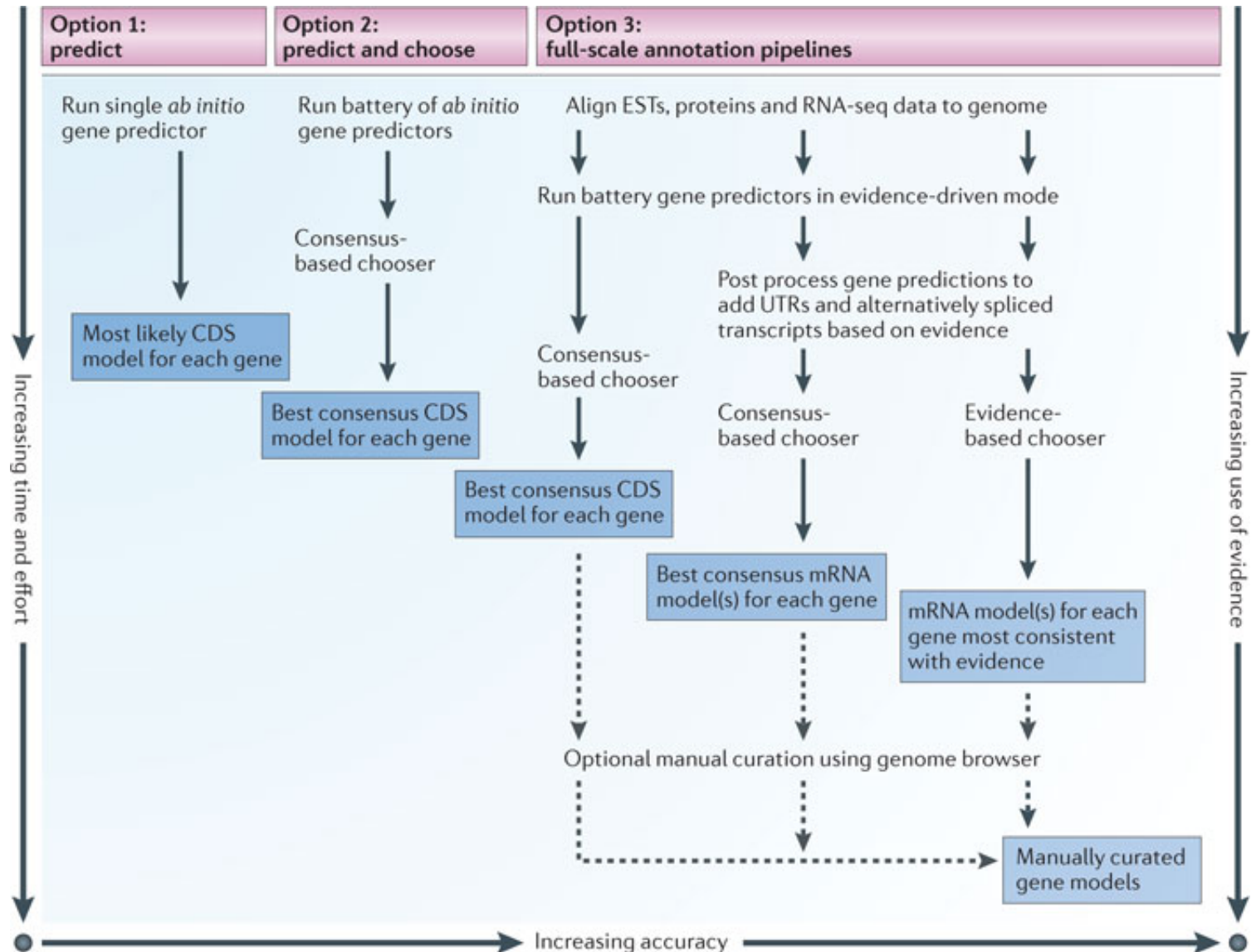
TWINSKAN vs. SLAM

- both use multiple genomes to predict genes
- both use generalized HMMs
- TWINSKAN
 - 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

Modern Genome Annotation

- RNA-Seq, mass spectrometry, and other technologies provide powerful information for genome annotation

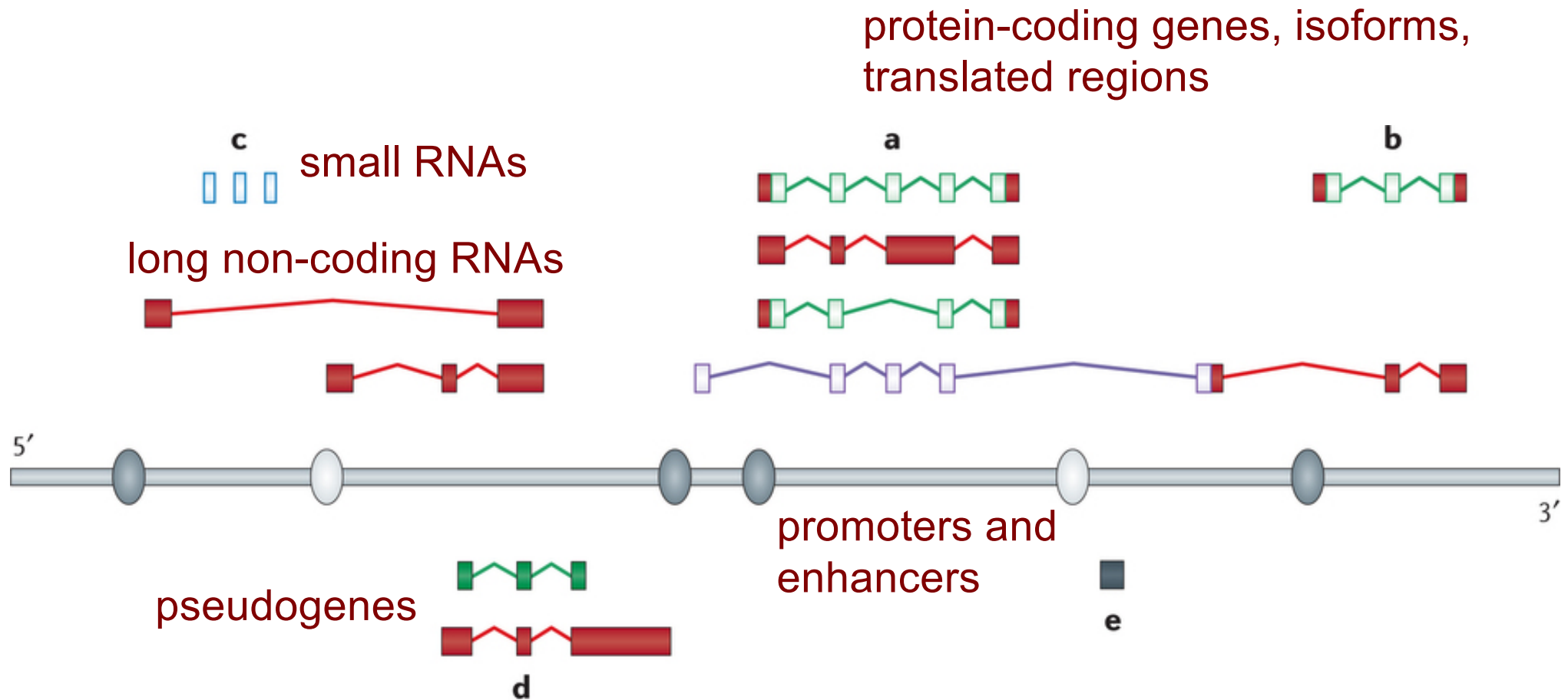
Modern Genome Annotation



Nature Reviews | Genetics

Yandell et al. *Nature Reviews Genetics* 2012

Modern Genome Annotation



Nature Reviews | **Genetics**