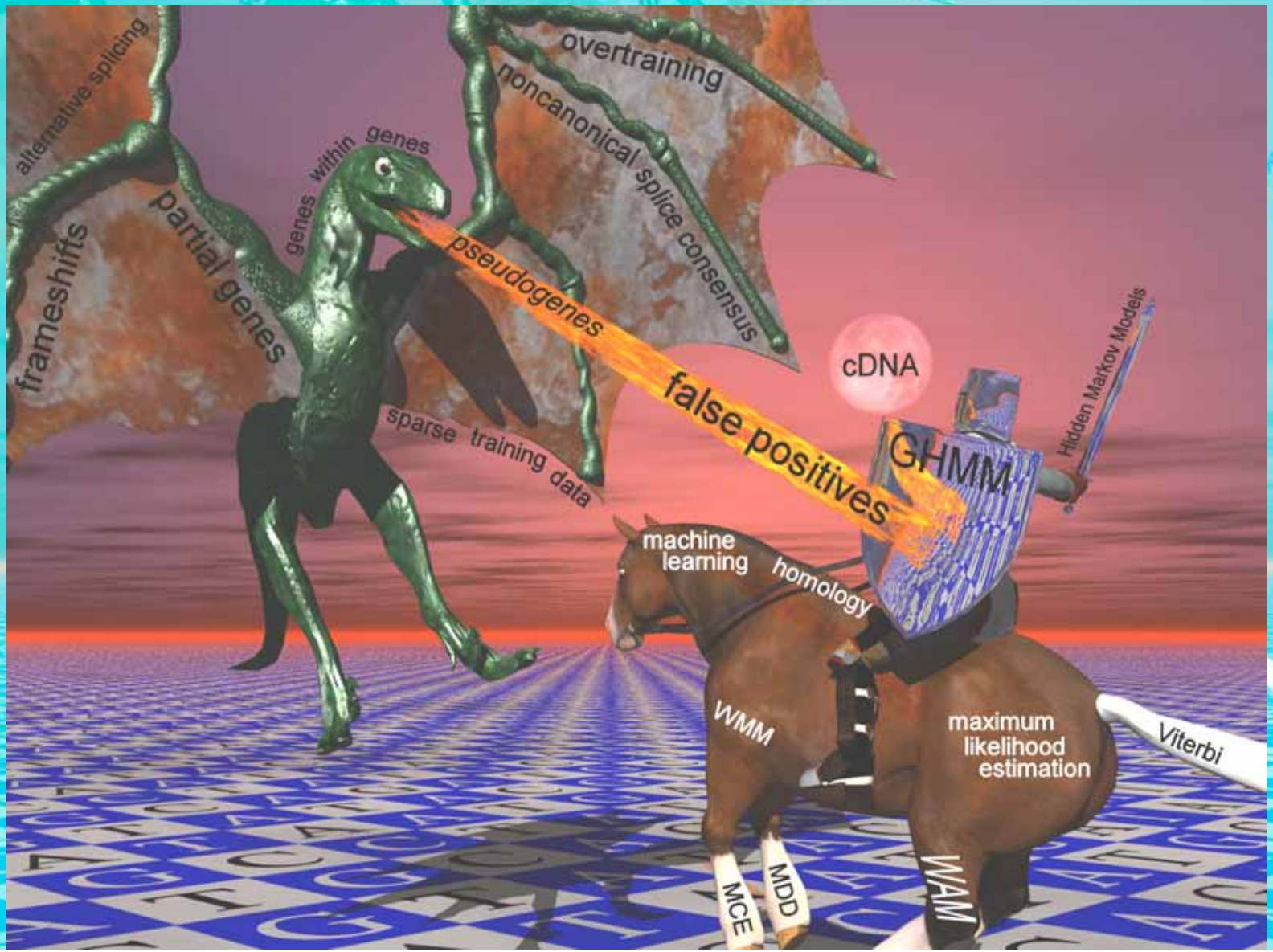
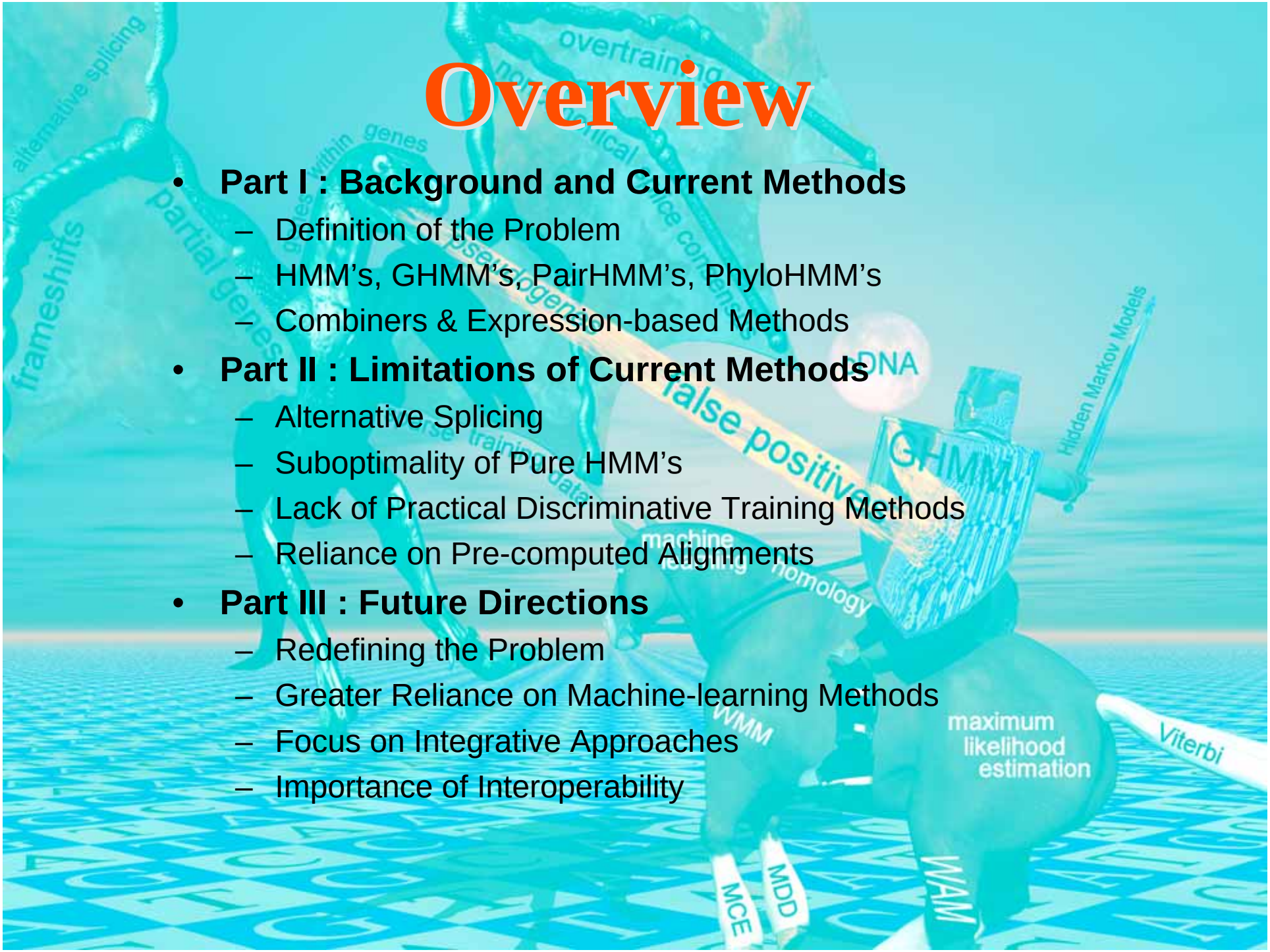
Advancing the State-of-the-art in **Computational Gene Prediction**

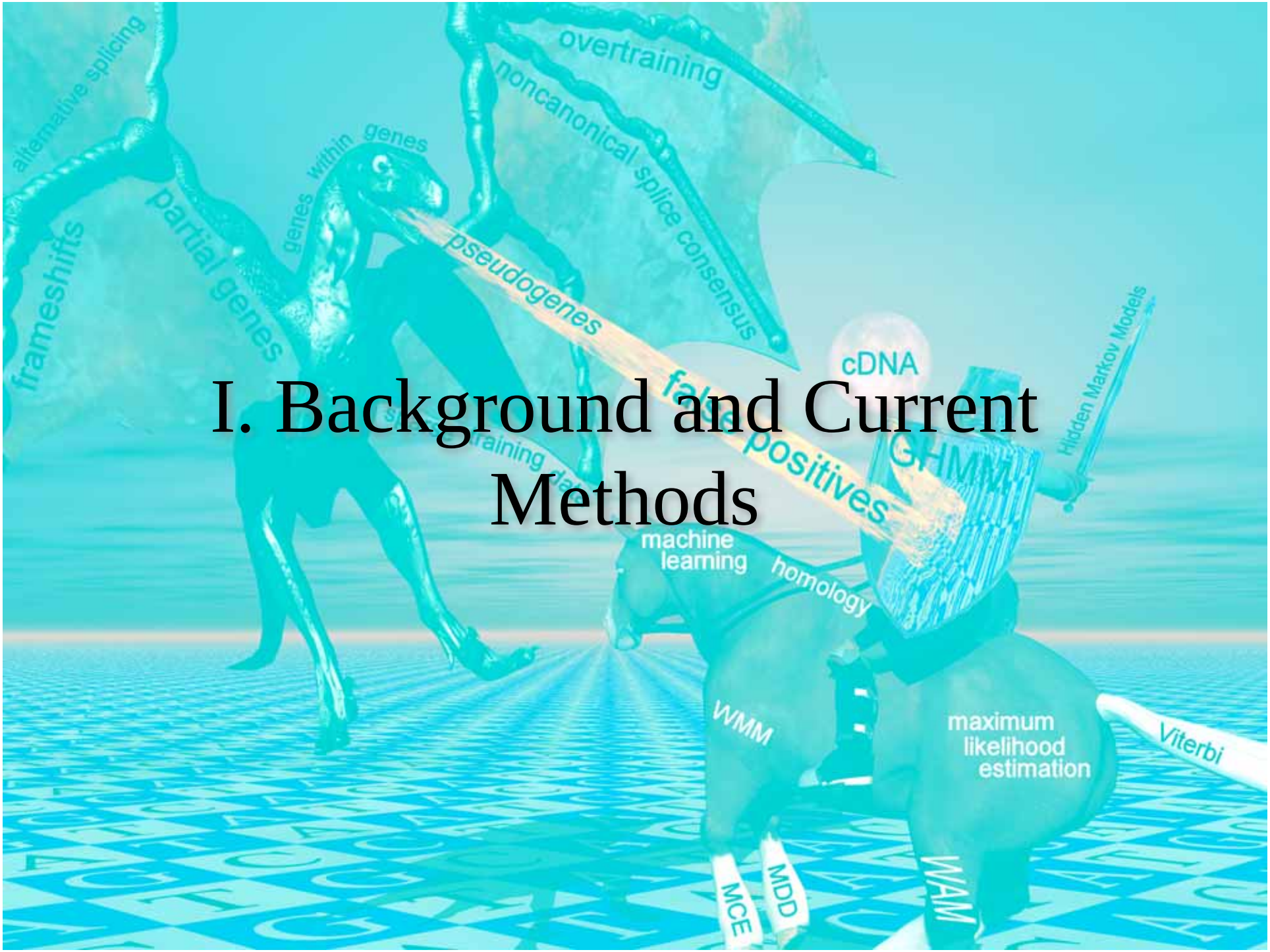
Bill Majoros (bmajoros@duke.edu)



Overview

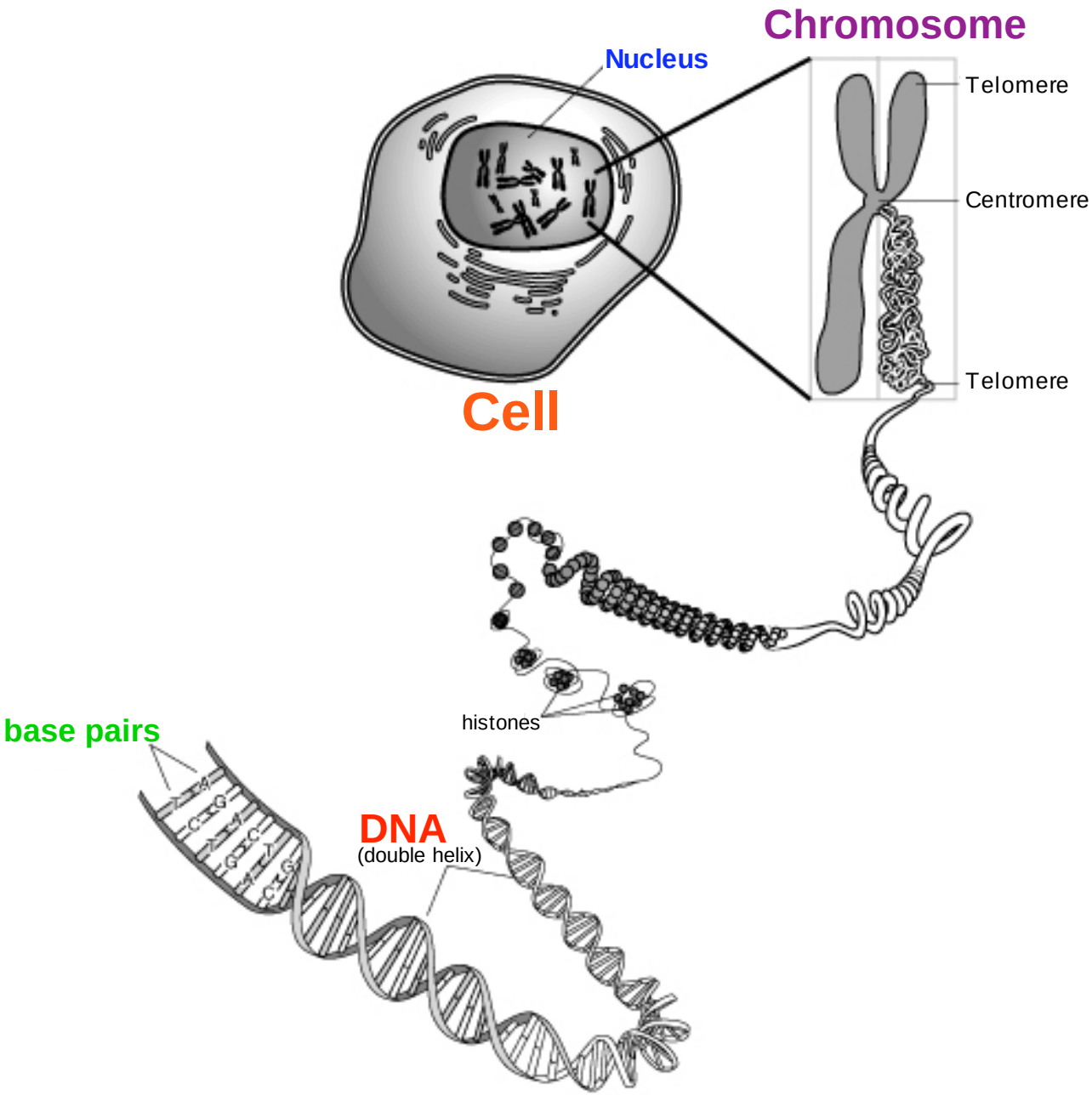
- **Part I : Background and Current Methods**
  - Definition of the Problem
  - HMM's, GHMM's, PairHMM's, PhyloHMM's
  - Combiners & Expression-based Methods
- **Part II : Limitations of Current Methods**
  - Alternative Splicing
  - Suboptimality of Pure HMM's
  - Lack of Practical Discriminative Training Methods
  - Reliance on Pre-computed Alignments
- **Part III : Future Directions**
  - Redefining the Problem
  - Greater Reliance on Machine-learning Methods
  - Focus on Integrative Approaches
  - Importance of Interoperability



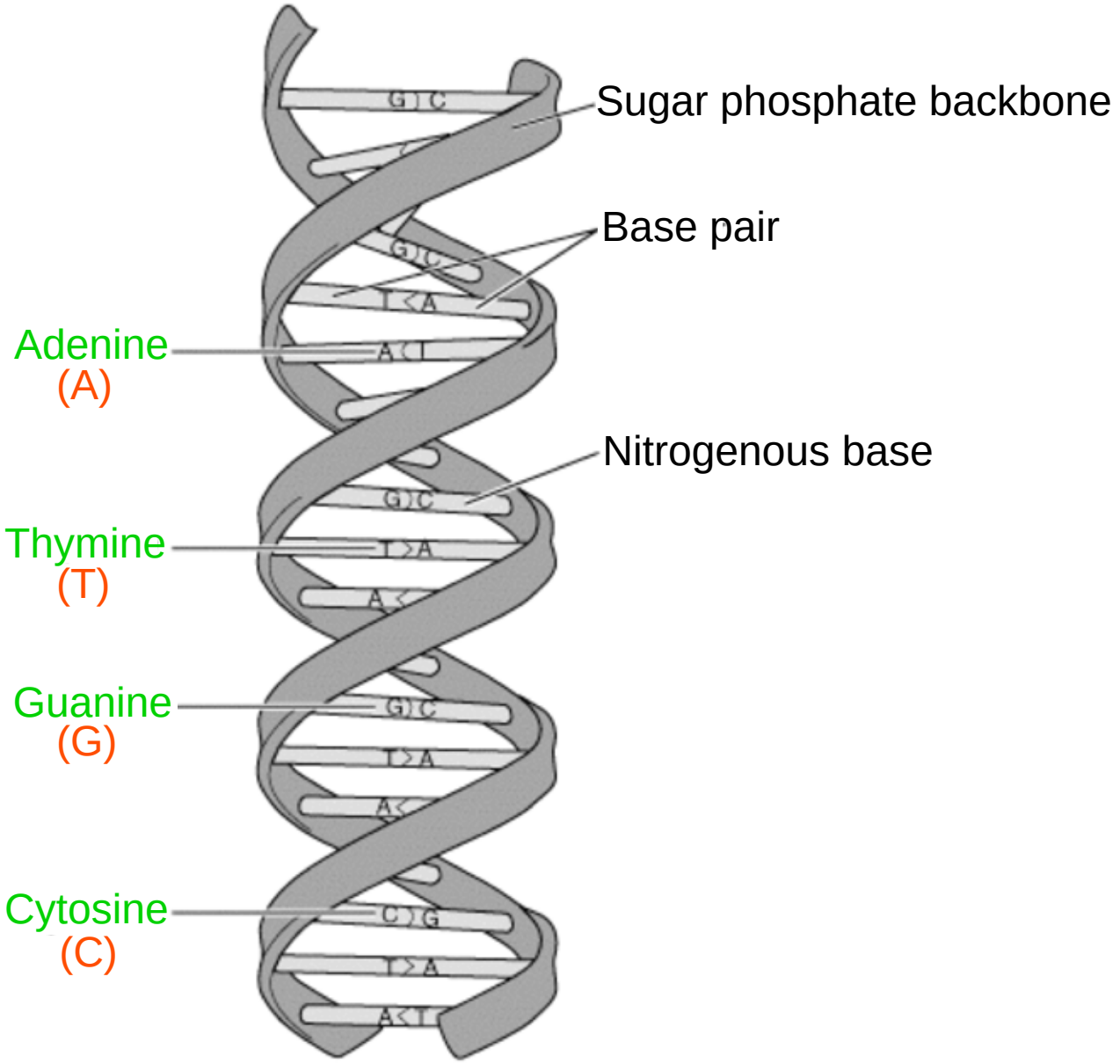


I. Background and Current Methods

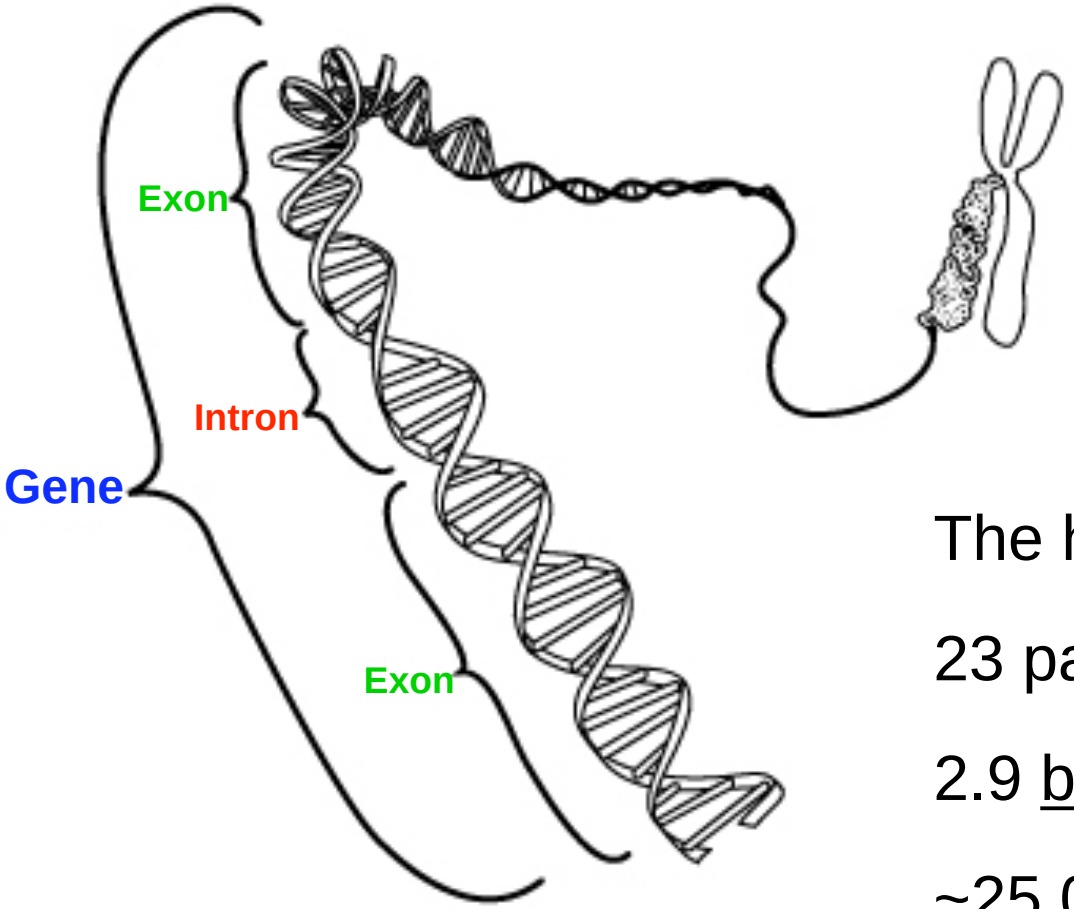
Cells, Chromosomes, and DNA



Nucleotide Composition of DNA



Exons, Introns, and Genes



The human genome:

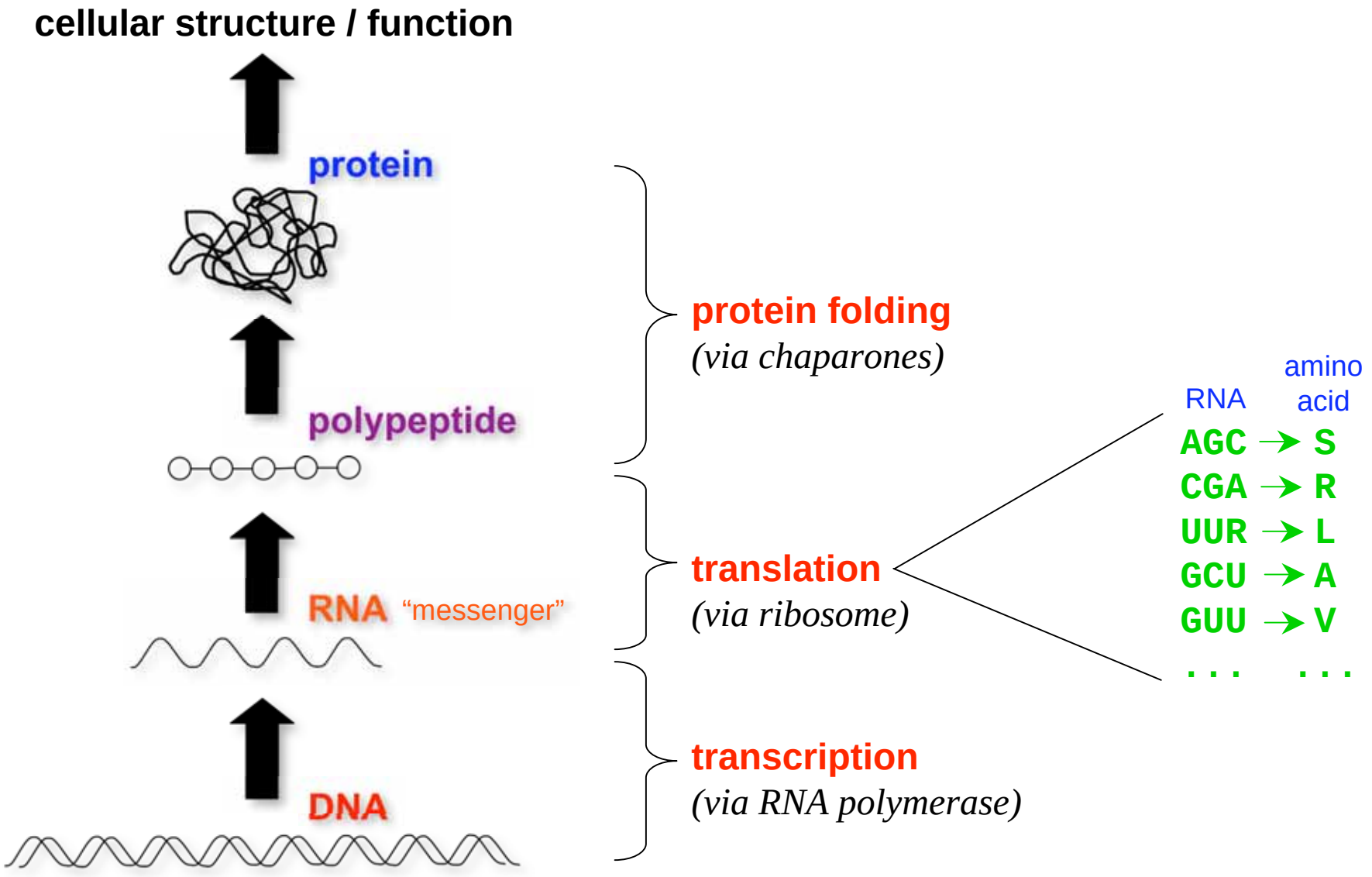
23 pairs of chromosomes

2.9 billion A's, T's, C's, G's

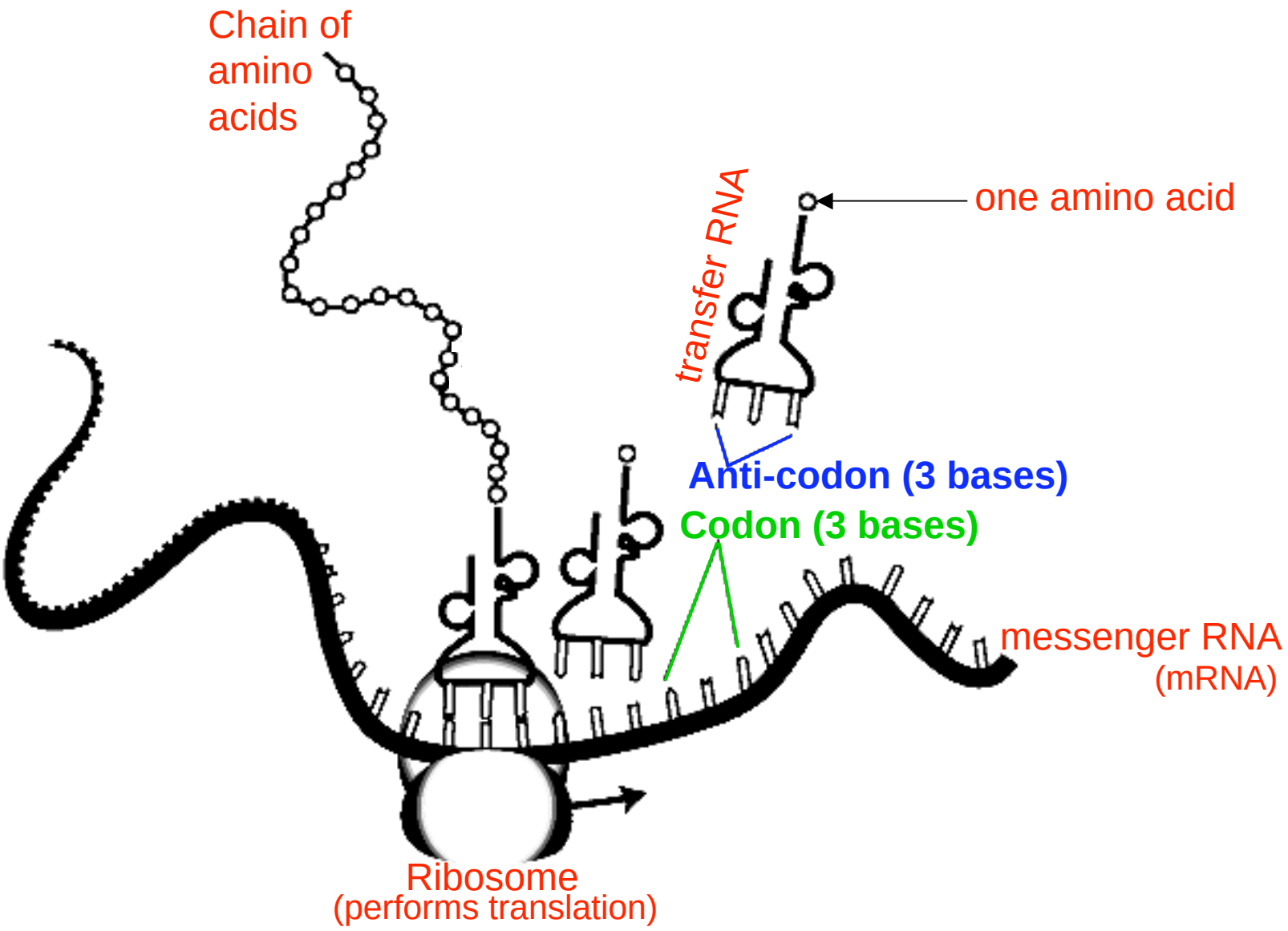
~25,000 genes (?)

~1.4% of genome is coding

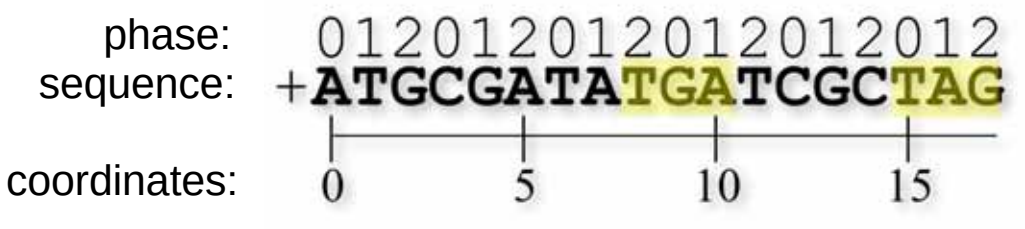
The Central Dogma of Molecular Biology



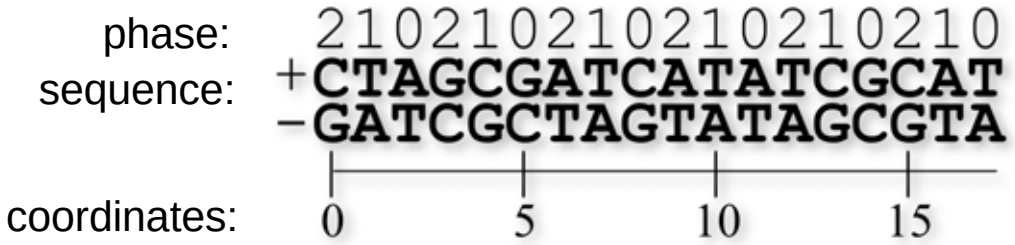
The mod-3 Nature of the Genetic Code



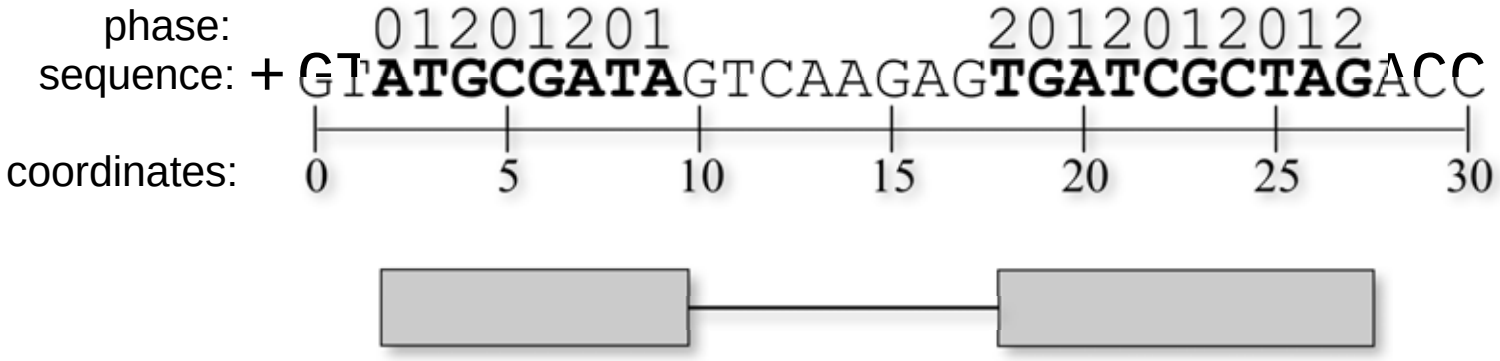
Phase Constraints



forward strand

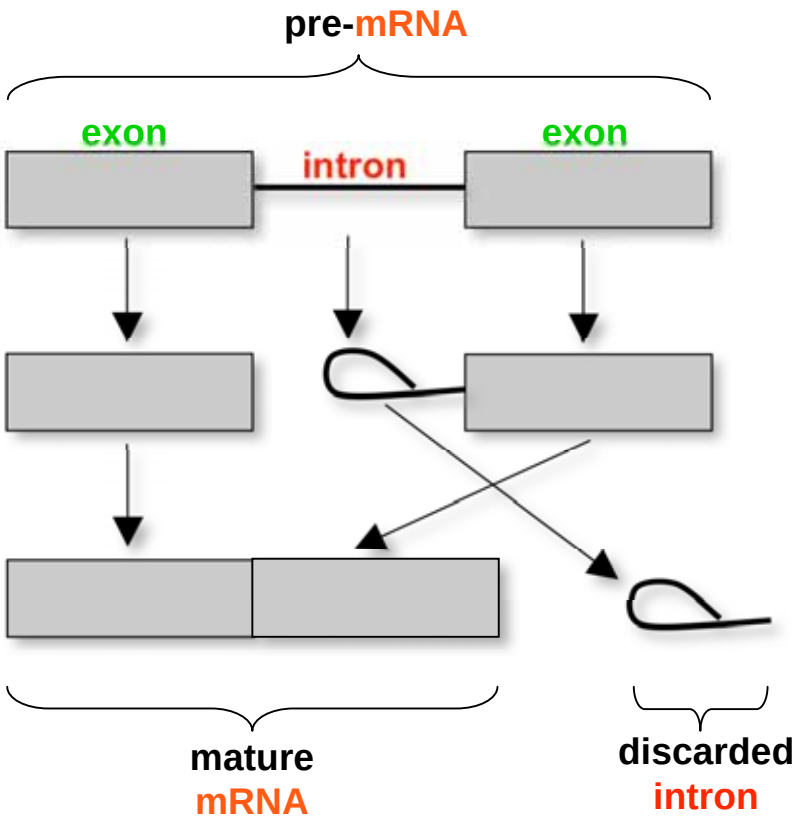


reverse strand

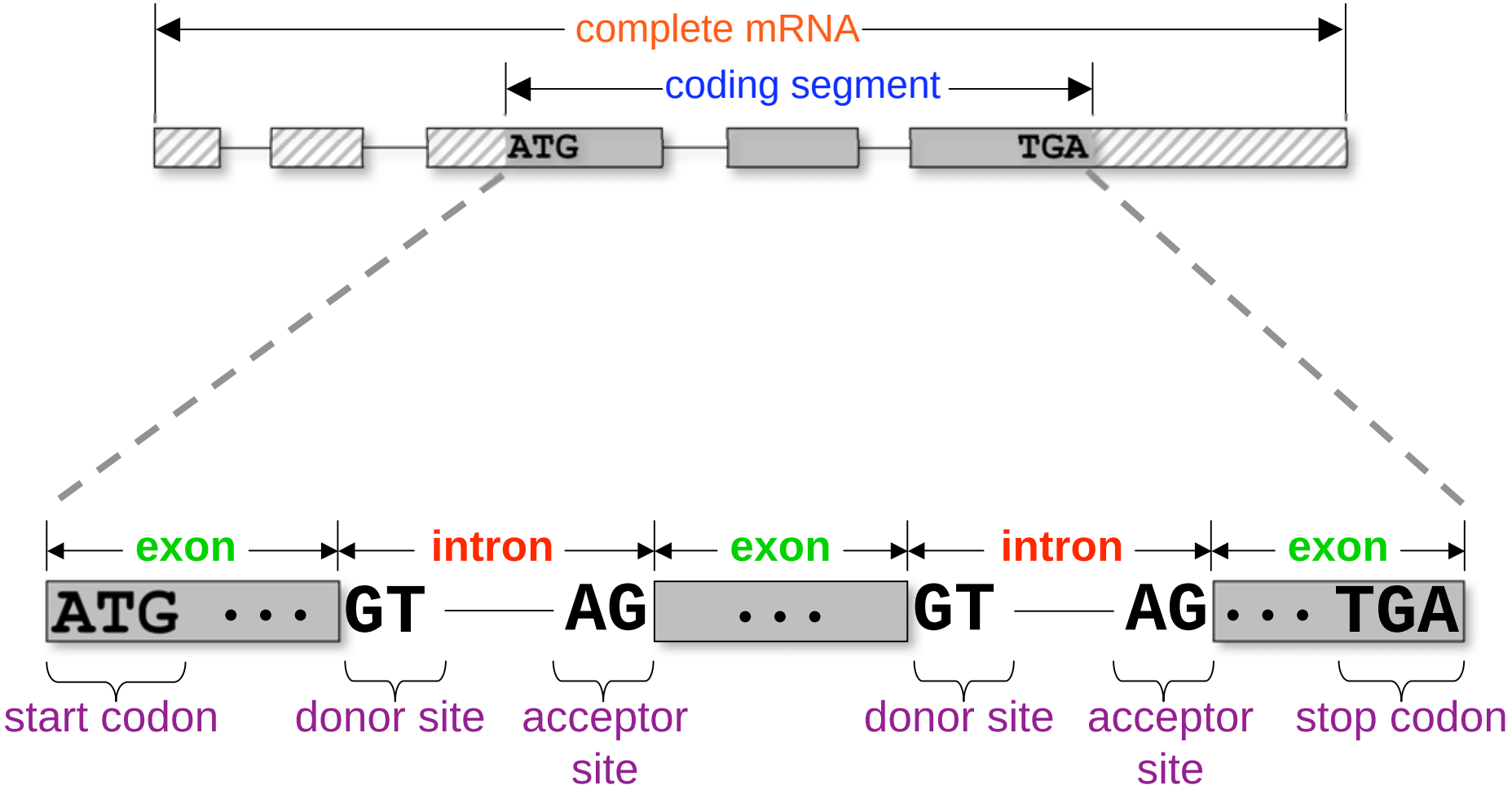


forward strand, spliced

Splicing of Eukaryotic Messenger RNAs



The Eukaryotic Gene Finding Problem



>4271863
GAATTCTCCAACCCCAGGTGAGGATCTGACTACCTGGAACAGAACCCCGCTCTTCCAGGTGAGAATATGACAGAATAAAGCAG
CACCCTGCACCCCCAGTTGAGCATCTGACAGCCTGGGGCAGCACCCACACTCCCAGGTGAGCATCTGACAGCCTGGAGCAGCA
CCCACACCCCCAGGTGTGCATCTGACAGCCTGAAACAGCACCCTCCACCACCAGATGAGCATCTGACAACCAGAACCTGCACC
ACACACCCCAAGGTGGGCATCCGATGGCATGGAACAGCACCCCCACTCACAGGTGATGTGACTGCGTGGAACAGCACATCCCC
TCAGGTGAGCATCTGACAGCATAAAACAGCACCCCACAACCCCAGGTGATCATTTGCCAGCCAGGAACGGCAACCCACATCCC
CAGGTAAGTGTCTGACAGCCTAGAGCGGCACCTGCACACTTAGGTAAGAATCTGAAAGCCTGGATCAACACTCGAACCCCCAG
GTGAGCATCTGACAGCCTGGAGCAGCACCCCACACCTCCAGGGGAGCATCTGACATCCTGGAACAGCACCCCACACCCCCAGT
GAGCATCTGACAACCTGGAGCAGCACCCCACACCTCCAGGGGAGCATCTGACATCCTGGAACAGCACCCCACACCTCCAGGGG
GAGCATCTGACAGCCTGGAGCAACACCCCACACCTCCAGGGGAGCATCTGACAGCCTGGAACAGCACCCCACACCTCCAGGGG
AGCATCTGACAGCATGGACAAGTCCTGCCCCCCGGTTAGTGTCTGAATTCCTGGAATATGTGCTGTCCTTTTCCACCAGGTGA
GCATATGACCGCCTGGAAGAAGCACCCCTGCATGTTACCTGTGGTGAAACCAAGGCTGAGAGACAGGACAGGGTTGTTGGCCA
GGAGGAGGGGCCTGCTGCTGAGCCCCAGCGCTGAGTCAGAGCTCACAGCCTTGAGCCTGTGCCATGCCTCCTCTCAGGGTGAA
GAGGCAGAGGGCATGGGGGGGGGGGGCGCAGCATCAGCCCATATCCTGGATGCACAAATTCACAGGCATGACGGGGCGAGGGC
CTGCTGCTCCTACCGTCACTCCCACATGCTAGCCCTCCAACGTCCTGGCTGACTTTCCCTGCCTCTGGTCCTGCGGCCCTGGA
CACAGCGGGAGGAGGGGACAGGATCCTGTGGTACCCCTTGGAGGAGGTTCCGGCACCTGTAGGCAGTTTCCAAGGATCCCTTG
TGGCCACATCCAGCTGTTAAATGGGCGTGTCCCTGGCAGCCACAGATGTCTGTCCAGTGCAGGAAAGTCTGTCCAGTGCAGGA
AAGGGCAGGCAGAGAGCTGGCTCCCAGCCCCAAATGCATGTCTCCCTCCCTGGGGCCCAGGCTGGCACAGAAGTCAGGCCTGC
CAGTGGGAAACTTGGGGGAAACTTGGTGCCCTGGTCCAGCAGCCTGCCCTGCAGCAGCAAGCATGGCCTCTGGAGGCTGTCGT
CCTCCTGGCCTCCAGGATTGCTTTTCCTTTCTTCCTAGAACTCCAGCCCTTAAGAAAATCAGAAGCCCTGGCAGGCACATTGC
CTCTGTGCTGTGCTTTTACCCAGCGAAGCATCAGGGCAGACAGCCAATTTCAACACTGCTCTTGGCTGGGAAGTGCCCTCATC
TCTGGCAGCCCCCACAGAGAAAGTGCAGGGCCCCGGGGCTGTGGCTGCCTCAGGGCAGGTCTCCCTTGTGACAGCCTCTTGTC
ATGGGCCTGGGAGTGGACCCCTCCCATCCCTGCCGTGCATCCTGTTGAGTAGACAGCTCAGGCTAGTACCCAAGAGGGTGGCC
AGCAGATCACAGGGGATGTCCCTTTTGTCTTAGCTGTTTATGGGCTGGAGGAACCACTGTTCAGCCACATCTCCTCTCCTGCC
CCTCCTGCATTCACCCGAGGTTGATGGGAGCTCTGTGGGGGGAGACAGGCAGGGGAGGGGCCAGGCAACACCTGGGCATATCC
CCAGGGTGCCTCCTCTGATCCCCAGGAGGGGCAGCACCCACCCTCTTCTTTTTCCAATTTGTTTTTATTGTGGTTAAATATAC
ATAACGTAAAATTTACCATCTTAATTATCTTTATGTGTACGGTGTGGTGGCATTAAATACATTCATCATGTTGTGTGGCCGTC
ACCACCATCCCTTTCCAGAACTAGCTCATCTTCCCAAACTGAAACTCTGTCCCCGTTAAATACTAACTCTCCGTTCCCCAGGC
ACTCTCTGCCCCCAACCCCAGGCACCCACCATTCTGCTTTCTGTCTCTGTGATTCGATGACTCTAGGGACTTCATATAAGGGA
AATCACACAGTGTTTGTCCTTTTGTGGTGGCTGCTTATTTTGCTGAGCACAATGTCCTTGAGGTTCATCCATGTTGTAGTGTG
TACCAGGAATCCCTTCCTTTTTAATGTTGAATAATTCCCCATTGTATGGATGGATCATGTTTGGCTTATCCACCCATCCATCG
GTGGACACCTGGGTGCCTTCCACCTCCAAGCTCTTGTGAACAATGATGCTATCTATGAATATGGTGTACAAATGTCTCTAAAA

The Stochastic Nature of Signal Sensing

(start codons)

A T G

(stop codons)

T G A
T A A
T A G

(donor splice sites)

G T

(acceptor splice sites)

A G

Common Assumptions in Gene Finding

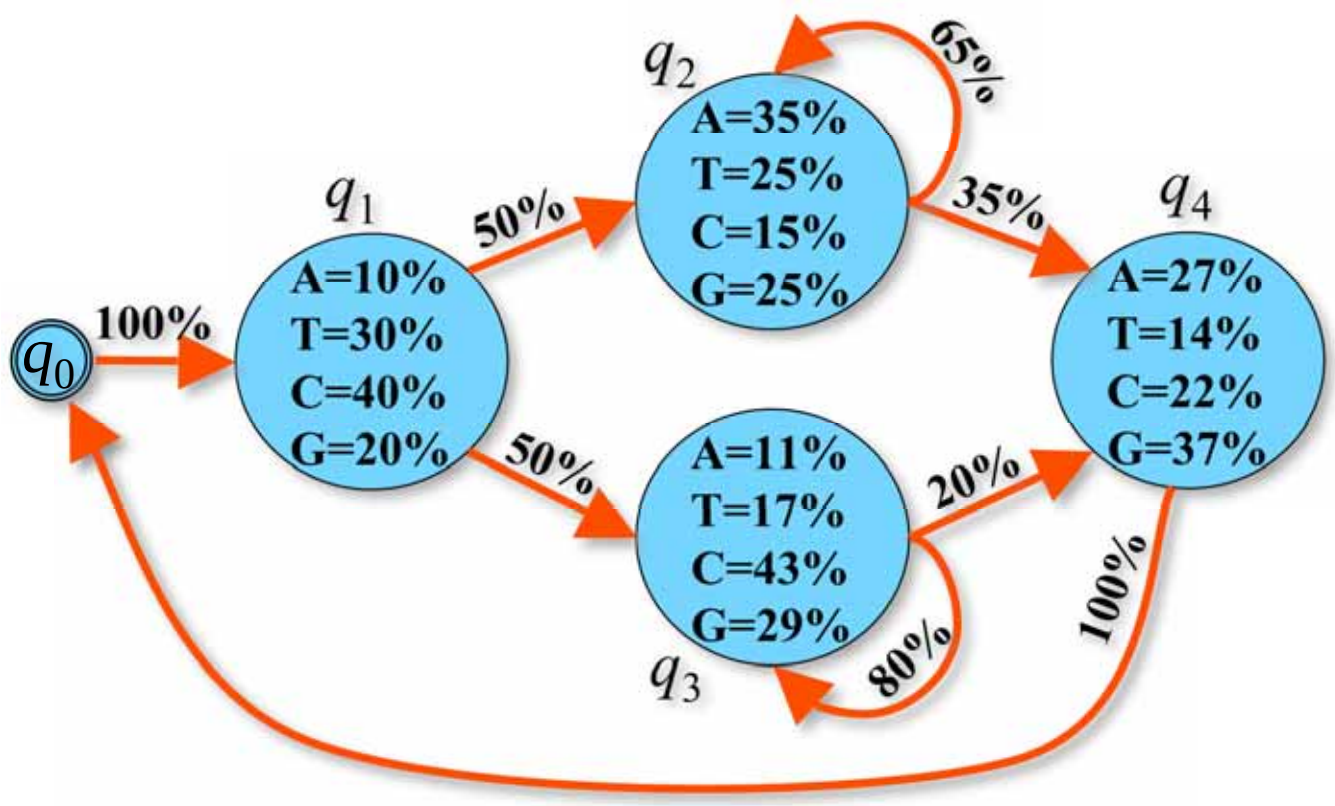
- No overlapping genes
- No nested genes
- No frame shifts or sequencing errors
- Optimal parse only
- No split start codons (ATGT...AGG)
- No split stop codons (TGT...AGAG)
- No alternative splicing
- No selenocysteine codons (TGA)
- No ambiguity codes (Y,R,N, etc.)

Definition of a Hidden Markov Model

$M = (Q, \alpha, P_t, q_0, P_e)$

$Q = (q_0, q_1, q_2, q_3, q_4)$

$\alpha = \{A, T, C, G\}$

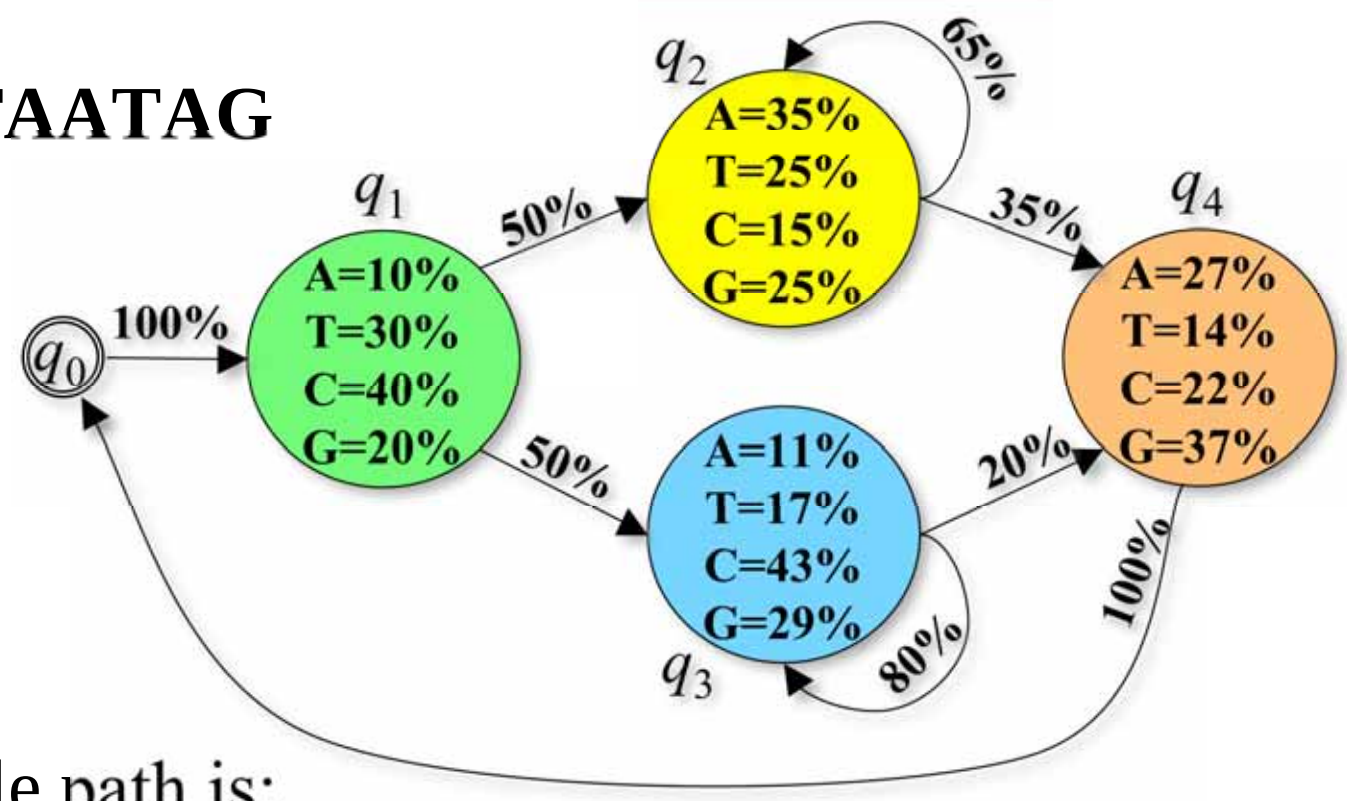


Finding the Most Probable Path

Example: **CATTAATAG**

top: 7.0×10^{-7}

bottom: 2.8×10^{-9}



The most probable path is:

States: 122222224

Sequence: CATTAATAG

resulting in this parse:

States: 122222224

Sequence: CATTAATAG

feature 1: C

feature 2: ATTAATA

feature 3: G

Decoding with an HMM

$$\phi_{\max} = \underset{\phi}{argmax}\, P(\phi \mid S) = \underset{\phi}{argmax}\, \frac{P(\phi \wedge S)}{P(S)}$$

$$= \underset{\phi}{argmax}\, P(\phi \wedge S)$$

$$= \underset{\phi}{argmax}\, P(S \mid \phi) P(\phi)$$

$$P(S \mid \phi) = \prod_{i=0}^{L-1} P_e(x_i \mid q_{(i+1)})$$

emission prob.

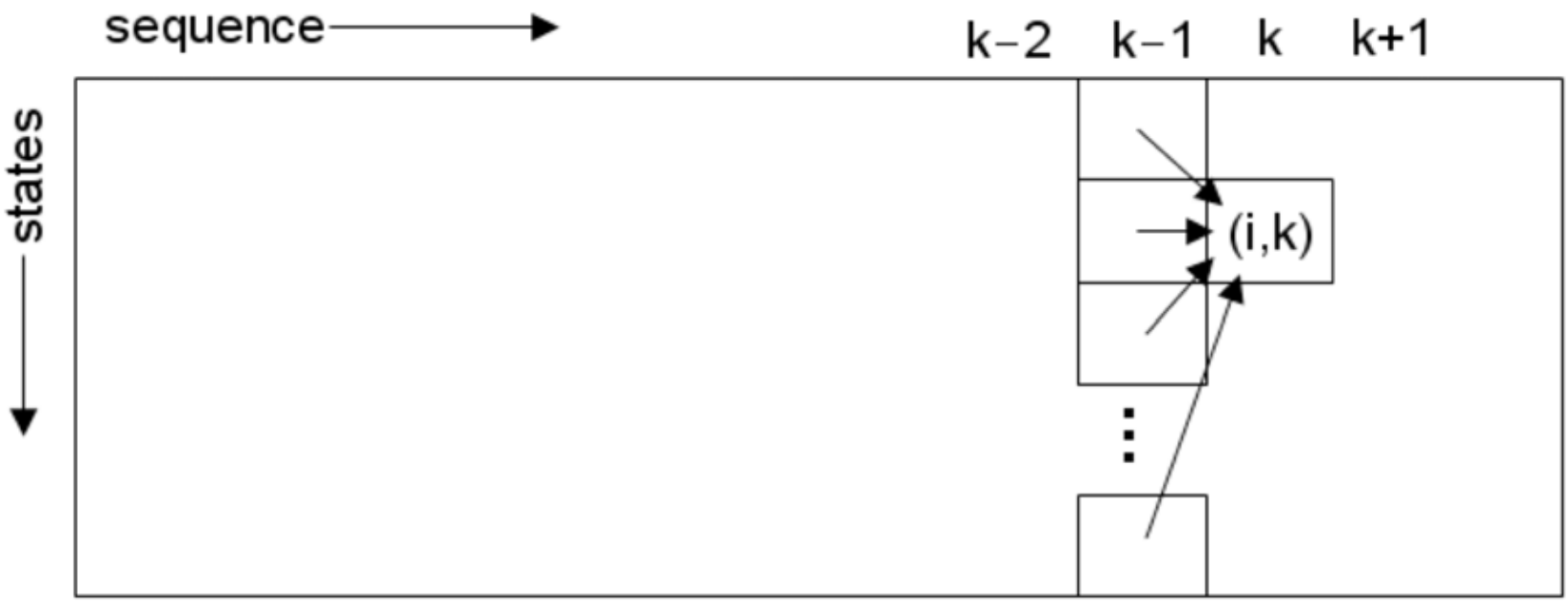
$$P(\phi) = \prod_{i=0}^{L} P_t(q_{(i+1)} \mid q_{(i)})$$

transition prob.

$$\phi_{\max} = \underset{\phi}{argmax}\, P_t(q_0 \mid q_{(L)}) \prod_{i=0}^{L-1} P_e(x_i \mid q_{(i+1)}) P_t(q_{(i+1)} \mid q_{(i)})$$

Decoding with the Viterbi Algorithm

$$V(i,k) = \begin{cases} \max\limits_{j} V(j,k-1) P_t(q_i \mid q_j) P_e(x_k, q_i) & \text{if } k > 0, \\ P_t(q_i \mid q_0) P_e(x_0 \mid q_i) & \text{if } k = 0. \end{cases}$$



$$\phi^* = \underset{\phi_{i,L-1}}{\arg\max}\, V(i, L-1) P_t(q_0 \mid q_i)$$

Training an HMM from Labeled Sequence

CGATATTCGATTCTACGCGCGTATACTAGCTTATCTGATC

0111111222222211111122221111111222211111 0

transitions

		to state		
		0	**1**	**2**
from state	**0**	0 (0%)	1 (100%)	0 (0%)
	1	1 (4%)	21 (84%)	3 (12%)
	2	0 (0%)	3 (20%)	12 (80%)

$$a_{i,j} = \frac{A_{i,j}}{\sum_{h=0}^{|Q|-1} A_{i,h}}$$

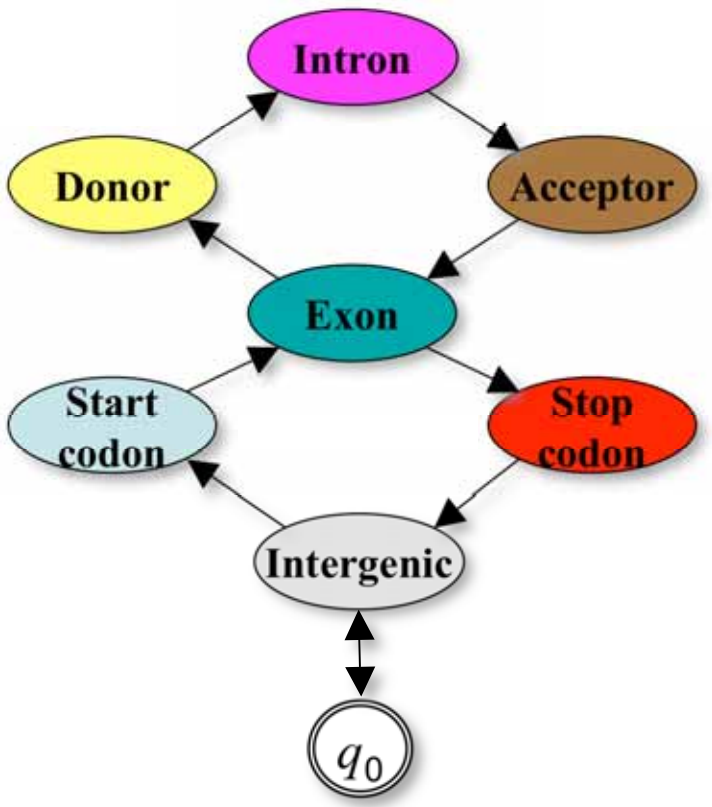
emissions

		symbol			
		A	**C**	**G**	**T**
in state	**1**	6 (24%)	7 (28%)	5 (20%)	7 (28%)
	2	3 (20%)	3 (20%)	2 (13%)	7 (47%)

$$e_{i,k} = \frac{E_{i,k}}{\sum_{h=0}^{|\Sigma|-1} E_{i,h}}$$

Using an HMM for Gene Prediction

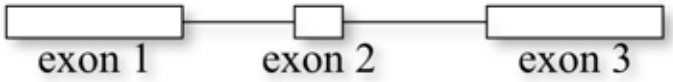
the Markov model:



the input sequence: AGCTAGCAGTATGTCATGGCATGTTCGGAGGTAGTACGTAGAGGTAGCTAGTATAGGTCGATAGTACGCGA

the most probable path:

the gene prediction:



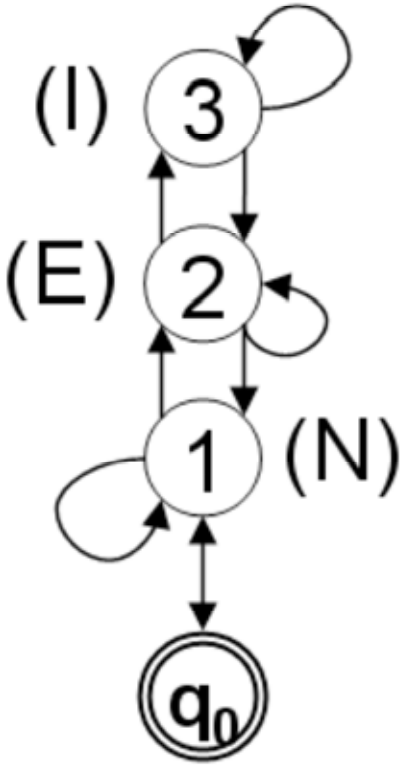
Higher Order Markovian Eukaryotic Recognizer (HOMER)

HOMER, version H_3

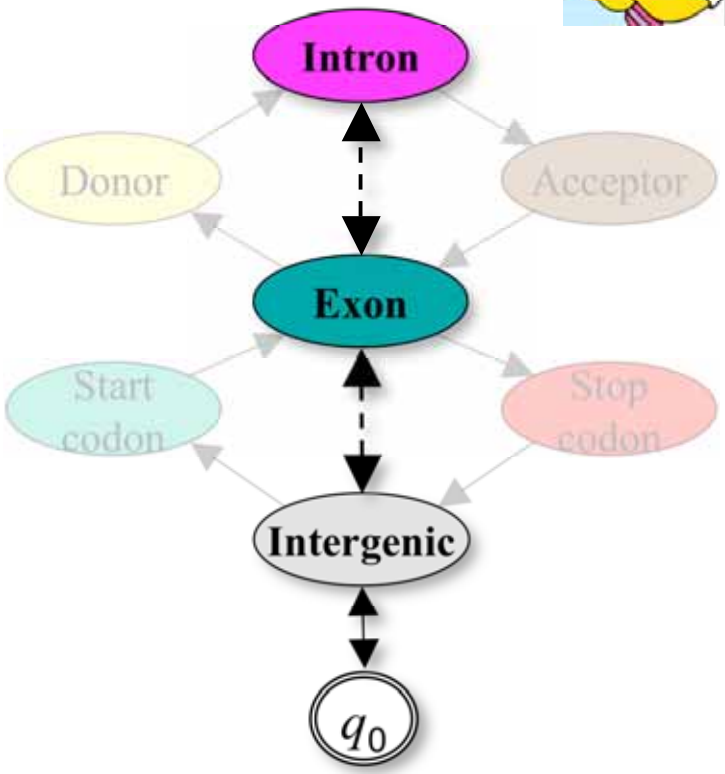
I=intron state

E=exon state

N=intergenic state



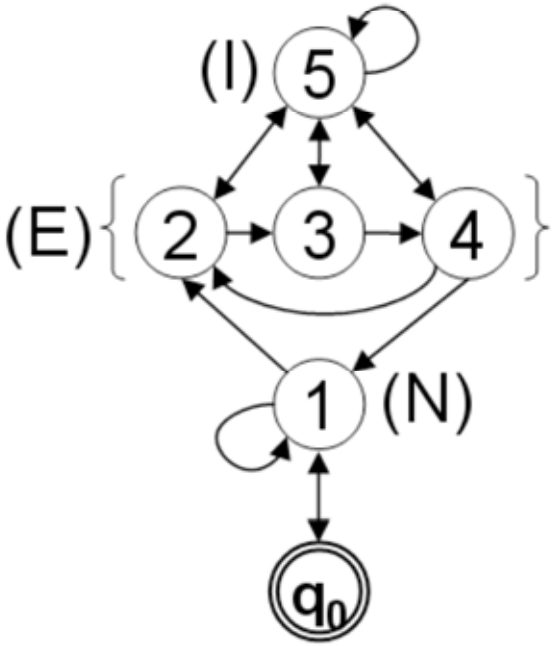
≈



tested on 500 *Arabidopsis* genes:

	nucleotides			**splice sites**		**start/stop codons**		**exons**			**genes**	
	Sn	*Sp*	*F*	*Sn*	*Sp*	*Sn*	*Sp*	*Sn*	*Sp*	*F*	*Sn*	#
baseline	50	28	44	0	0	0	0	0	0	0	0	0
H_3	53	88	66	0	0	0	0	0	0	0	0	0

HOMER, version H_5



three exon states, for the three codon positions

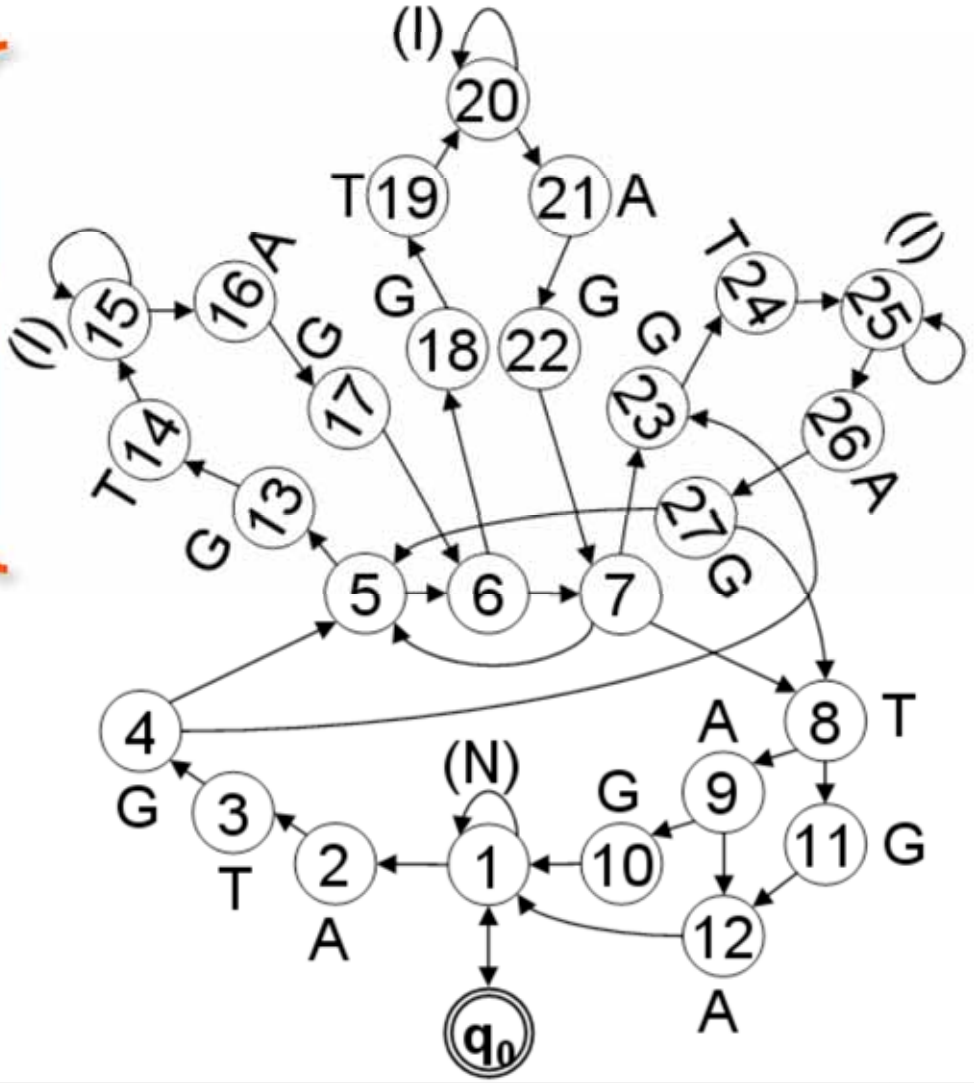
	nucleotides			splice sites		start/stop codons		exons			genes	
	Sn	*Sp*	*F*	*Sn*	*Sp*	*Sn*	*Sp*	*Sn*	*Sp*	*F*	*Sn*	#
H_3	53	88	66	0	0	0	0	0	0	0	0	0
H_5	65	91	76	1	3	3	3	0	0	0	0	0

HOMER version H_{17}

	nucleotides			splice sites		start/stop codons		exons			genes	
	Sn	*Sp*	*F*	*Sn*	*Sp*	*Sn*	*Sp*	*Sn*	*Sp*	*F*	*Sn*	#
H_5	65	91	76	1	3	3	3	0	0	0	0	0
H_{17}	81	93	87	34	48	43	37	19	24	21	7	35

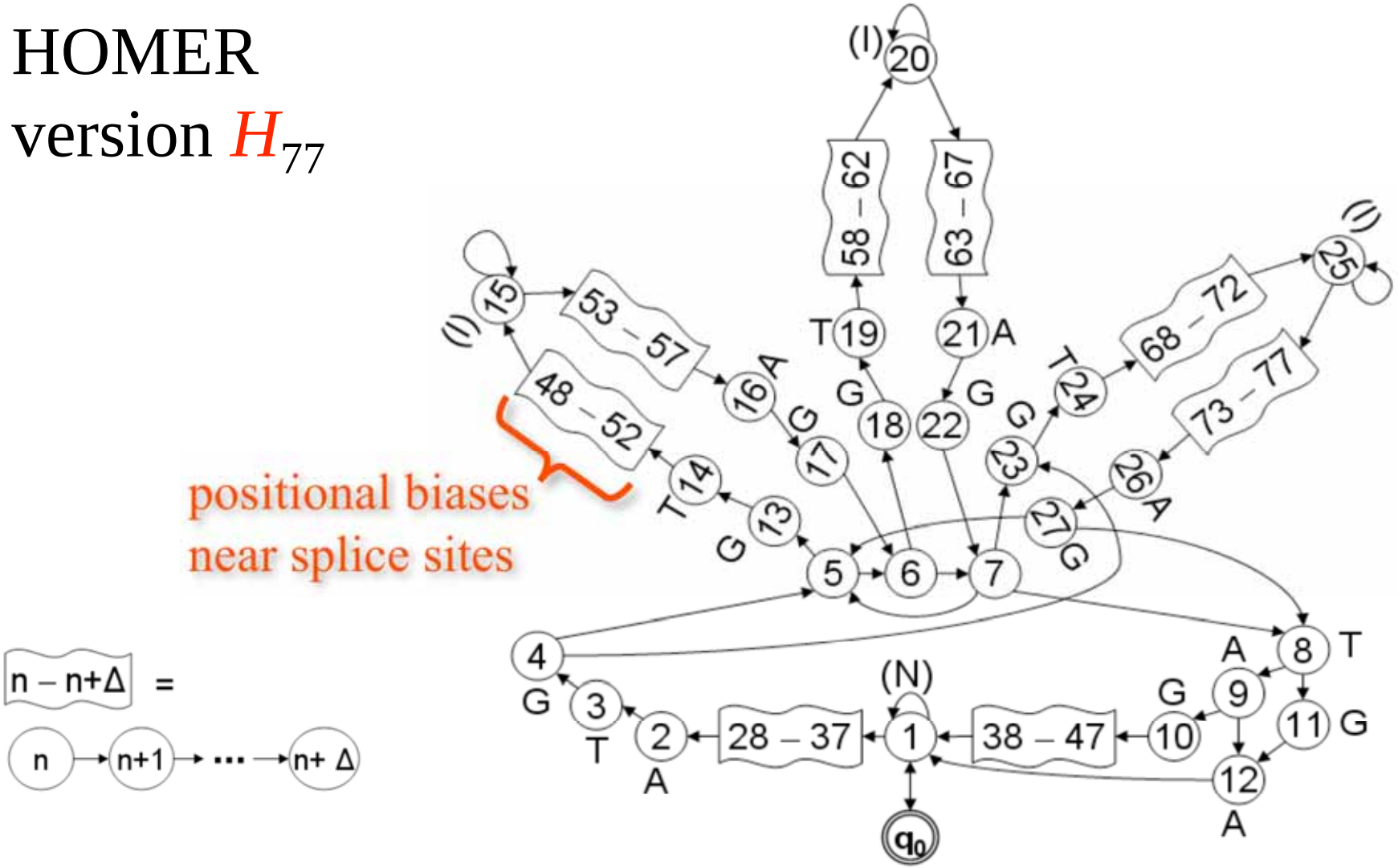
HOMER version H_{27}

three separate intron models

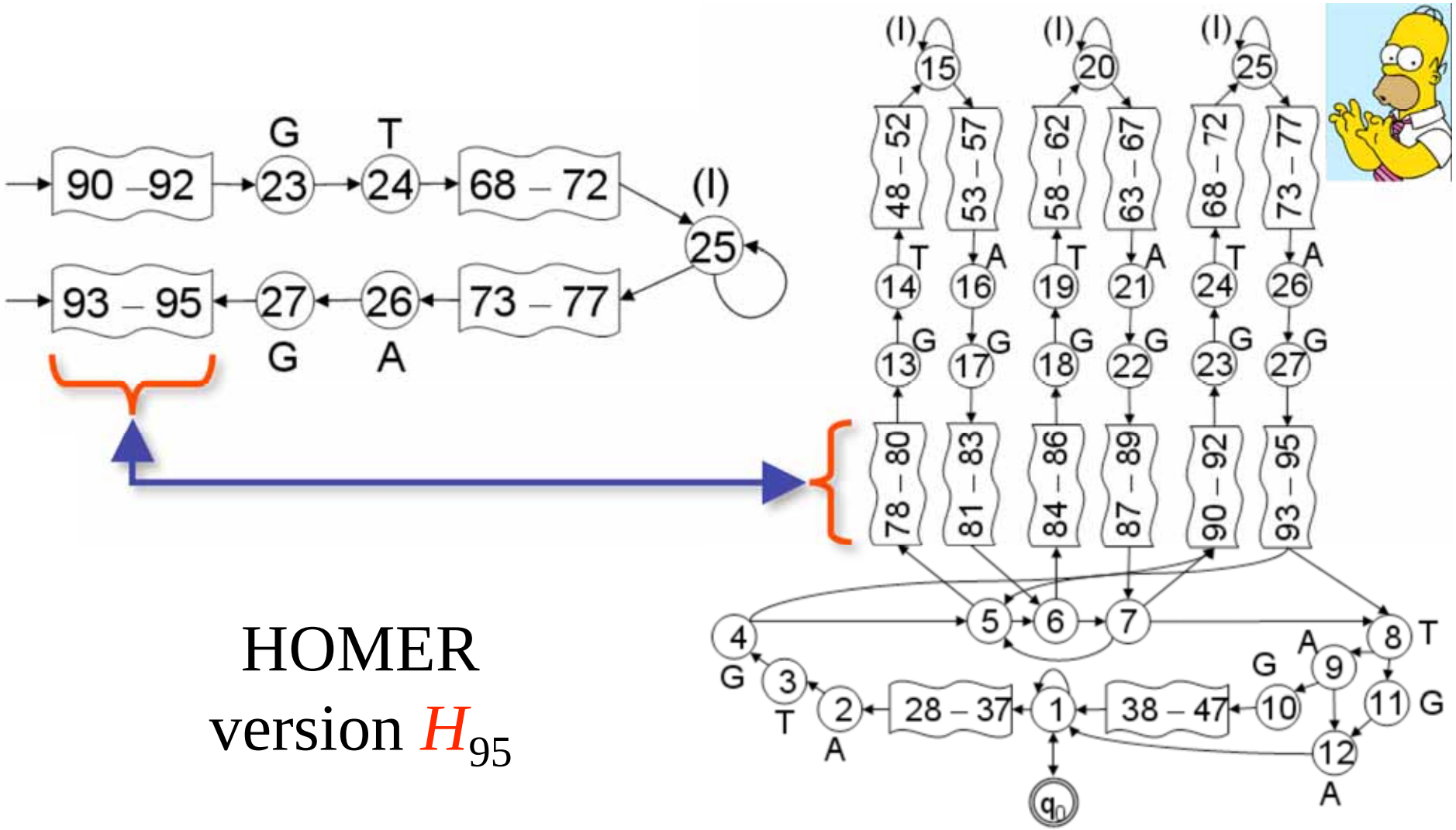


	nucleotides			splice		start/stop		exons			genes	
	Sn	*Sp*	*F*	*Sn*	*Sp*	*Sn*	*Sp*	*Sn*	*Sp*	*F*	*Sn*	#
H_{17}	81	93	87	34	48	43	37	19	24	21	7	35
H_{27}	83	93	88	40	49	41	36	23	27	25	8	38

HOMER
version H_{77}



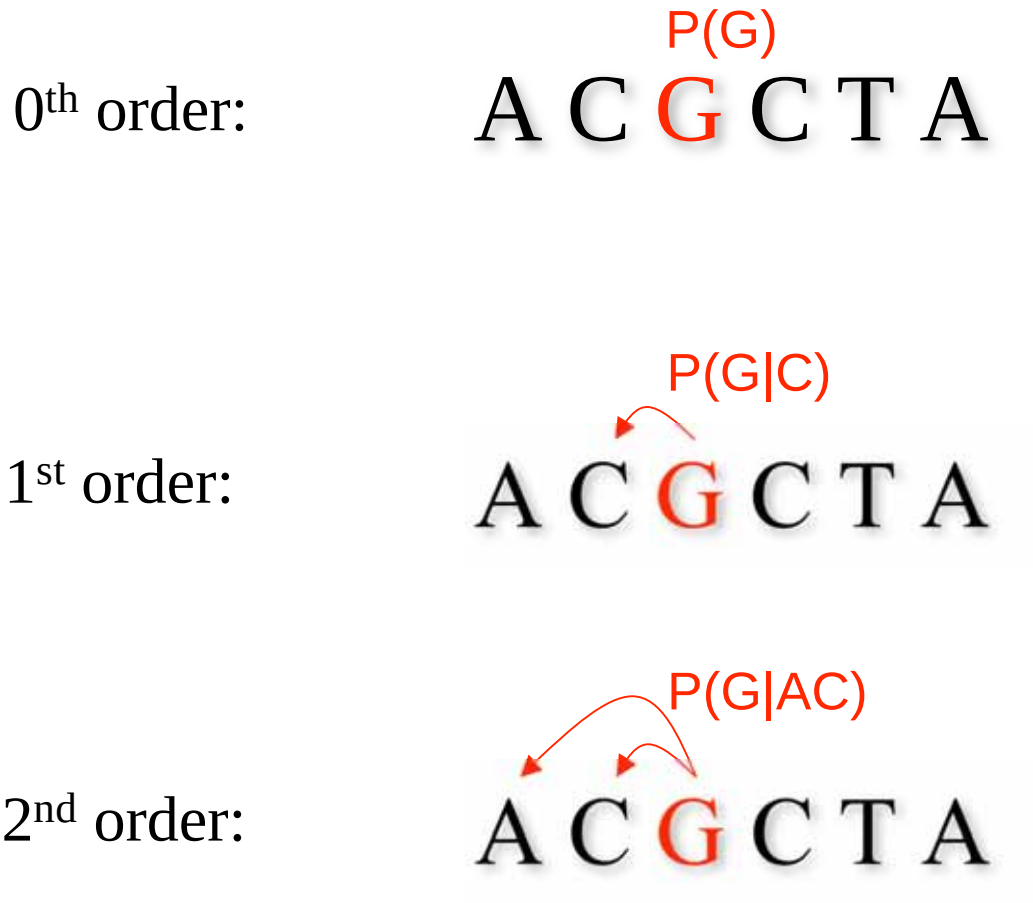
	nucleotides			splice		start/stop		exons			genes	
	Sn	*Sp*	*F*	*Sn*	*Sp*	*Sn*	*Sp*	*Sn*	*Sp*	*F*	*Sn*	#
H_{27}	83	93	88	40	49	41	36	23	27	25	8	38
H_{77}	88	96	92	66	67	51	46	47	46	46	13	65



HOMER
version H_{95}

	nucleotides			splice		start/stop		exons			genes	
	Sn	*Sp*	*F*	*Sn*	*Sp*	*Sn*	*Sp*	*Sn*	*Sp*	*F*	*Sn*	#
H_{77}	88	96	92	66	67	51	46	47	46	46	13	65
H_{95}	92	97	94	79	76	57	53	62	59	60	19	93

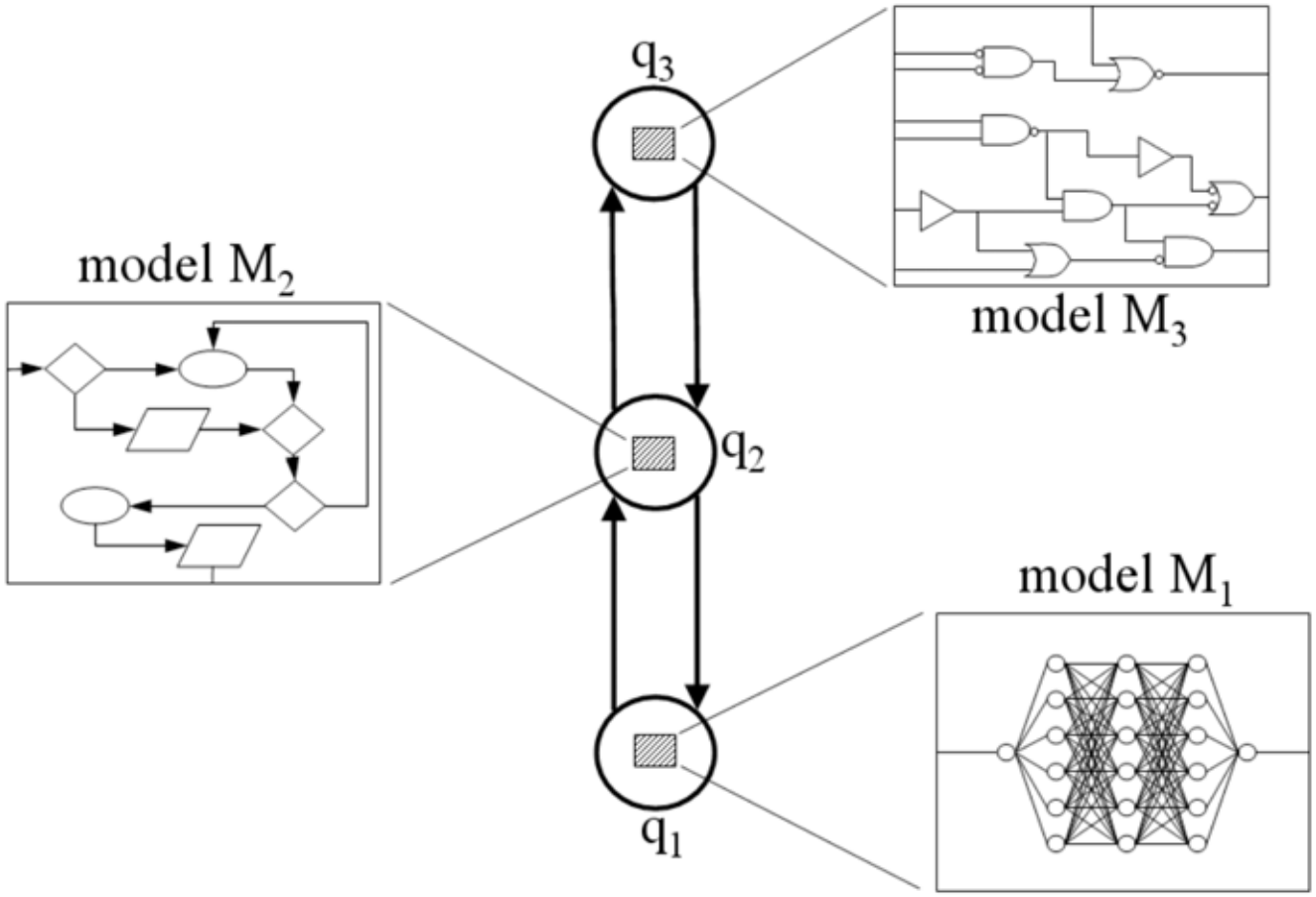
Higher-order Markov Models



Higher-order Markov Models

	order	nucleotides			splice sites		starts/ stops		exons			genes	
		Sn	*Sp*	*F*	*Sn*	*Sp*	*Sn*	*Sp*	*Sn*	*Sp*	*F*	*Sn*	#
H_{95}^{0}	0	92	97	94	79	76	57	53	62	59	60	19	93
H_{95}^{1}	1	95	98	97	87	81	64	61	72	68	70	25	127
H_{95}^{2}	2	98	98	98	91	82	65	62	76	69	72	27	136
H_{95}^{3}	3	98	98	98	91	82	67	63	76	69	72	28	140
H_{95}^{4}	4	98	97	98	90	81	69	64	76	68	72	29	143
H_{95}^{5}	5	98	97	98	90	81	66	62	74	67	70	27	137

Generalized Hidden Markov Models

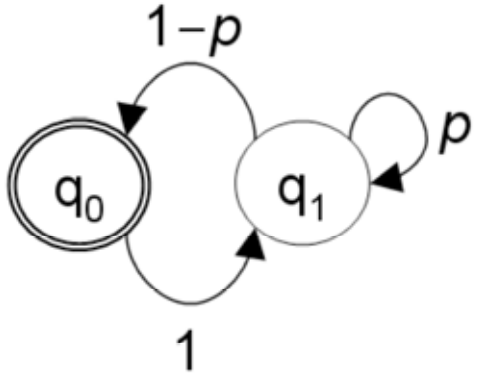


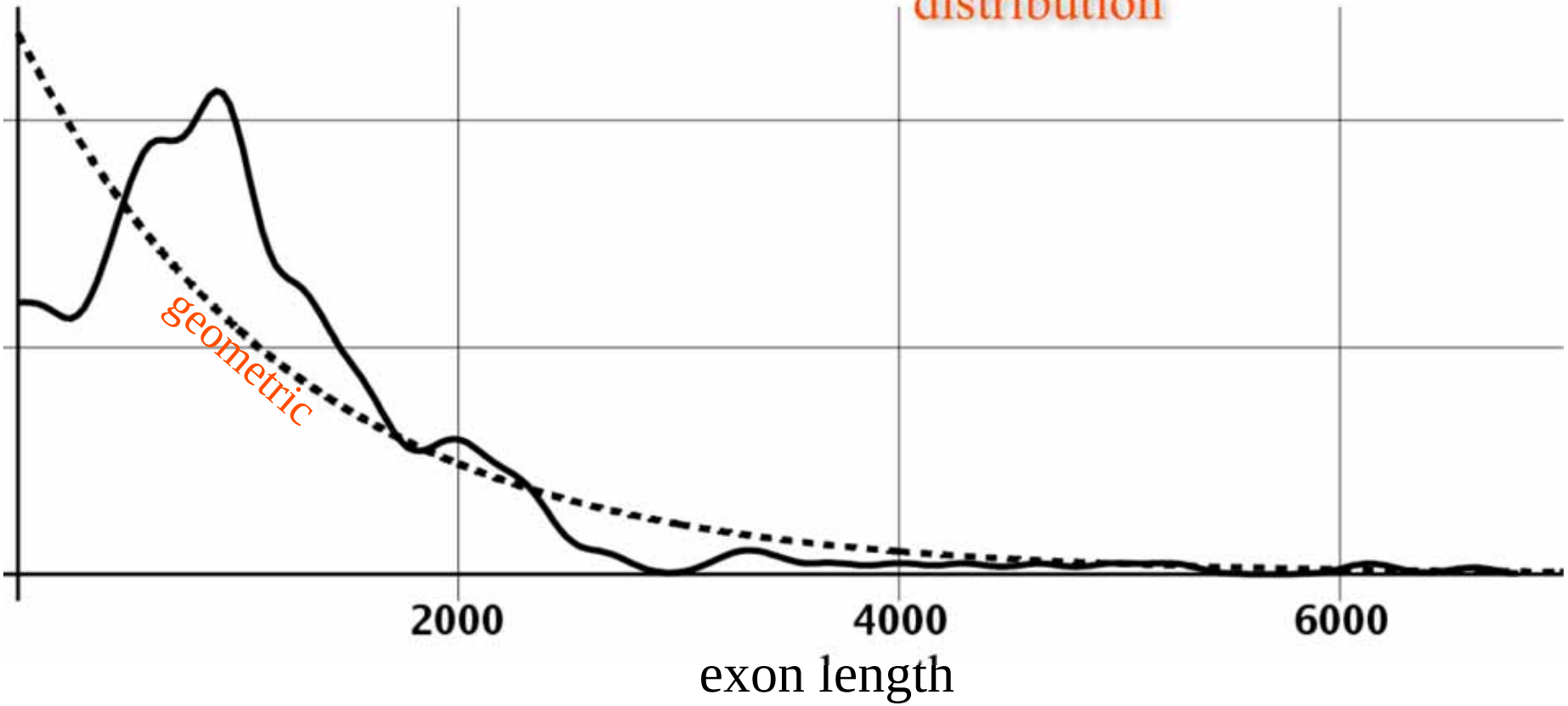
Advantages:
* Submodel abstraction
* Architectural simplicity
* State duration modeling

Disadvantages:
* Decoding complexity

HMMs & Geometric Feature Lengths

$$P(x_0 \ldots x_{d-1} \mid \theta) = \left(\prod_{i=0}^{d-1} P_e(x_i \mid \theta) \right) \underbrace{p^{d-1}(1-p)}_{\text{geometric distribution}}$$

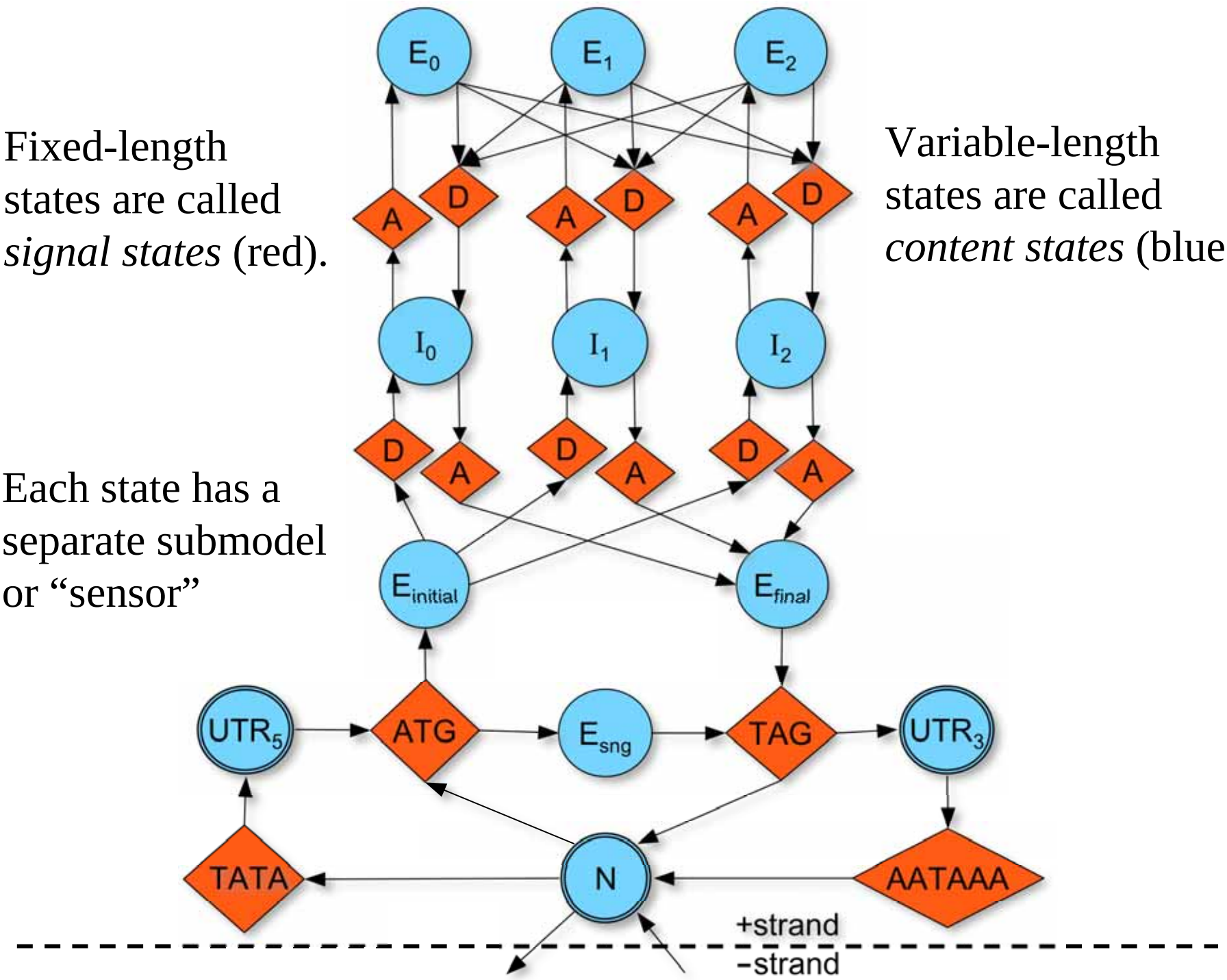




Fixed-length states are called *signal states* (red).

Variable-length states are called *content states* (blue).

Each state has a separate submodel or “sensor”



Some GHMM Submodel Types

1. WMM (Weight Matrix) $\prod_{i=0}^{L-1} P_i(x_i)$

2. Nth-order Markov Chain (MC) $\prod_{i=0}^{n-1} P(x_i \mid x_0 \ldots x_{i-1}) \prod_{i=n}^{L-1} P(x_i \mid x_{i-n} \ldots x_{i-1})$

3. Three-Periodic Markov Chain (3PMC) $\prod_{i=0}^{L-1} P_{(f+i)(\mathrm{mod}3)}(x_i)$

4. Nonstationary Markov Chain (NSMC) $max \prod_{i=1}^{n} P_i(x_{\sum_{j=1}^{i-1} d_j} \ldots x_{(\sum_{j=1}^{i} d_j)-1})$

5. Codon Bias $\prod_{i=0}^{n-1} P(x_{\alpha+3i} x_{\alpha+3i+1} x_{\alpha+3i+2})$

6. MDD

7. Interpolated Markov Models

Recall: Decoding with an HMM

$$\phi_{\max} = \underset{\phi}{argmax}\, P(\phi \mid S) = \underset{\phi}{argmax}\, \frac{P(\phi \wedge S)}{P(S)}$$

$$= \underset{\phi}{argmax}\, P(\phi \wedge S)$$

$$= \underset{\phi}{argmax}\, P(S \mid \phi) P(\phi)$$

$$P(S \mid \phi) = \prod_{i=0}^{L-1} P_e(x_i \mid q_{(i+1)})$$

emission prob.

$$P(\phi) = \prod_{i=0}^{L} P_t(q_{(i+1)} \mid q_{(i)})$$

transition prob.

$$\phi_{\max} = \underset{\phi}{argmax}\, P_t(q_0 \mid q_{(L)}) \prod_{i=0}^{L-1} P_e(x_i \mid q_{(i+1)}) P_t(q_{(i+1)} \mid q_{(i)})$$

Decoding with a GHMM

$$\phi_{\max} = \underset{\phi}{argmax}\, P(\phi \mid S) = \underset{\phi}{argmax}\, \frac{P(\phi \wedge S)}{P(S)}$$

$$= \underset{\phi}{argmax}\, P(\phi \wedge S)$$

$$= \underset{\phi}{argmax}\, P(S \mid \phi) P(\phi)$$

$$P(S \mid \phi) = \prod_{i=1}^{|\phi|-2} P_e(S_i \mid q_{(i)}, d_i)$$

emission prob.

$$P(\phi) = \prod_{i=0}^{|\phi|-2} P_t(q_{(i+1)} \mid q_{(i)}) P_d(d_i \mid q_{(i)})$$

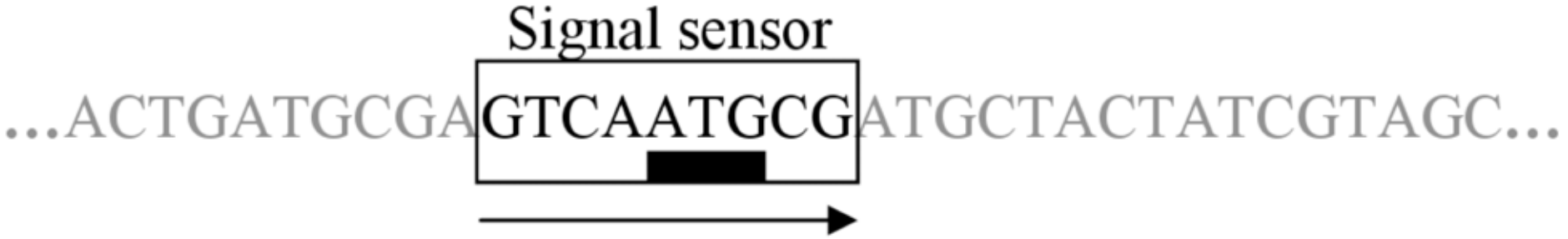
transition prob.

duration prob.

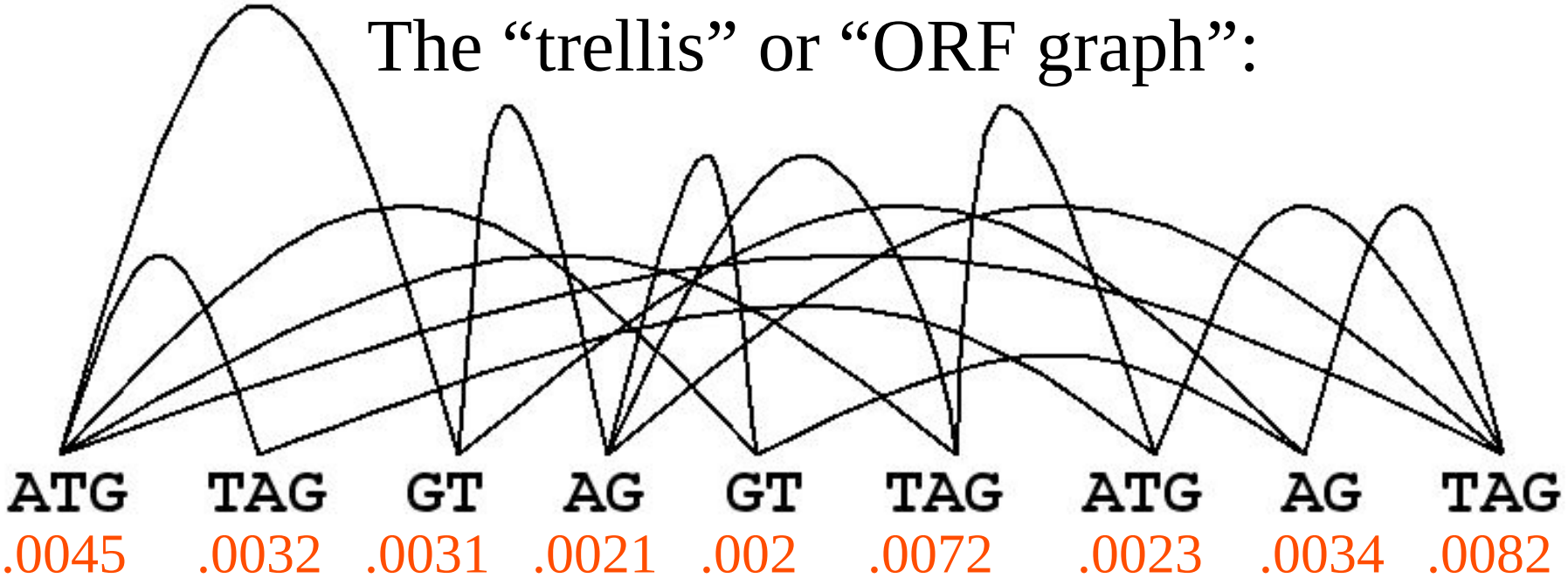
$$\phi_{\max} = \underset{\phi}{argmax} \prod_{i=0}^{|\phi|-2} P_e(S_i \mid q_{(i)}, d_i) P_t(q_{(i+1)} \mid q_{(i)}) P_d(d_i \mid q_{(i)})$$

Efficient Decoding via Signal Sensors

Each signal state has a *signal sensor*:



The “trellis” or “ORF graph”:

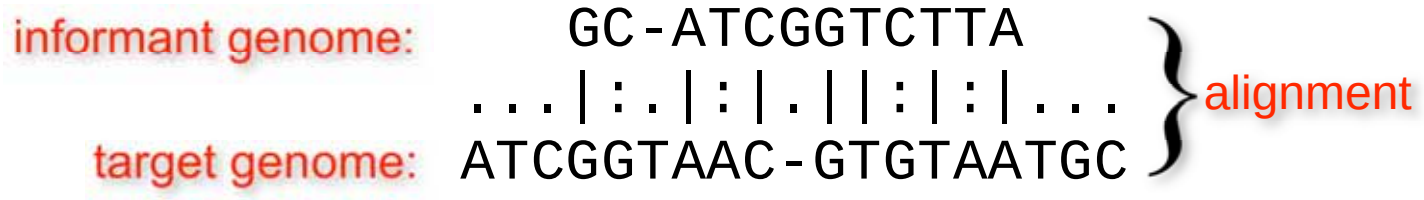


Accuracy

	nucleotides			**splice**		**start/stop**		**exons**			**genes**	
	Sn	*Sp*	*F*	*Sn*	*Sp*	*Sn*	*Sp*	*Sn*	*Sp*	*F*	*Sn*	#
HMM	98	97	98	90	81	66	62	74	67	70	27	
GHMM	94	96	95	87	89	77	74	79	80	80	45	

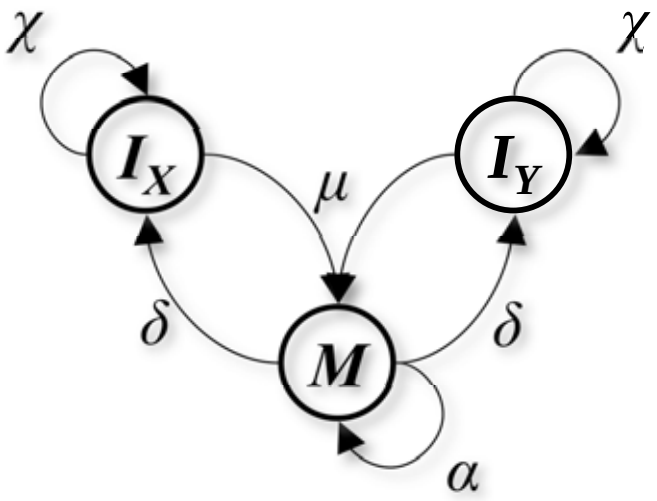
Comparative Methods

Problem: Predict genes in a *target genome* G based on the contents of G and also based on the contents of one or more *informant genomes* $I^{(1)}$... $I^{(n)}$:



Rationale: *Natural selection should operate more strongly on protein-coding DNA than on non-functional DNA such as introns.*

Pair HMM's (PHMM's)



I_X : emit a symbol into output channel X

I_Y : emit a symbol into output channel Y

M : emit a symbol into both X and Y

I_X can be called an *insertion state*

I_Y can be called a *deletion state*

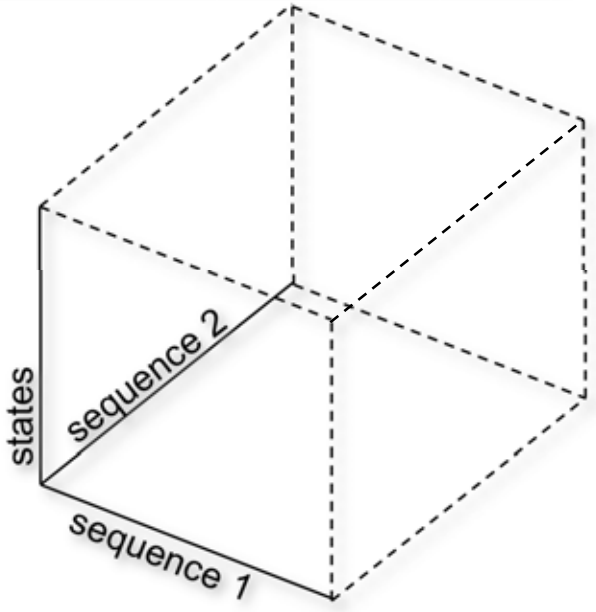
M can be called a *match state*

Decoding for PHMM's

$$\phi^* = \underset{\phi=\{q^0=q_0,\ldots,q_{m-1}=q^0\}}{\arg\max} P_t(q^0 \mid q_{m-2}) \prod_{i=1}^{m-2} P_e(a_{i,1}, a_{i,2} \mid q_i) P_t(q_i \mid q_{i-1})$$

$$V_{i,j,k} = \begin{cases} \max\limits_h V_{i-1,j-1,h} P_t(q_k \mid q_h) P_e(s_{i-1,1}, s_{j-1,2} \mid q_k) & \text{for } q_k \in Q_M \\ \max\limits_h V_{i-1,j,h} P_t(q_k \mid q_h) P_e(s_{i-1,1}, - \mid q_k) & \text{for } q_k \in Q_I \\ \max\limits_h V_{i,j-1,h} P_t(q_k \mid q_h) P_e(-, s_{j-1,2} \mid q_k) & \text{for } q_k \in Q_D \end{cases}$$

$$T_{i,j,k} = \begin{cases} \underset{(i-1,j-1,h)}{\arg\max} V_{i-1,j-1,h} P_t(q_k \mid q_h) P_e(s_{i-1,1}, s_{j-1,2} \mid q_k) & \text{for } q_k \in Q_M \\ \underset{(i-1,j,h)}{\arg\max} V_{i-1,j,h} P_t(q_k \mid q_h) P_e(s_{i-1,1}, - \mid q_k) & \text{for } q_k \in Q_I \\ \underset{(i,j-1,h)}{\arg\max} V_{i,j-1,h} P_t(q_k \mid q_h) P_e(-, s_{j-1,2} \mid q_k) & \text{for } q_k \in Q_D \end{cases}$$



$$V_{0,0,k} = \begin{cases} 1 & \text{for } q_k = q^0 \\ 0 & \text{otherwise} \end{cases}$$

$$V_{i>0,0,k} = \begin{cases} \max\limits_h V_{i-1,0,h} P_t(q_k \mid q_h) P_e(s_{i-1,1}, - \mid q_k) & \text{for } q_k \in Q_I \\ 0 & \text{otherwise} \end{cases}$$

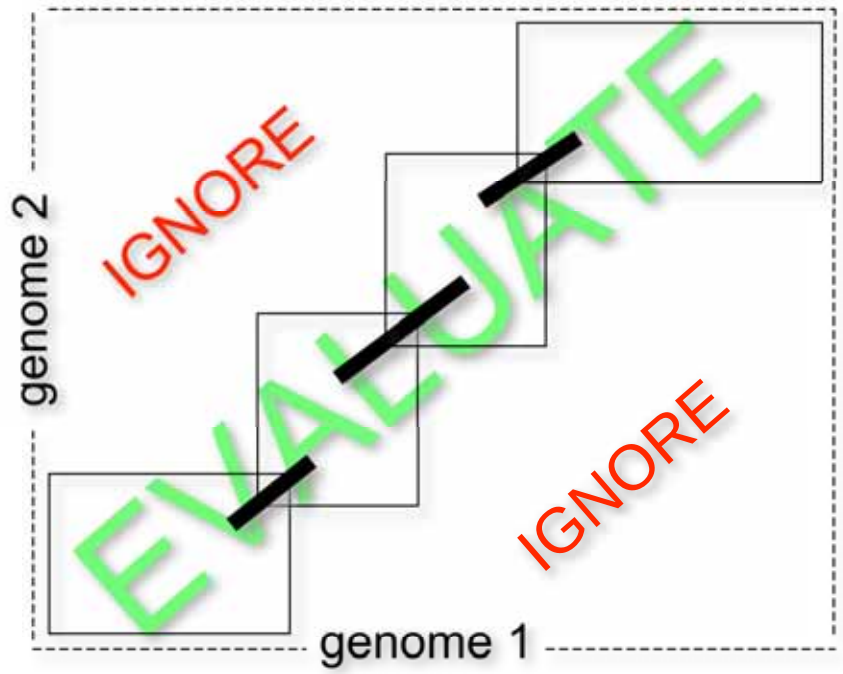
$$V_{0,j>0,k} = \begin{cases} \max\limits_h V_{0,j-1,h} P_t(q_k \mid q_h) P_e(-, s_{j-1,2} \mid q_k) & \text{for } q_k \in Q_D \\ 0 & \text{otherwise} \end{cases}$$

$$T_{i>0,0,k} = \begin{cases} \underset{(i-1,0,h)}{\arg\max} V_{i-1,0,h} P_t(q_k \mid q_h) P_e(s_{i-1,1}, - \mid q_k) & \text{for } q_k \in Q_I \\ NIL & \text{otherwise} \end{cases}$$

$$T_{0,j>0,k} = \begin{cases} \underset{(0,j-1,h)}{\arg\max} V_{0,j-1,h} P_t(q_k \mid q_h) P_e(-, s_{j-1,2} \mid q_k) & \text{for } q_k \in Q_D \\ NIL & \text{otherwise} \end{cases}$$

$$T_{0,0,k} = NIL$$

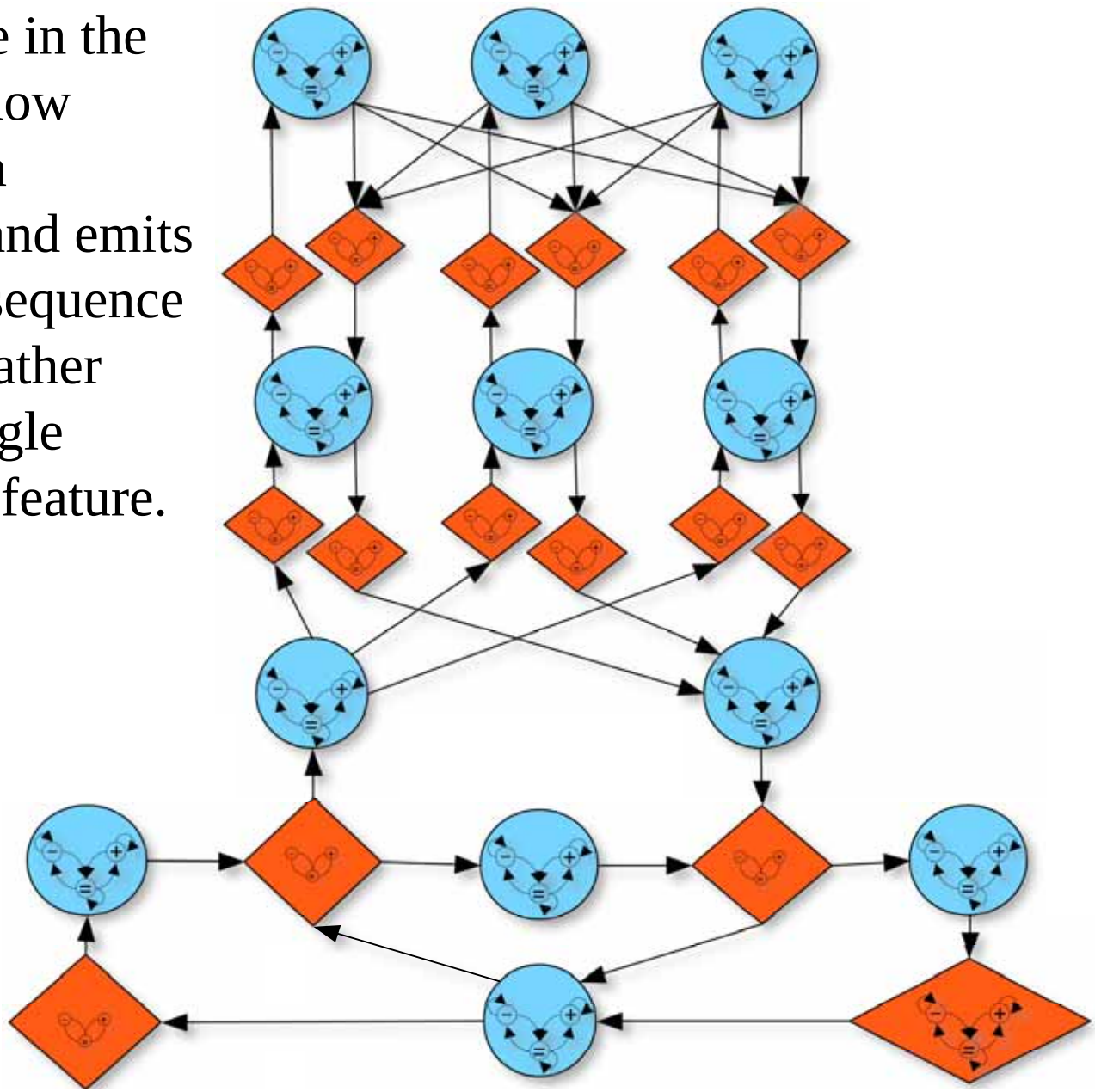
Pruning the Search Space for PHMM's



- Find significantly conserved regions (thick bars) using BLAST
- Force the DP algorithm to select a path which passes through these regions
- Allow more flexibility in the regions not aligned
- Do not evaluate regions of the matrix far from the conserved regions

Pair GHMM's (PGHMM's)

Each state in the GHMM now contains a PHMM, and emits a pair of sequence features rather than a single sequence feature.



Recall: GHMM Decoding

Finding the optimal parse, $\phi_{\max}$:

$$\phi_{\max} = \arg\max_{\phi} P(\phi \mid S) = \arg\max_{\phi} \frac{P(\phi, S)}{P(S)}$$

$$= \arg\max_{\phi} P(\phi, S) = \arg\max_{\phi} P(S \mid \phi) P(\phi)$$

$$= \arg\max_{\phi} \prod_{i=1}^{n-1} P_e(S_i \mid q_i, d_i) P_t(q_i \mid q_{i-1}) P_d(d_i \mid q_i)$$

P_e =emission, P_t =transition, P_d =duration

Decoding with a GPHMM

$$\phi^* = P_t(q^0 \mid q_{n-2}) \underset{\phi}{\arg\max} \prod_{i=1}^{n-2} P_e(S_{i,1}, S_{i,2} \mid q_i, d_{i,1}, d_{i,2})$$

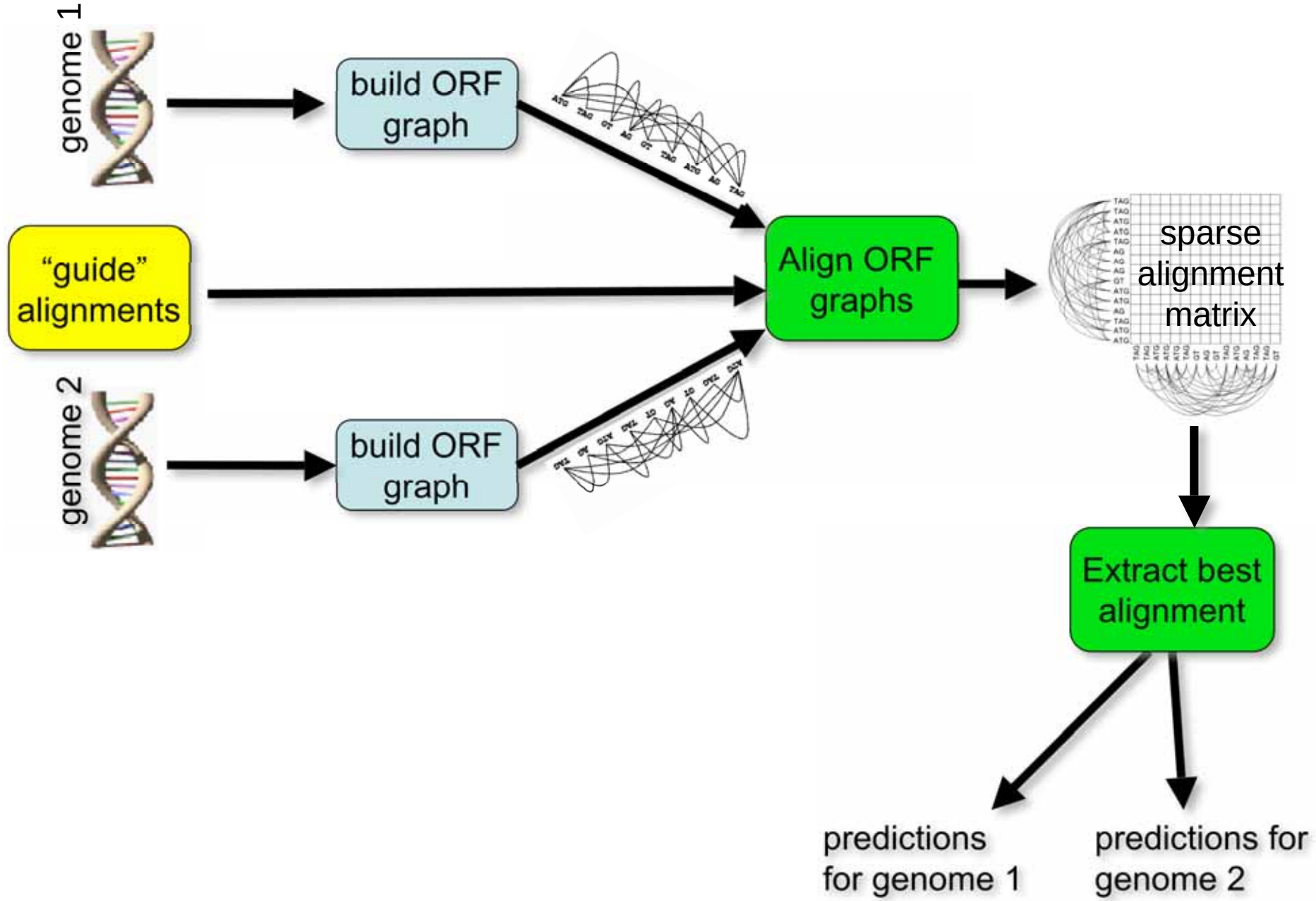
$$\underbrace{P_t(q_i \mid q_{i-1})}_{\text{=transition}} P_d(d_{i,1}, d_{i,2} \mid q_i)$$

$$P_e(S_{i,1}, S_{i,2} \mid q_i, d_{i,1}, d_{i,2}) \approx \underbrace{P_e(S_{i,1} \mid q_i, d_{i,1})}_{\text{=single-genome emission score}} \underbrace{P_e(S_{i,2} \mid S_{i,1}, q_i, d_{i,2})}_{\text{≈alignment score}}$$

$$P_d(d_{i,1}, d_{i,2} \mid q_i) = P(d_{i,1} \mid q_i) P(d_{i,2} \mid d_{i,1}, q_i)$$

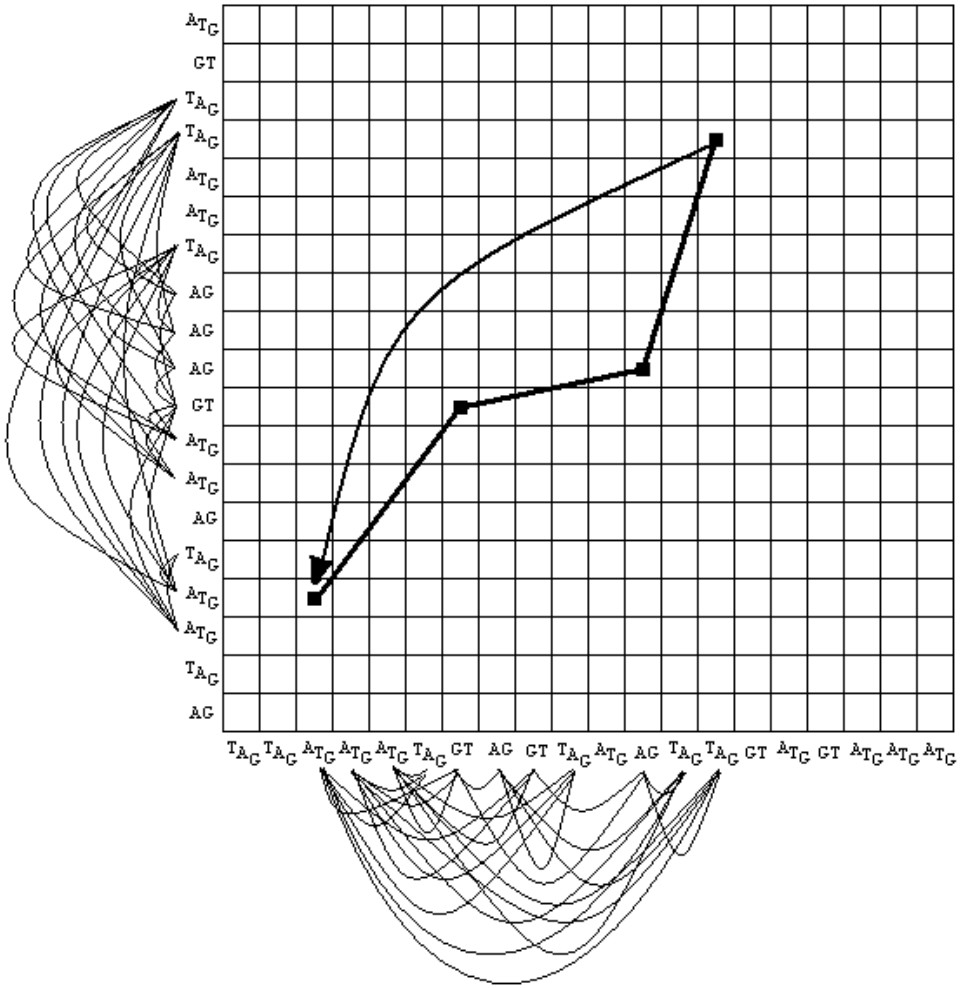
$$\approx \underbrace{P(d_{i,1} \mid q_i)}_{\text{=single-genome duration score}} \underbrace{P(\Delta_d \mid q_i)}_{\text{ignore (implicit in alignment score)}}, \quad \Delta_d = d_{i,2} - d_{i,1}$$

Practical GPHMM Decoding



Aligning Parse Graphs

The two parse graphs can be aligned using a global alignment algorithm. The optimal alignment corresponds to the chosen pair of orthologous gene predictions.



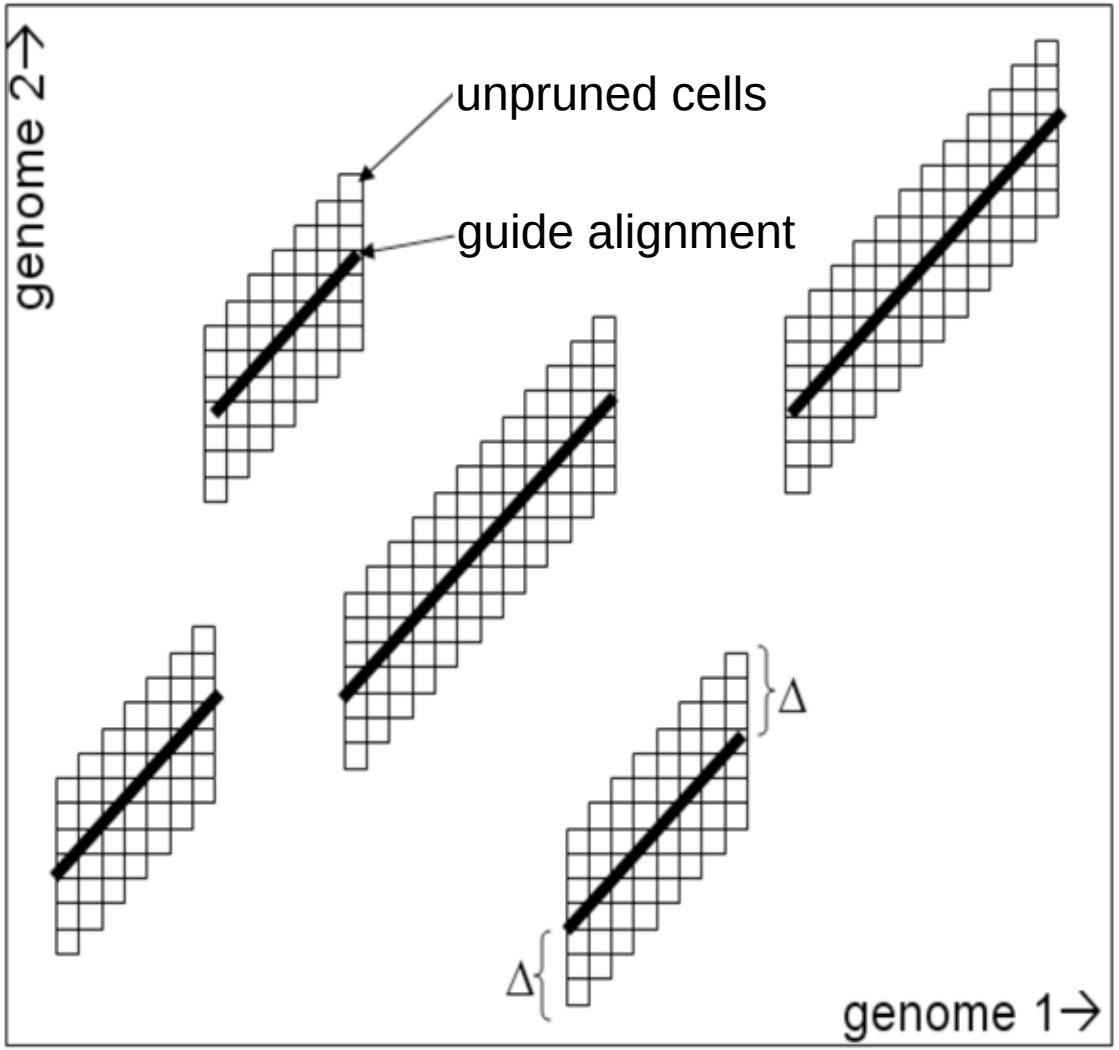
The alignment is constrained by the topologies of the two parse graphs:

1.Only like signals can align,

2.Two signals can align only if they have neighbors which also align,

3.Standard phase constraints apply.

Using a Sparse Alignment Matrix

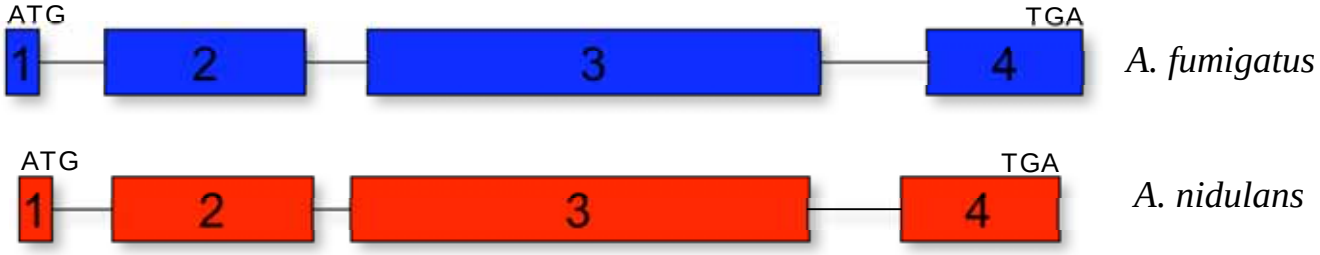


Accuracy

Data set: 147 high-confidence *Aspergillus fumigatus* × *A. nidulans* orthologs (493 exons, 564kb).

	nucleotide accuracy	exon sensitivity	exon specificity	exact genes
GHMM	99%	78%	73%	54%
GPHMM	99%	89%	85%	74%

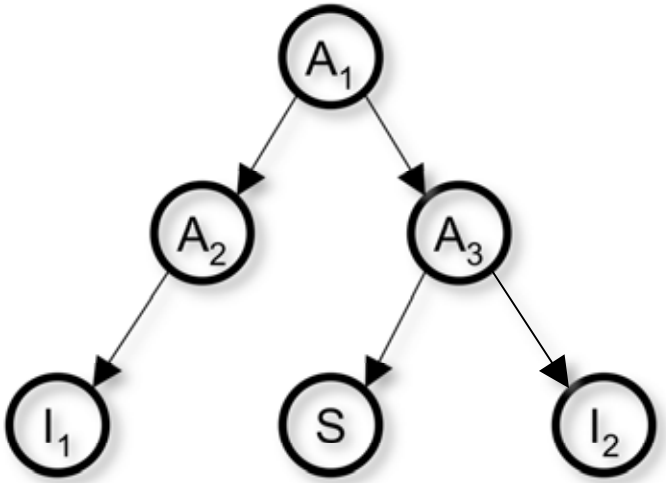
How Does Homology Help?



feature	amino acid alignment score	<,>	nucleotide alignment score
exon 1	100%	>	71%
intron 1	14%	<	51%
exon 2	98%	>	85%
intron 2	29%	<	49%
exon 3	97%	>	82%
intron 3	9%	<	49%
exon 4	96%	>	83%

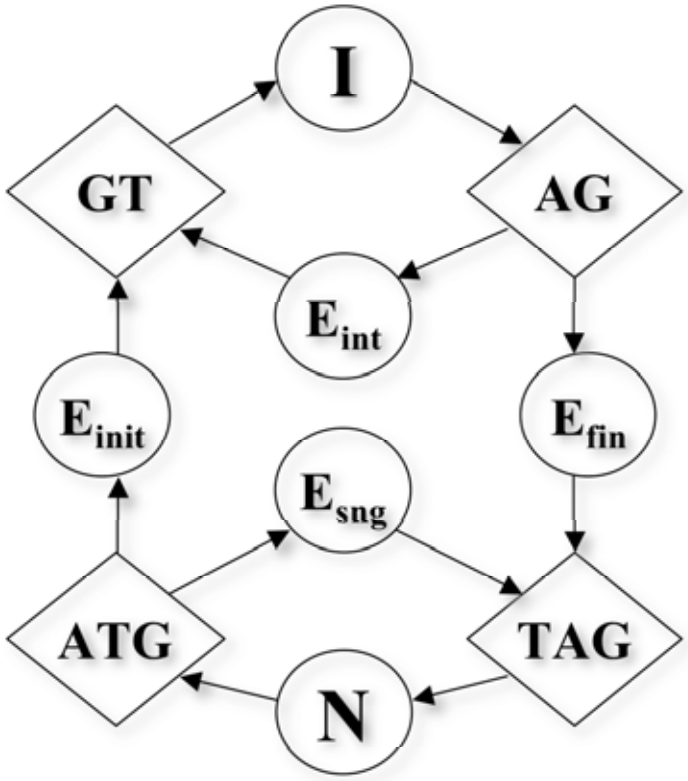
Phylogenetic HMM's (PhyloHMM's)

model of phylogeny



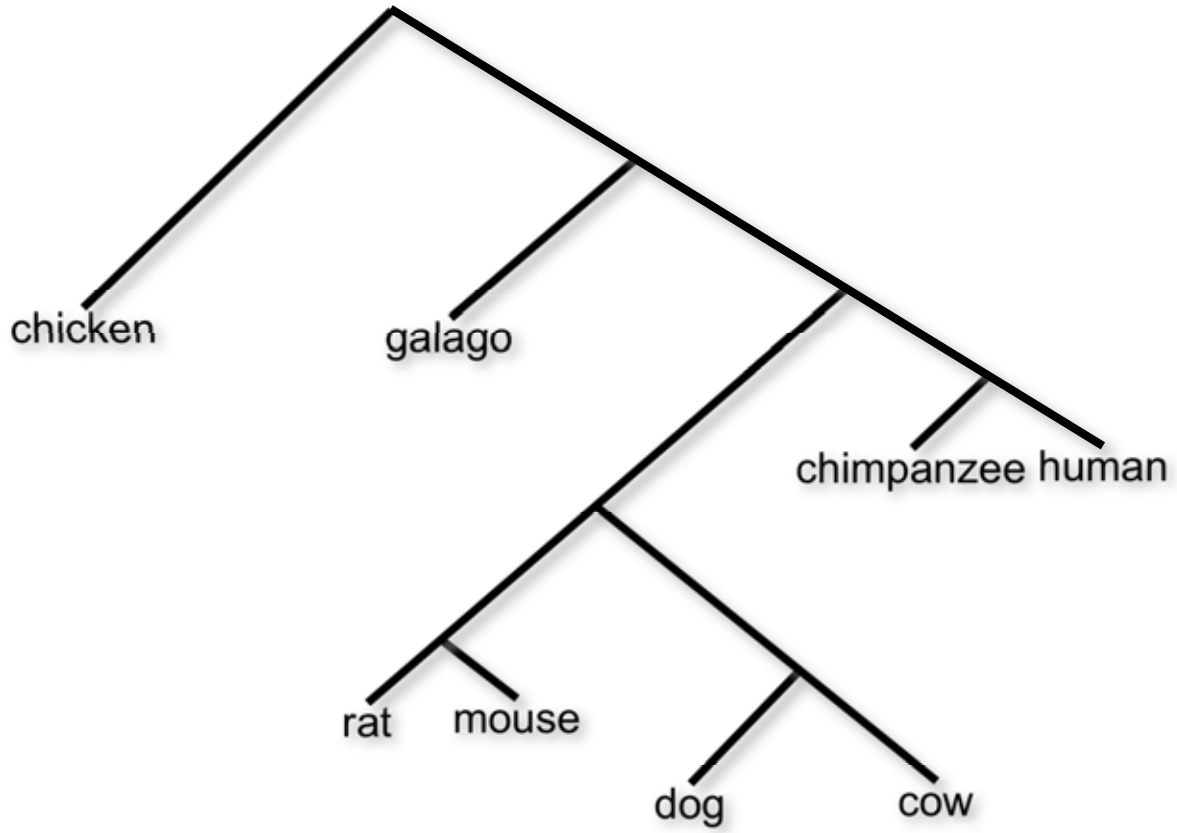
+

model of gene structure



= a model of gene structure informed by observed evolutionary divergence

Evolutionary Sequence Conservation



- Using multiple genomes increases effective sample size
- However, we have to control for the non-independence of informant genomes
- The “ideal evolutionary distance” for informant genomes usually is not known

```
human:  AAGGGAAGACAGGTGAGGGTCAAGCCCCAGCAAGTGCACCCAG----------ACACC
chimp:  AAGGGAAGACAGGTGAGGGTCAAGCCCCAGCAAGTGCACCCAG----------ACACC
cow:    AAGGGAAGACATTTACGAGTCAAGCCACAGAAAGAGCCCTGAG----------GTGCC
dog:    AAAGGAGGACATGTGAGGGCCAAACTACTGAAGGTTCAACCAGG----------ATGCT
galago: AAGGGGAGACAGGGGAGGGTCACACCATGGCAGAGG--CCAAG----------ACAGC
rat:    AAAGGAAACAATGGGAAGGTTA-TCAACTCCAAGTATGCCCAAGATCAAGGGAACCCCTT
mouse:  AAAGGAAACCACTGGGAGGTTA-GAAATCACAGGTGCACCCAAGATCAAGGAA--CCCCT
```

Decoding with a PhyloHMM

$$\phi^* = \arg\max_{\phi} P(\phi \mid S, I^{(1)}, \ldots, I^{(n)})$$

$$= \arg\max_{\phi} \frac{P(\phi, S, I^{(1)}, \ldots, I^{(n)})}{P(S, I^{(1)}, \ldots, I^{(n)})}$$

$$= \arg\max_{\phi} P(\phi, S, I^{(1)}, \ldots, I^{(n)})$$

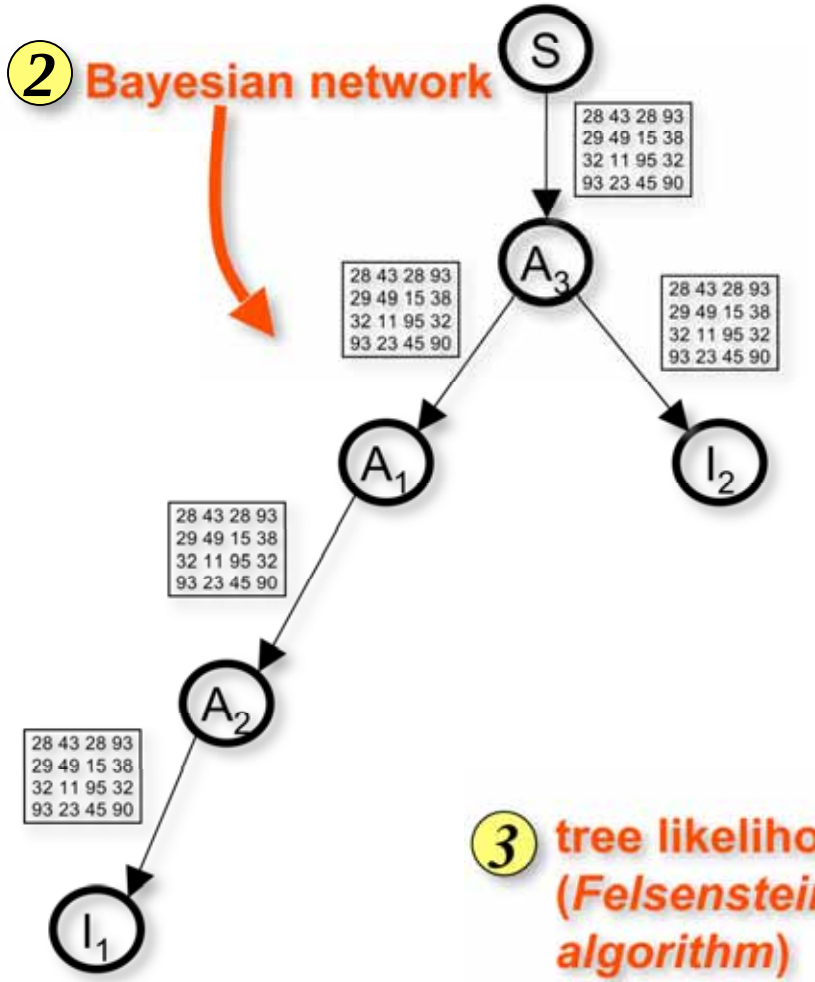
$$= \arg\max_{\phi} P(\phi) P(S, I^{(1)}, \ldots, I^{(n)} \mid \phi)$$

$$= \arg\max_{\phi} P(\phi) P(S \mid \phi) P(I^{(1)}, \ldots, I^{(n)} \mid S, \phi)$$

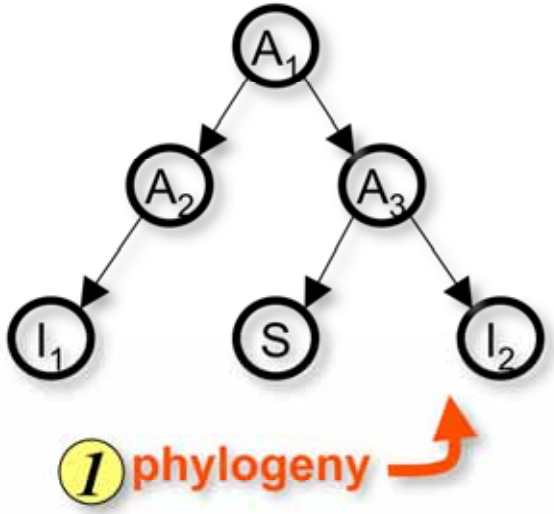
standard GHMM computation

tree likelihood (Felsenstein's algorithm)

Phylogenies as Bayes Networks



re-root and attach rate matrices



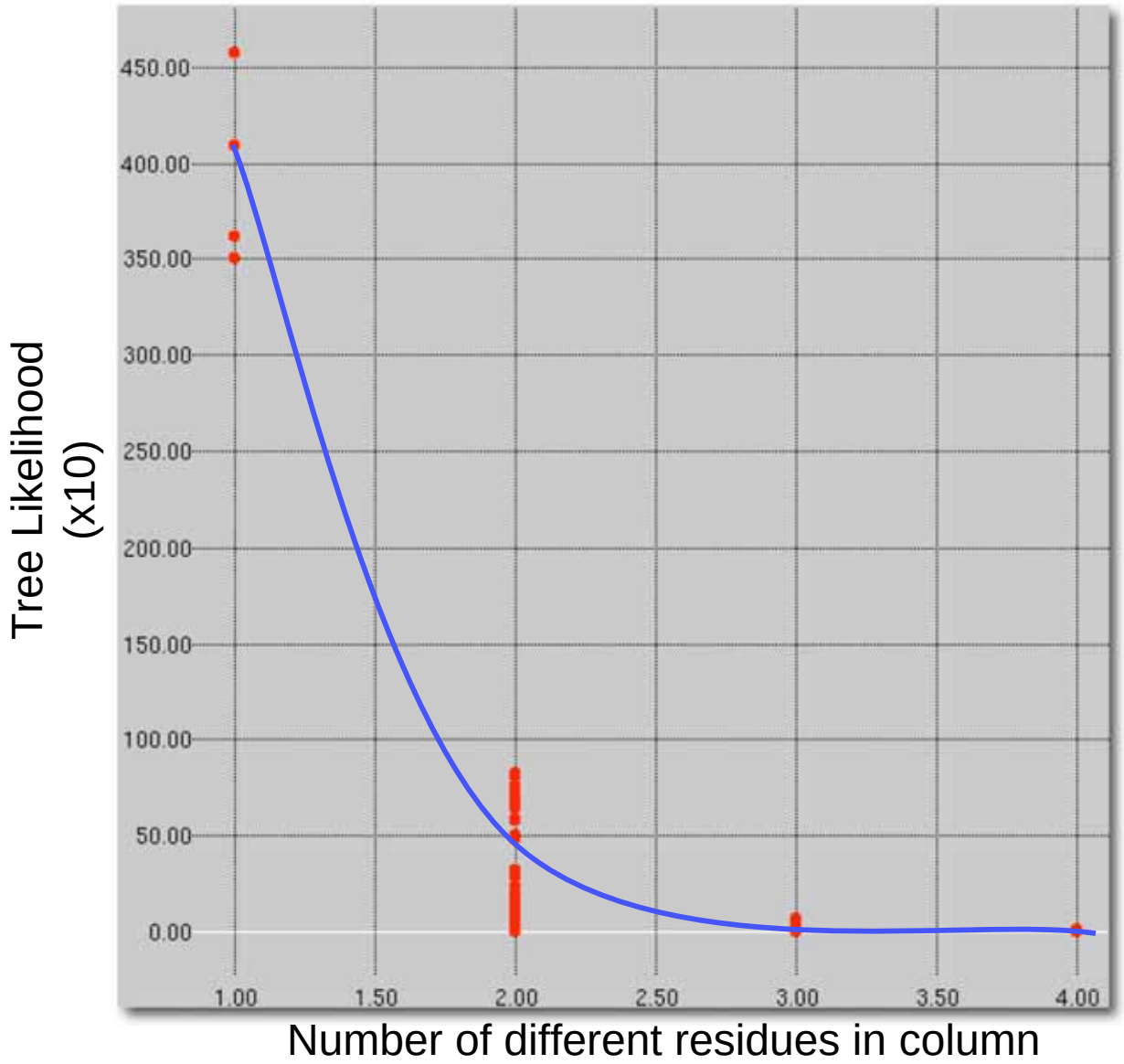
$$P(I^{(1)},\ldots,I^{(n)} \mid S,\phi) =$$

3 tree likelihood (Felsenstein's algorithm)

$$\sum_{unobservables} \left(\prod_{\substack{\text{nonroot} \\ v}} P(v \mid parent(v),\phi) \right)$$

$$= \sum_{A_1,A_2,A_3} P(I_1 \mid A_2)P(A_2 \mid A_1)P(A_1 \mid A_3)P(I_2 \mid A_3)P(A_3 \mid S)$$

Likelihood rapidly decreases with larger numbers of mutations



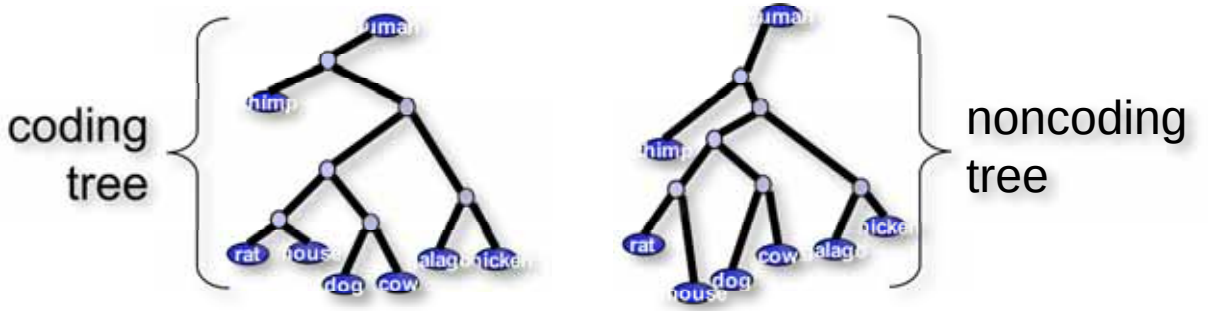
(five species aligned over 5000 columns)

Modeling Evolutionary Change in both Nucleotides and Amino Acids

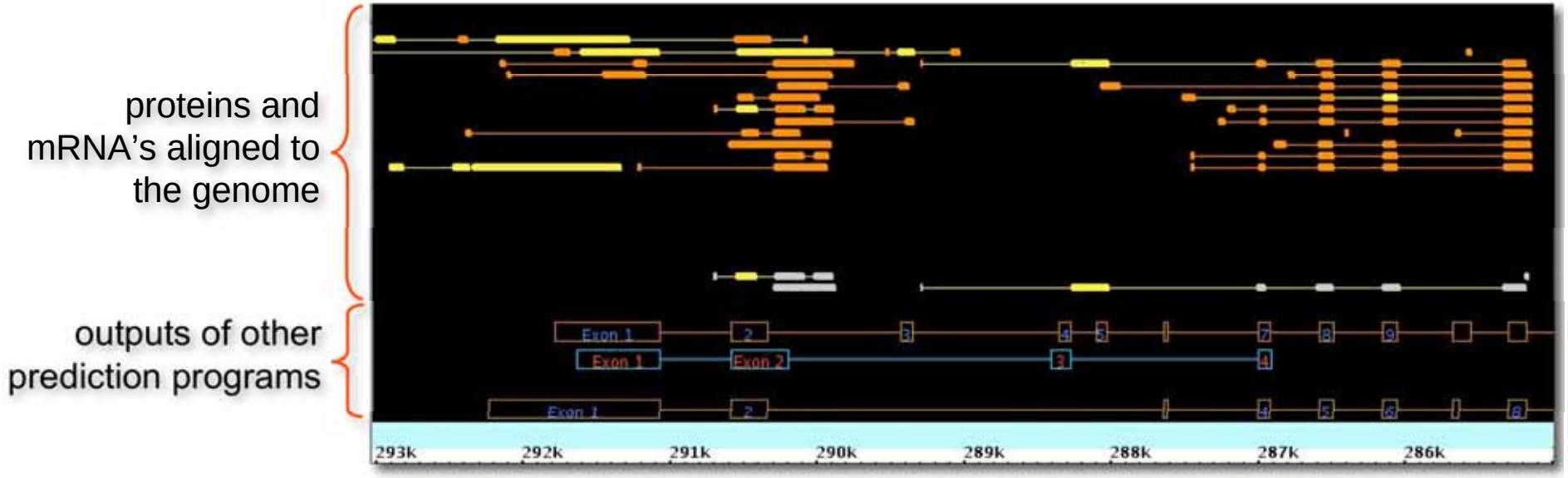
Recall from the earlier GHMM example:

feature	amino acid alignment score	<,>	nucleotide alignment score
exon 1	100%	>	71%
intron 1	14%	<	51%
exon 2	98%	>	85%
intron 2	29%	<	49%
exon 3	97%	>	82%
intron 3	9%	<	49%
exon 4	96%	>	83%

Thus, we model separately the rate of evolutionary change in coding regions (ideally at the amino acid level) and noncoding regions (at the nucleotide level).



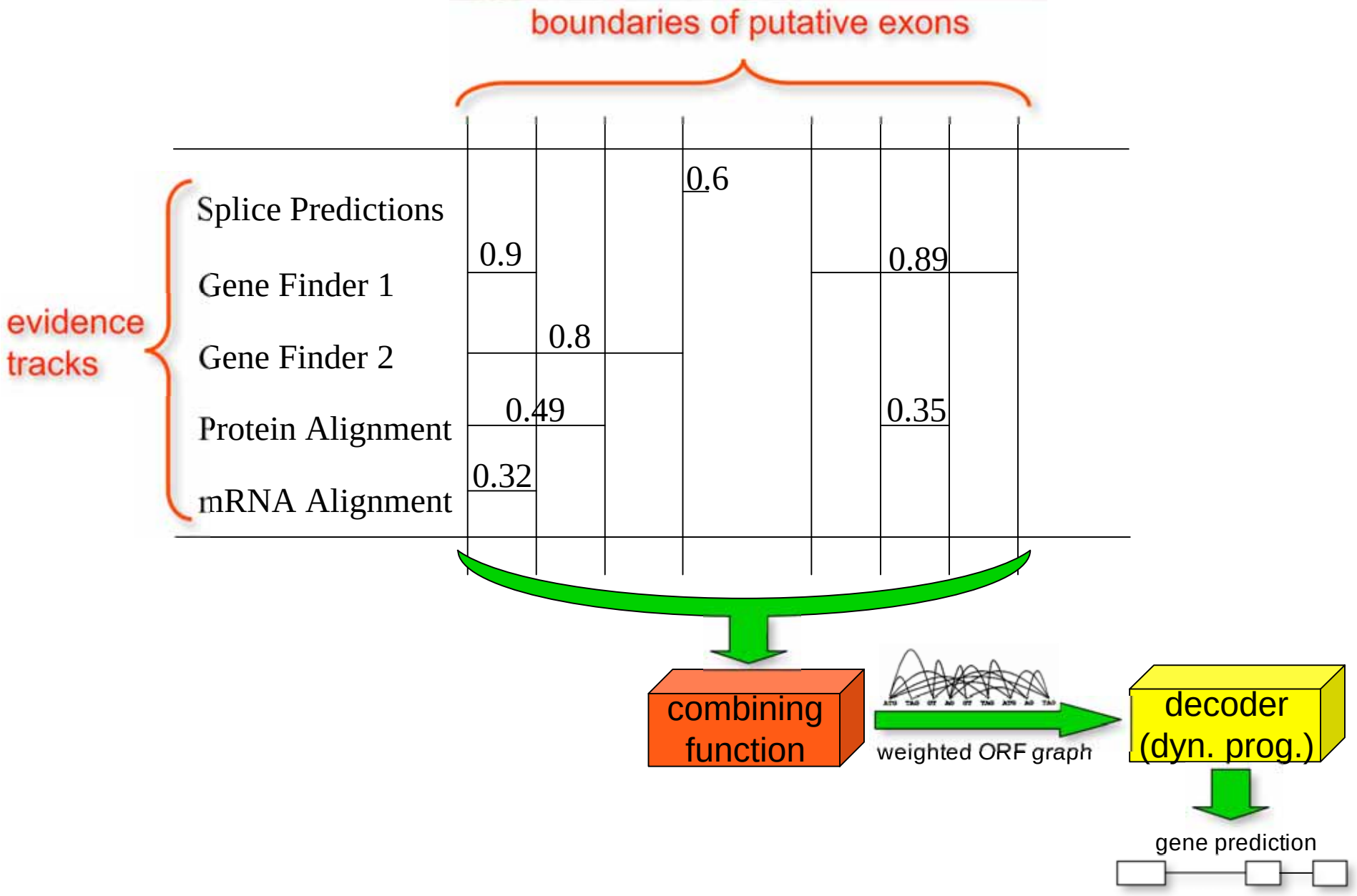
Expression-based Methods



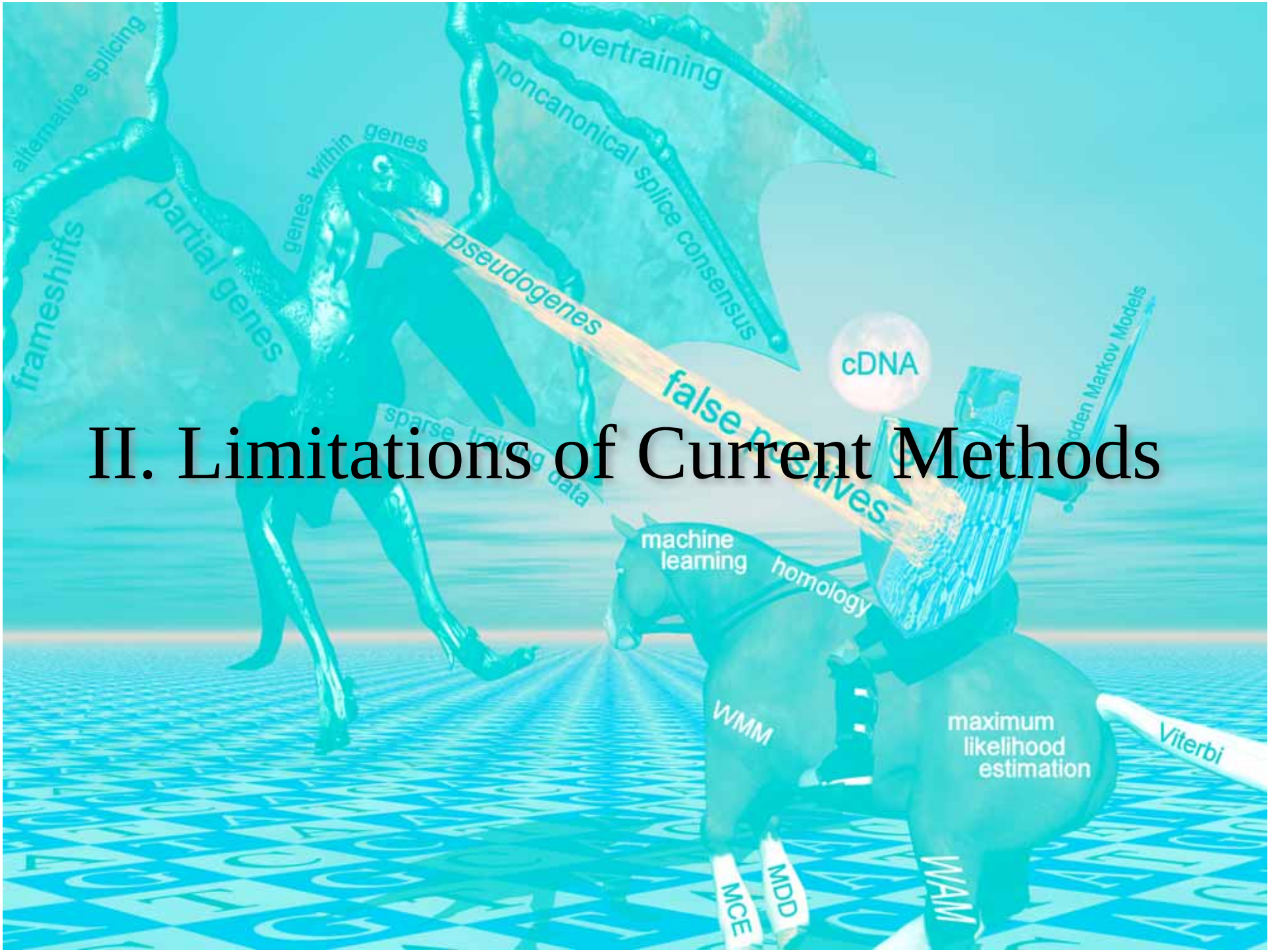
- Proteins and sequenced mRNA's can be aligned to the genome using dynamic programming algorithms.
- Various *ad hoc* methods have been explored for utilizing this information during gene prediction.
- Outputs from other gene finders can also be used as input to a "combiner" program.

(figure courtesy of J. Allen, University of Maryland)

Ad hoc "Combiner" Methods



(J. Allen, University of Maryland)



II. Limitations of Current Methods

1. MLE+Viterbi Is Not Optimal

Currently, most gene finders are trained via maximum likelihood estimation (MLE):

$$\theta^* = \arg\max_{\theta} \left(\prod_{(S,\phi)\in T} P(S,\phi \mid \theta) \right)$$

$$= \arg\max_{\theta} \left(\prod_{(S,\phi)\in T} \prod_{i=1}^{N-1} P_\theta(\pi_i \mid q_i, d_i) P_\theta(q_i \mid q_{i-1}) P_\theta(d_i \mid q_i) \right)$$

The advantage of MLE is that it is easy to perform, since the transition, emission, and duration parameters can be optimized separately:

$$\sum_{(S_i \phi_i)\in T} \log \prod_{q_j \in \phi_i, j>0} P(q_j \mid q_{j-1}) \qquad \text{(transitions)}$$

$$\sum_{\pi_j \in S_i \in T} \log P(\pi_j \mid q) \qquad \text{(emissions)}$$

$$\sum_{(S_i \phi_i)\in T} \log \prod_{(q_j, d_j)\in \phi_i} P(d_j \mid q_j) \qquad \text{(durations)}$$

anecdotal evidence

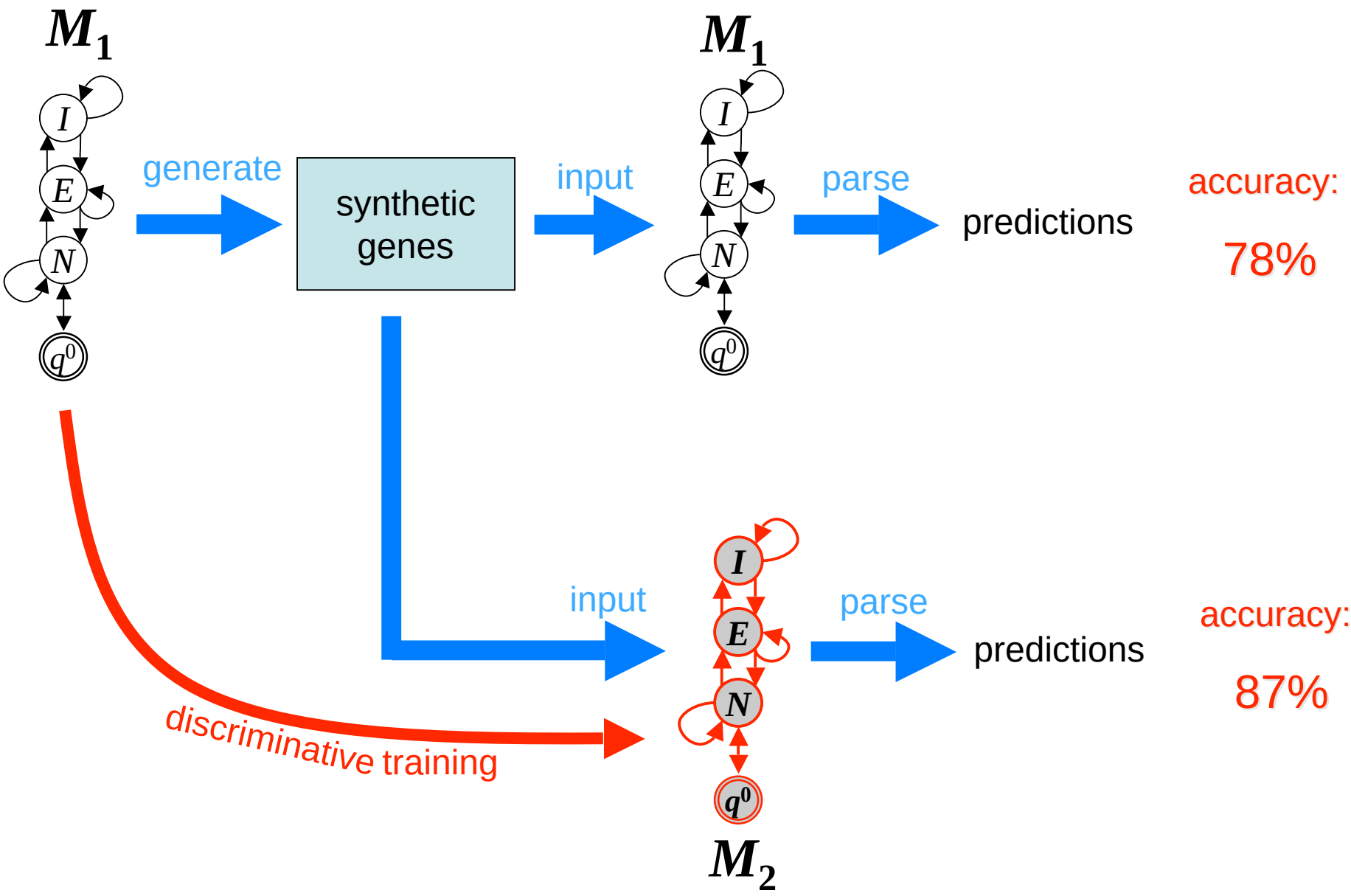
Anecdotal evidence:

- folklore about $1/\sqrt{3}$ in the source code of a certain popular gene finder*
- fudge factor in: NSCAN (“conservation score coefficient”; Gross & Brent, 2005)
- fudge factor in: ExoniPhy (“tuning parameter”; Siepel & Haussler, 2004)
- fudge factor in TWAIN (“percent identity”; Majoros *et al.*, 2005)
- fudge factor in GlimmerHMM (“optimism”; M. Pertea, *pers. communication*)
- fudge factor in TIGRscan (“optimism”; Majoros *et al.*, 2004)
- *lack* of fudge factors in EvoGene (Pedersen & Hein, 2003)

If MLE+Viterbi produced optimal parsers, why would “tweaking” & the use of “fudge factors” be necessary? (...apart from sample size issues...)

** folklore also states that this programs’s author made pact with the devil in exchange for gene-finding accuracy, but I have encountered no independent verification of this.*

a simple experiment



changes selected by the hill-climber

changes to the emission parameters

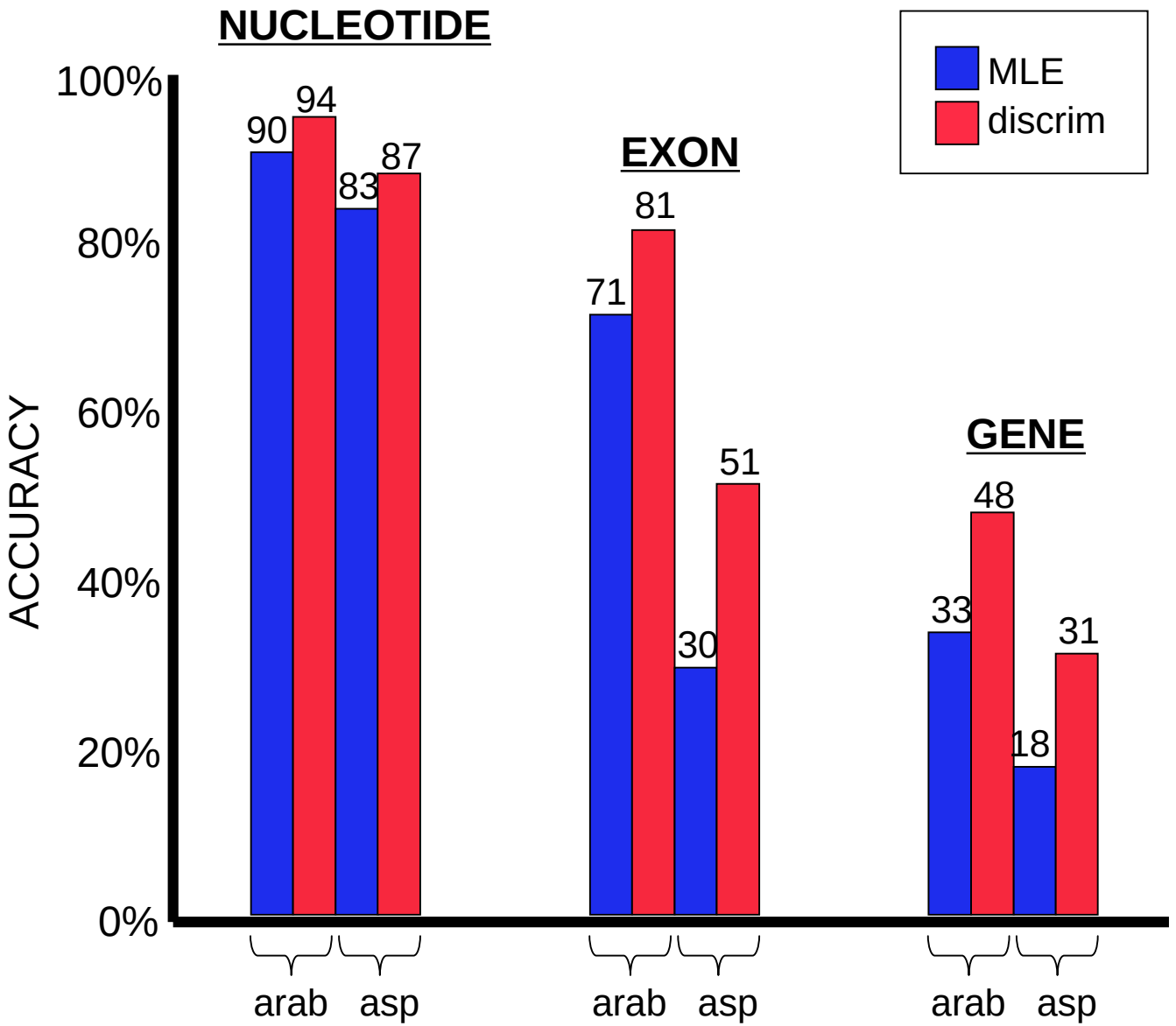
		symbol			
state		A	C	G	T
	inter-genic	–12%	+5%	–3%	+10%
	exon	+1%	+2%	+3%	–6%
	intron	+3%	–1%	–6%	+4%

changes to the transition parameters

		to state			
from state		0	1	2	3
	0	·	·	·	·
	1	·	–1.5%	+1.5%	·
	2	·	+0.4%	–0.8%	+0.4%
	3	·	·	+0.9%	–0.9%

"These changes are consistent with the notion that superior discrimination requires exaggeration of (or emphasis on) those model parameters which most reliably separate the classes of interest. In this case, the *Arabidopsis* training genes showed both lower T content and higher C and G content in the exonic regions versus the other two regions, and these biases, which were already present in the same proportions in the MLE model (data not shown) are obviously exaggerated in the discriminative model (Table 3). The most extreme exaggeration, the -12% for A in the intergenic state, is among the most subtle differences in the training data, in which the intergenic regions showed ~29% A versus ~27% in the exonic and intronic regions; in this way, the discriminative trainer has identified a subtle but consistent difference and boosted the "weight" of this feature in the model by exaggerating the corresponding emission probabilities."

another experiment: GHMM



what about *conditional* maximum likelihood?

Instead of: $\theta^* = \arg\max_{\theta} \left(\prod_{(S,\phi)\in T} P(S,\phi \mid \theta) \right)$ we might consider: $\theta^* = \arg\max_{\theta} \left(\prod_{(S,\phi)\in T} P(\phi \mid S,\theta) \right)$

This is called *Conditional Maximum Likelihood*. It expands into:

$$\theta^* = \arg\max_{\theta} \left(\prod_{(S,\phi)\in T} P(\phi \mid S,\theta) \right)$$

$$= \arg\max_{\theta} \left(\prod_{(S,\phi)\in T} \frac{P(S,\phi \mid \theta)}{P(S \mid \theta)} \right)$$

$$= \arg\max_{\theta} \left(\prod_{(S,\phi)\in T} \frac{\prod_{i=1}^{N-1} P_\theta(\pi_i \mid q_i, d_i) P_\theta(q_i \mid q_{i-1}) P_\theta(d_i \mid q_i)}{P_\theta(S)} \right)$$

Unfortunately, EM-like update equations which have been derived for pure HMM's tend to be unstable (e.g., Reichl and Ruske, 1995; Normandin, 1996). Update equations for GHMM's, PHMM's, GPHMM's, and PhyloHMM's have (as far as I know) not yet been derived.

2. Newer Decoders Also Not Optimal

Posterior Viterbi (Fariselli *et al.*, 2005), OAD (Käll *et al.*, 2005):

$$\underset{\phi}{argmax} \prod_{\lambda_i \in \phi} P_e(\lambda_i \mid s_i)\delta(q_i \mid q_{i-1}) \qquad \underset{\phi}{argmax} \sum_{\lambda_i \in \phi} P_e(\lambda_i \mid s_i)\delta(q_i \mid q_{i-1})$$

200 training genes, 50 test genes

algorithm	*nuc*	*exon*	*gene*
A_{Vit}	85%	42%	24%
A_{PV}	89%	41%	26%
A_{OAD}	89%	41%	26%

(*Oryza sativa*)

1000 training genes, 200 test genes

algorithm	*nuc*	*exon*	*gene*
A_{Vit}	96%	68%	36%
A_{PV}	97%	64%	33%
A_{OAD}	97%	61%	30%

(*Arabidopsis thaliana*)

200 training genes, 50 test genes

algorithm	*nuc*	*exon*	*gene*
A_{Vit}	94%	50%	30%
A_{PV}	96%	39%	22%
A_{OAD}	96%	39%	22%

(*Aspergillus fumigatus*)

...even for GHMM's

G/OAD and G/PV dynamic programming update formulas:

$$W_j = \begin{cases} 0 & \text{if } deg_{in}(q_j) = 0 \\ \max\limits_{i \in pred(j)} W_i + f_i b_j \lambda_{i \to j} P(s_j \mid q_j) & \text{otherwise} \end{cases}$$

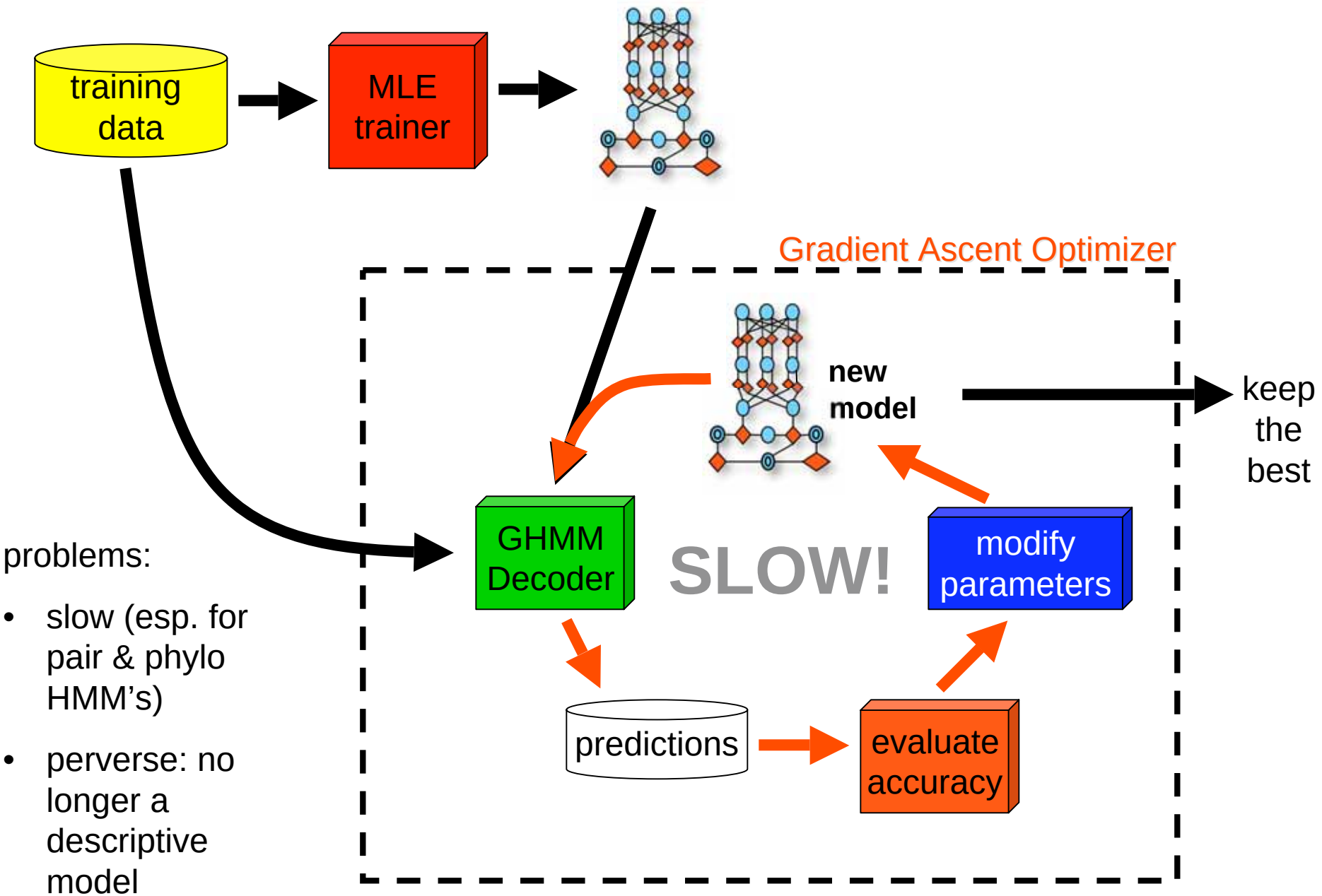
$$V_j = \begin{cases} 1 & \text{if } deg_{in}(q_j) = 0 \\ \max\limits_{i \in pred(j)} V_i f_i b_j \lambda_{i \to j} P(s_j \mid q_j) & \text{otherwise} \end{cases}$$

decoding system	nuc	exon	gene
$(\xi_{MLE}, \mathcal{A}_{Vit})$	97%±1%	80%±1%	48%±3%
$(\xi_{MLE}, \mathcal{A}_{GPV})$	96%±0%	79%±2%	48%±3%
$(\xi_{MLE}, \mathcal{A}_{GOAD})$	98%±0%	79%±1%	47%±2%
$(\xi_{\mathcal{D}(Vit)}, \mathcal{A}_{Vit})$	98%±0%	83%±1%	51%±3%
$(\xi_{\mathcal{D}(GPV)}, \mathcal{A}_{GPV})$	98%±0%	83%±3%	53%±3%
$(\xi_{\mathcal{D}(GOAD)}, \mathcal{A}_{GOAD})$	98%±1%	83%±3%	51%±5%

5-fold cross validation; (600 training genes & 200 test genes per run)

(*Arabidopsis thaliana*)

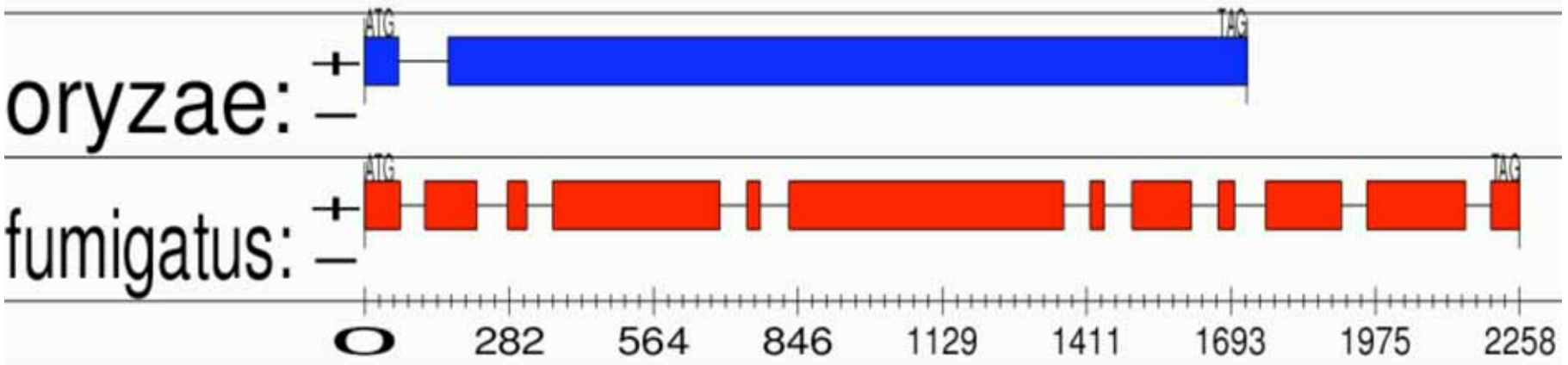
3. Discriminative Training is Expensive



problems:

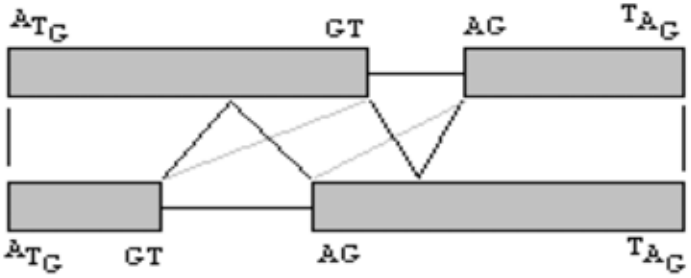
- slow (esp. for pair & phylo HMM's)
- perverse: no longer a descriptive model

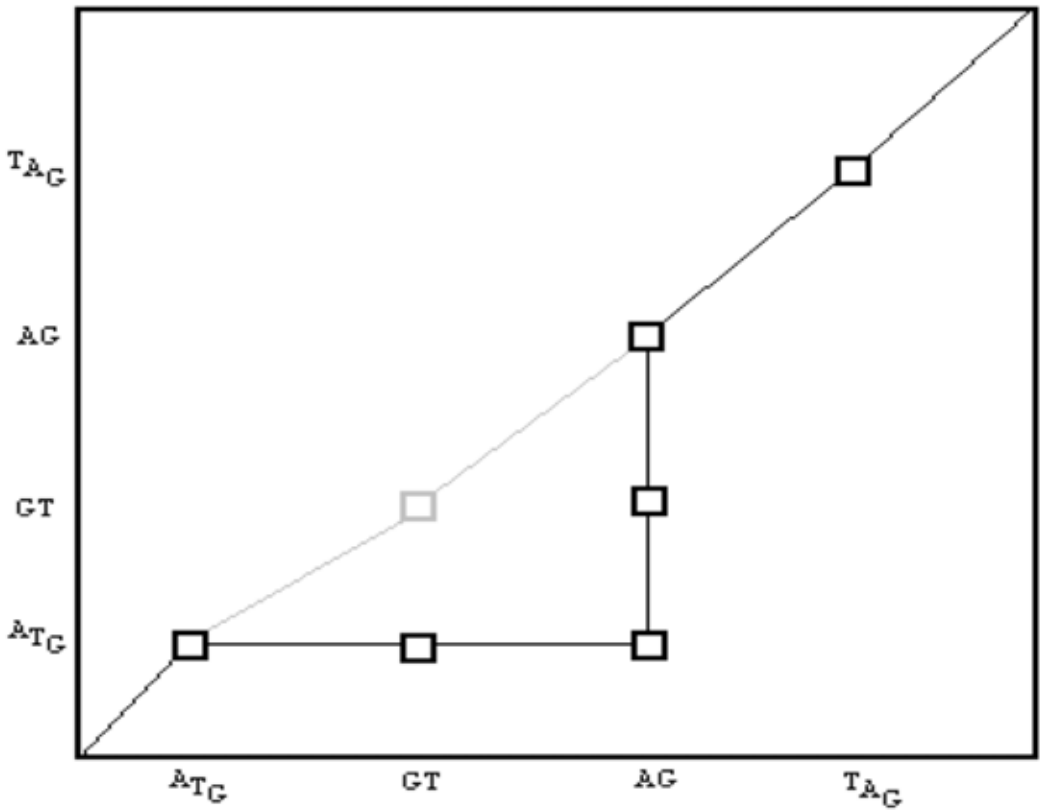
4. Intron Evolution is Rarely Modeled



Many *Aspergillus* orthologs have unequal numbers of exons.

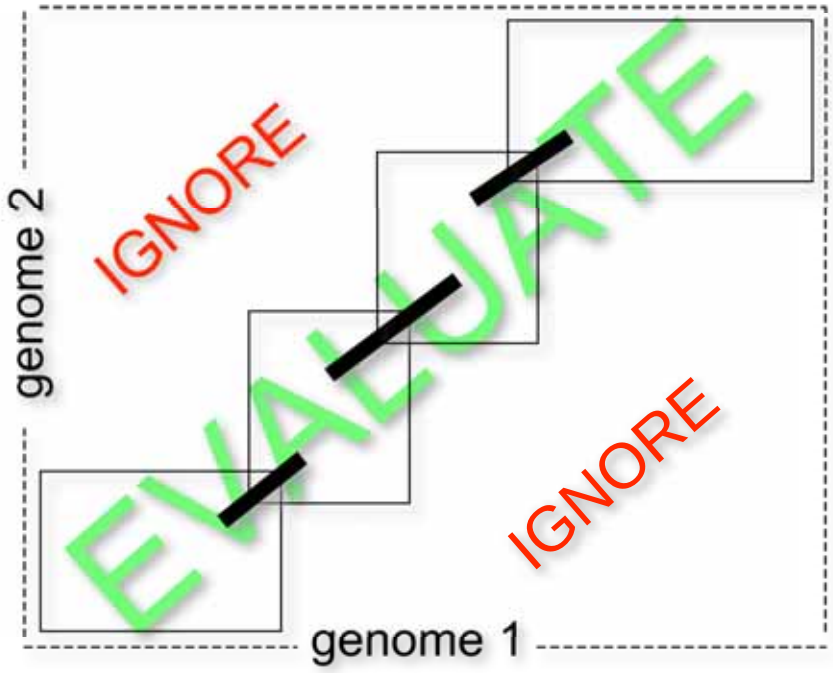
(intron evolution, cont.)



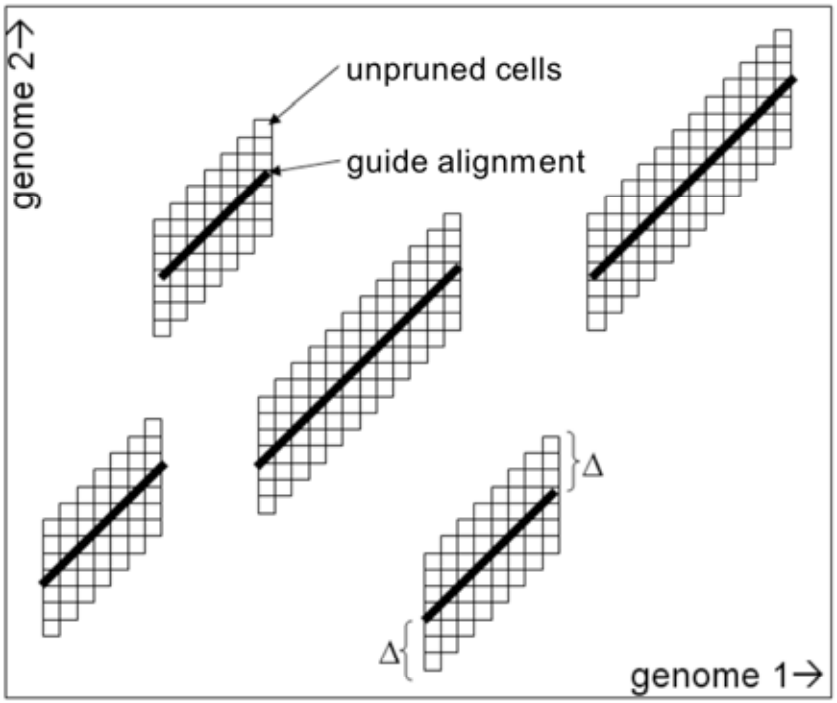


5. Dependence on Pre-computed Alignments

The PHMM case:

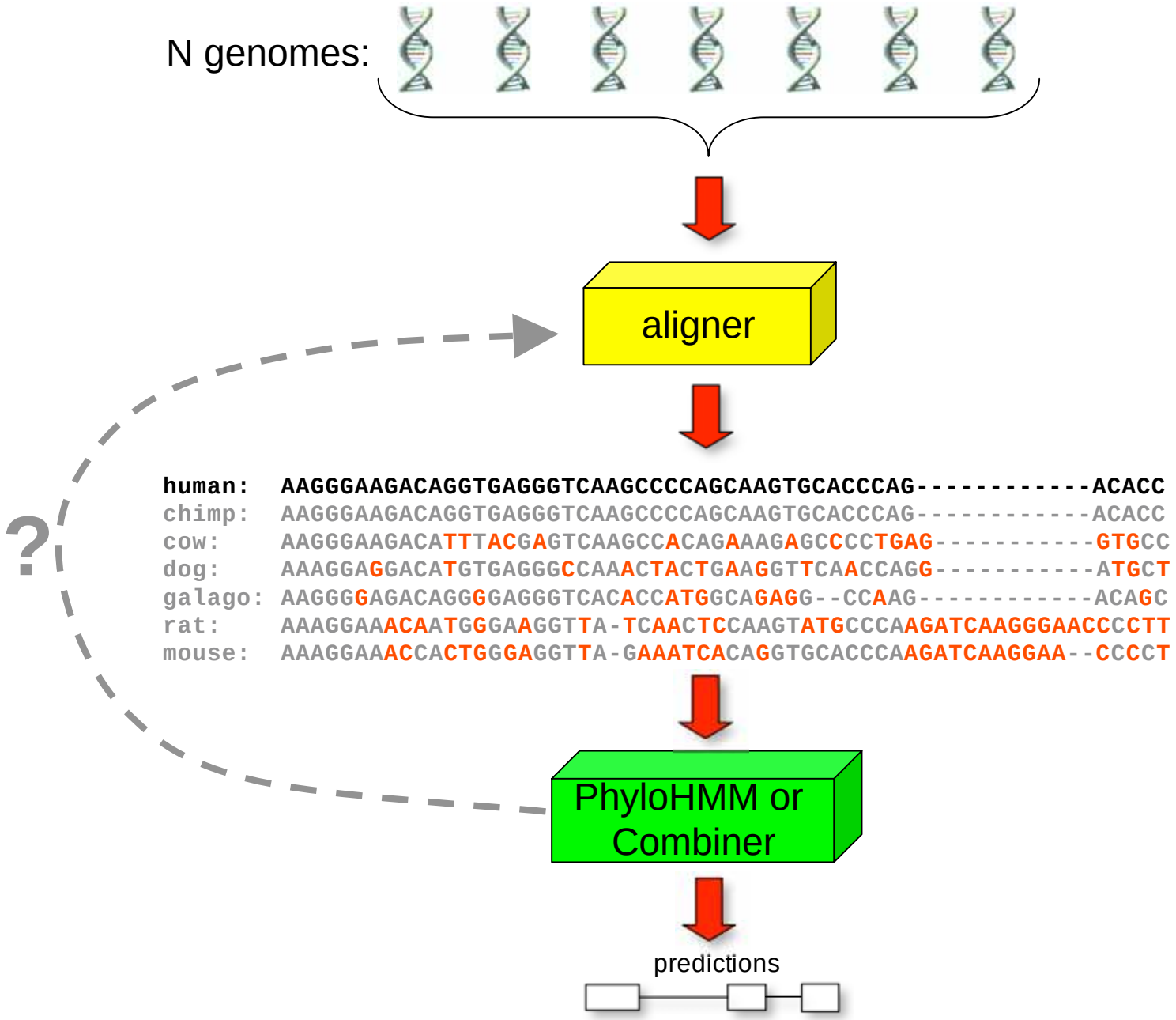


The GPHMM case:

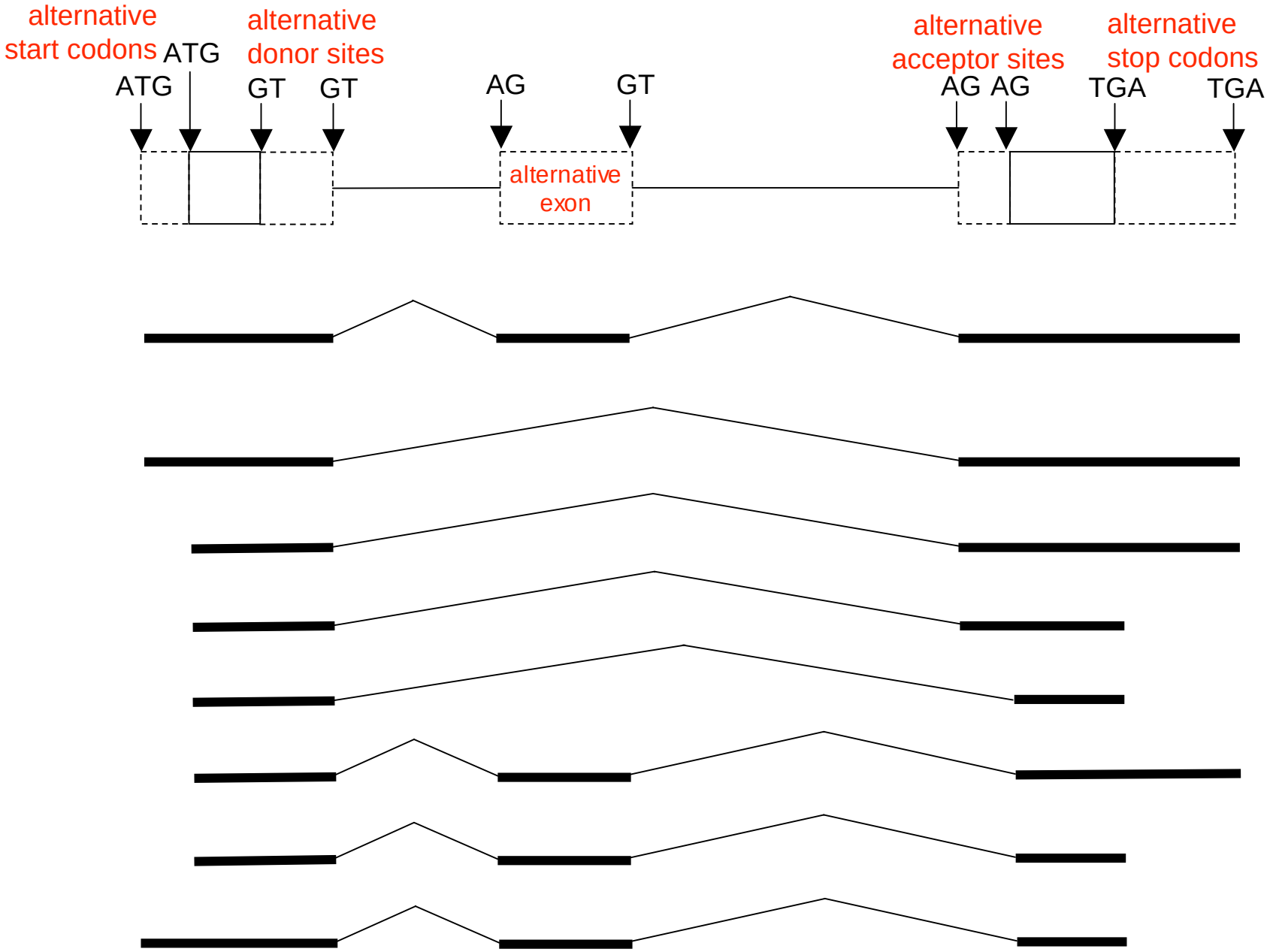


As for the PhyloHMM & Combiner cases...

(pre-computed alignments, cont.)



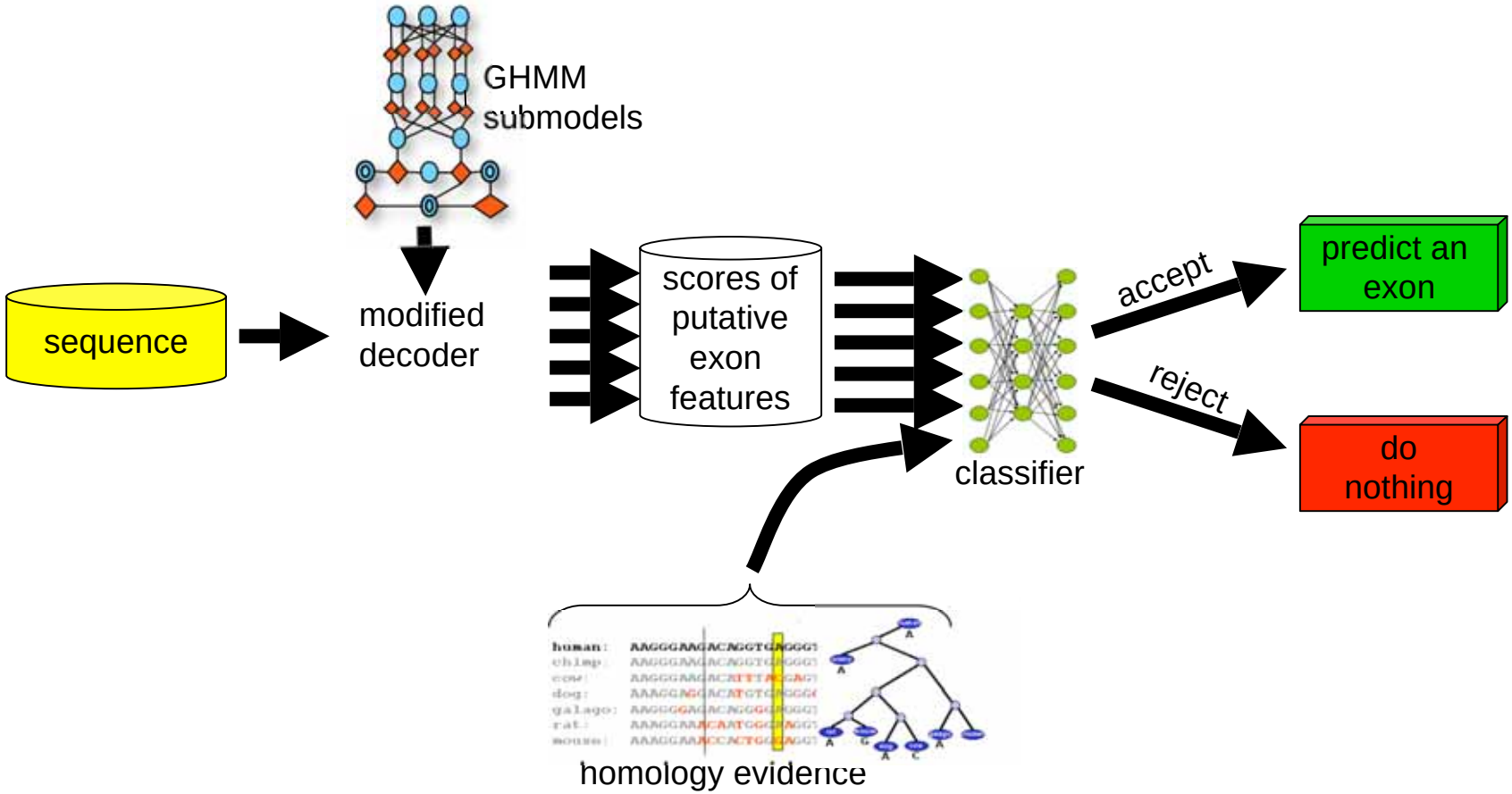
6. Alternative Splicing



III. Future Directions

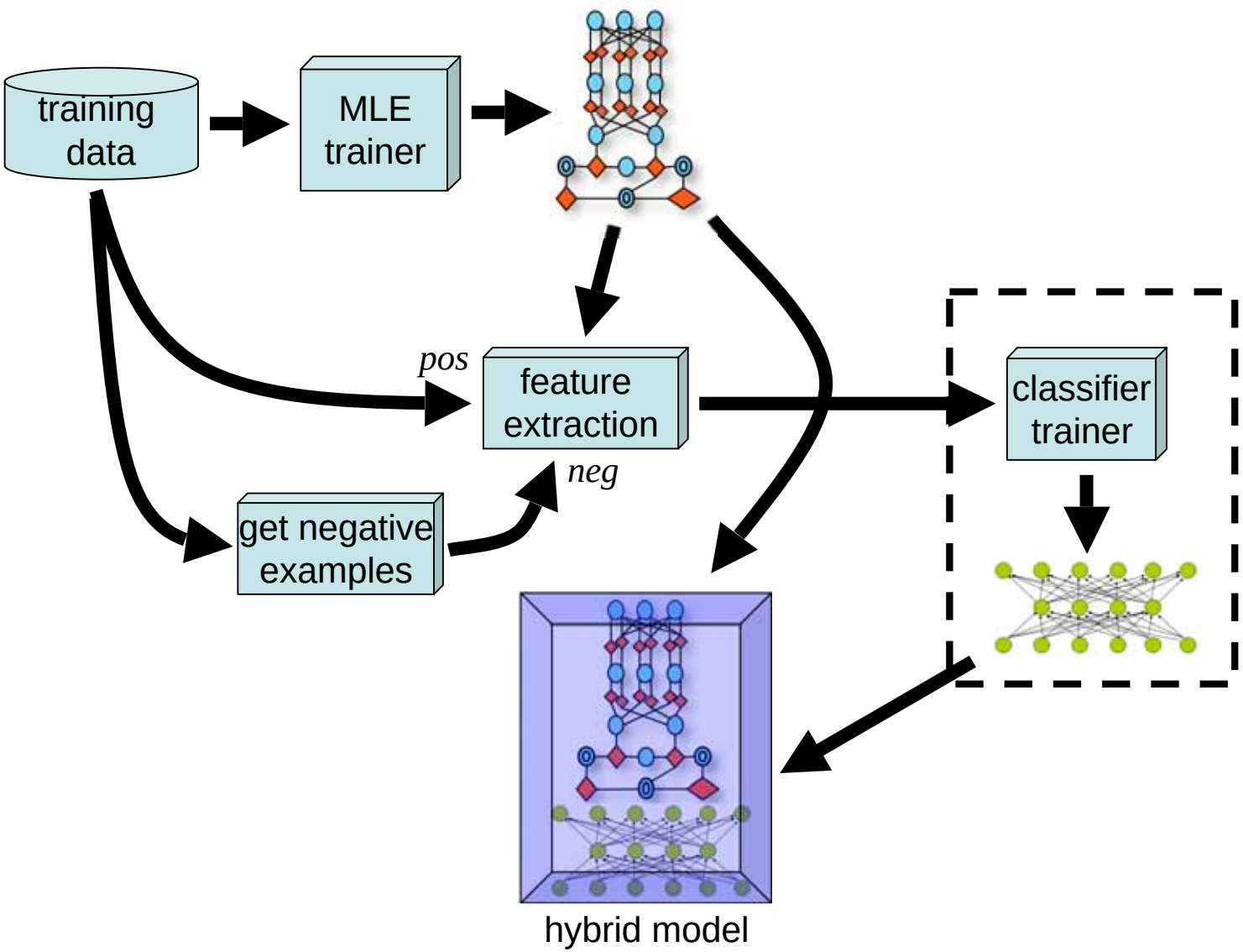
1. Redefine the Problem

- Predict exons rather than whole genes -- a step backward?
- Classification rather than parsing
- This “solves” the alternative splicing problem (in a manner of speaking...)
- Possibly use a statistical ensemble representation of parses, and develop more sophisticated browsers that can use such a representation



2. A Greater Role for Machine Learning

- Use established, general-purpose ML algorithms
- Let the learner attend to the goal of maximal discrimination



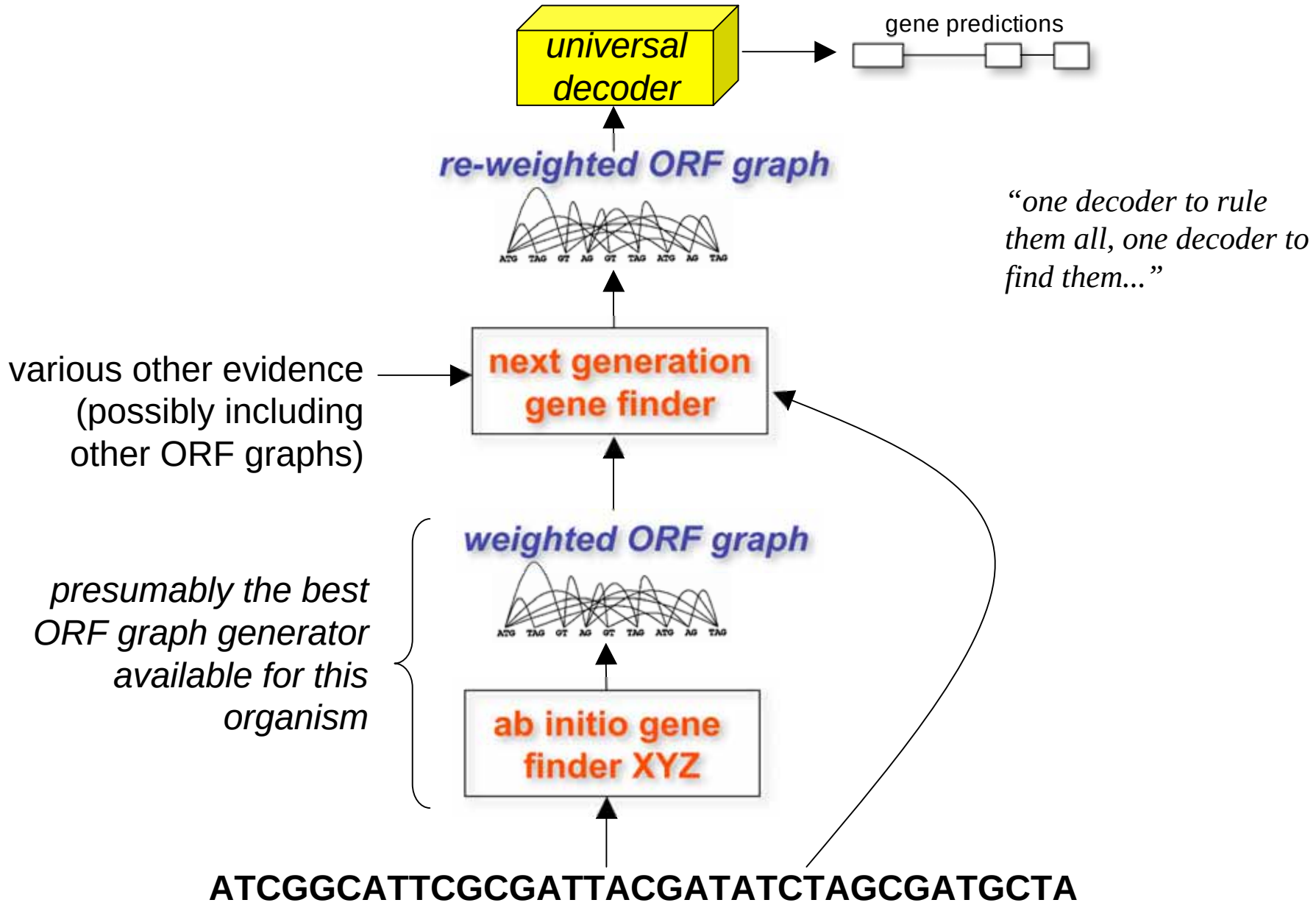
3. Focus on Combiners

- Combiners integrate all available evidence
- Results from EGASP:

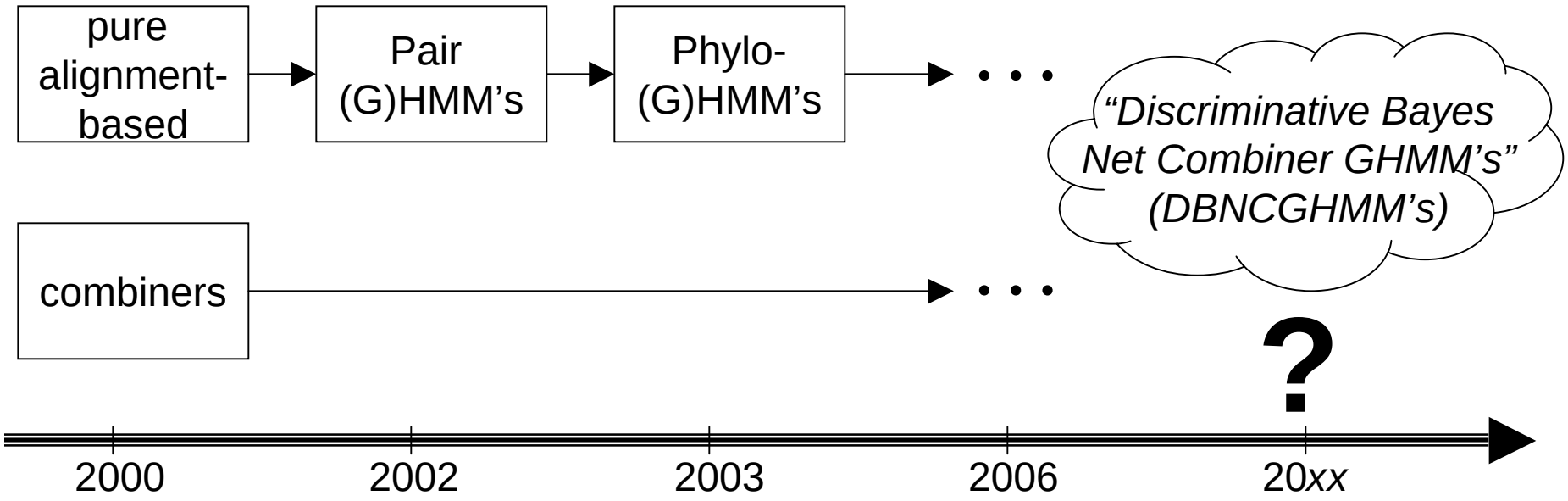
	NUCLEOTIDE			EXON		GENE	
	NS*n*	NS*p*	N *CC*	ES*n*	ES*p*	GS*n*	GS*p*
Combiners							
AUGUSTUS-any	94.42%	82.43%	0.88	74.67%	76.76%	47.97%	35.59%
FGENESH++	91.09%	76.89%	0.83	75.18%	69.31%	69.93%	42.09%
JIGSAW	94.56%	92.19%	0.93	80.61%	89.33%	72.64%	65.95%
PAIRAGON-any	87.77%	92.78%	0.90	76.85%	88.91%	69.59%	61.32%
HMM's & GHMM's							
AUGUSTUS-abinit	78.65%	75.29%	0.76	52.39%	62.93%	24.32%	17.22%
GENEMARK.hmm	78.43%	37.97%	0.53	50.58%	29.01%	15.20%	3.24%
GENEZILLA	87.56%	50.93%	0.66	62.08%	50.25%	19.59%	8.84%
Expression-based							
ACEVIEW	90.94%	79.14%	0.84	85.75%	56.98%	63.51%	48.65%
AUGUSTUS-EST	92.62%	83.45%	0.88	74.10%	77.40%	47.64%	37.01%
ENSEMBL	90.18%	92.02%	0.91	77.53%	82.65%	71.62%	67.32%
EXOGEAN	84.18%	94.33%	0.89	79.34%	83.45%	63.18%	80.82%
EXONHUNTER	90.46%	59.67%	0.73	64.44%	41.77%	21.96%	6.33%
PAIRAGON+NSCAN	87.56%	92.77%	0.90	76.63%	88.95%	69.59%	61.71%
Pair & Phylo HMM's							
AUGUSTUS-dual	88.86%	80.15%	0.84	63.06%	69.14%	26.01%	18.64%
DOGFISH	64.81%	88.24%	0.74	53.11%	77.34%	10.81%	14.61%
MARS	84.25%	74.13%	0.78	65.56%	61.65%	33.45%	24.94%
NSCAN	85.38%	89.02%	0.87	67.66%	82.05%	35.47%	36.71%
SAGA	52.54%	81.39%	0.65	38.82%	50.73%	4.39%	3.44%

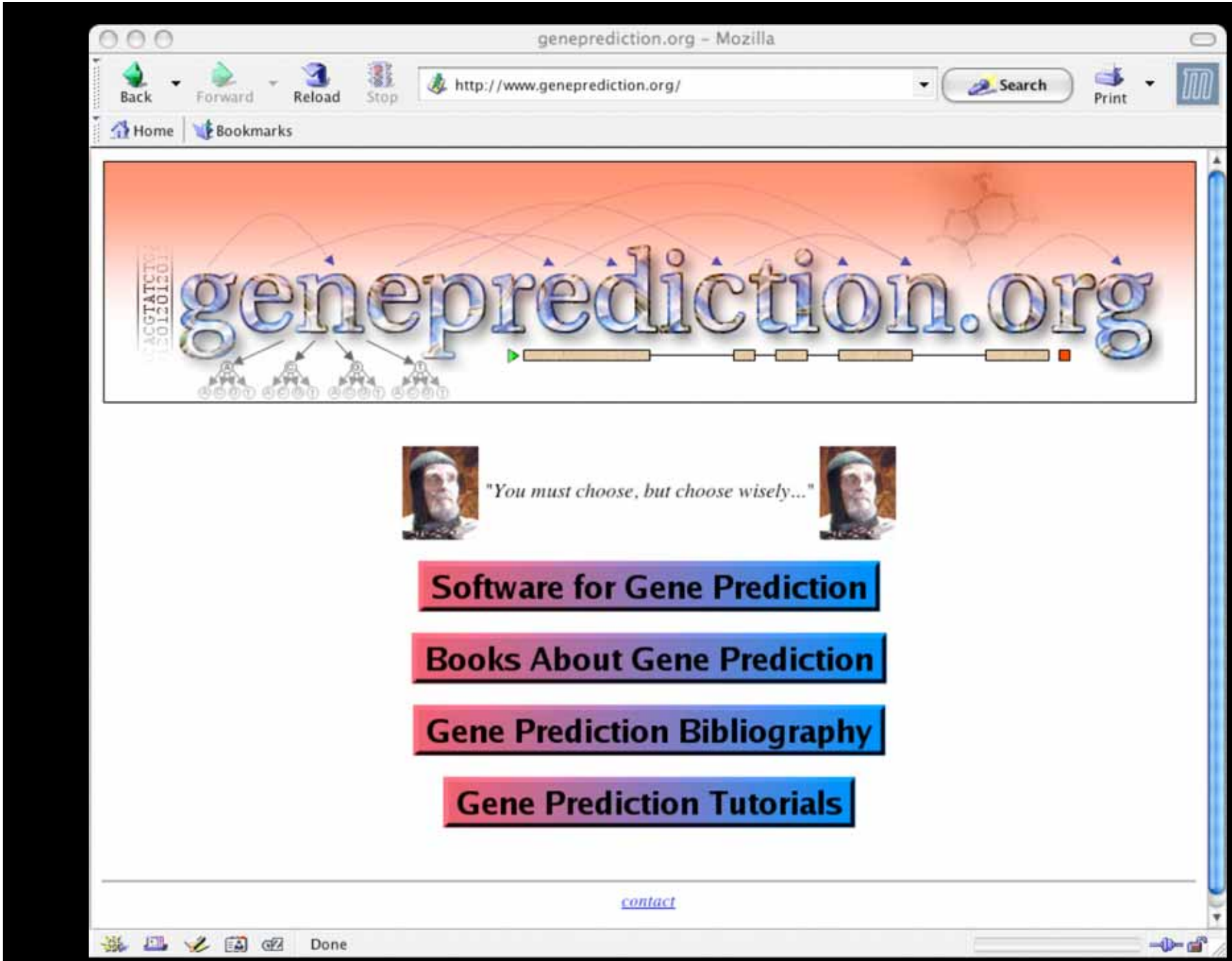
Source: Guigo et al., 2006 (to appear in Genome Biology)

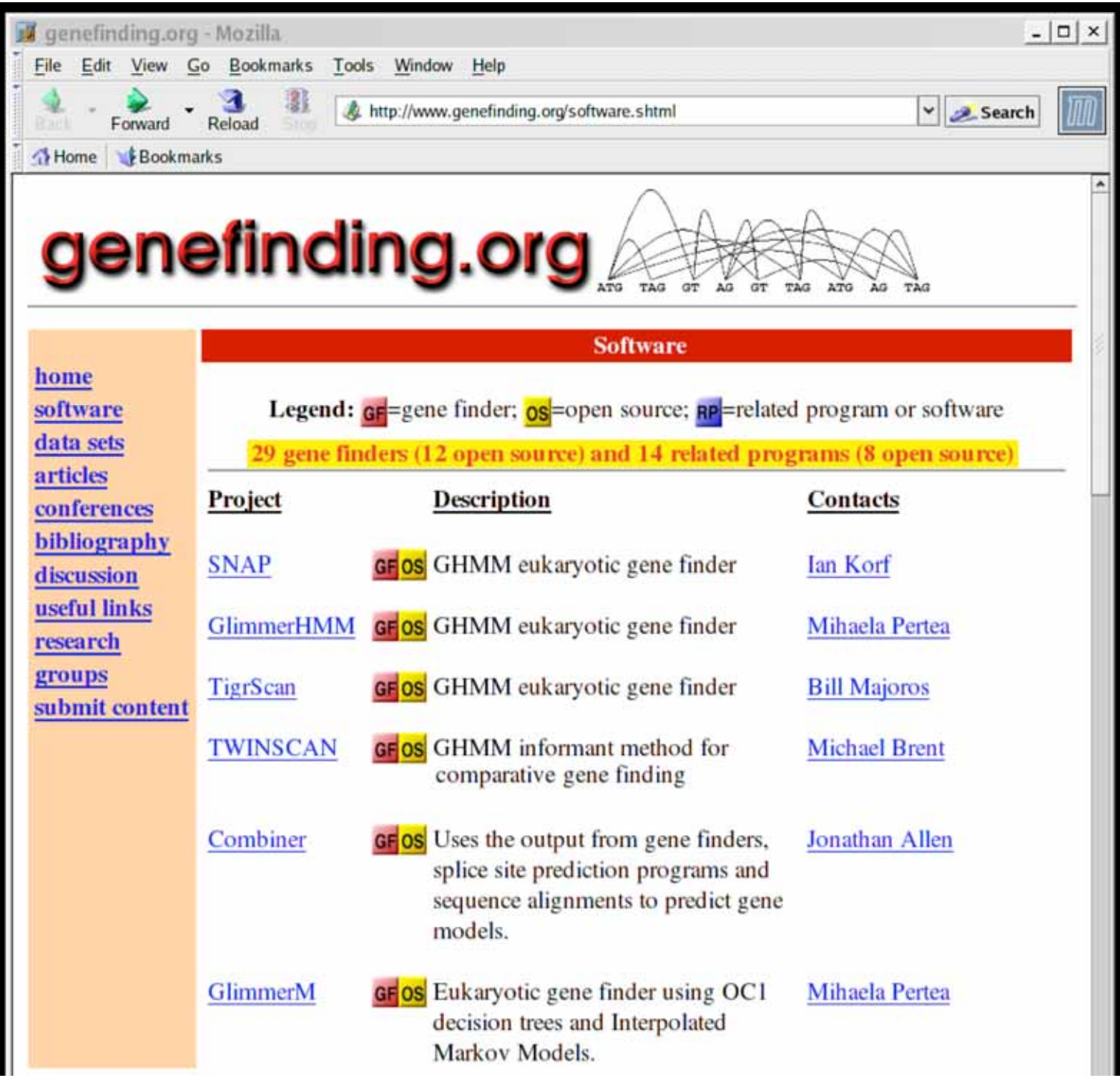
4. Interoperability



Contemplating the Future




geneprediction.org – Mozilla
Back
Forward
Reload
Stop
http://www.geneprediction.org/
Search
Print
Home
Bookmarks
geneprediction.org
"You must choose, but choose wisely..."
Software for Gene Prediction
Books About Gene Prediction
Gene Prediction Bibliography
Gene Prediction Tutorials
contact
Done


genefinding.org - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://www.genefinding.org/software.shtml
Search
Home Bookmarks
genefinding.org
ATG TAG GT AG GT TAG ATG AG TAG
home
software
data sets
articles
conferences
bibliography
discussion
useful links
research
groups
submit content
Software
Legend: GF=gene finder; OS=open source; RP=related program or software
29 gene finders (12 open source) and 14 related programs (8 open source)
Project
Description
Contacts
SNAP
GF OS GHMM eukaryotic gene finder
Ian Korf
GlimmerHMM
GF OS GHMM eukaryotic gene finder
Mihaela Pertea
TigrScan
GF OS GHMM eukaryotic gene finder
Bill Majoros
TWINSCAN
GF OS GHMM informant method for comparative gene finding
Michael Brent
Combiner
GF OS Uses the output from gene finders, splice site prediction programs and sequence alignments to predict gene models.
Jonathan Allen
GlimmerM
GF OS Eukaryotic gene finder using OC1 decision trees and Interpolated Markov Models.
Mihaela Pertea

ACKNOWLEDGEMENTS

TIGR: *S. Salzberg, M. Pertea, J. Allen, A. Delcher, J. Wortman, B. Haas*

Duke: *U. Ohler, S. Mukherjee*

Elsewhere: *M. Yandell, I. Korf, J. Eisen*

Methods for Computational Gene Prediction

W.H. Majoros

Expected publication date:

late 2006

www.geneprediction.org/book