

Acknowledgements

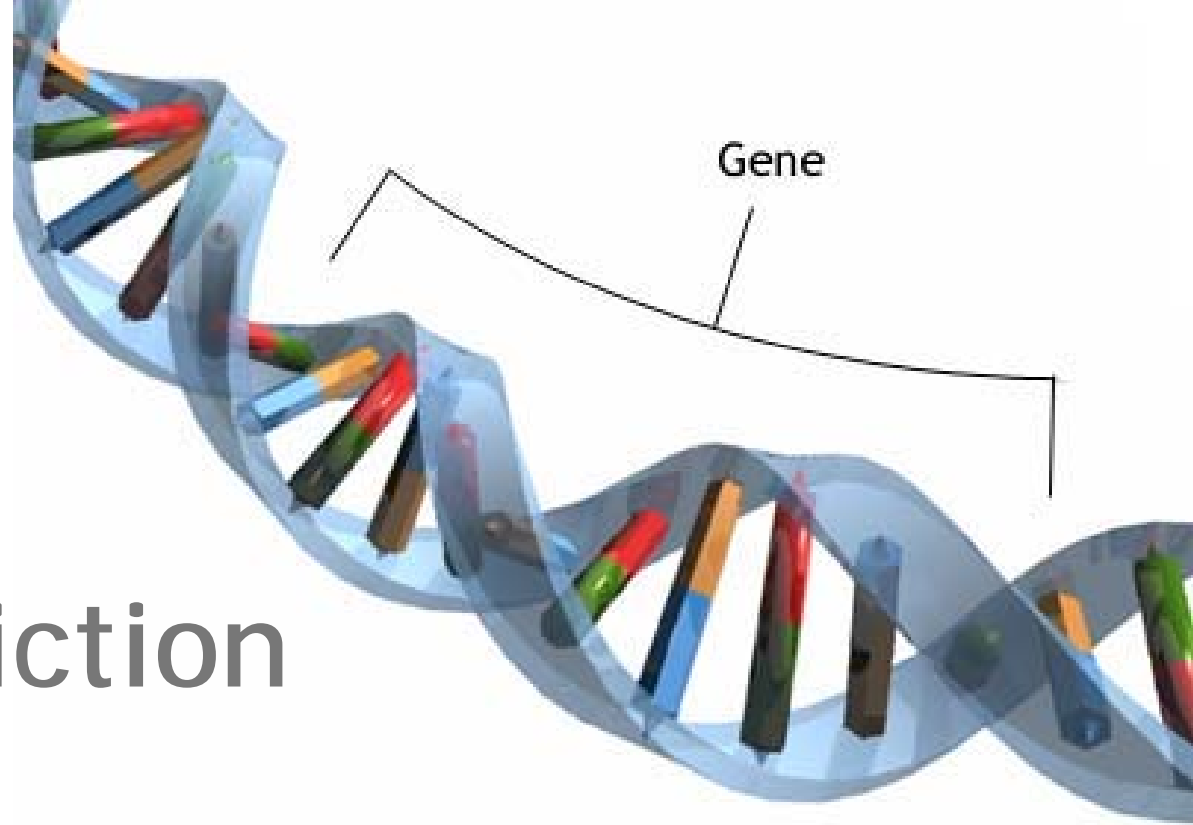
This presentation contains much material that was copied from material found on the internet or in scientific literature.

No explicit permission was asked to use this material and not all material is properly referenced.

However, I fully acknowledge the authors of all the material that I used in this presentation.

Antoine van Kampen

Gene prediction



Antoine van Kampen
a.h.vankampen@amc.uva.nl

Prediction of (human) gene structure

Gene identification

- ✓ Use and development of computational approaches to accurately predict gene structure.
- ✓ Important due to progress of large-scale sequencing projects.
- ✓ Part of detailed automatic (functional) annotation of genes and genomes (sequences).
- ✓ Ultimate goal: near 100% accuracy.
- ✓ Reduce amount of experimental verification work.

Genomes

According to National Center for Biotechnology Information (NCBI; dec. 2007)

Complete microbial genomes: 623

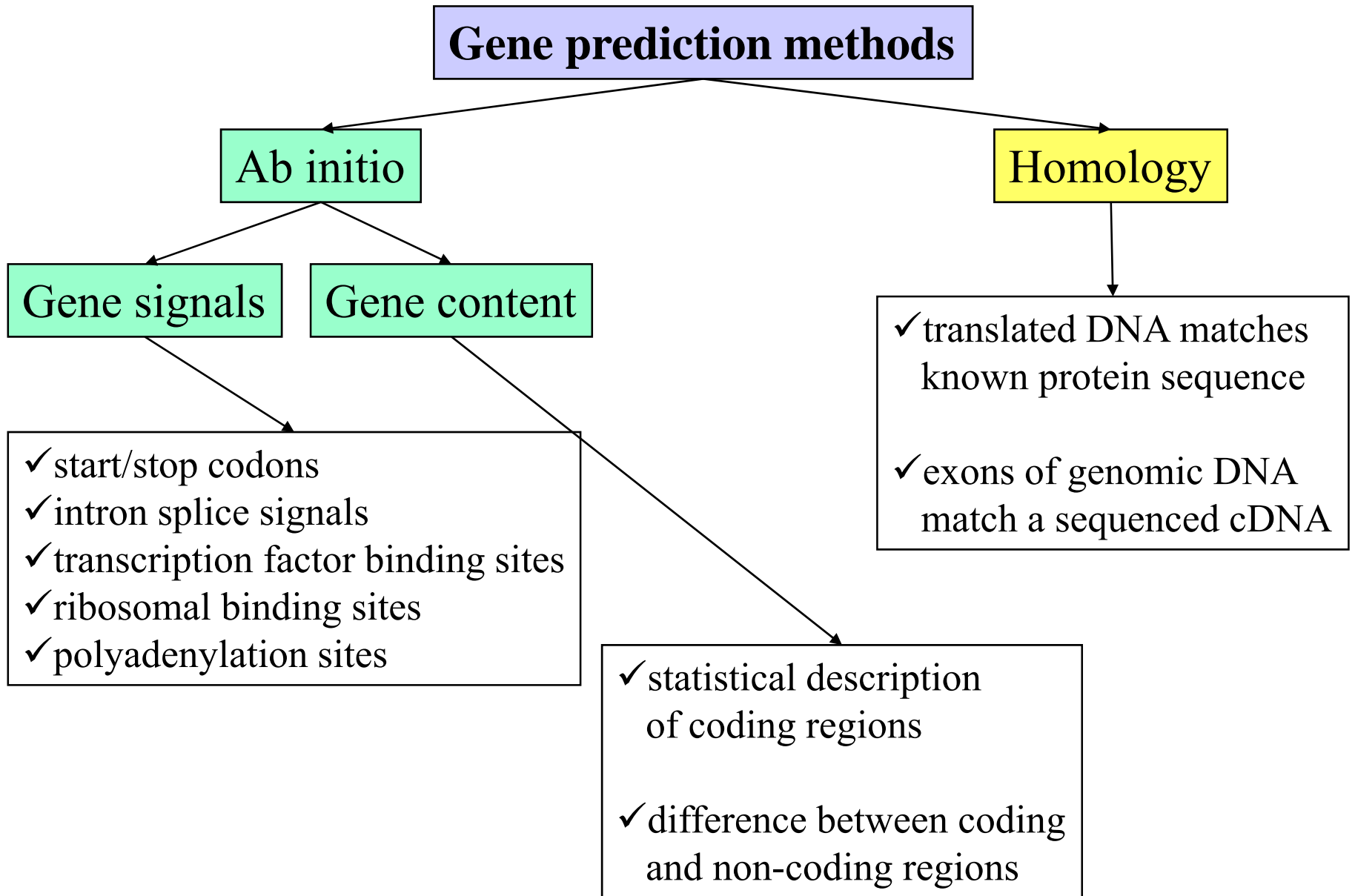
Microbial genomes in progress: 912

Complete Eukaryotic genomes: 23

Assembly of Eukaryotic genomes: 183

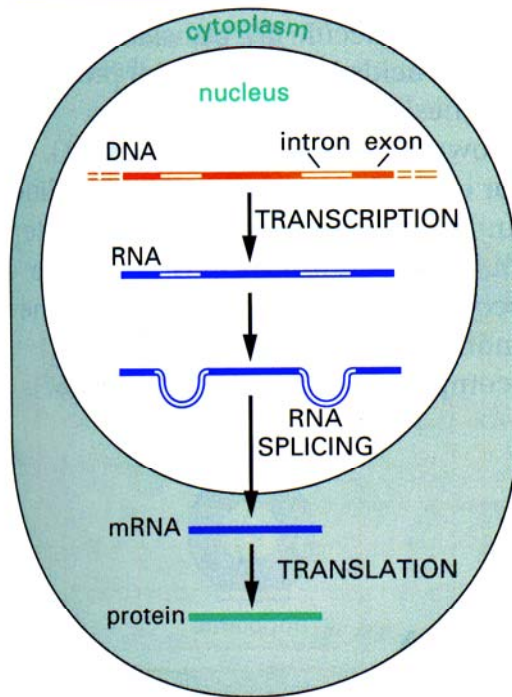
Eukaryotic genomes in progress: 230

Categories of gene prediction programs

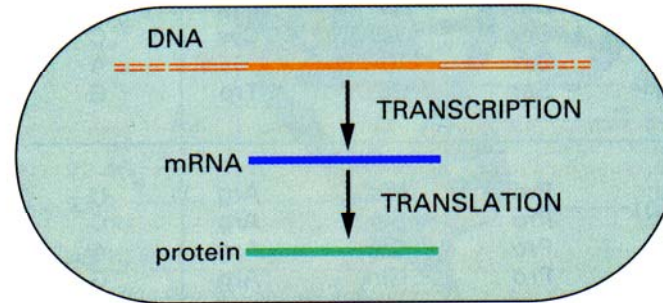


Difference between eukaryotes and prokaryotes

EUCARYOTES



PROCARYOTES



Nucleus

Genome: 10Mbp-670Gbp

Human: 3Gbp

3% protein coding

Many repetitive sequences

Gene: exon structure

No nucleus

Genome: 0.5-10Mbp

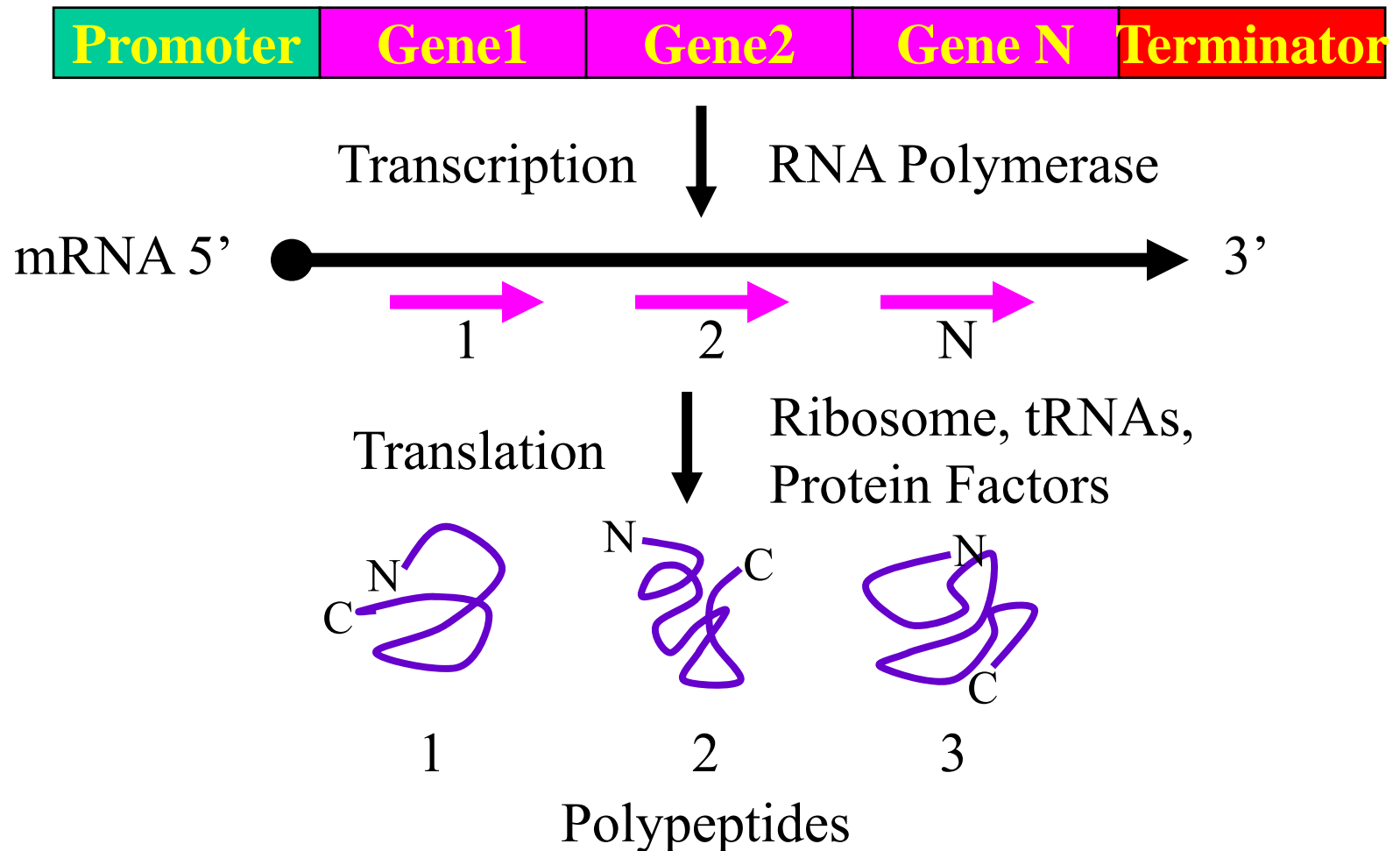
>90% protein coding

Few repetitive sequences

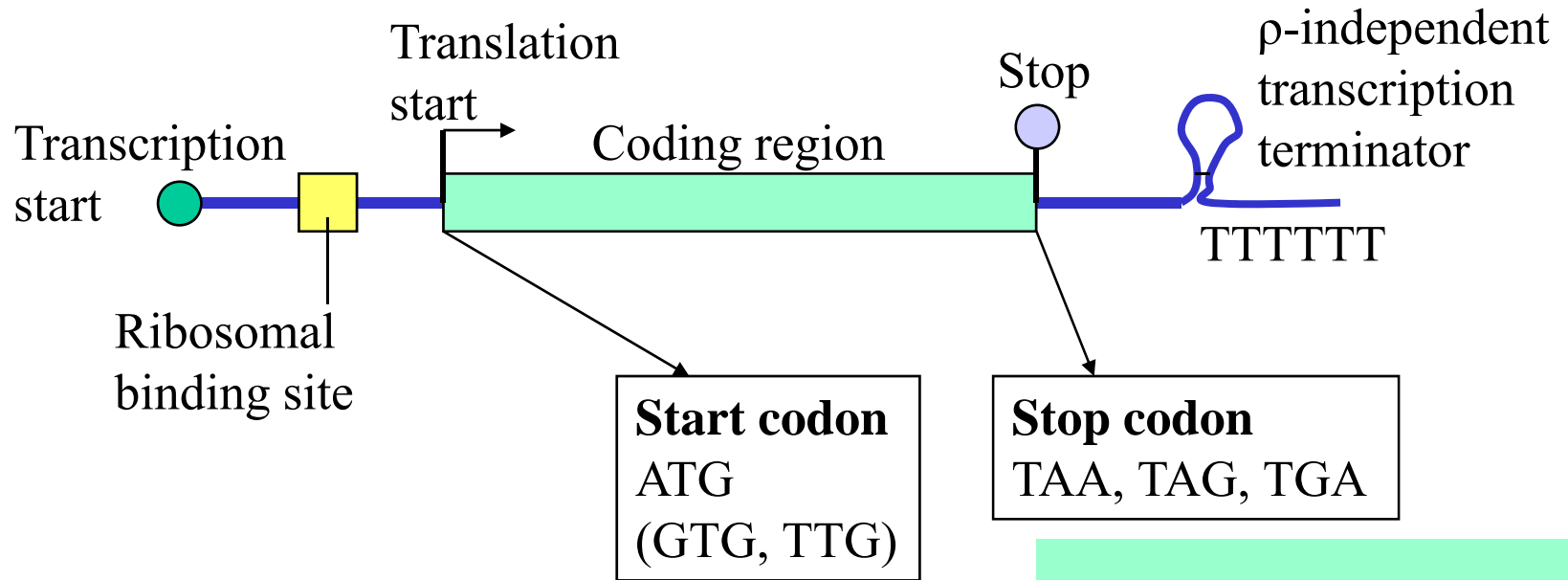
Gene: single contiguous stretch

Gene prediction in prokaryotes

Prokaryotes stack multiple genes together for expression (“operons”)

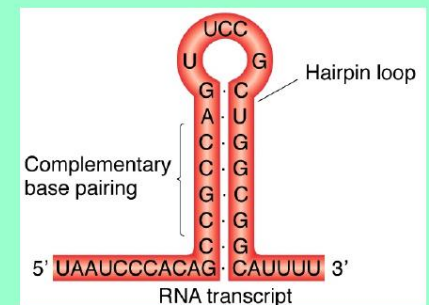


Gene structure of prokaryotes

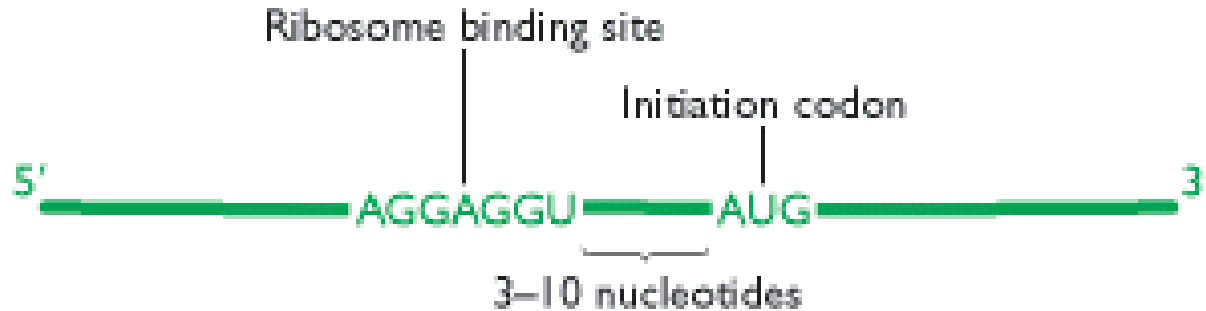


Identification of these features help to identify the (structure of the) gene

rho-independent terminator



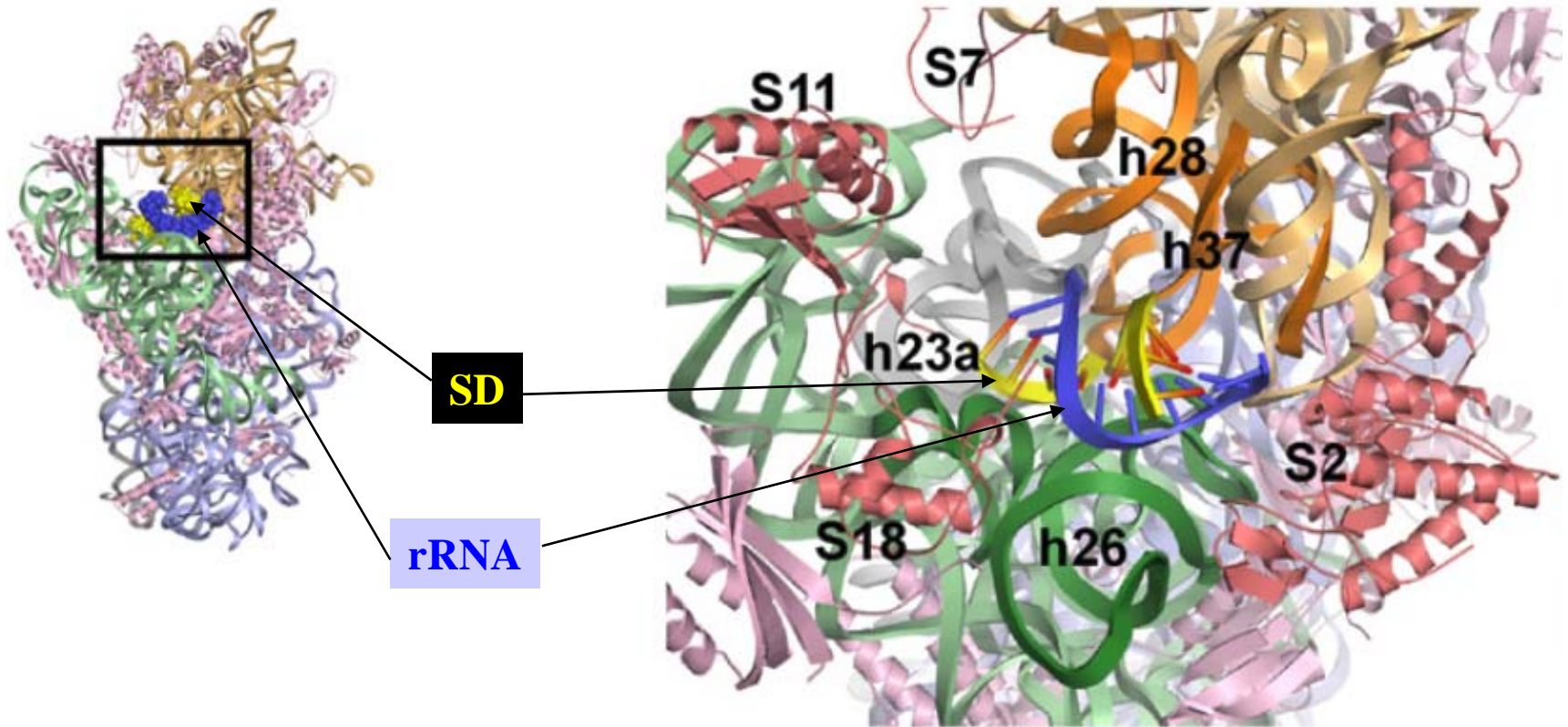
Ribosomal binding site: Shine-Delgarno sequence



The **ribosome binding** site for bacterial translation. In *Escherichia coli*, the ribosome binding site has the consensus sequence 5'-AGGAGGU-3' and is located between 3 and 10 nucleotides upstream of the initiation codon.

The sequence is complementary to `gaucACCUCCUuaOH` at the 3' end of 16S rRNA

Shine-Delgarno sequence



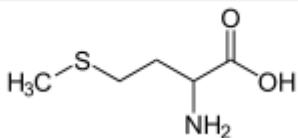
Shine-Dalgarno (SD) sequence which base pairs with a complementary anti-SD sequence at the 3' terminus of 16S rRNA in the 30S subunit

Genetic code

1st position (5' end) ↓	2nd position				3rd position (3' end) ↓
	U	C	A	G	
U	Phe Phe Leu Leu	Ser Ser Ser Ser	Tyr Tyr STOP STOP	Cys Cys STOP Trp	U C A G
C	Leu Leu Leu Leu	Pro Pro Pro Pro	His His Gln Gln	Arg Arg Arg Arg	U C A G
A	Ile Ile Ile Met	Thr Thr Thr Thr	Asn Asn Lys Lys	Ser Ser Arg Arg	U C A G
G	Val Val Val Val	Ala Ala Ala Ala	Asp Asp Glu Glu	Gly Gly Gly Gly	U C A G

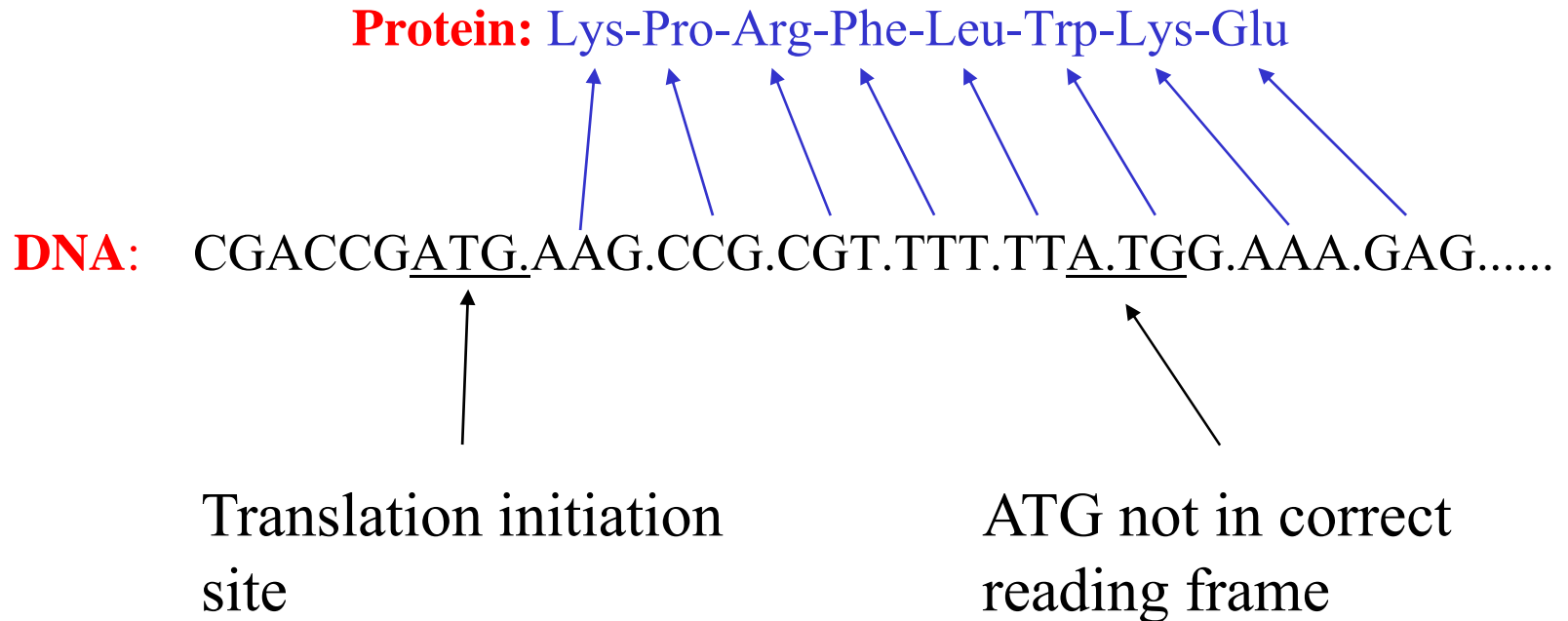
Synonymous
codons

Methionine

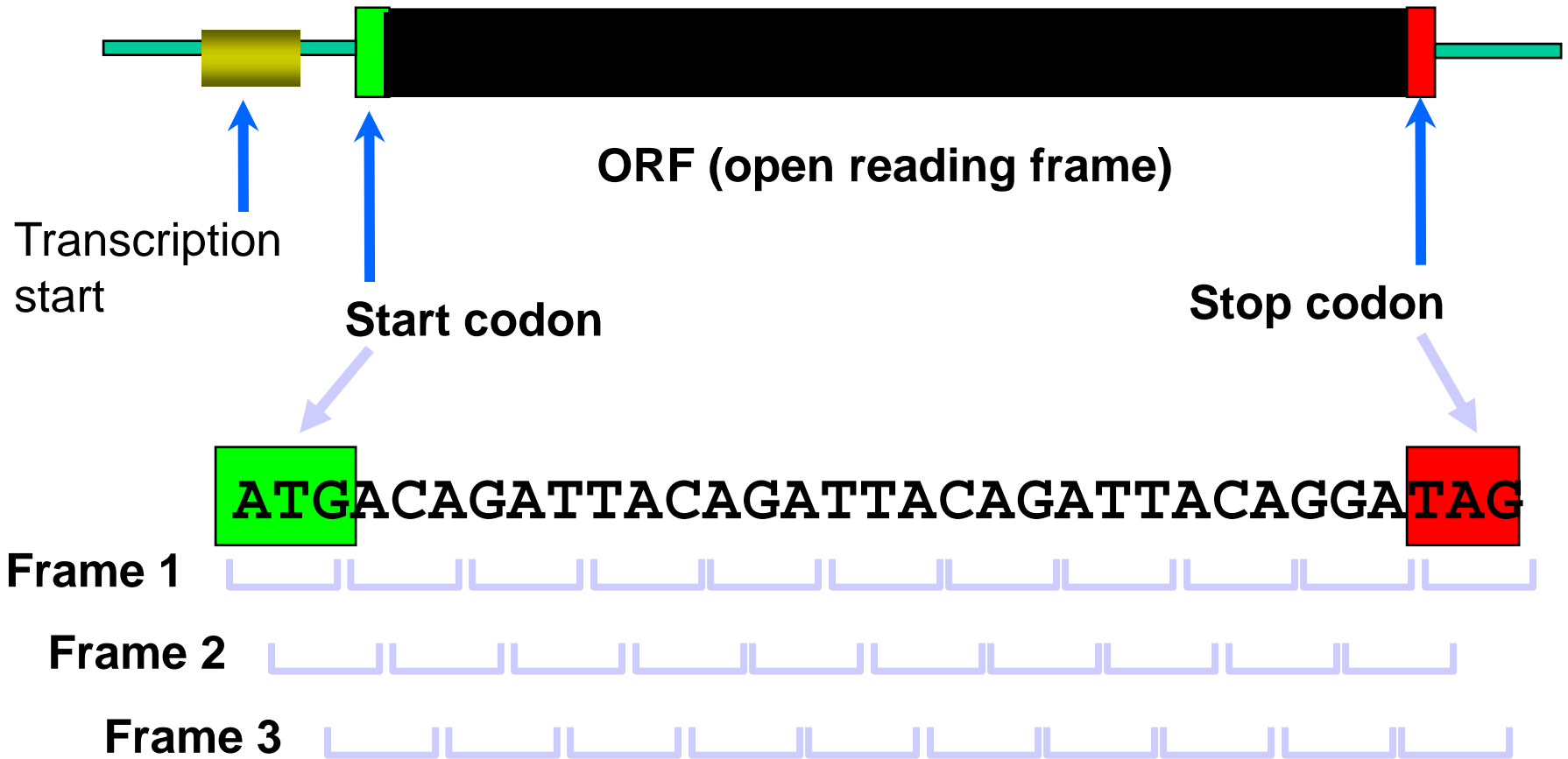


Determination of Open Reading Frames (ORF)

ATG must be in correct 'reading frame'



Determination of Open Reading Frames (ORF)



Determination of Open Reading Frames (ORF)

Each sequence has 6 possible **reading frames** that potentially encodes a proteins

- 3 in each direction (sense and anti-sense)

Problems:

- ✓ There will be many "ORFs" occurring by chance
- ✓ Some will be short - how do we know which are true?
- ✓ Introns make this useless in Eukaryotic DNA

Finding ORFs

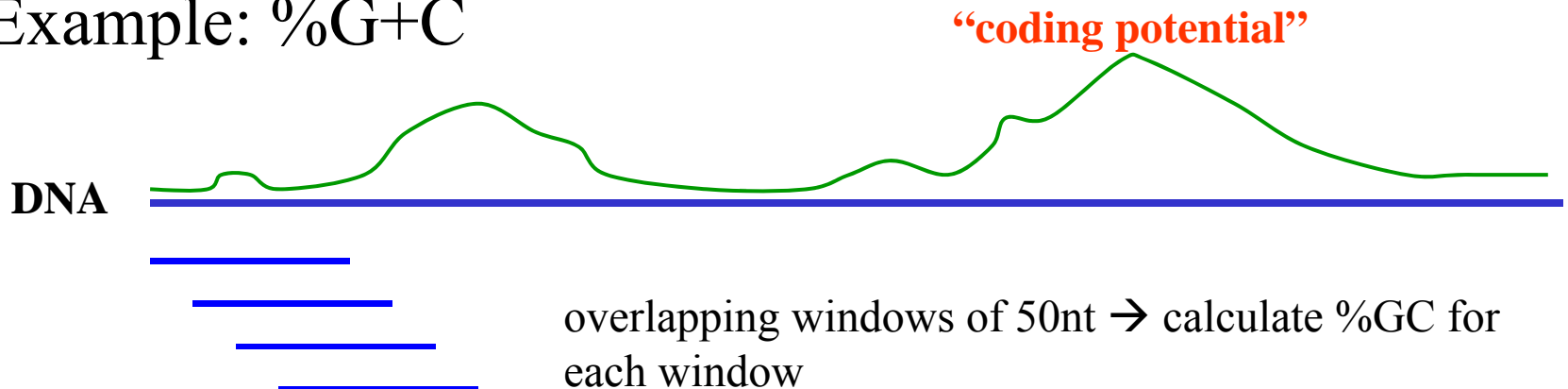
- Many more ORFs than genes
 - In E.Coli one finds 6500 ORFs while there are 4290 genes.
- In random DNA, one stop codon every $64/3=21$ codons on average.
- Average protein is ~ 300 codons long.
=> search long ORFs.
- Problems:
 - Short genes
 - Overlapping long ORFs on opposite strands

Statistical approaches

Many sequence analyses require calculating some statistic over a long sequence looking for regions where the statistic is unusually high or low

To do this, we define a *window size* to be the width of the region over which each calculation is to be done

Example: %G+C

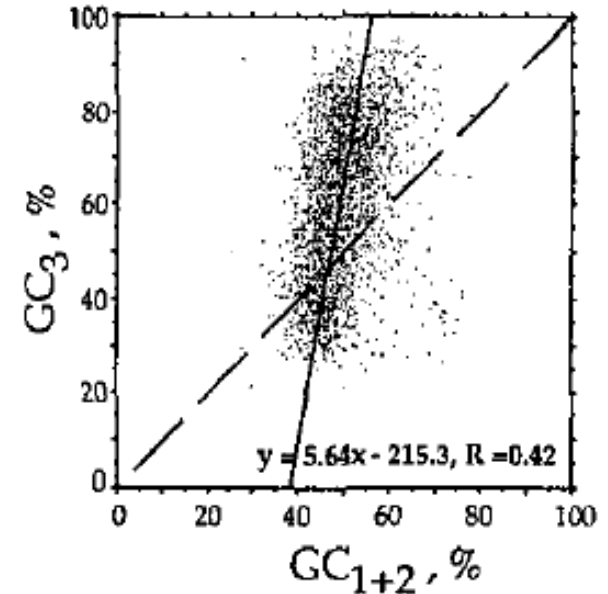
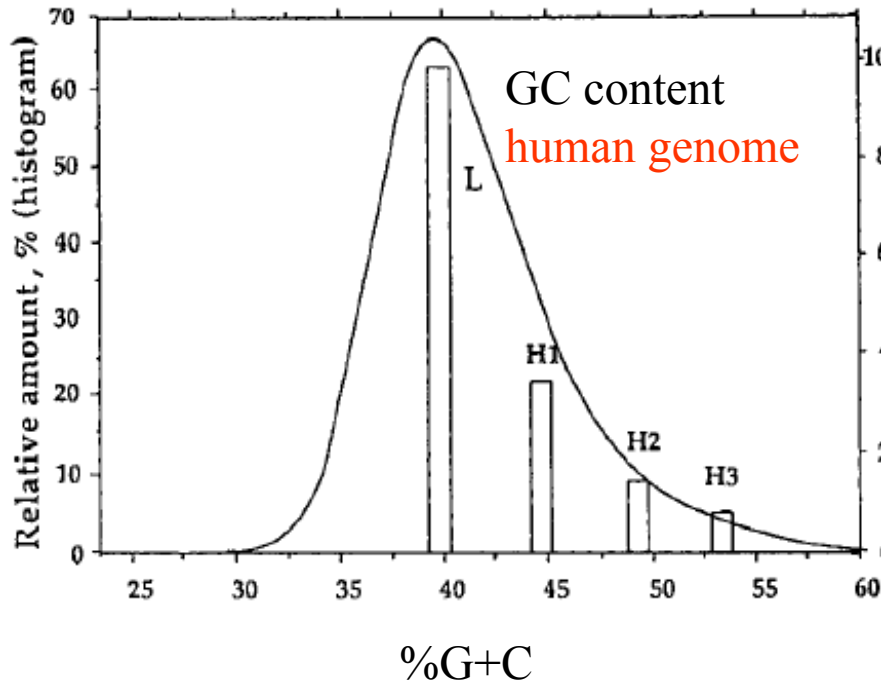


Compositional differences between coding and non-coding regions

Detect regular but very diffuse patterns in DNA sequences

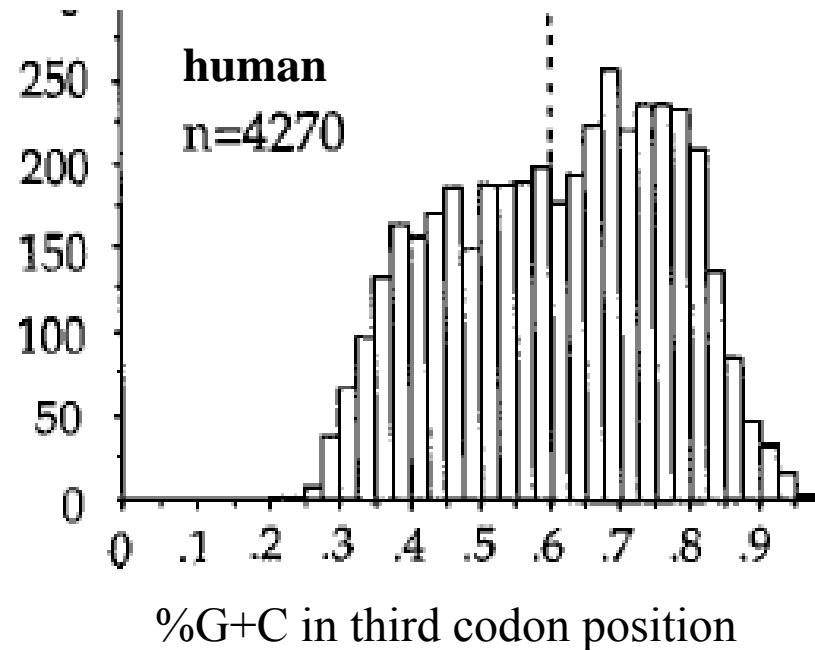
- **The patterns arise because:**
 - Organisms have a detectable preference for G or C over A and T in the third position in a codon (in coding region).
 - The genetic code is degenerate
 - Synonymous codons are not used with same frequency (codon bias)
 - Codon usage varies from
 - gene to gene
 - organism to organism
 - Unequal usage of amino acids in proteins sufficient to cause bias in all three positions of codons.
- **Biological basis, e.g.,**
 - Avoidance of codons similar to stop
 - Preference for codons that correspond to abundant tRNAs within the organism

G+C bias in third position



Isochores: > 300 Kbd DNA segments with homogeneous GC composition:
 GC-poor (L)
 GC-rich (H1, H2)
 GC-very rich (H3)

Number of genes



Codon usage in E. coli genes

CODON USAGE IN *E. COLI* GENES¹

	Codon	Amino acid ²	% ³	Ratio ⁴	Codon	Amino acid	%	Ratio	Codon	Amino acid	%	Ratio	Codon	Amino acid	%	Ratio		
U	UUU	Phe (F)	1.9	0.51	UCU	Ser (S)	1.1	0.19	UAU	Tyr (Y)	1.6	0.53	UGU	Cys (C)	0.4	0.43	U	
	UUC	Phe (F)	1.8	0.49	UCC	Ser (S)	1.0	0.17	UAC	Tyr (Y)	1.4	0.47	UGC	Cys (C)	0.6	0.57		C
	UUA	Leu (L)	1.0	0.11	UCA	Ser (S)	0.7	0.12	UAA	STOP	0.2	0.62	UGA	STOP	0.1	0.30		
	UUG	Leu (L)	1.1	0.11	UCG	Ser (S)	0.8	0.13	UAG	STOP	0.03	0.09	UGG	Trp (W)	1.4	1.00		G
C	CUU	Leu (L)	1.0	0.10	CCU	Pro (P)	0.7	0.16	CAU	His (H)	1.2	0.52	CGU	Arg (R)	2.4	0.42	U	
	CUC	Leu (L)	0.9	0.10	CCC	Pro (P)	0.4	0.10	CAC	His (H)	1.1	0.48	CGC	Arg (R)	2.2	0.37		C
	CUA	Leu (L)	0.3	0.03	CCA	Pro (P)	0.8	0.20	CAA	Gln (Q)	1.3	0.31	CGA	Arg (R)	0.3	0.05		
	CUG	Leu (L)	5.2	0.55	CCG	Pro (P)	2.4	0.55	CAG	Gln (Q)	2.9	0.69	CGG	Arg (R)	0.5	0.08		G
A	AUU	Ile (I)	2.7	0.47	ACU	Thr (T)	1.2	0.21	AAU	Asn (N)	1.6	0.39	AGU	Ser (S)	0.7	0.13	U	
	AUC	Ile (I)	2.7	0.46	ACC	Thr (T)	2.4	0.43	AAC	Asn (N)	2.6	0.61	AGC	Ser (S)	1.5	0.27		C
	AUA	Ile (I)	0.4	0.07	ACA	Thr (T)	0.1	0.30	AAA	Lys (K)	3.8	0.76	AGA	Arg (R)	0.2	0.04		
	AUG	Met (M)	2.6	1.00	ACG	Thr (T)	1.3	0.23	AAG	Lys (K)	1.2	0.24	AGG	Arg (R)	0.2	0.03		G
G	GUU	Val (V)	2.0	0.29	GCU	Ala (A)	1.8	0.19	GAU	Asp (D)	3.3	0.59	GGU	Gly (G)	2.8	0.38	U	
	GUC	Val (V)	1.4	0.20	GCC	Ala (A)	2.3	0.25	GAC	Asp (D)	2.3	0.41	GGC	Gly (G)	3.0	0.40		C
	GUA	Val (V)	1.2	0.17	GCA	Ala (A)	2.1	0.22	GAA	Glu (E)	4.4	0.70	GGA	Gly (G)	0.7	0.09		
	GUG	Val (V)	2.4	0.34	GCG	Ala (A)	3.2	0.34	GAG	Glu (E)	1.9	0.30	GGG	Gly (G)	0.9	0.13		G
	U				C				A				G					

¹ The data shown in this table is from the Arabidopsis Research Companion on the World Wide Web (<http://weeds/mgh.harvard.edu>). Codon frequencies for many other bacteria can be found at <http://morgan.angis.su.oz.au/Angis/Tables.html>.

² The letter in parenthesis represents the one-letter code for the amino acid.

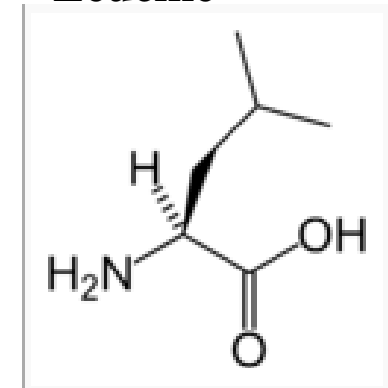
³ % represents the average frequency this codon is used per 100 codons.

⁴ Ratio represents the abundance of that codon relative to all of the codons for that particular amino acid.

Codon usage in E. coli genes

	Codon	Amino acid ²	% ³	Ratio ⁴
U	UUU	Phe (F)	1.9	0.51
	UUC	Phe (F)	1.8	0.49
	UUA	Leu (L)	1.0	0.11
	UUG	Leu (L)	1.1	0.11
C	CUU	Leu (L)	1.0	0.10
	CUC	Leu (L)	0.9	0.10
	CUA	Leu (L)	0.3	0.03
	CUG	Leu (L)	5.2	0.55

Leucine



Average frequency that this codon is used per 100 codons

Abundance of that codon relative to all codons for that amino acid

Codon preference

A codon preference parameter is calculated for

- each codon in the reading frame based on the codon's frequency of occurrence (f) and the total number of occurrences of its synonymous family (F) in the codon frequency table,
- and the calculated occurrences of the codon (r) and its synonymous family (R) in a random sequence with the same base composition as the sequence being analyzed.

Preference parameter for codon p is given by:

$$p = \frac{f/F}{r/R}$$

Codon **preference statistic**:

(w = number of codons in window)

$$P = e^{(\sum_{i=0}^w \log p_i) / w} = (\prod_{i=0}^w p_i)^{1/w}$$

Codon preference *E. coli* rpoBC operon

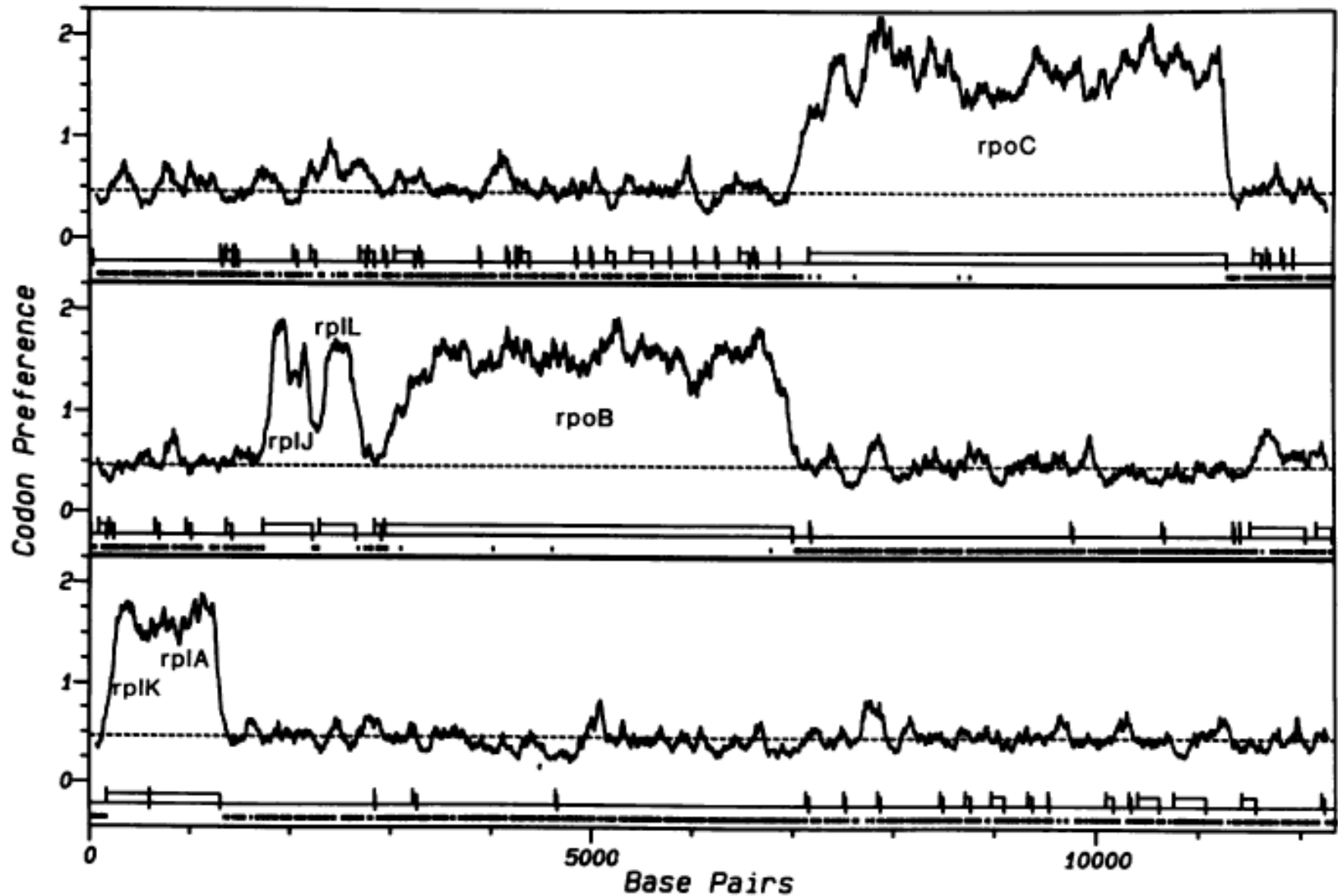


Fig. 2. Codon preference plot of the *E. coli* *rpoBC* operon. Rare codons, 2.5% of synonymous family or less, are marked, $w = 50$.

Testing for non-randomness: TESTCODE

- Test for period-three compositional bias in DNA
 - Test for bias in every position in a codon
 - This exploits any bias in the third codon position
- Use statistic developed by J. Fickett's (**TESTCODE**)
 - Independent of reading frame

Position parameter for the 'A' nucleotide

Calculate position parameter for sequence in a window

$$\text{A-position} = \frac{\max (fA_{pos1}, fA_{pos2}, fA_{pos3},)}{\min (fA_{pos1}, fA_{pos2}, fA_{pos3},)+1}$$

CATTAGACAGTAAGAGATCATAGAACAGAAATATAATA
1,2,2 $\rightarrow 2/(1+1)=1$

CATTAGACAGTAAGAGATCATAGAACAGAAATATAATA
2,2,2 $\rightarrow 2/(2+1)=0.66$

CATTAGACAGTAAGAGATCATAGAACAGAAATATAATA
2,2,1 $\rightarrow 2/(1+1)=1$

Count the frequency of 'A' at position 1, 2 and 3 for all codons

Position parameter

Calculate four position parameters for sequence

$$\text{A-position} = \frac{\max (fA_{\text{pos1}}, fA_{\text{pos2}}, fA_{\text{pos3}},)}{\min (fA_{\text{pos1}}, fA_{\text{pos2}}, fA_{\text{pos3}},)+1}$$

$$\text{C-position} = \frac{\max (fC_{\text{pos1}}, fC_{\text{pos2}}, fC_{\text{pos3}})}{\min (fC_{\text{pos1}}, fC_{\text{pos2}}, fC_{\text{pos3}})+1}$$

$$\text{G-position} = \frac{\max (fG_{\text{pos1}}, fG_{\text{pos2}}, fG_{\text{pos3}})}{\min (fG_{\text{pos1}}, fG_{\text{pos2}}, fG_{\text{pos3}})+1}$$

$$\text{T-position} = \frac{\max (fT_{\text{pos1}}, fT_{\text{pos2}}, fT_{\text{pos3}})}{\min (fT_{\text{pos1}}, fT_{\text{pos2}}, fT_{\text{pos3}})+1}$$

**Provides information
about bias for nucleotide
to be at specific position**

Content parameters

Calculate the four content parameters for sequence in the window

A-content = %A (expressed as a fraction)

C-content = %C

G-content = %G

T-content = %T

“Probability of Coding” table

For each value of the position/content parameter you can obtain a probability P that the sequence in the window is coding.

<u>Position Parameter</u>			<u>Probability of Coding</u>			
0.0	to	1.1	A: .22	C: .23	G: .08	T: .09
1.1		1.2	.20	.30	.08	.09
1.2		1.3	.34	.33	.16	.20
1.3		1.4	.45	.51	.27	.54
1.4		1.5	.68	.48	.48	.44
1.5		1.6	.58	.66	.53	.69
1.6		1.7	.93	.81	.64	.68
1.7		1.8	.84	.70	.74	.91
1.8		1.9	.68	.70	.88	.97
1.9		2.0+	.94	.80	.90	.97

<u>Content Parameter</u>			<u>Probability of Coding</u>			
.00	to	.17	A: .21	C: .31	G: .29	T: .58
.17		.19	.81	.39	.33	.51
.19		.21	.65	.44	.41	.69
.21		.23	.67	.43	.41	.56
.23		.25	.49	.59	.73	.75
.25		.27	.62	.59	.64	.55
.27		.29	.55	.64	.64	.40
.29		.31	.44	.51	.47	.39
.31		.33	.49	.64	.54	.24
.33		.99	.28	.82	.40	.28

“Probability of Coding” table

Example 1. 0.45 (45%) of the time a sequence with an A-position parameter = 1.35 was a coding sequence.

Example 2. 0.27 (27%) of the time a sequence with a G-position parameter = 1.35 was a coding sequence.

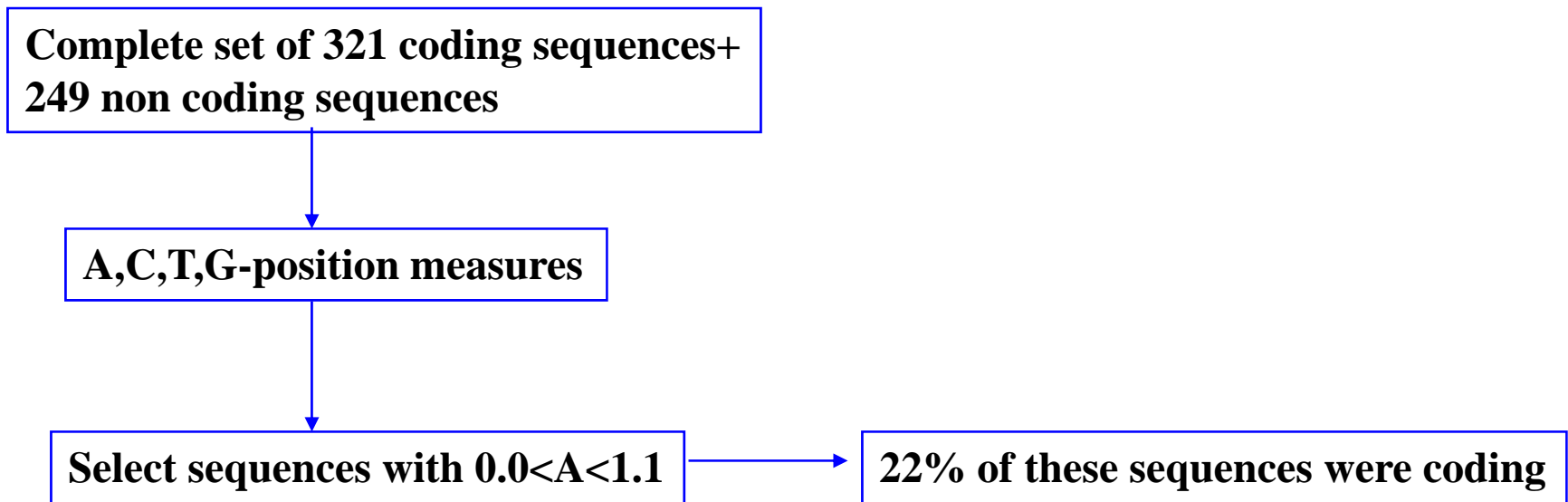
<u>Position Parameter</u>		<u>Probability of Coding</u>			
0.0	to 1.1	A: .22	C: .23	G: .08	T: .09
1.1	1.2	.20	.30	.08	.09
1.2	1.3	.34	.33	.16	.20
1.3	1.4	.45	.51	.27	.54
1.4	1.5	.68	.48	.48	.44
1.5	1.6	.58	.66	.53	.69
1.6	1.7	.93	.81	.64	.68
1.7	1.8	.84	.70	.74	.91
1.8	1.9	.68	.70	.88	.97
1.9	2.0+	.94	.80	.90	.97

<u>Content Parameter</u>		<u>Probability of Coding</u>			
.00	to .17	A: .21	C: .31	G: .29	T: .58
.17	.19	.81	.39	.33	.51
.19	.21	.65	.44	.41	.69
.21	.23	.67	.43	.41	.56
.23	.25	.49	.59	.73	.75
.25	.27	.62	.59	.64	.55
.27	.29	.55	.64	.64	.40
.29	.31	.44	.51	.47	.39
.31	.33	.49	.64	.54	.24
.33	.99	.28	.82	.40	.28

Generation of "Probability of Coding" table

-Originally, 321 coding and 249 non-coding sequences were analyzed

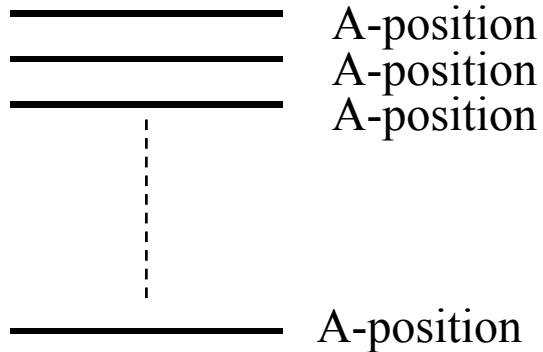
-Table contains the frequency that a coding region occurred when the parameter was within the defined limits.



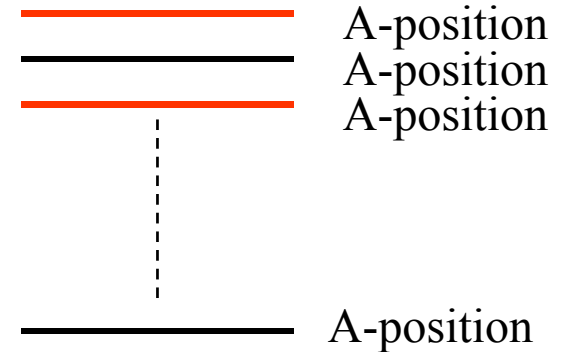
This is value in table

Generation of "Probability of Coding" table

321 coding sequences



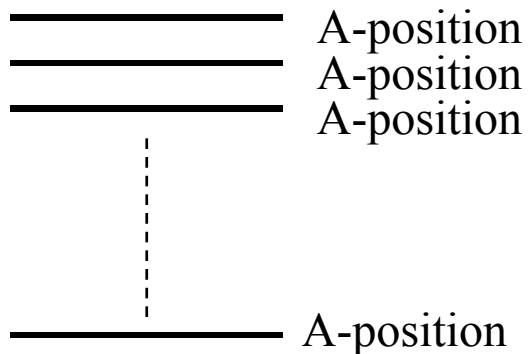
321 coding sequences



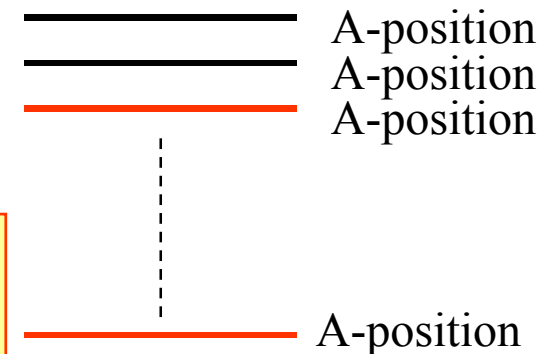
Select sequences
with $0.0 < A < 1.1$



249 noncoding sequences



249 noncoding sequences



22% of all red
sequences
is coding

Prediction rule

Prediction

If the **Probability of Coding** > 0.5 then the position parameter for this region is said to indicate that the region is coding.

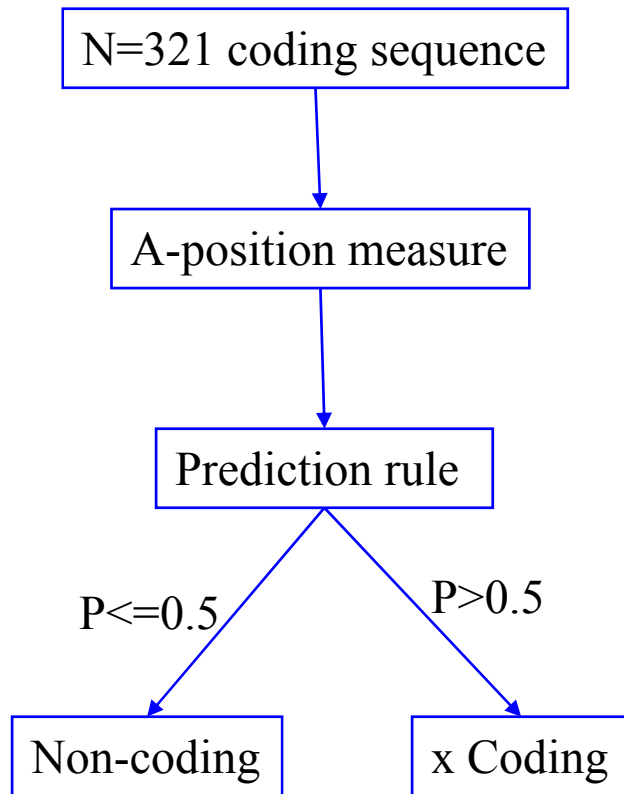
Very simple rule!

<u>Position Parameter</u>	<u>Probability of Coding</u>			
0.0 to 1.1	A: .22	C: .23	G: .08	T: .09
1.1 to 1.2	.20	.30	.08	.09
1.2 to 1.3	.34	.33	.16	.20
1.3 to 1.4	.45	.51	.27	.54
1.4 to 1.5	.68	.48	.48	.44
1.5 to 1.6	.58	.66	.53	.69
1.6 to 1.7	.93	.81	.64	.68
1.7 to 1.8	.84	.70	.74	.91
1.8 to 1.9	.68	.70	.88	.97
1.9 to 2.0+	.94	.80	.90	.97

<u>Content Parameter</u>	<u>Probability of Coding</u>			
.00 to .17	A: .21	C: .31	G: .29	T: .58
.17 to .19	.81	.39	.33	.51
.19 to .21	.65	.44	.41	.69
.21 to .23	.67	.43	.41	.56
.23 to .25	.49	.59	.73	.75
.25 to .27	.62	.59	.64	.55
.27 to .29	.55	.64	.64	.40
.29 to .31	.44	.51	.47	.39
.31 to .33	.49	.64	.54	.24
.33 to .99	.28	.82	.40	.28

Determine weights for position/content parameters

-Test 321 coding sequences again to obtain **weights**



$$\text{Weight } W_a = x/N$$

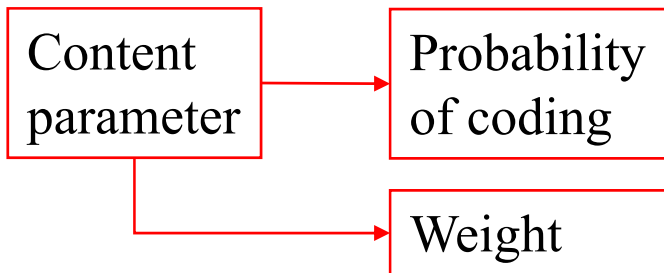
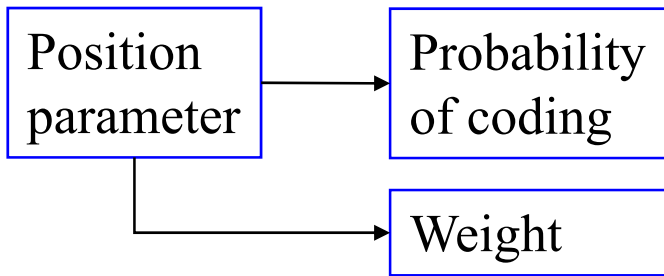
Similarly we obtain weights for the other position measures and content measures.

8 weights in total

Weight is fraction of sequences for which simple prediction rule ($P > 0.5$) is correct.

Weight table

	Weights	
	Position	Content
A	0.26	0.11
C	0.18	0.12
G	0.31	0.15
T	0.33	0.14



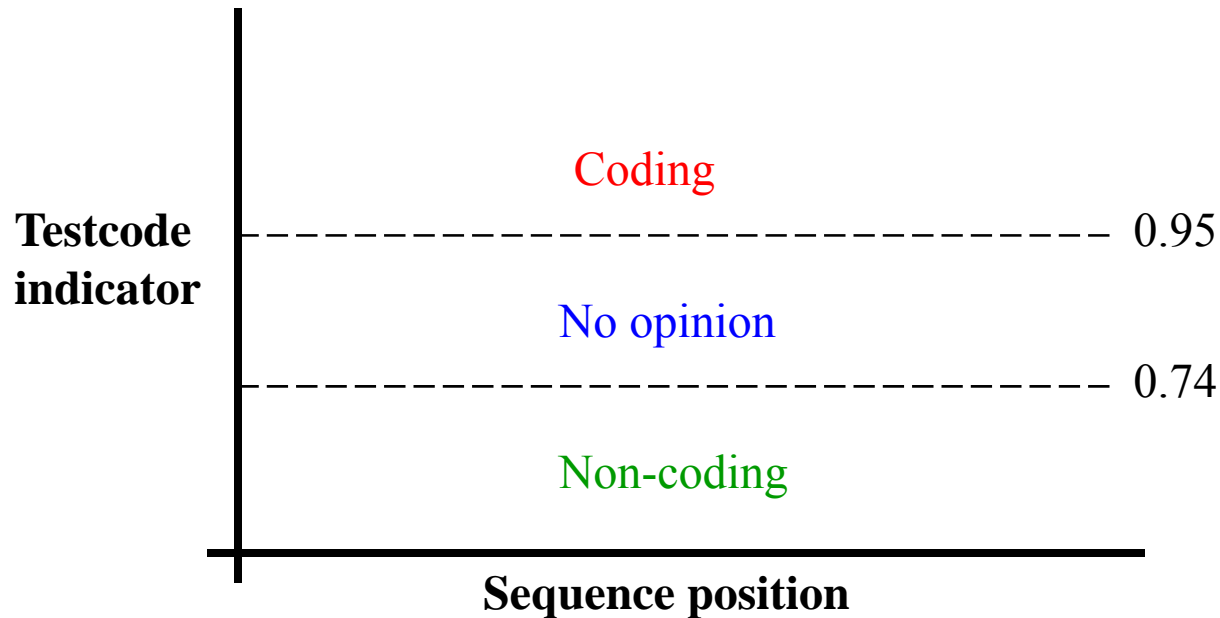
Example. A-position parameter is 1.35 \rightarrow probability of coding=0.45.

Weight (confidence) for A-position parameter = 0.26

Testcode Indicator

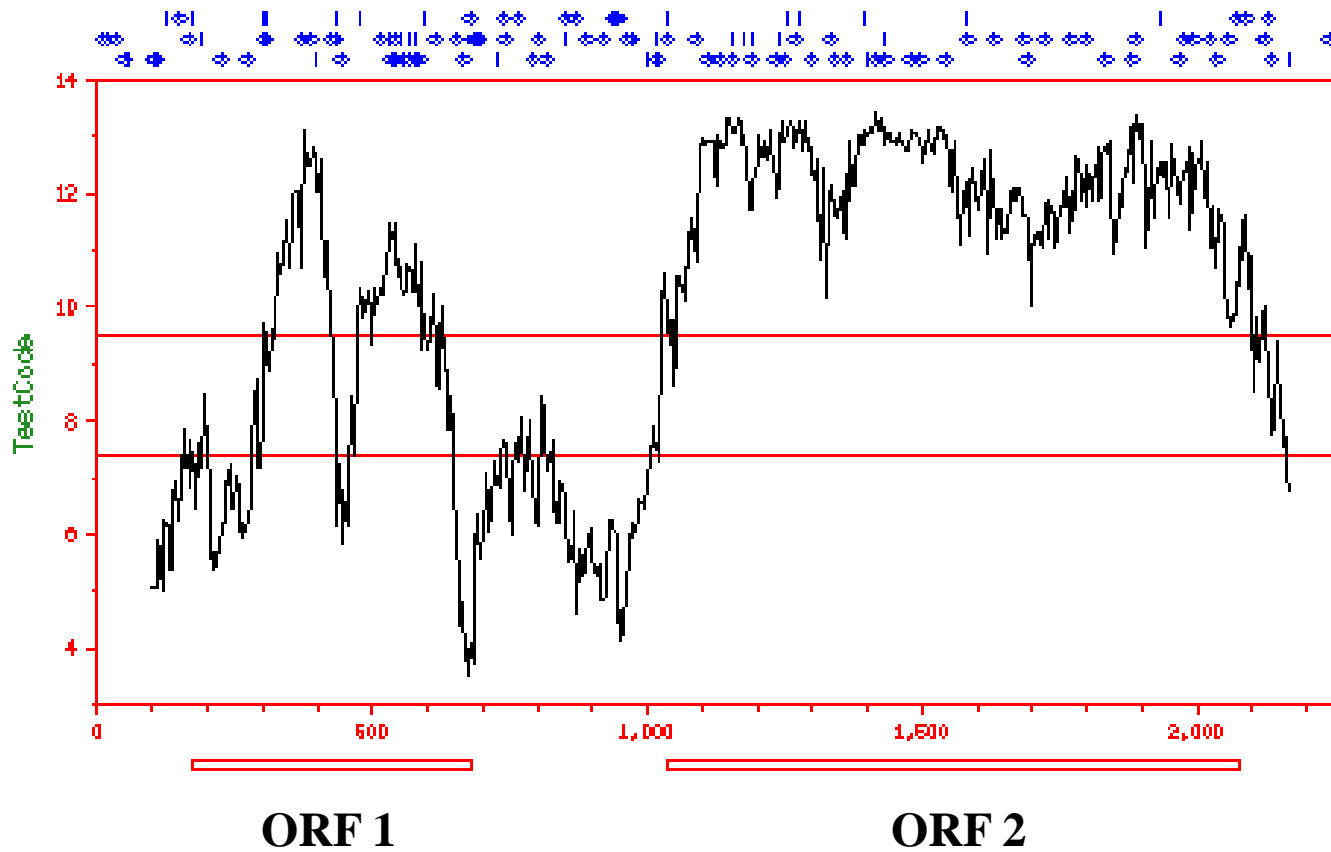
$$\text{Testcode Indicator} = P_1 W_1 + P_2 W_2 + \dots + P_8 W_8$$

This values is traditionally plotted for a moving window over the sequence



Fickett's Testcode

TESTCODE of: gb_ba:Ecoamp1 ck: 778, 1 to: 2270
Window: 200 bp October 6, 1998 10:54



Statistical models

These statistical methods, which are based on empirical rules identify only typical genes and tend to miss atypical genes.

We need more sophisticated statistical models

Markov chain models

Gene sequences

ctgagtctcgaatgccgagatac

Count transitions
between nucleotides

Non-gene sequences

ctgagagagacacacagactacag

Unknown sequence:
gene or not?

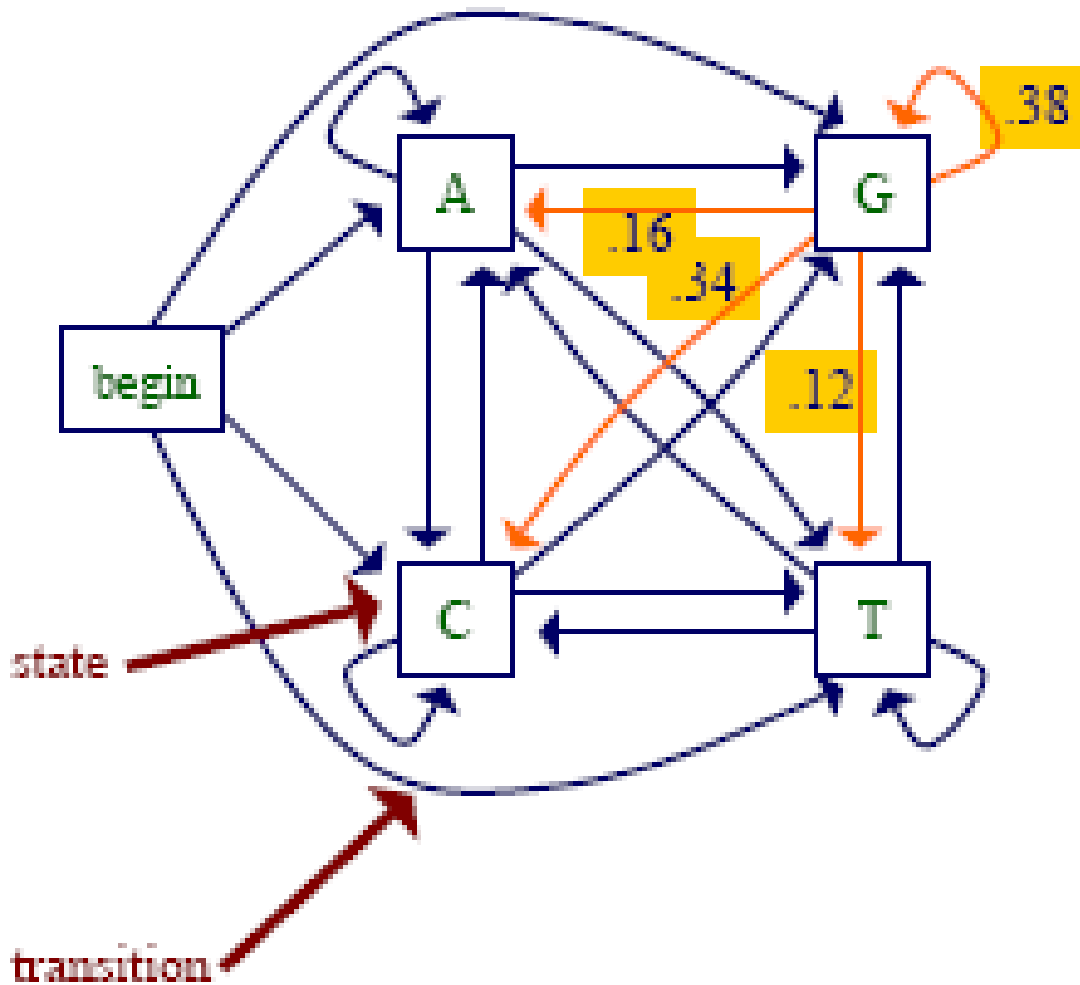
compare

<u>dinucleotide</u>	<u>f(gene)</u>	<u>f(non-gene)</u>
aa	1000	500
at	600	610
ac		
ag		
...		
gg		

Markov chain models

- a Markov chain model is defined by:
 - a set of **states**
 - some states *emit* symbols
 - other states (e.g. the *begin* state) are *silent*
 - a set of **transitions** with associated probabilities

Markov chain models



transition probabilities

$$\Pr(x_i = a \mid x_{i-1} = g) = 0.16$$

$$\Pr(x_i = c \mid x_{i-1} = g) = 0.34$$

$$\Pr(x_i = g \mid x_{i-1} = g) = 0.38$$

$$\Pr(x_i = t \mid x_{i-1} = g) = 0.12$$

Markov chain models

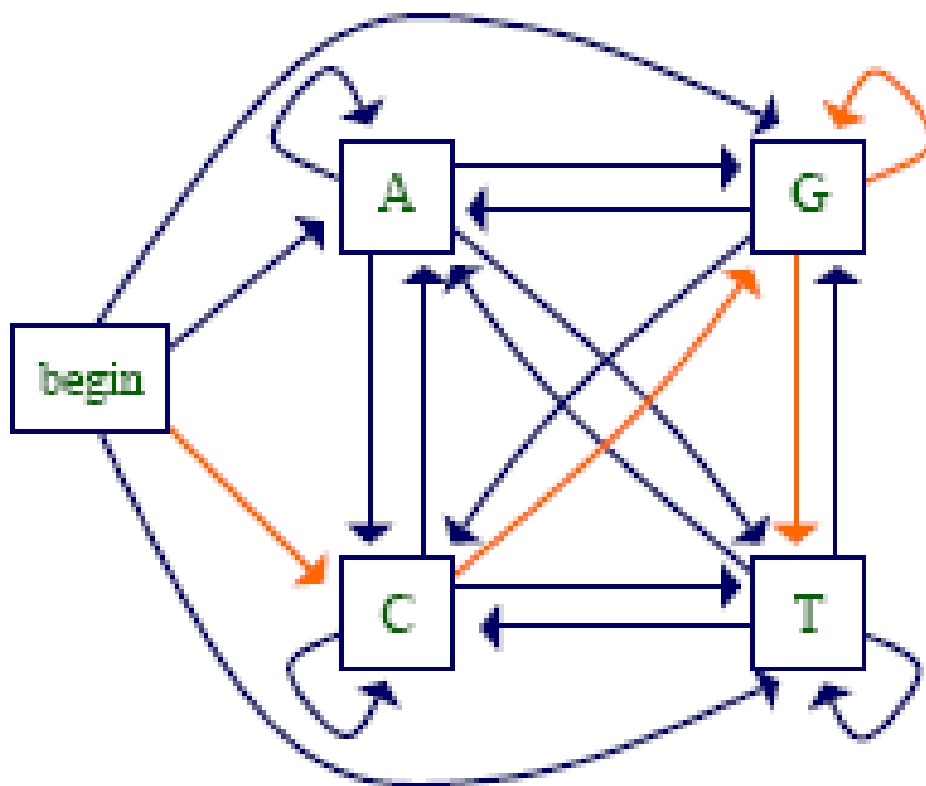
- given some **sequence x of length L** , we can ask how probable the sequence is given our (gene) model
- for any **probabilistic model** of sequences, we can write this probability as

$$\begin{aligned}\Pr(x) &= \Pr(x_L, x_{L-1}, \dots, x_1) \\ &= \Pr(x_L | x_{L-1}, \dots, x_1) \Pr(x_{L-1} | x_{L-2}, \dots, x_1) \dots \Pr(x_1)\end{aligned}$$

- key property of a **1st order Markov chain**: the probability of each X_i depends only on X_{i-1}

$$\begin{aligned}\Pr(x) &= \Pr(x_L | x_{L-1}) \Pr(x_{L-1} | x_{L-2}) \dots \Pr(x_2 | x_1) \Pr(x_1) \\ &= \Pr(x_1) \prod_{i=2}^L \Pr(x_i | x_{i-1})\end{aligned}$$

Markov chain models

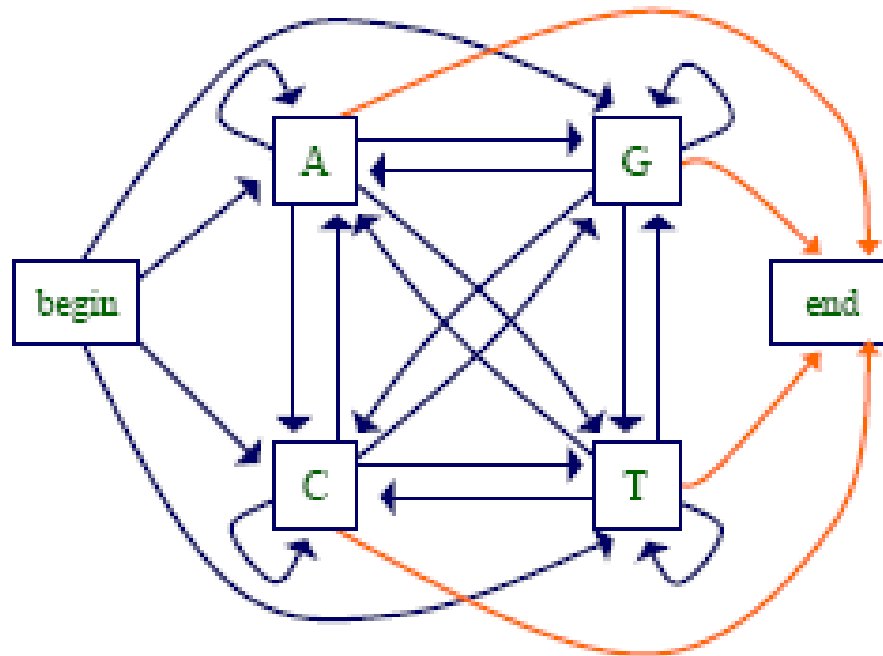


$$\Pr(cggt) = \Pr(c)\Pr(g|c)\Pr(g/g)\Pr(t/g)$$

Markov chain models

Can also have an *end state*, allowing the model to represent:

- Sequences of different lengths
- Preferences for sequences ending with particular symbols



Higher order Markov Chains

- The Markov property specifies that the probability of a state depends only on the probability of the previous state.
- We can build more ‘memory’ into our states by using a higher order Markov model
- *n*th order Markov model:

$$\Pr(x_i | x_{i-1}, x_{i-2}, \dots, x_1) = \Pr(x_i | x_{i-1}, \dots, x_{i-n})$$

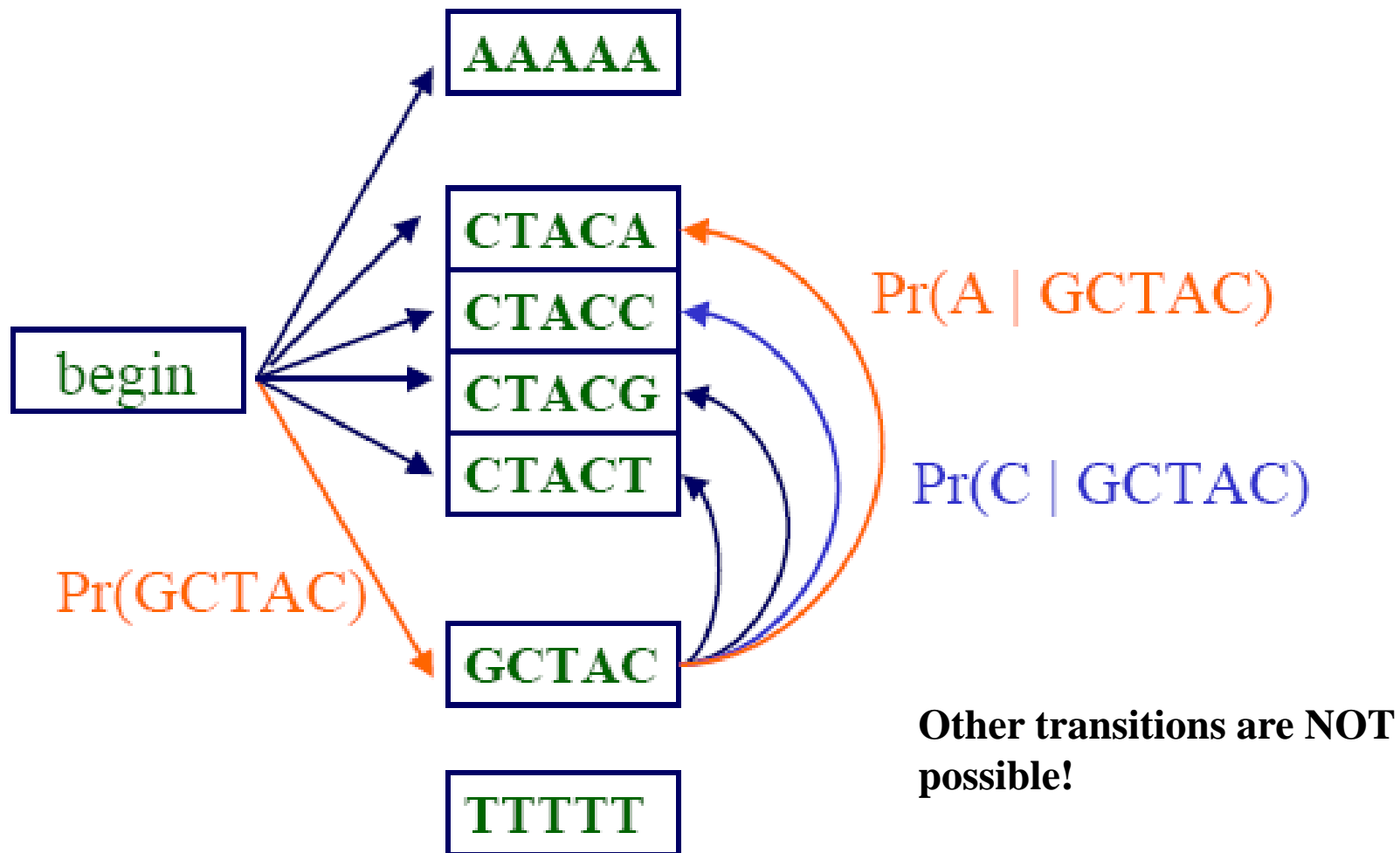
Higher order Markov chains

- An n th order Markov chain over some alphabet is equivalent to a first order Markov chain over the alphabet of n -tuples

Example:

- a 2nd order Markov model for DNA can be treated as a 1st order Markov model over alphabet:
AA, AC, AG, AT, CA, CC, CG, CT, GA, GC, GG,
GT, TA, TC, TG, and TT
(i.e. all possible dinucleotides)

A fifth order Markov chain

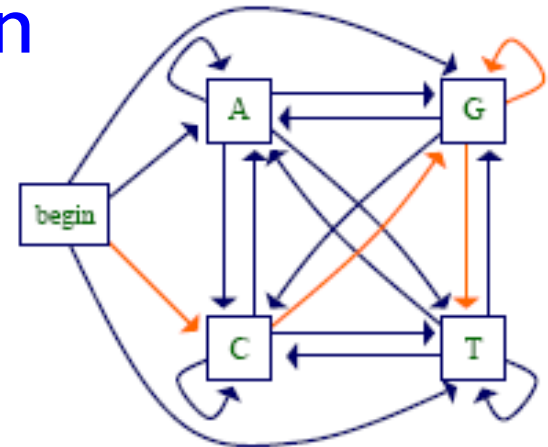
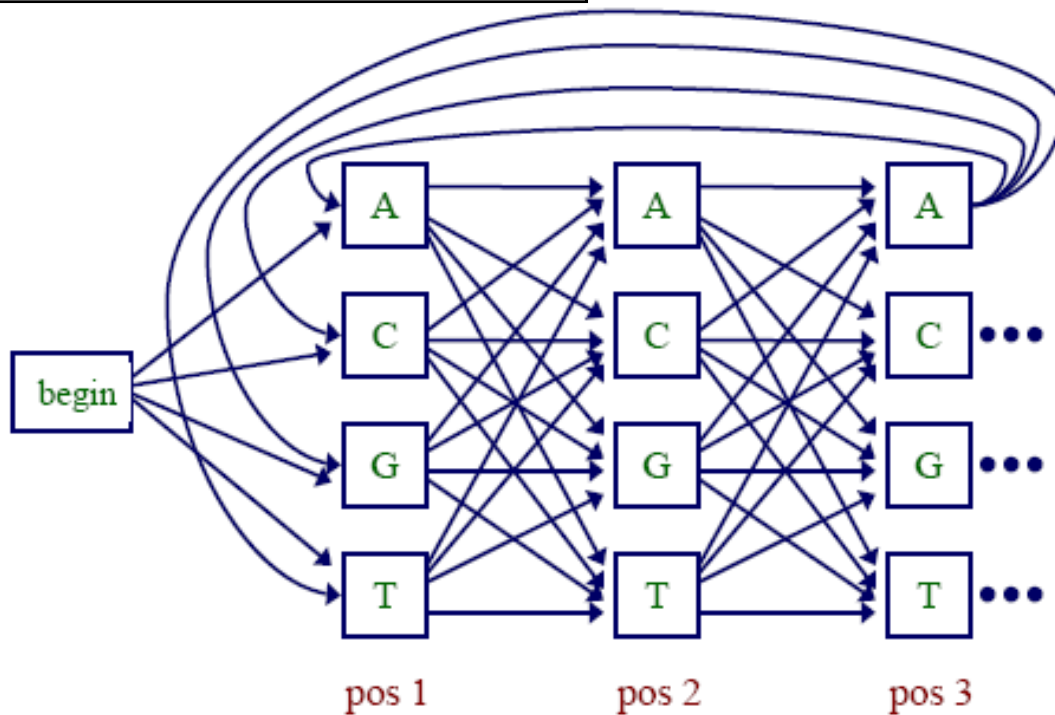


Inhomogeneous Markov Chains

- In the Markov chain models we have considered so far, the probabilities do not depend on where we are in a given sequence
- In an *inhomogeneous Markov model*, we can have different distributions at different positions in the sequence
- Consider modeling codons in protein coding regions

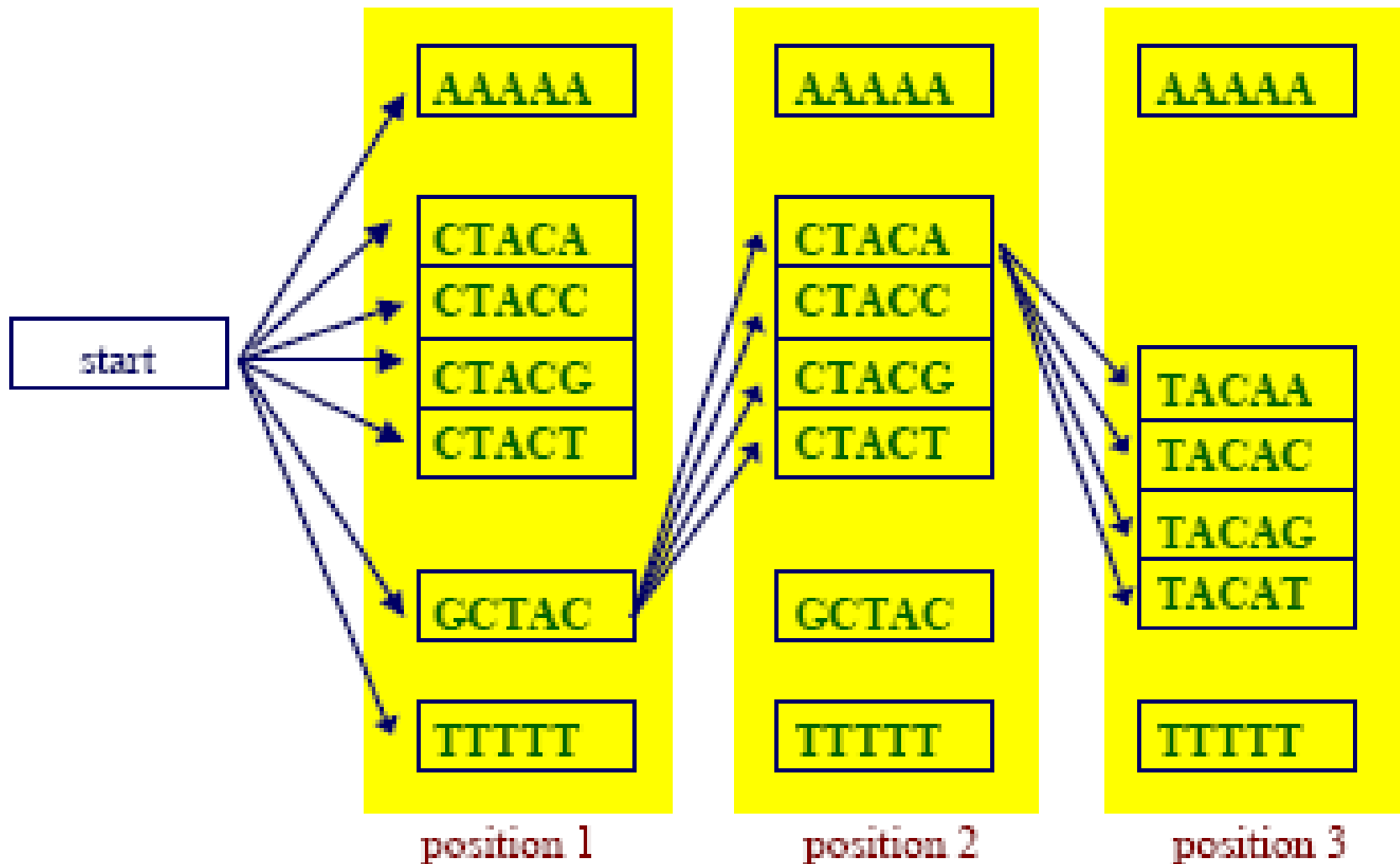
Inhomogeneous Markov Chain

Transition probability $A \rightarrow T$ depends on position in codon



Always same transition probability from e.g., $A \rightarrow T$

A fifth order inhomogeneous Markov chain



Selecting the order of a Markov chain model

- Higher order models remember more “history”
- Additional history can have predictive value

Example

Predict the next word in this sentence fragment

“...finish ___” (up, it, first, last, ...?)

Selecting the order of a Markov chain model

- Higher order models remember more “history”
- Additional history can have predictive value

Example

Predict the next word in this sentence fragment

“...finish ___” (up, it, first, last, ...?)

Now predict it given more history

“Fast guys finish ___”

Selecting the order of a Markov chain model

- However, the number of parameters we need to estimate grows exponentially with the order (n)
 - For modeling DNA we need parameters for an n th order model; with $n \geq 5$ normally
- The higher the order, the less reliable we can expect our parameter estimates to be
 - Estimating the parameters of a 2nd order homogenous Markov chain from the complete genome of E. Coli, we would see each word $> 72,000$ times on average
 - Estimating the parameters of an 8th order chain, we would see each word ~ 5 times on average (too few examples to estimate transitions probabilities)

Interpolated Markov models

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

$$\begin{aligned} \Pr_{\text{IMM}}(x_i | x_{i-1}, \dots, x_{i-n}) &= \lambda_0 \Pr(x_i) \\ &+ \lambda_1 \Pr(x_i | x_{i-1}) \\ &\dots \\ &+ \lambda_n \Pr(x_i | x_{i-1}, \dots, x_{i-n}) \end{aligned}$$

- 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

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

GLIMMER

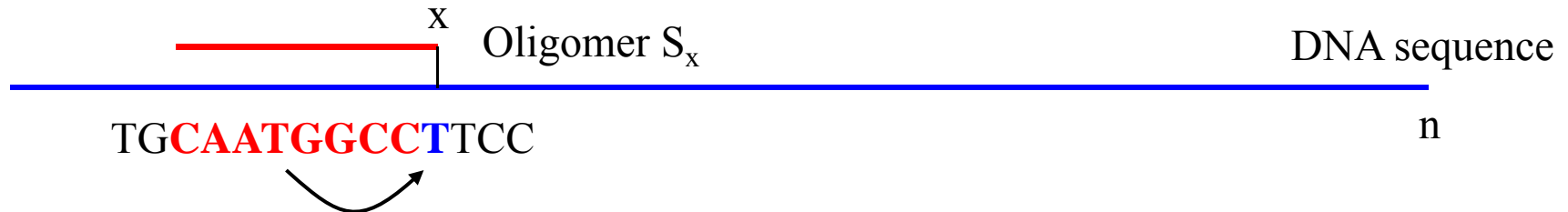
Probability that model M generated sequence S :

$$P(S|M) = \sum_{x=1}^n \text{IMM}_8(S_x)$$

S_x = oligomer ending at position x

n = length of sequence

$\text{IMM}_8(S_x)$ = 8th-order IMM



Normal Markov Model:

8th-order MM: probability $P(T|CAATGGCC)$ that these 8 bases predict base T at position x .

GLIMMER: recursive definition of IMM



recursive definition

$$\text{IMM}_k(S_x) = \lambda_k(S_{x-1}) \cdot P_k(S_x) + [1 - \lambda_k(S_{x-1})] \cdot \text{IMM}_{k-1}(S_x)$$

$$\text{IMM}_8(S_{100}) = \lambda_8(S_{99}) \times P_8(S_{100}) + [1 - \lambda_8(S_{99})] \times \text{IMM}_7(S_{100})$$

weight associated
with *kmer* (k=8) ending
at position 99

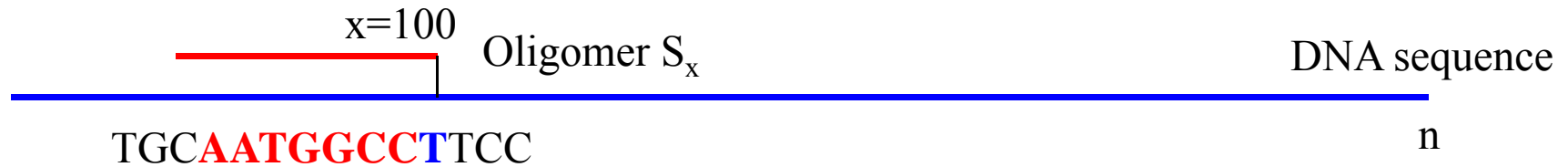
is **zero** if probability
associated with CAATGGCC
could be estimated with
sufficient reliability

Glimmer computes the probabilities
of each base a,c,t,g following all
kmers for $0 \leq k \leq 8$

0th-order: prior probabilities

For each *kmer* it computes a weight

GLIMMER



$$\text{IMM}_k(S_x) = \lambda_k(S_{x-1}) \cdot P_k(S_x) + [1 - \lambda_k(S_{x-1})] \cdot \text{IMM}_{k-1}(S_x)$$

$$\text{IMM}_8(S_{100}) = \lambda_8(S_{99}) \times P_8(S_{100}) + [1 - \lambda_8(S_{99})] \times \text{IMM}_7(S_{100})$$

$$\text{IMM}_7(S_{100}) = \lambda_7(S_{99}) \times P_7(S_{100}) + [1 - \lambda_7(S_{99})] \times \text{IMM}_6(S_{100})$$

$$\text{IMM}_1(S_{100}) = \lambda_1(S_{99}) \times P_1(S_{100}) + [1 - \lambda_1(S_{99})] \times \text{IMM}_0(S_{100})$$

$$\text{IMM}_0(S_{100}) = P_0(S_{100}) \text{ (prior probabilities)}$$

How does GLIMMER determine weights?

The **weight values** of $\lambda_k(S_k)$ associated with $P_k(S_k)$ can be regarded as a **measure of confidence** in the accuracy of this value as an estimate of the true probability.

IF the number of times we see a certain history in our training set

$$x_{i-n}, \dots, x_{i-1} > 400 \quad \text{THEN} \quad \lambda_k(x_{i-n}, \dots, x_{i-1}) = 1$$

In this case lower order *kmers* are not considered

How does GLIMMER determine weights?

If we haven't seen $x_{i-1} \dots x_{i-n}$ more than 400 times, then compare these counts with the next shorter history:

n^{th} -order history	$(n-1)^{\text{th}}$ -order history
$x_{i-n}, \dots, x_{i-1}, a$	$x_{i-n+1}, \dots, x_{i-1}, a$
$x_{i-n}, \dots, x_{i-1}, c$	$x_{i-n+1}, \dots, x_{i-1}, c$
$x_{i-n}, \dots, x_{i-1}, g$	$x_{i-n+1}, \dots, x_{i-1}, g$
$x_{i-n}, \dots, x_{i-1}, t$	$x_{i-n+1}, \dots, x_{i-1}, t$

Use a statistical test (χ^2) to get a value d indicating our confidence that the distributions represented by the two sets of counts are different \rightarrow prefer shorter history

How does GLIMMER determine weights?

- putting it all together

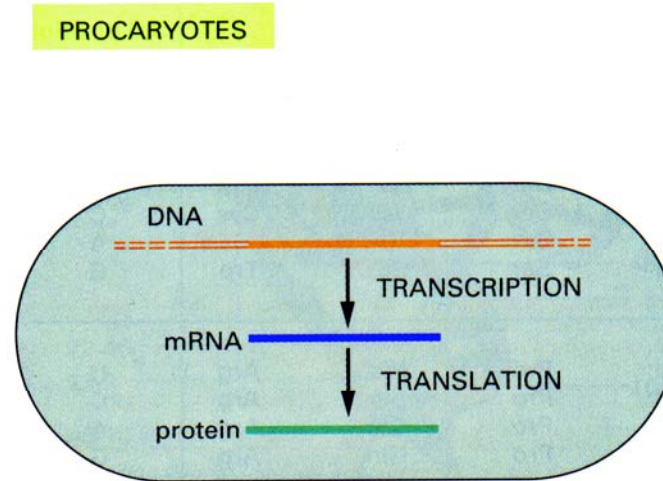
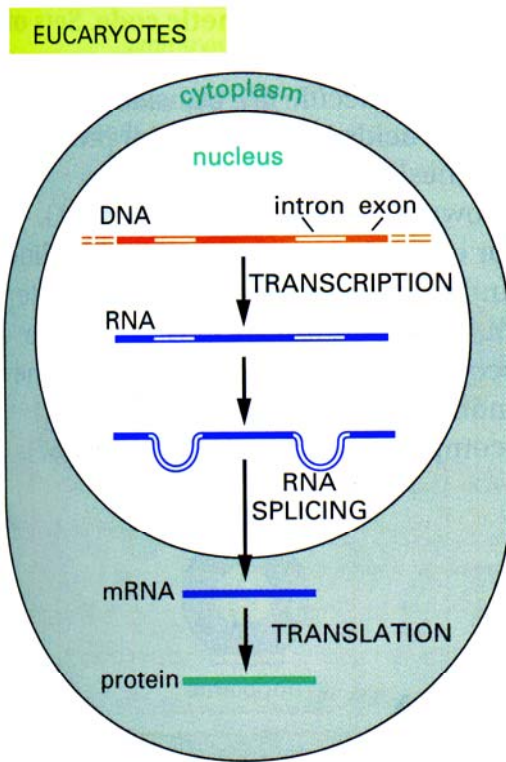
$$\lambda_n(x_{i-1}, \dots, x_{i-n}) = \begin{cases} 1 & \text{if } c(x_{i-1}, \dots, x_{i-n}) > 400 \\ d \times \frac{c(x_{i-1}, \dots, x_{i-n})}{400} & \text{else if } d \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

where $d \in (0,1)$

χ^2 score when comparing n^{th} -order with $n-1^{\text{th}}$ -order Markov model

Gene prediction in Eukaryotes

Gene prediction in Eukaryotes



Nucleus

Genome: 10Mbp-670Gbp

Human: 3Gbp

3% protein coding

Many repetitive sequences

Gene: exon structure

No nucleus

Genome: 0.5-10Mbp

>90% protein coding

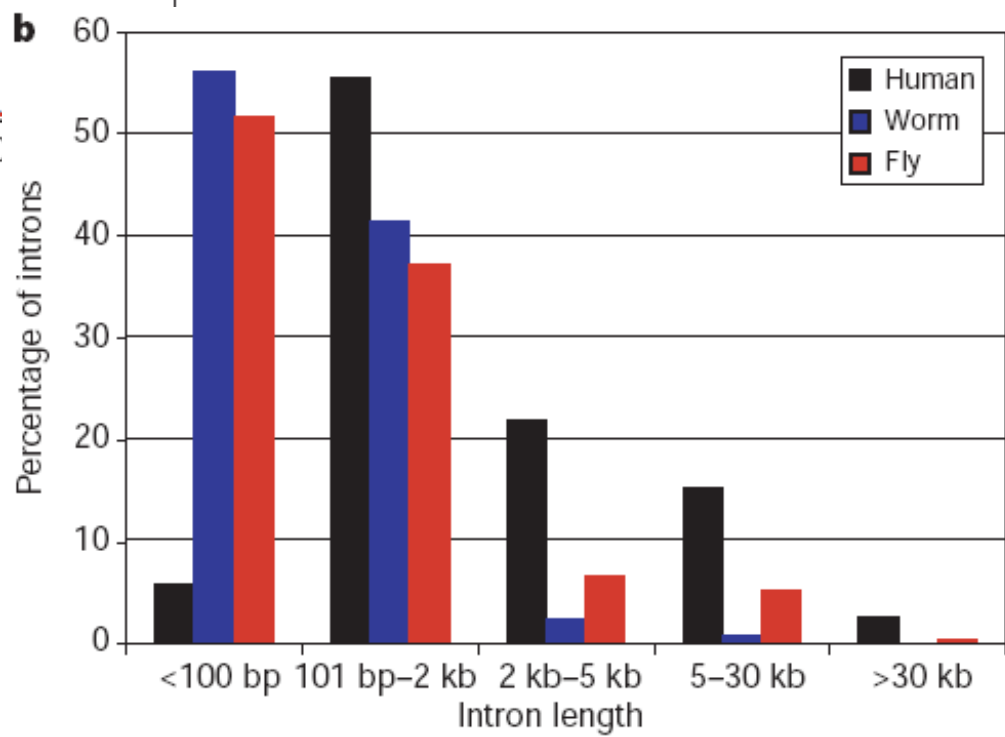
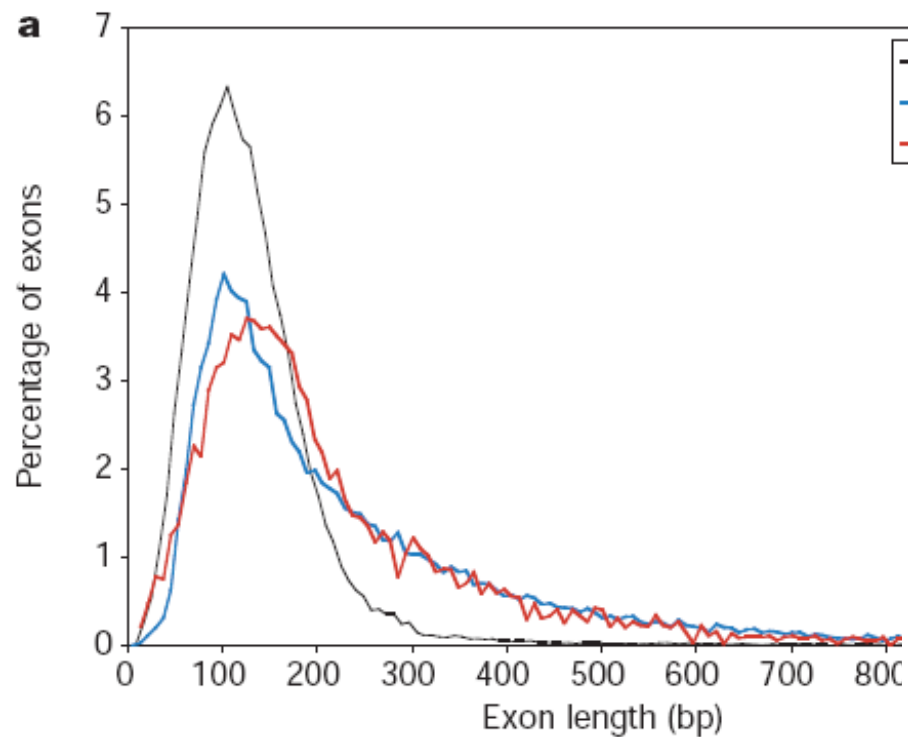
Few repetitive sequences

Gene: single contiguous stretch

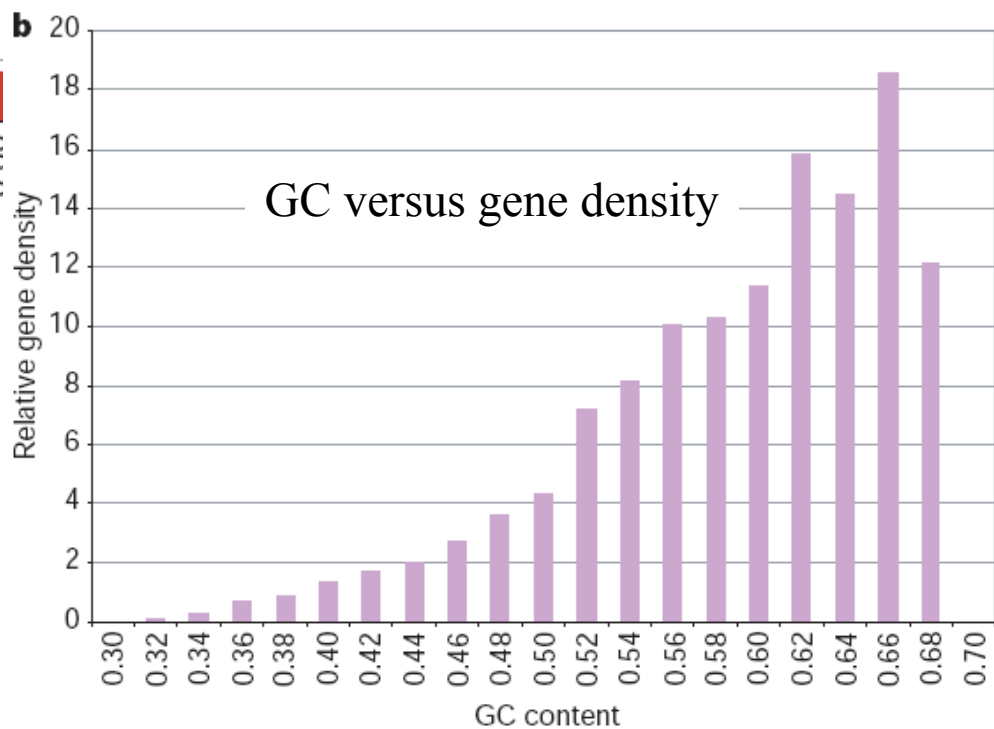
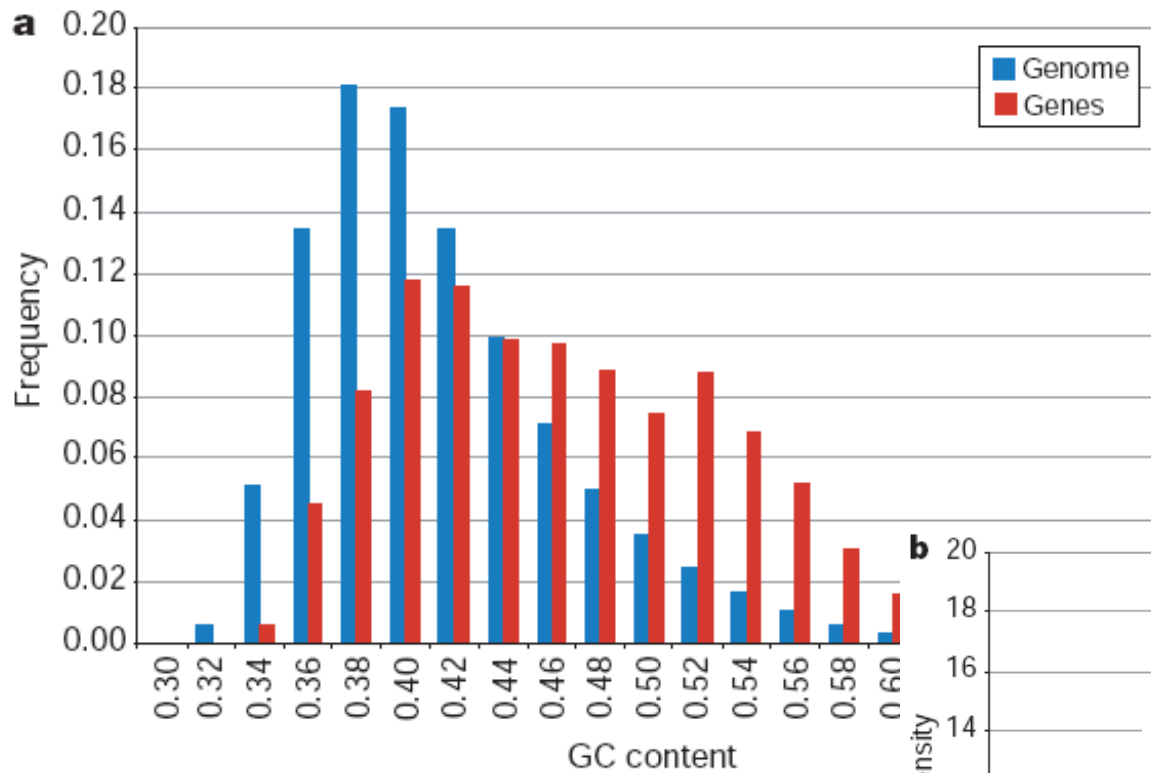
Complexity Eukaryotes

- Finding genes in Eukaryotes is difficult due to variation in gene structure
- **Example:**
 - Average vertebrate gene is 30Kb long out of which coding sequence is only about 1Kb
 - Average coding region consists of 6 exons of about 150bp
 - Dystrophin: 2.4Mb long
 - Blood coagulation factor VIII: 26 exons (69bp to 3106bp)
 - Intron 22 produces 2 transcripts unrelated to this gene.

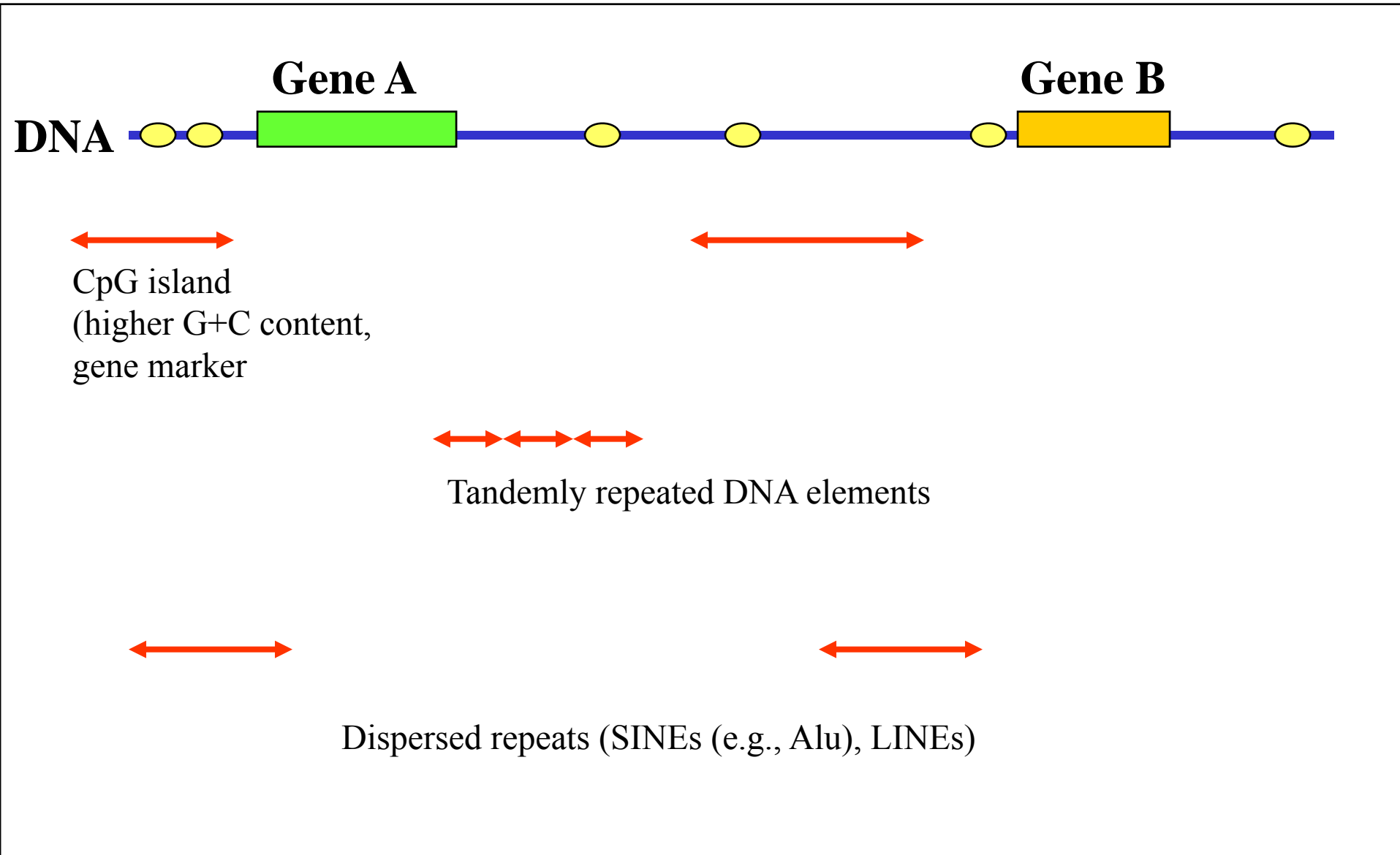
Intron and exon length



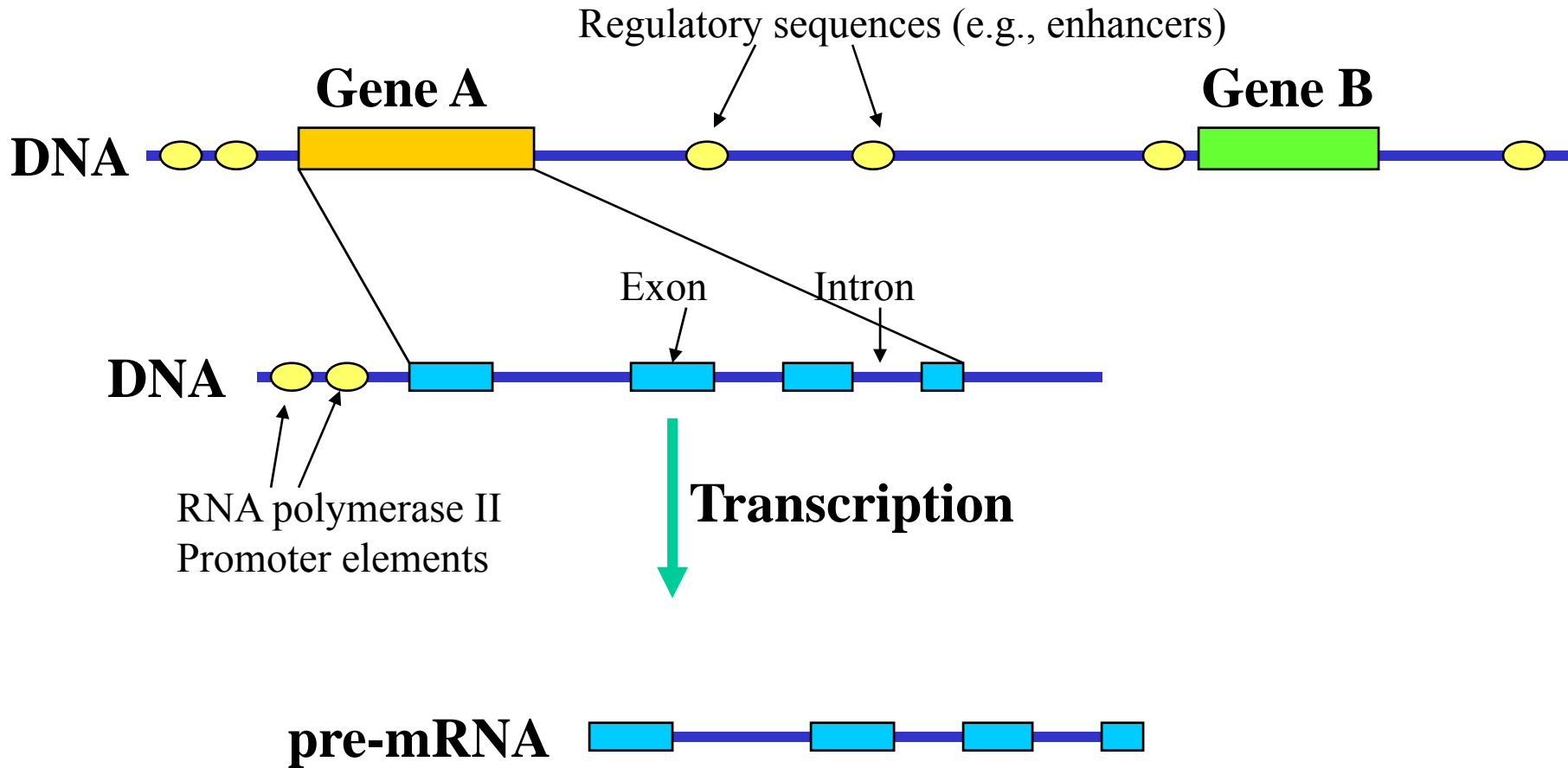
GC - content



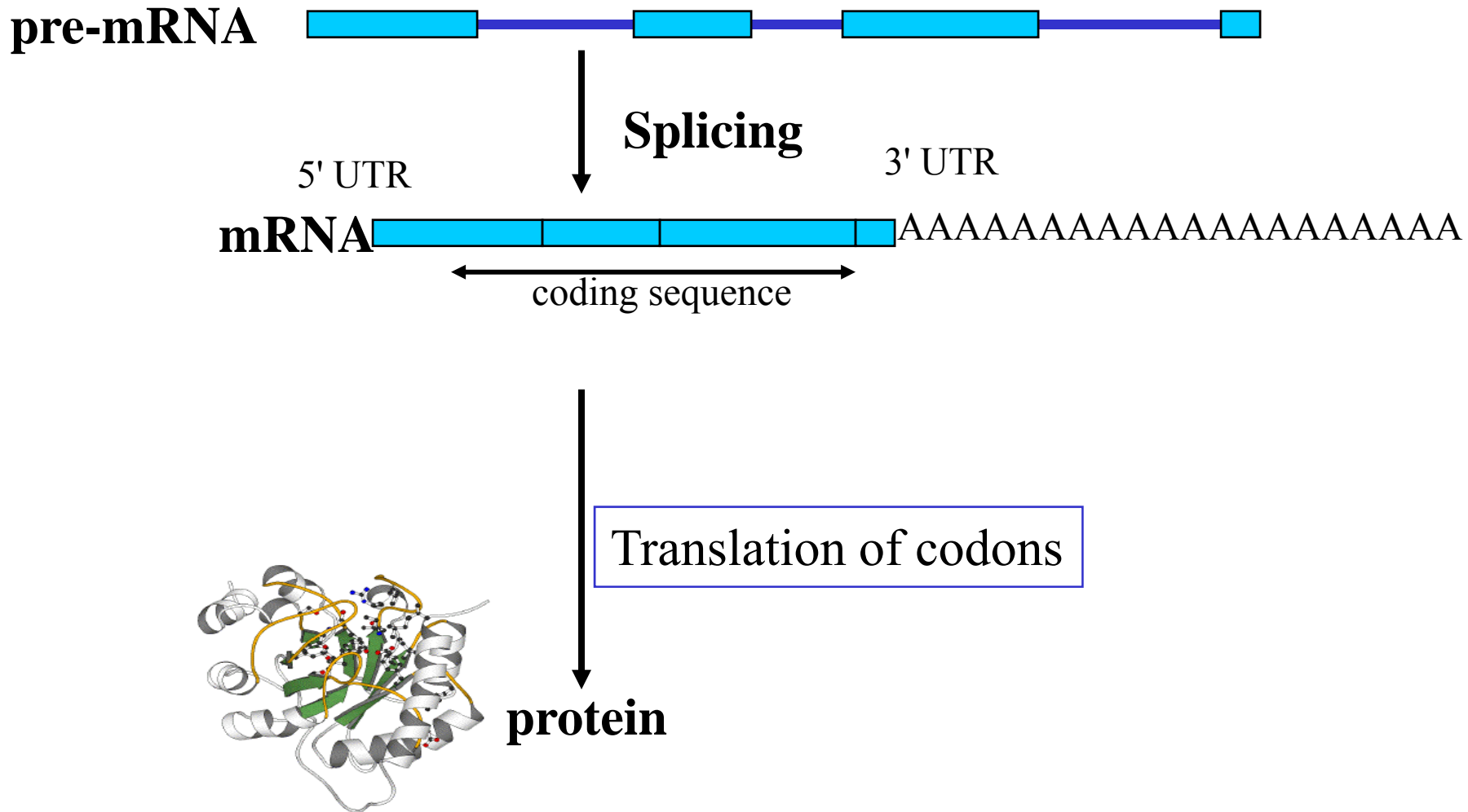
Eukaryotic genome structure



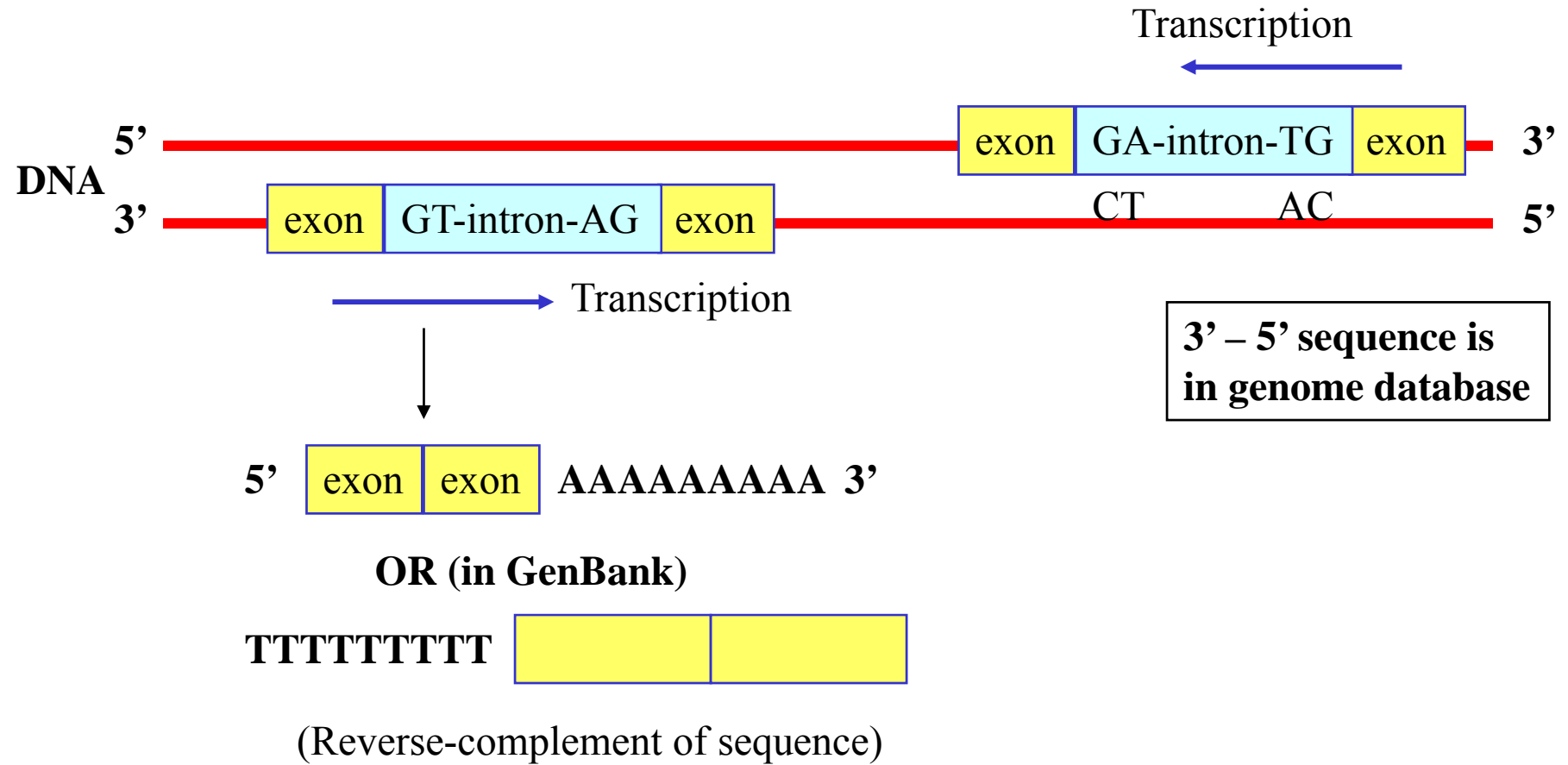
Eukaryotic genome structure



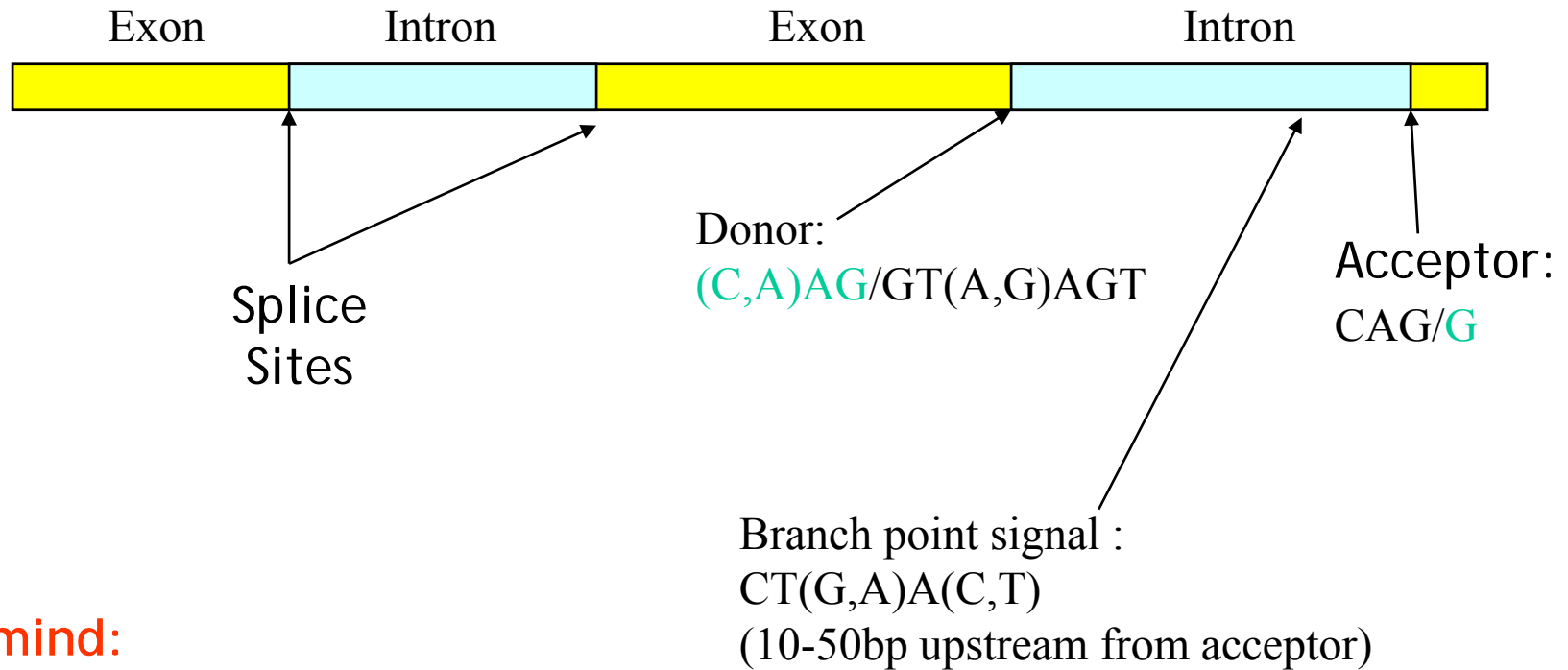
Eukaryotic genome structure



Exon - Intron structure



Exon - Intron structure



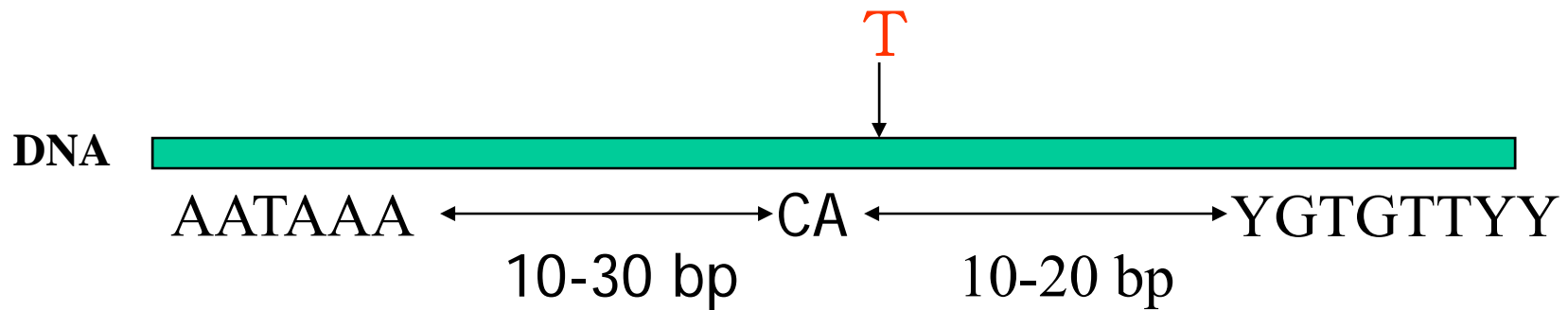
Remind:

- Alternative splicing
- AT-AC Introns (rare)

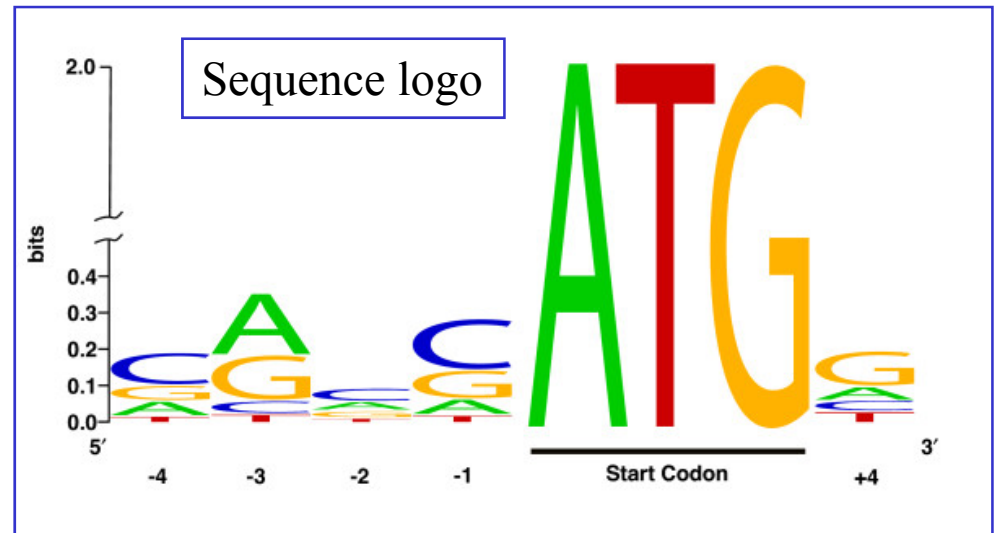
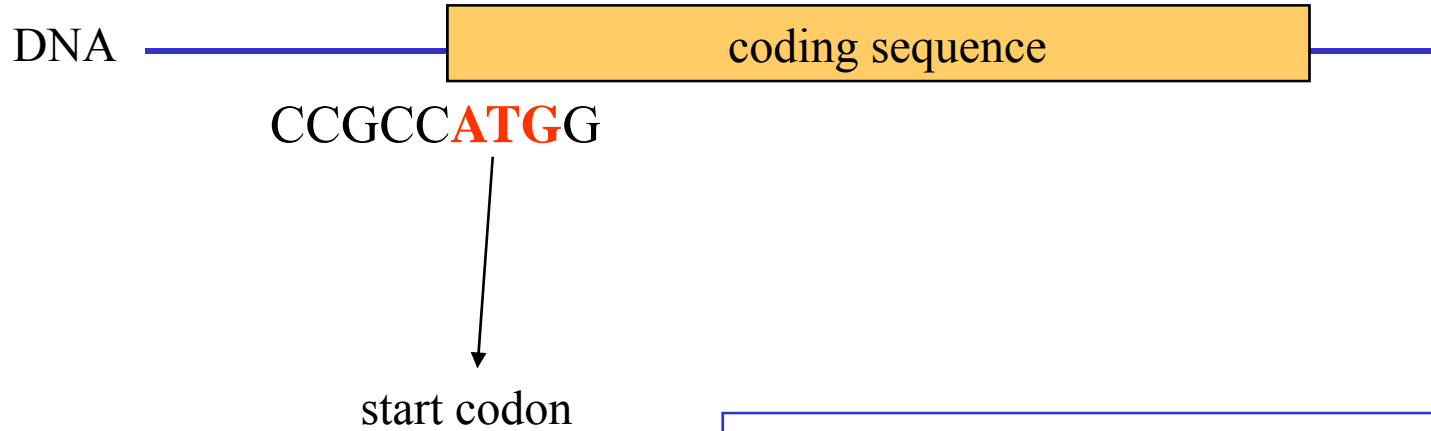
Polyadenylation signal

Eukaryotic mRNAs are polyadenylated, i.e., have up to 250 A's added to their 3' end after transcription terminates.

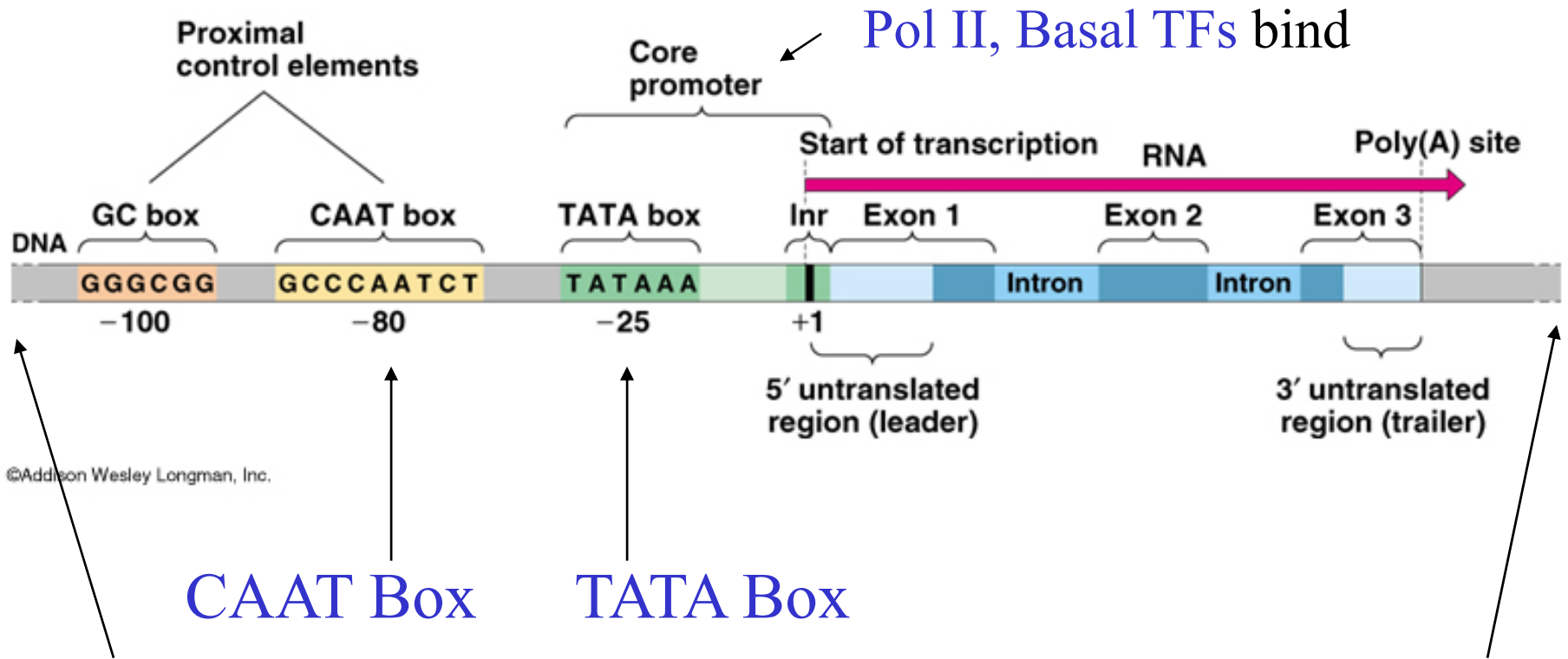
Signals:



Kozak sequence

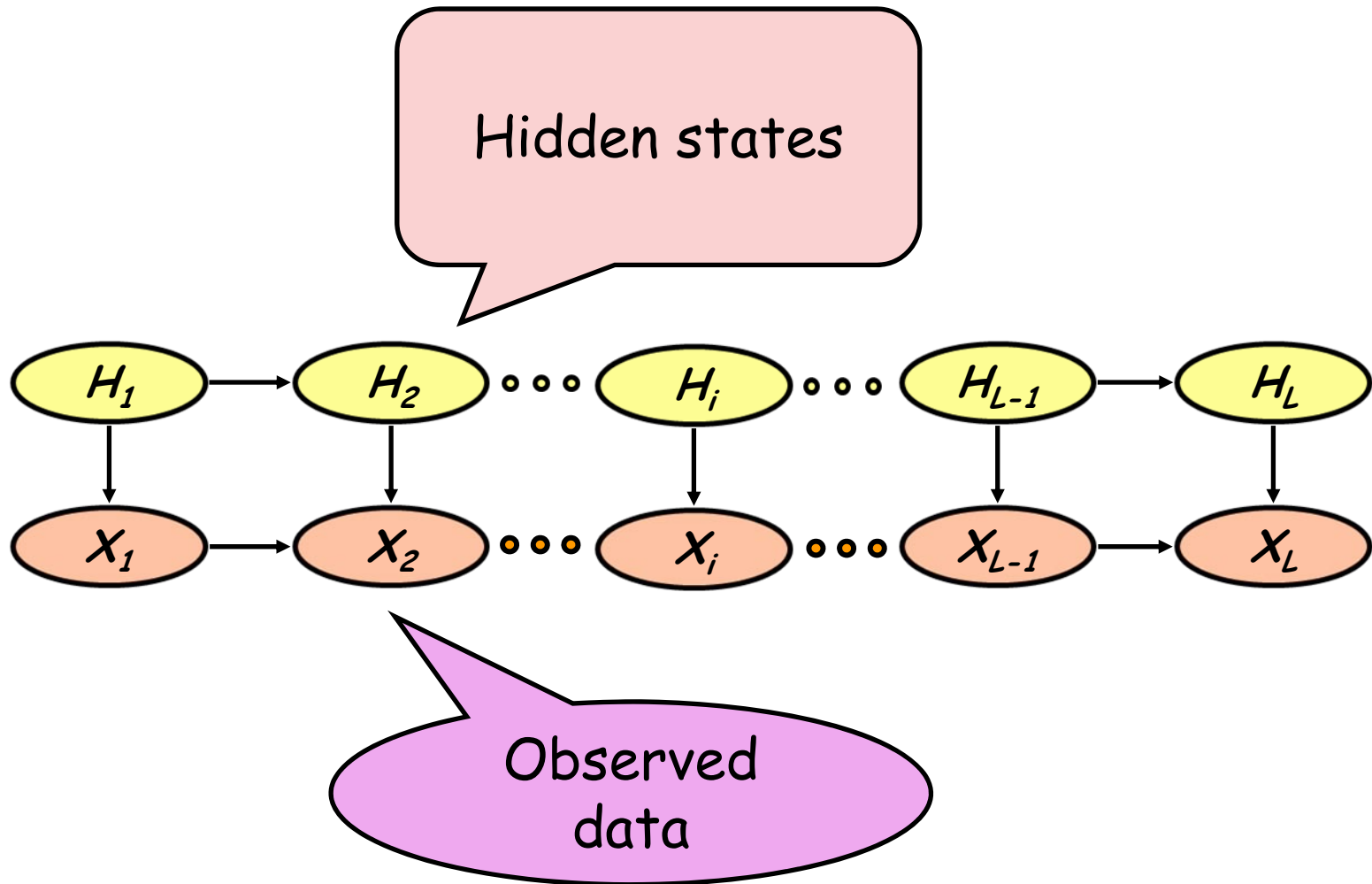


Anatomy of a Eukaryotic Gene



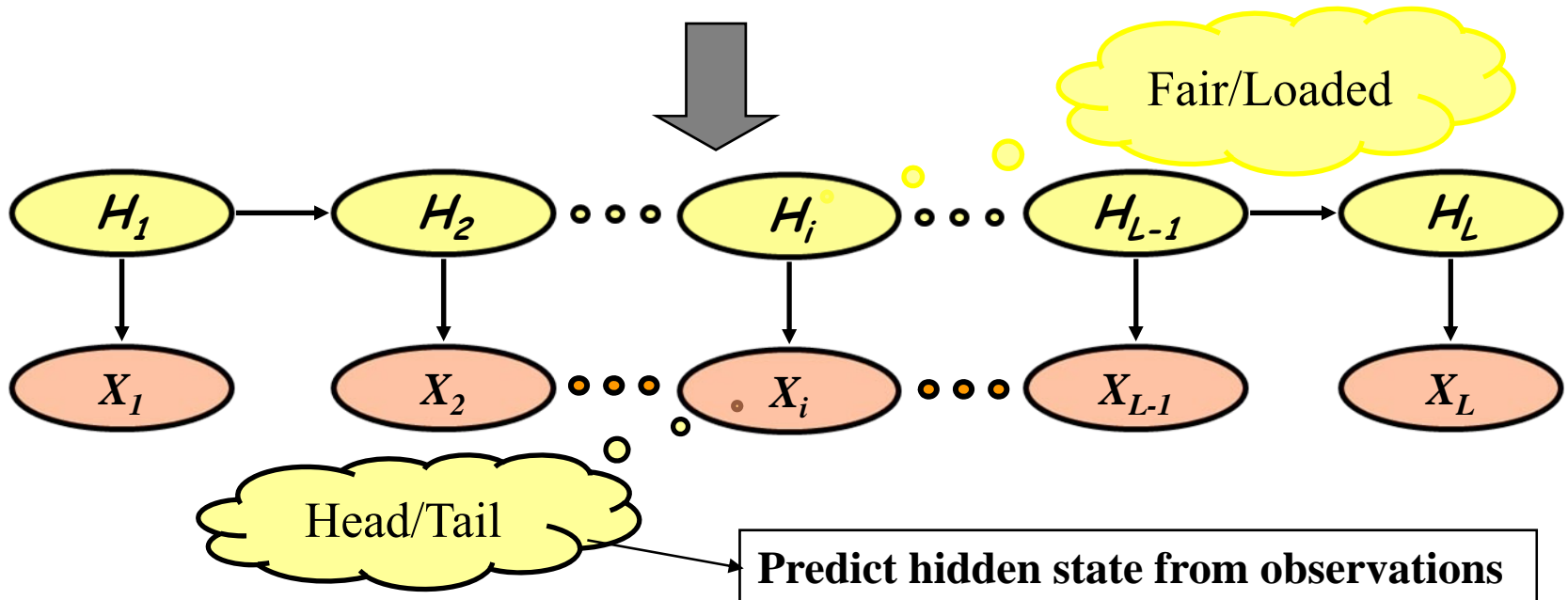
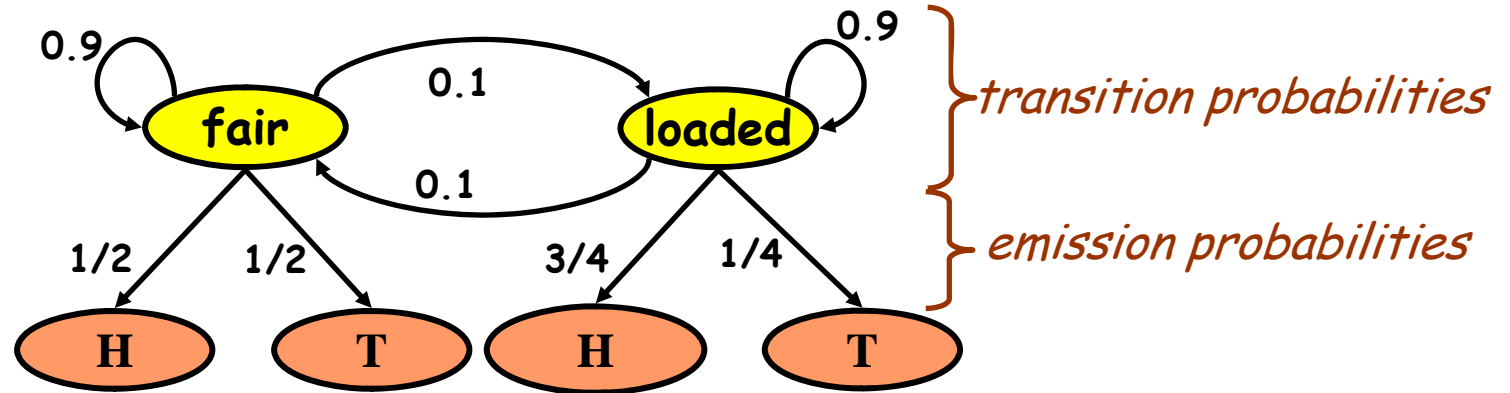
Cis-regulatory Elements may be located thousands of bases away; Regulatory TFs bind.

Hidden Markov Models - HMM



Hidden Markov Models - HMM

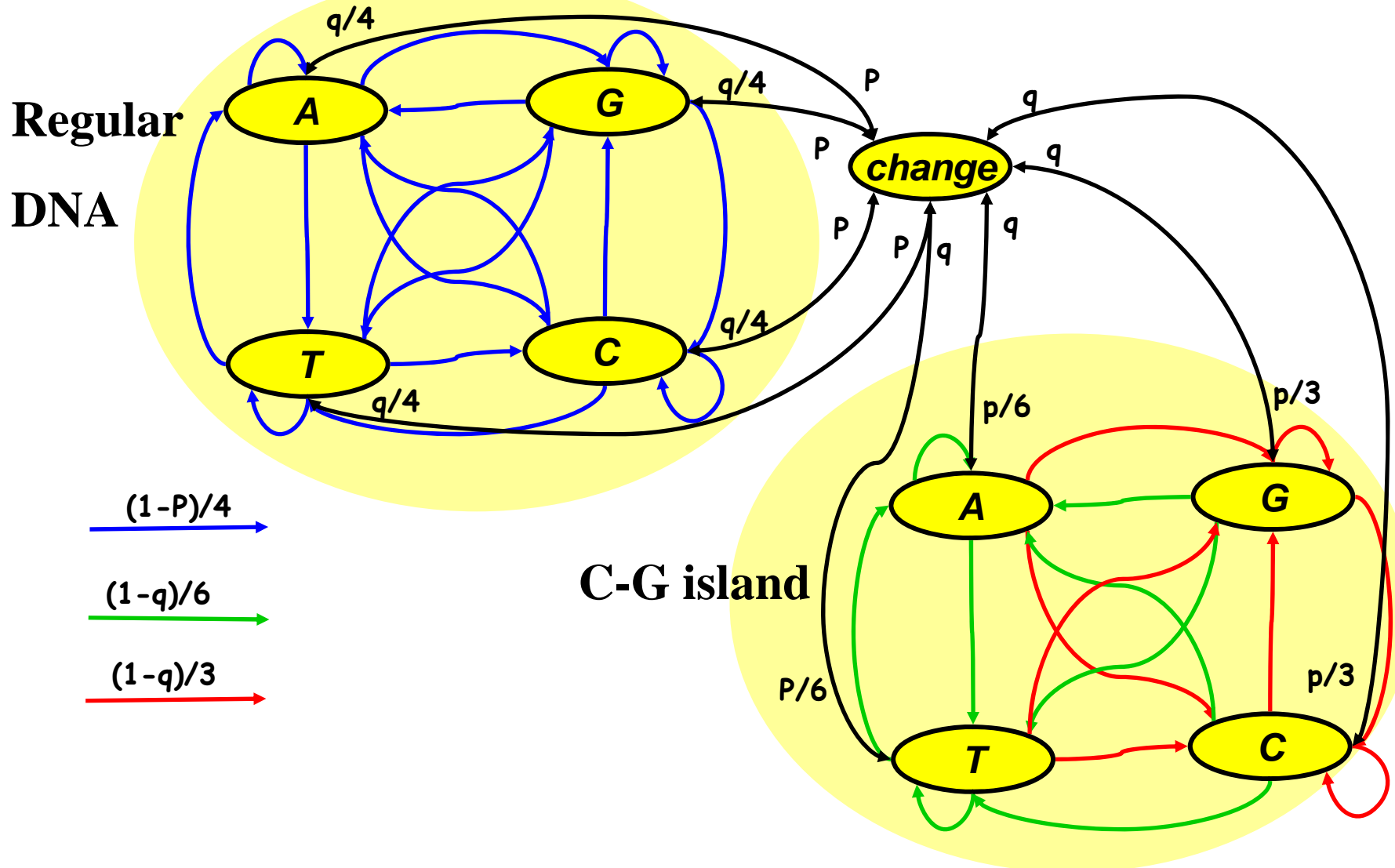
Coin-Tossing Example



Hidden Markov Models - HMM

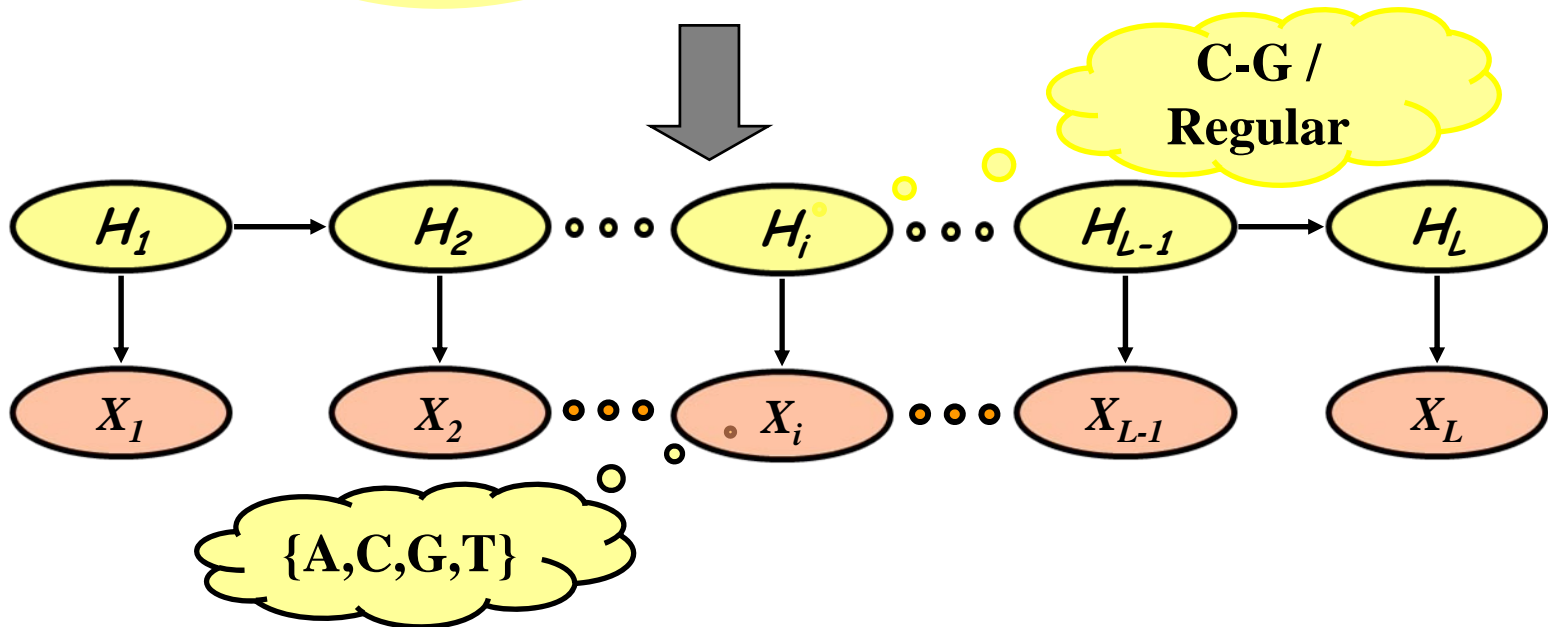
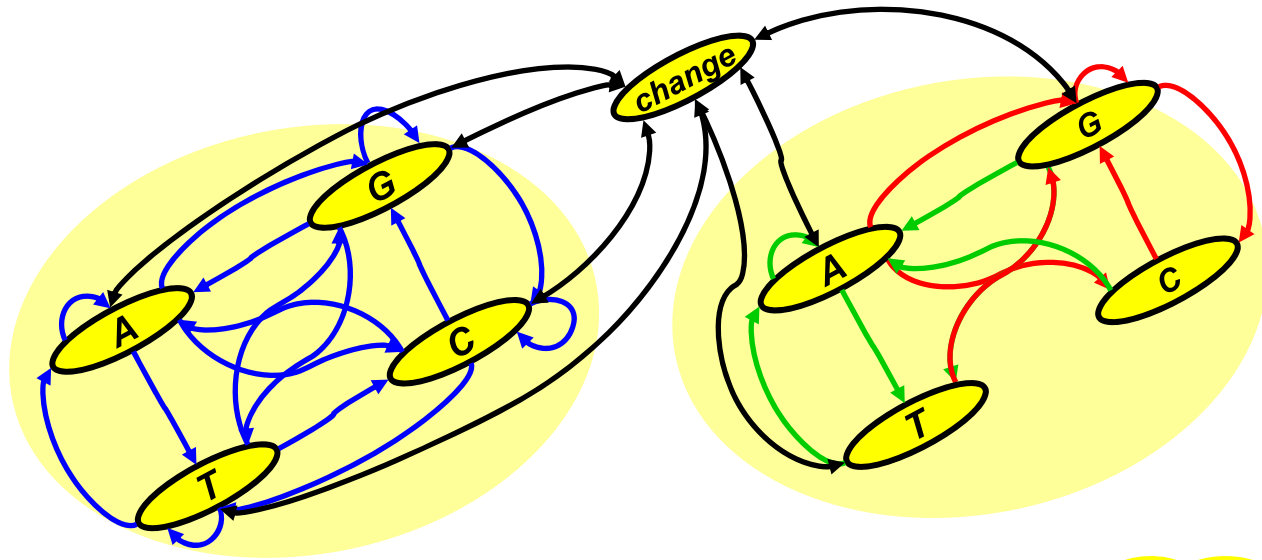
C-G Islands Example

C-G islands: Genome regions which are very rich in C and G

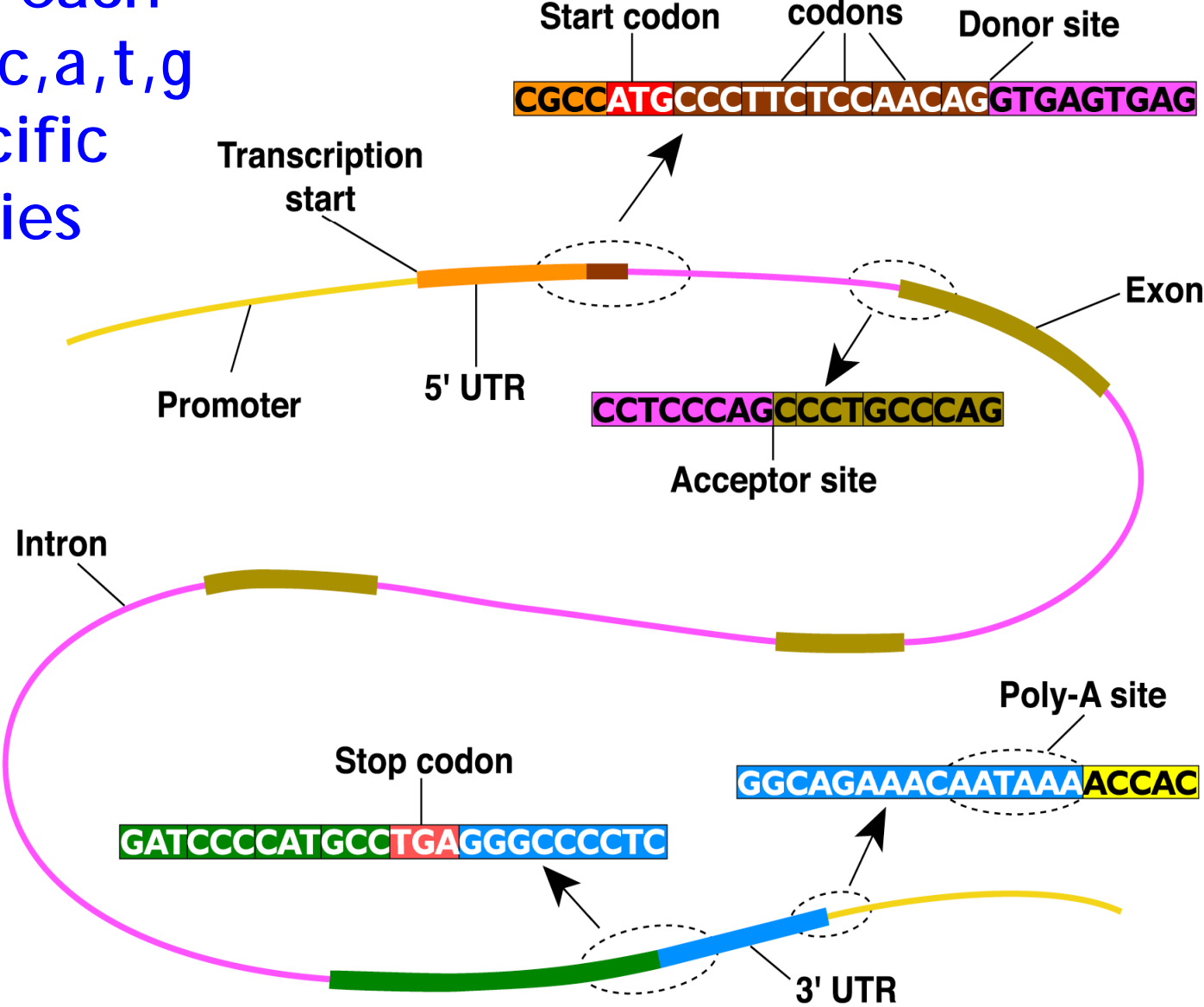


Hidden Markov Models - HMM

C-G Islands Example



States (colors)
in a gene each
emitting c,a,t,g
with specific
frequencies



Hidden Markov Models (HMMs)

Hidden State

Distinguish between the *observed parts* of a problem and the *hidden parts*

In the Markov models we have considered previously, it is clear which state accounts for each part of the observed sequence

In the HMM there are *multiple states that could account for each part of the observed sequence*

- this is the hidden part of the problem
- states are decoupled from sequence symbols

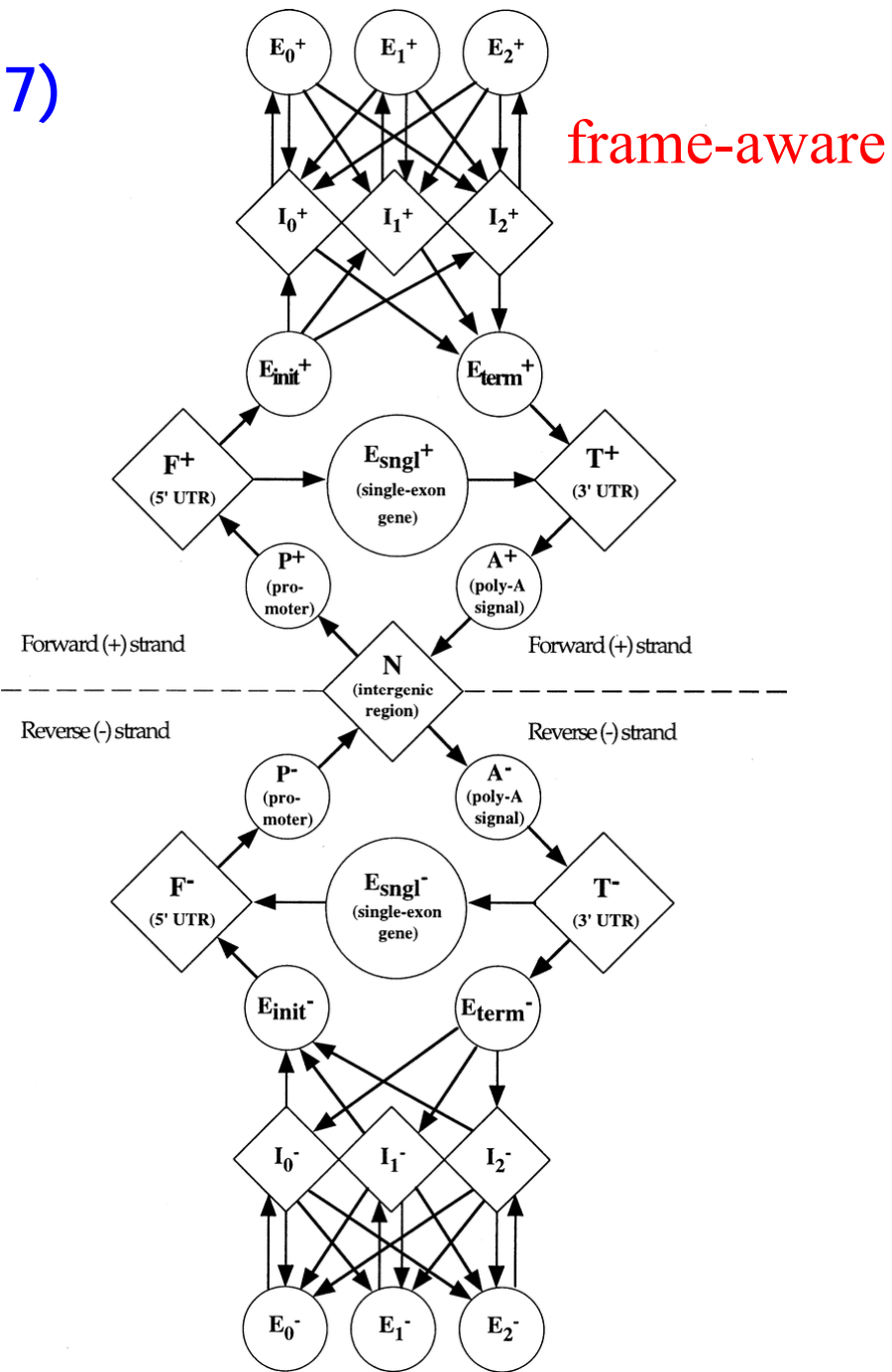
GeneScan (Burge & Karlin, 1997)

Generalized HMM (GHMM)

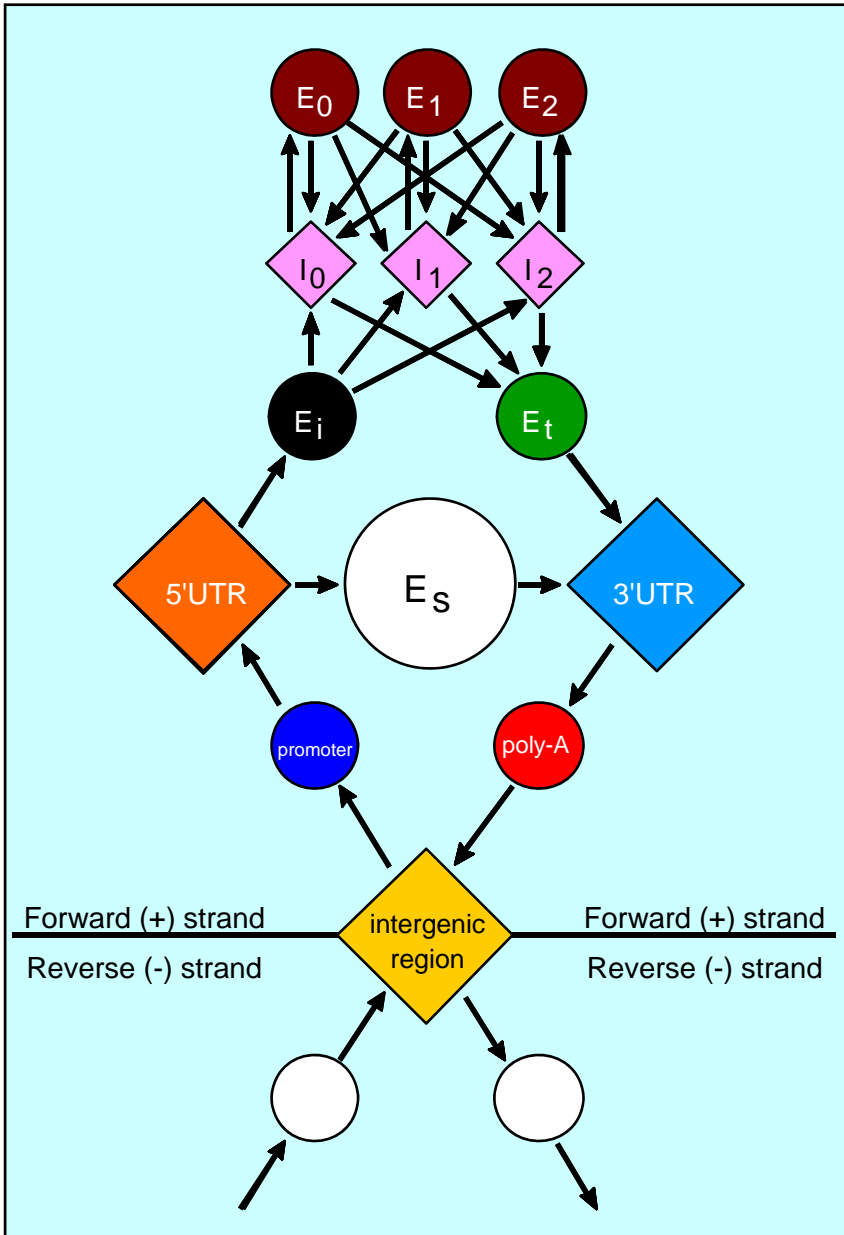
Each state may output a **string of symbols** (according to some probability distribution).

both strands

Exons: separate states for initial, terminal, single and internal exons



GENESCAN



62001	AGGACAGGTA	CGGCTGTCAT	CACTTAGACC	TCACCCTGTG	GAGCCACACC	
62051	CTAGGGTTGG	CCAATCTACT	CCCAGGAGCA	GGGAGGGCAG	GAGCCAGGGC	
62101	TGGGCATAAA	AGTCAGGGCA	GAGCCATCTA	TTGCTTACAT	TTGCTTCTGA	
62151	CACAACGTGT	TTCACTAGCA	ACCTCAAACA	GACAC		
62201						
62251				GGT TGGTATCAAG	GTTACAAGAC	
62301	AGGTTTAAGG	AGACCAATAG	AAACTGGGCA	TGTGGAGACA	GAGAAGACTC	
62351	TTGGGTTTCT	GATAGGCACT	GACTCTCTCT	GCCTATTGGT	CTATTTTCCC	
62401	ACCTTAGGC	TGCTGGTGGT	CTACCCTTGG	ACCCAGAGGT	TCTTTGAGTC	
62451	CTTTGGGGAT	CTGTCCACTC	CTGATGCTGT	TATGGGCAAC	CCTAAGGTGA	
62501	AGGCTCATGG	CAAGAAAGTG	CTCGGTGCCT	TTAGTGATGG	CCTGGCTCAC	
62551	CTGGACAACC	TCAAGGGCAC	CTTTGCCACA	CTGAGTGAGC	TGCACTGTGA	
62601	CAAGCTGCAC	GTGGATCCTG	AGAACTTCAG	GGTGAGTCTA	TGGGACCCTT	
62651	GATGTTTCT	TTCCCCTTCT	TTTCTATGGT	TAAGTTCATG	TCATAGGAAG	
62701	GGGAGAAGTA	ACAGGGTACA	GTTTAGAATG	GGAACAGAC	GAATGATTC	
62751	ATCAGTGTGG	AAGTCTCAGG	ATCGTTTTAG	TTTCTTTTAT	TTGCTGTICA	
62801	TAACAATGT	TTTCTTTTGT	TAAATCTGTG	CTTCTTTTTT	TTTTCTTCTC	
62851	CGCAATTTTT	ACTATTATAC	TTAATGCCTT	AACATTGTGT	ATAACAAAAG	
62901	GAAATATCTC	TGAGATACAT	TAAGTAACTT	AAAAAAAAC	TTTACACAGT	
62951	CTGCCTAGTA	CATTACTATT	TGGAATATAT	GTGTGCTTAT	TTGCATATTC	
63001	ATAATCTCCC	TACTTTATTT	TCTTTTATTT	TTAATTGATA	CATAATCATT	
63051	ATACATATTT	ATGGGTTAAA	GTGTAATGTT	TTAATATGTG	TACACATATT	
63101	GACCAAATCA	GGTAATTTTT	GCATTTGTAA	TTTTAAAAAA	TGCTTTCTTC	
63151	TTTTAATATA	CTTTTTTGTG	TATCTTATTT	CTAATACTTT	CCCTAATCTC	
63201	TTTCTTTCAG	GGCAATAATG	ATACAATGTA	TCATGCCTCT	TTGCACCATT	
63251	CTAAAGAATA	ACAGTGATAA	TTTCTGGGTT	AAGGCAATAG	CAATATTTCT	
63301	GCATATAAAT	ATTTCTGCAT	ATAAAITGTA	ACTGATGTAA	GAGGTTTCAT	
63351	ATTGCTAATA	GCAGCTACAA	TCCAGCTACC	ATTCTGCTTT	TATTTTATGG	
63401	TTGGGATAAG	GCTGGATTAT	TCTGAGTCCA	AGCTAGGCC	TTTTGCTAAT	
63451	CAITGTCATA	CCTCTTATCT	TCCCTCCACA	CCTCCTGGGC	AACGTGCTGG	
63501	TCTGTGTGCT	GGCCATCAC	TTTGGCAAAG	AATTCACCCC	ACCAGTGCAG	
63551	GCTGCCTATC	AGAAAGTGGT	GGCTGGTGTG	GCTAATGCC	TGGCCACAA	
63601	GTATCACTAA	GCTCGCTTTC	TTGCTGTCCA	ATTTCTATTA	AAGGTCCCTT	
63651	TGTTCCCTAA	GTCCAACACT	TAAACTGGG	GATATTATGA	AGGGCCTTGA	
63701	GCATCTGGAT	TCTGCC	TAAAT	AAAAACATT	TATTTTCATT	GCAATGATGT

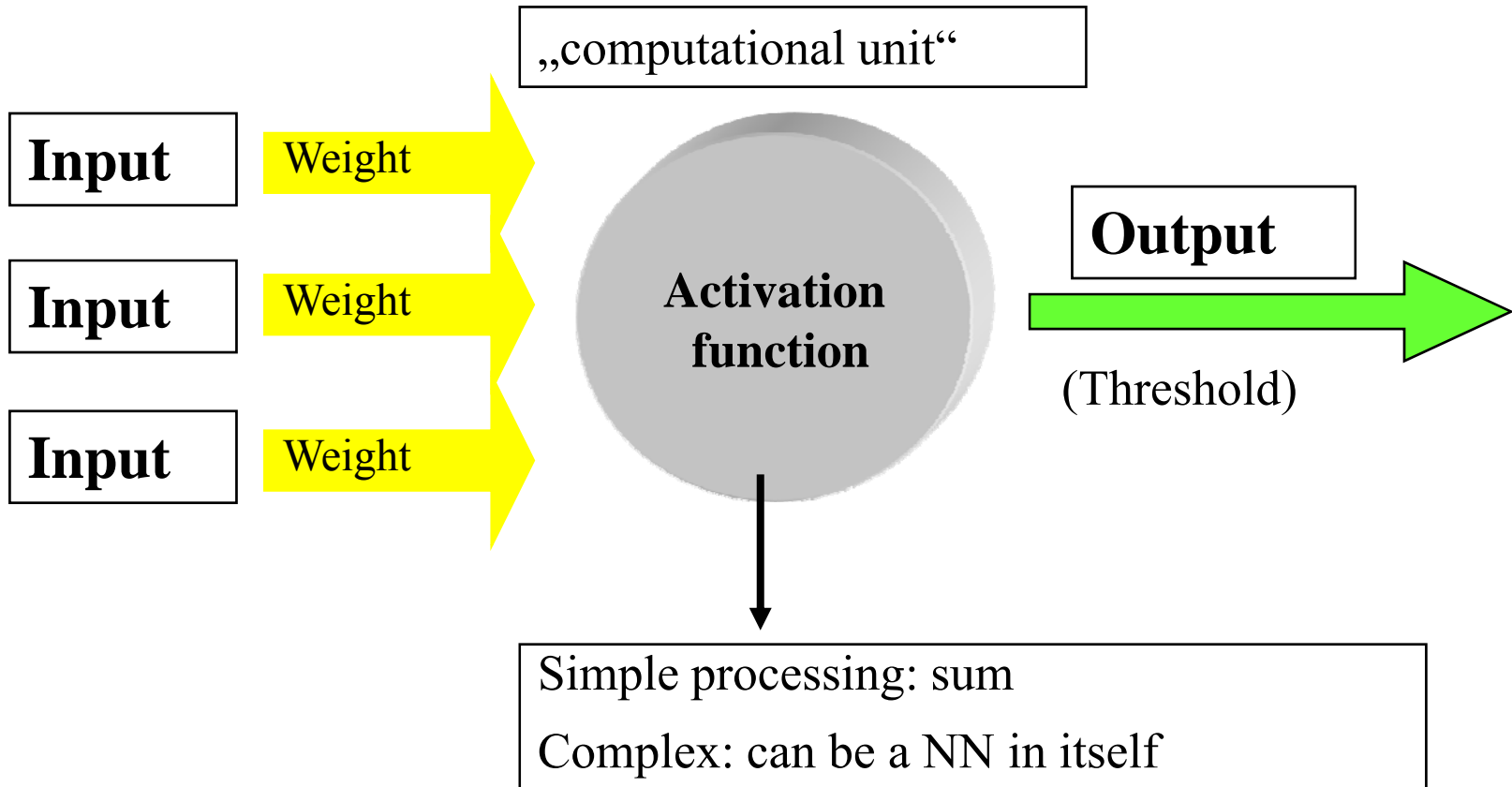
Neural Networks for gene prediction

What are Neural Networks?

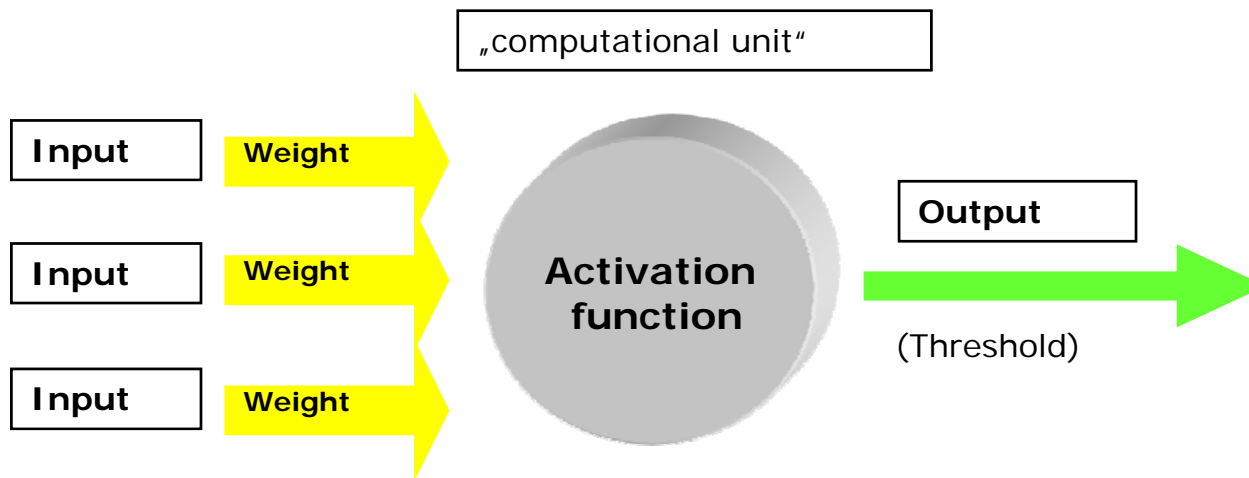
- Neural Network is a computer program that learns to recognize patterns from a given a training set of data. Subsequently, NN are used to determine whether such patterns exist in new data.
- The name derives from the fact that originally they were intended to imitate human brain.

Neural Networks for gene prediction

- Artificial neurons: the nodes of the network



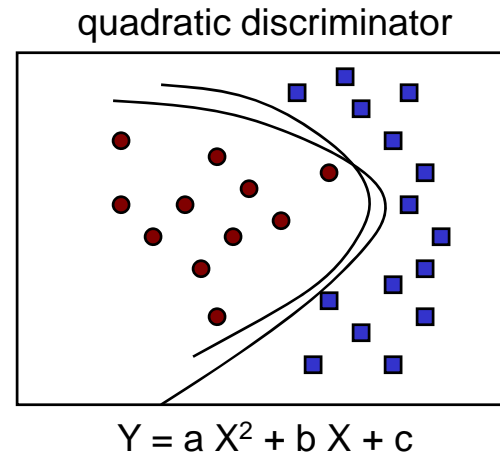
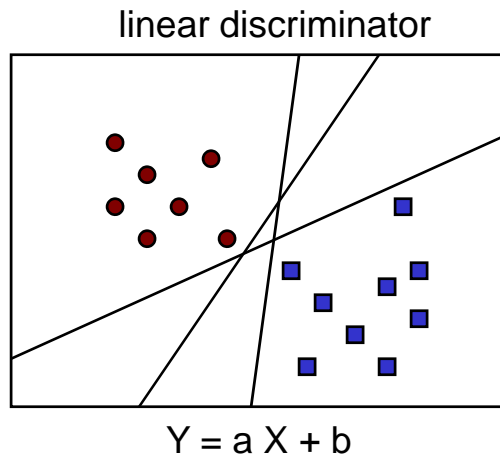
Neural Networks for gene prediction



- **Weighting factor** – A neuron receives many simultaneous inputs. Each input has its own relative weight (w)
- **Summation function** – Processing in the usual artificial neuron consists of computing weighted sum.
- **Transfer function** – the result of the summing function is transferred via transfer function. Transfer function usually compares the weighted sum against some threshold value and may transfer no signal if the value is below the threshold.

Pattern discrimination methods

- Represent a DNA segment as a list of scores (features), including preference score, Markov score, splice junction scores, length, G+C composition,
- Find simple mathematical functions that can best separate different classes of training data



Pattern discrimination methods

Linear discriminant functions

FGENEH

Solovyev VV, Salamov AA, Lawrence CB (1995) Proc Int Conf Intell Syst Mol Biol., 3, 367.

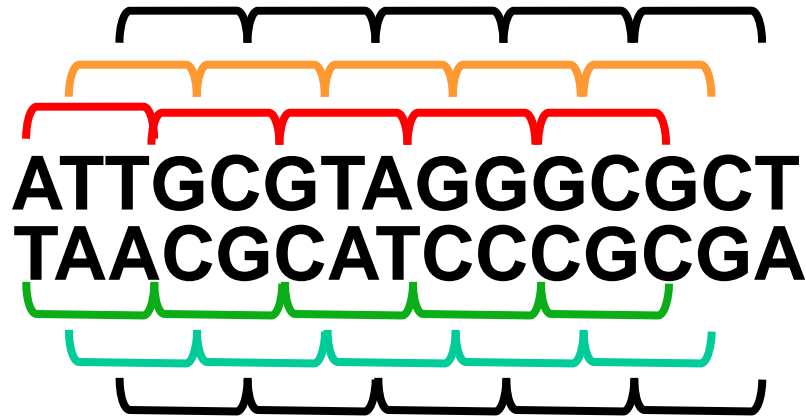
Quadratic discriminant function

MZEF (Michael Zhang's Exon Finder)

Zhang MQ (1997) Proc Natl Acad Sci U S A., 94(2), :565.

Similarity-Based Methods: Database Search

In different genomes: *Translate* DNA into all 6 reading frames and search against proteins (TBLASTX,BLASTX, etc.)



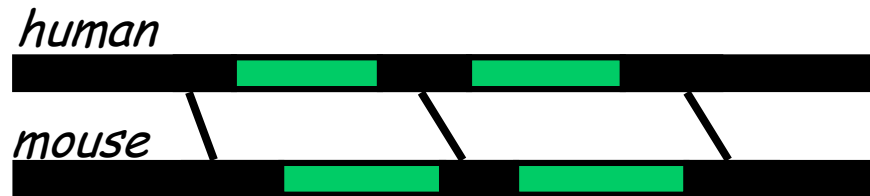
Within same genome: Search with EST/cDNA database (EST2genome, BLAT, etc.).

Problems:

- Will not find “new” or RNA genes (non-coding genes).
- Limits of similarity are hard to define
- Small exons might be overlooked

Similarity-Based Methods: Comparative Genomics

Idea: Functional regions are more conserved than non-functional ones; high similarity in alignment indicates gene



```
GGTTTT--ATGAGTAAAGTAGACACTCCAGTAACGCGGTGAGTAC----ATTAA
|         ||||| ||||| |||         ||||| ||||| ||||| |||||
C-TCAGGAATGAGCAAAGTCGAC---CCAGTAACGCGGTAAGTACATTAAACGA-
```

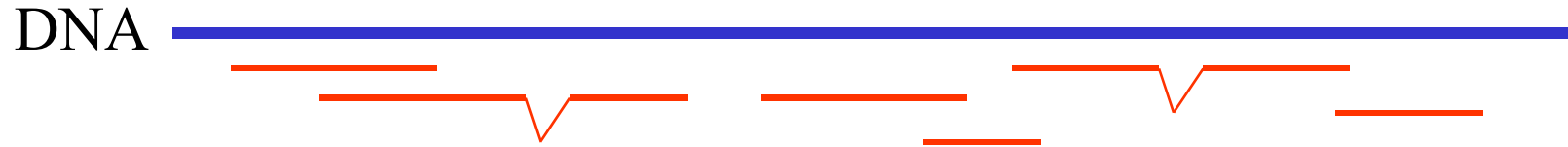
Advantages:

- May find uncharacterized or RNA genes

Problems:

- Finding suitable evolutionary distance
- Finding limits of high similarity (functional regions)

EST2Genome: principle

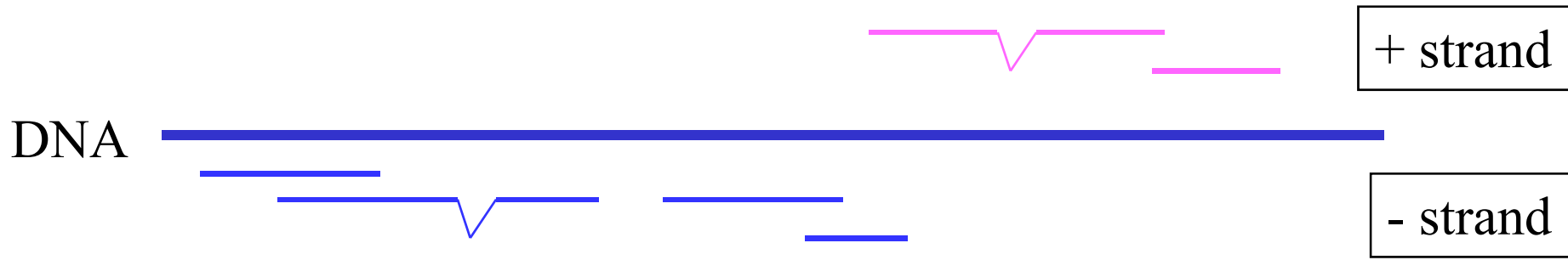


BLAT Alignments of mRNA/ESTs against genome

mRNA / EST sequences from GenBank (NCBI)

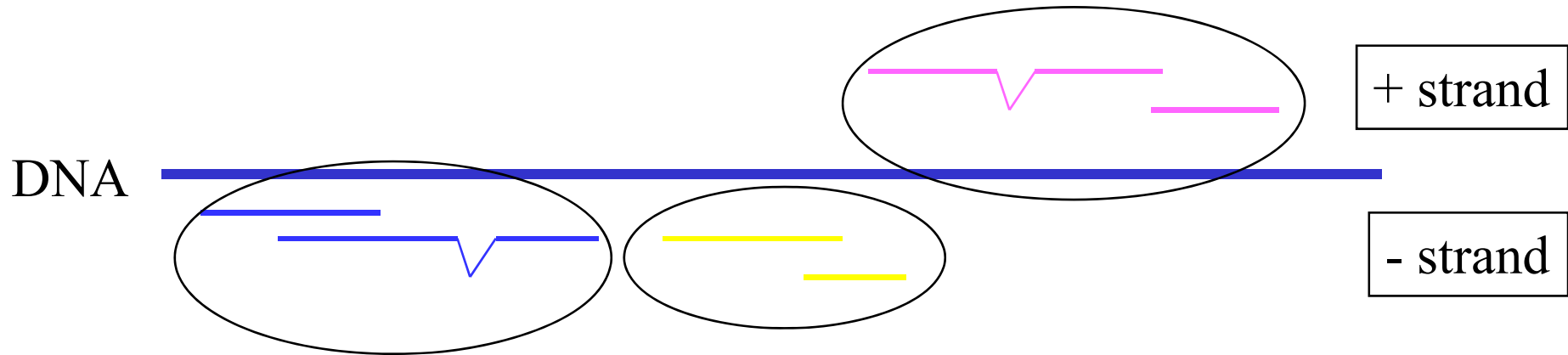
Alignments of these sequences to the genome (UCSC)

EST2Genome: principle



Assign orientation (polyA signal/tail, exon boundaries, annotation)

EST2Genome: principle



Determine overlap: 3 genes

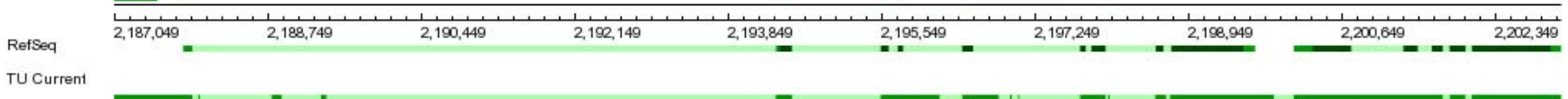
Hybrid transcriptional unit

Erroneous transcriptional unit that contains both the splicing factor 3a, subunit 2 (SF3A2) and anti-Mullerian hormone (AMH) gene.

anti-Mullerian hormone

splicing factor 3a, subunit 2

artifact sequence

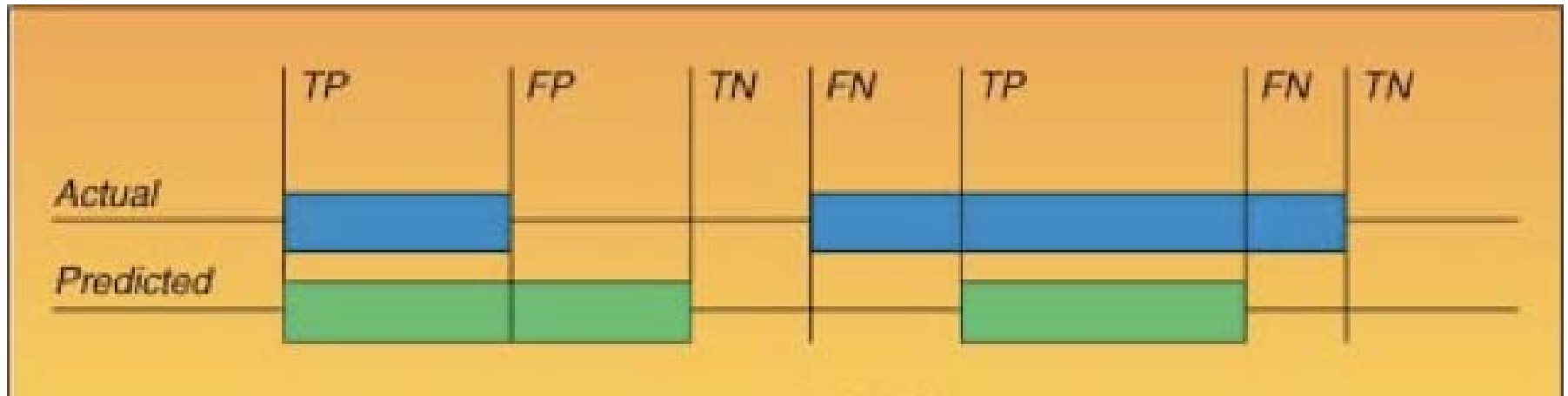


Performance evaluation

Measuring Predictive Accuracy

- Several measures for accuracy have been proposed:
 - Coding nucleotide.
 - For the test sequence the predicted coding value (coding, non-coding) is compared with true coding
 - Exonic structure
 - For test sequence the predicted exons are compared with the true exons.
 - Protein product
 - The predicted and true protein product are compared

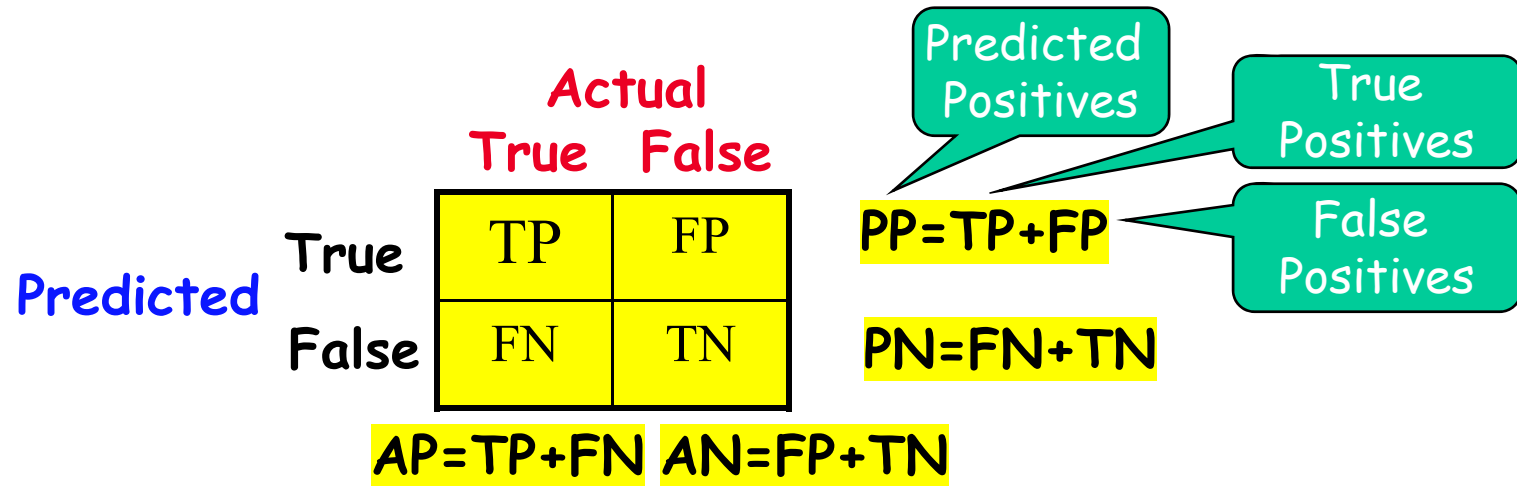
Evaluation of nucleotides



Right!

- TP = positive instance correctly predicted as positive
- FP = negative instance incorrectly predicted as positive
- TN = negative instance correctly predicted as negative
- FN = positive instance incorrectly predicted as negative

Evaluation of Predictions



- **Sensitivity:** $S_n = TP / AP$
- **Specificity:** $S_p = TN / PN$

Evaluation of Predictions

		Actual		
		True	False	
Predicted	True	TP	FP	PP=TP+FP
	False	FN	TN	PN=FN+TN
		AP=TP+FN		AN=FP+TN

- **Sensitivity:** $S_n = TP / AP$ = Coverage

In English? Sensitivity is the fraction of all positive instances having a *true positive prediction*.

IMPORTANT: Sensitivity alone does not tell us much about performance because a *100% sensitivity can be achieved trivially by labeling all test cases positive!*

- **Specificity:** $S_p = TP / PP$ = Recall

In English? Specificity is the fraction of all predicted positives that are, in fact, *true positives*.

Exon evaluation

must have identical start/stop coordinates

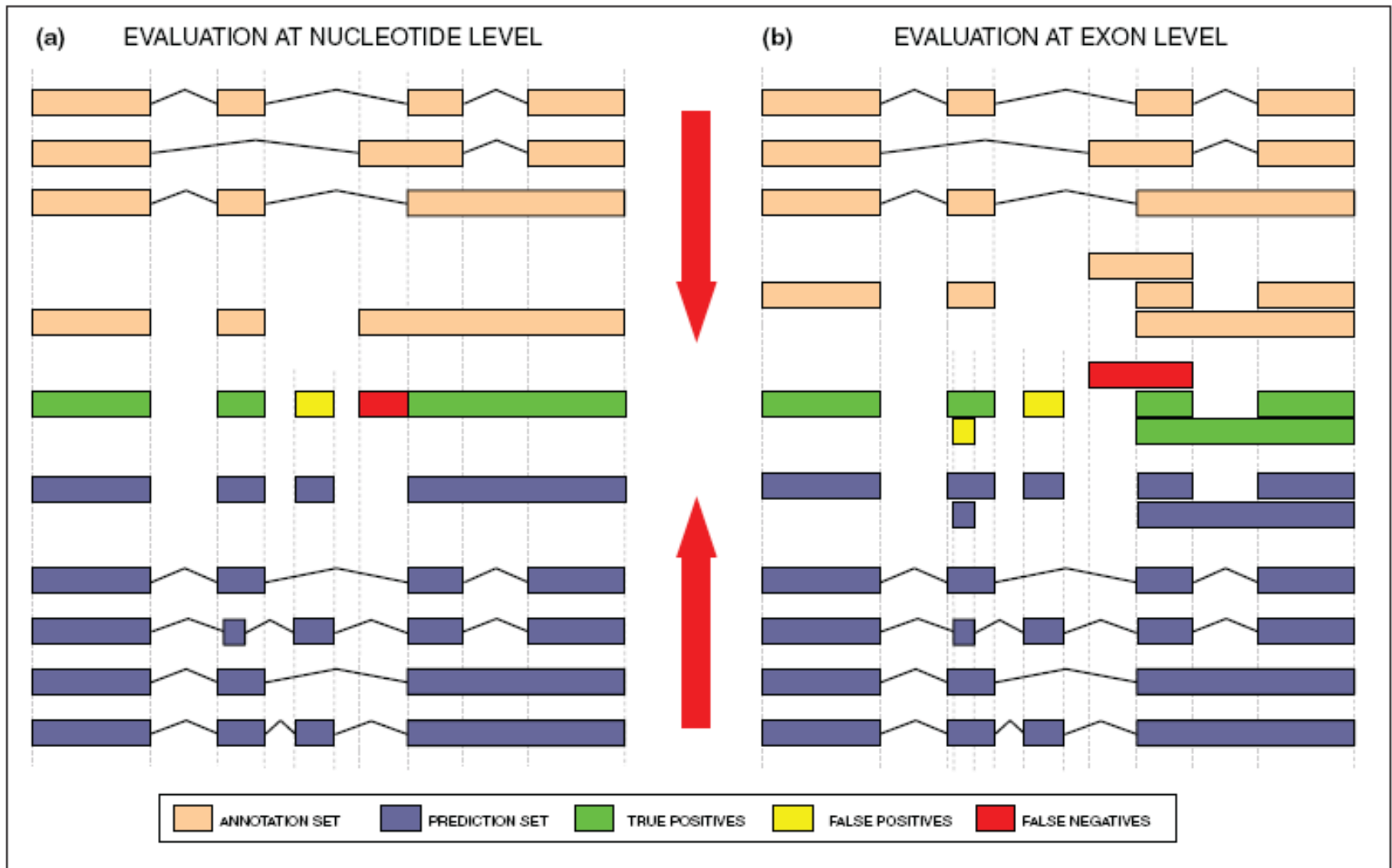
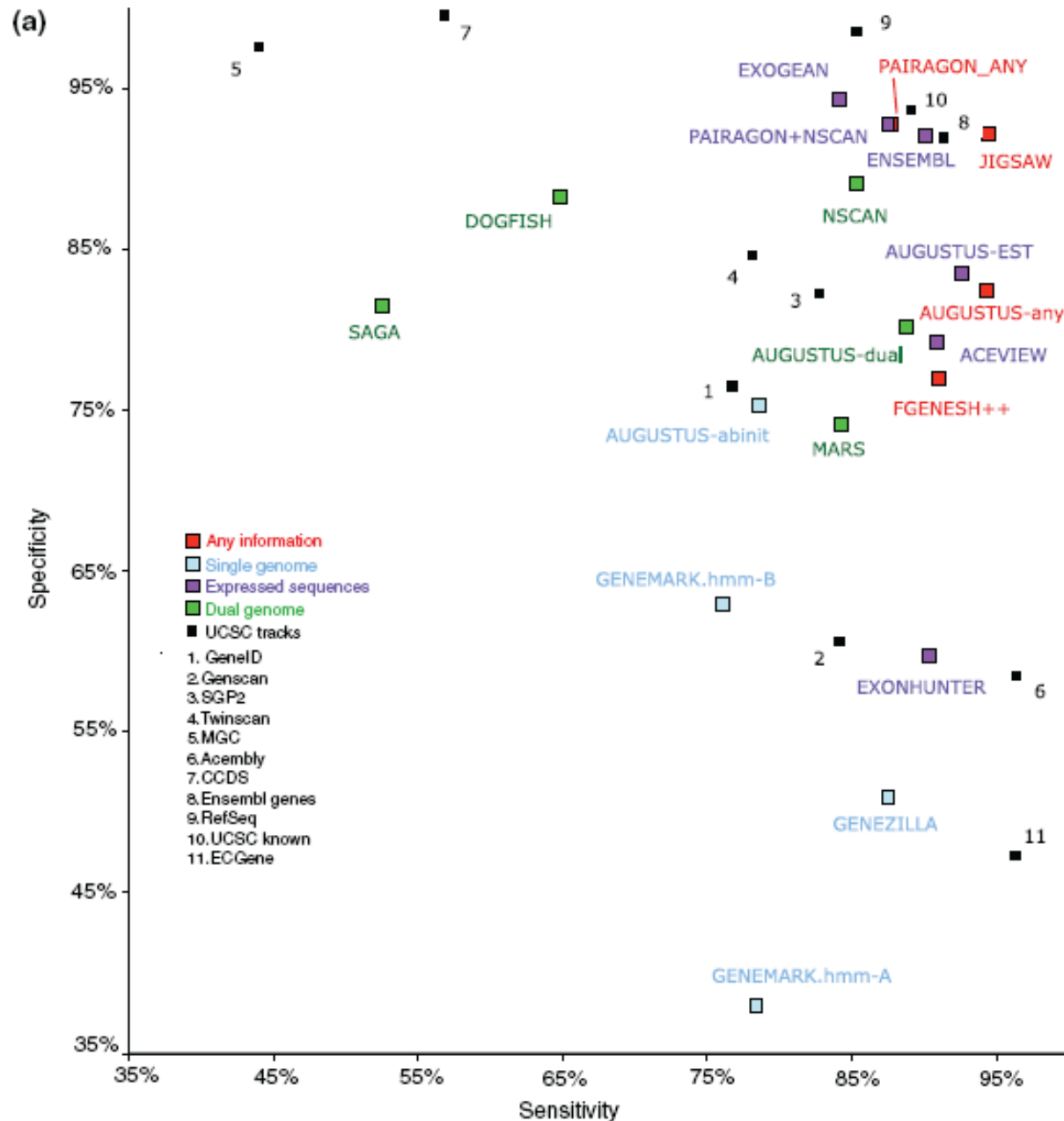


Figure 3

Gene Feature Projection for evaluation. The process of projecting genic features into unique nucleotide and exon coordinates in order to compute the accuracy values (see text for details).

Evaluation of different programs at nucleotide level



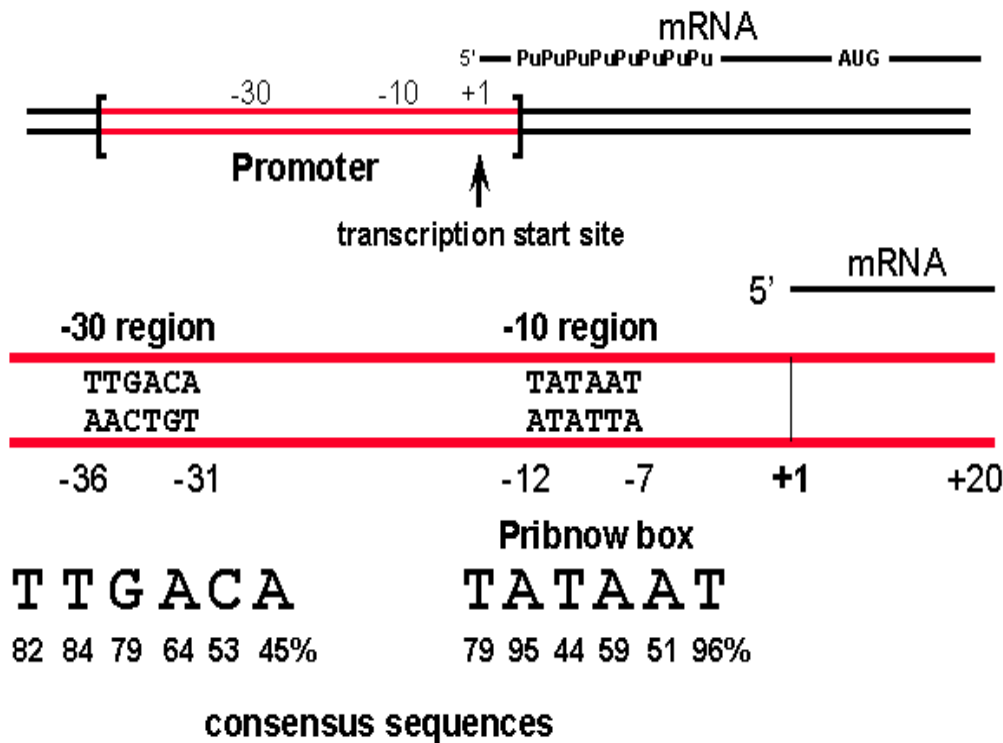
Promoter and Regulatory Element Prediction

Prokaryotic Promoters

- RNA polymerase complex recognizes promoter sequences located very close to and on 5' side (“upstream”) of transcription initiation site
- Prokaryotic RNA polymerase complex binds directly to promoter, by virtue of its *sigma subunit* - no requirement for “transcription factors” binding first
- Prokaryotic promoter sequences are highly conserved:
 - » **-10 region**
 - » **-35 region**

Promoter structure in prokaryotes (E. coli)

Promoter structure in prokaryotes



Transcription starts at offset 0.

- Pribnow Box (-10)
- Gilbert Box (-30)

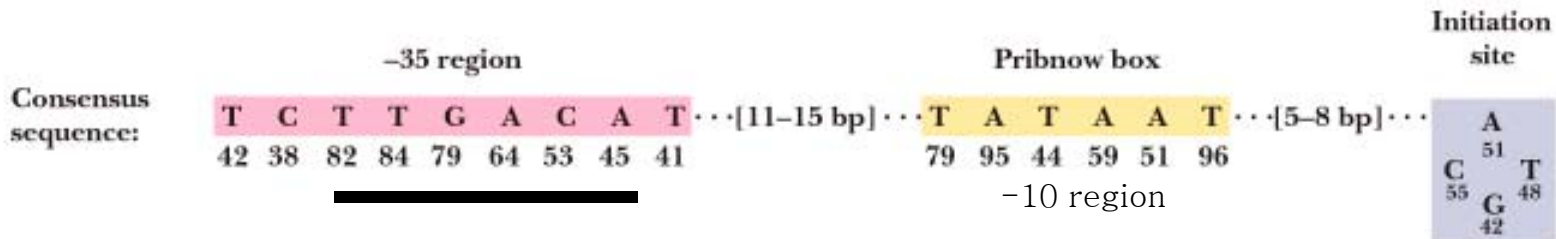
Promoter prediction in E. coli

Method most often used:

1. **Align** set of promoter sequence by the position that marks the known transcription start site
2. Search for **conserved regions**

Promoter structure in prokaryotes (E. coli)

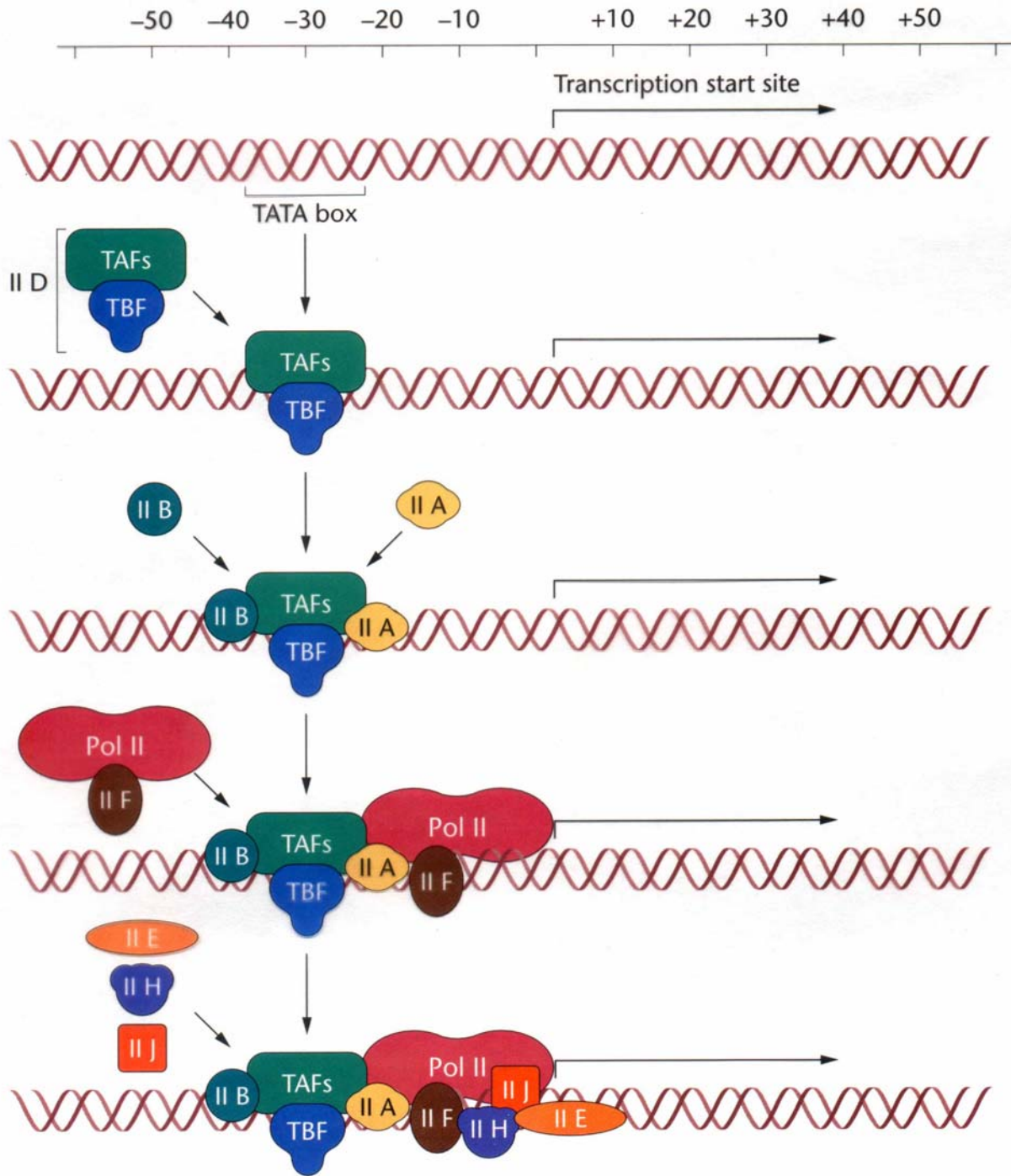
Gene	-35 region	Pribnow box (-10 region)	Initiation site (+1)
<i>araBAD</i>	GGATCCTACCTGACGCTTTT	TTATCGCAACTCTCTACTGTTTCTCCATA	CCCGTTTTT
<i>araC</i>	GCCGTGATTATAGACACTTTT	TGTTACGCGTTTTTGTTCATGGC	TTTGGTCCCGCTTTG
<i>bioA</i>	TTCCAAAACGTGTTTTTTGTTG	TTAATTCGGTGTAGACTTGTAA	ACCTAAATCTTTT
<i>bioB</i>	CATAATCGACTTGTAACCA	AATTGAAAAGATTTAGGTTT	TACAAGTCTACACCGAAT
<i>galP2</i>	ATTTATTCATGTCACACTTTT	TCGCATCTTTGTATGCTATGGTTA	TTTCATACCAT
<i>lac</i>	ACCCAGGCTTTACACTTTA	TGCTTCCGGCTCGTATGTTGTGTG	GAATTGTGAGCGG
<i>lacI</i>	CCATCGAATGGCGCAAAAC	CTTTCGCGGTATGGCATGATAGCG	CCCGGAAGAGAGTC
<i>rrnA1</i>	AAAATAAATGCTTGACTCTGT	AGCGGGAAGGCGTATTATCACAC	CCCGCGCCGCTG
<i>rrnD1</i>	CAAAAAAATACTTGTGCA	AAAAAATTGGGATCCCTATAATGCG	CCTCCGTTGAGACGA
<i>rrnE1</i>	CAATTTTTCTATTGCGGC	CTGCGGAGAACTCCCTATAATGCG	CCTCCATCGACACGG
<i>tRNA^{Tyr}</i>	CAACGTAAACACTTTACAG	CGGCGCGTCAATTTGATATGATGCG	CCCCGCTTCCCGATA
<i>trp</i>	AAATGAGCTGTTGACAATTA	AATCATCGAACTAGTTAACTAGTAC	GCAAGTTTCACGTA



TTGACA...16-19 bp... TATAAT
 “-35” spacer “-10”

Eukaryotic Promoters

- Eukaryotic RNA polymerase complexes do not bind directly to promoter sequences
- Transcription factors must bind first and serve as landmarks recognized by RNA polymerase complexes
- Eukaryotic promoter sequences are less highly conserved, but many promoters (for RNA polymerase II) contain :
 - » -30 region "TATA" box
 - » -100 region "CCAAT" box



Describing signals by consensus sequences

Consensus sequence - sequence representative of a sequence alignment

Often: The most common base in each position of an alignment

Use IUPAC-IUB code to describe ambiguous nucleotides in the consensus

A,T,G,C

R=A or G

K=G or T

H=A or T or C

Y=T or C

M=A or C

D=A or T or G

S=G or C

B=T or G or C

N=A or T or G or C

W=A or T

V=A or G or C

TATA box

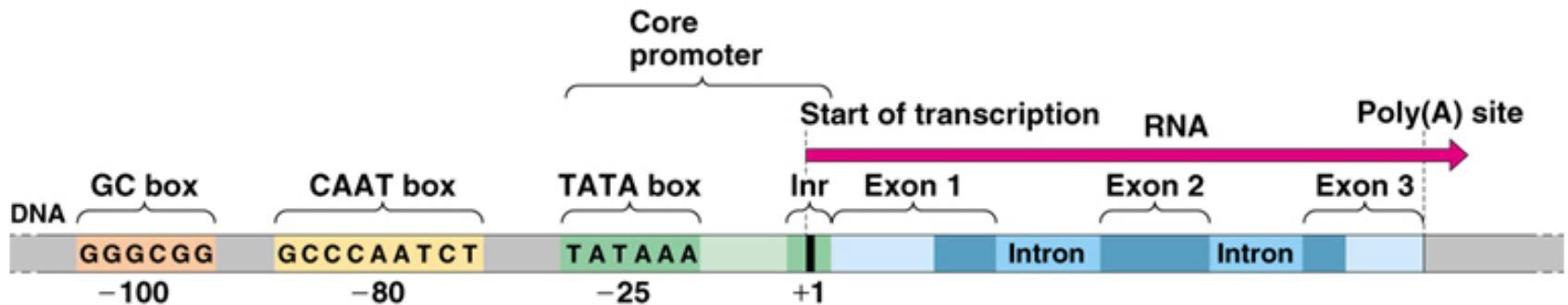
STWTTWAWRSSSSSS

-25

STWTTWAWRSSSSSS

GTATAAAAGGGGGG

C T TT TGCCCCCC



Automated Promoter Prediction Strategies

- 1) Pattern-driven algorithms (ab initio)
- 2) Sequence-driven algorithms (homology based)
- 3) Combined "evidence-based"

BEST RESULTS? Combined, sequential

Pattern-driven Algorithms

- Success depends on availability of collections of annotated transcription factor binding sites (TFBSs)
- Tend to produce very large numbers of false positives (FPs)
- Why?
 - Binding sites for specific TFs are often **variable**
 - Binding sites are **short** (typically 6-10 bp)

EXAMPLE

bicoid
(transcription
factor) site
in fly hunchback
genes

DmA1	C	G	T	A	A	T	C	C	C	C	A	T	A	G
DmA2	T	C	T	A	A	T	C	C	A	G	A	A	T	G
DmA3	T	C	T	A	A	T	C	C	C	T	T	G	A	C
Dmx1	G	C	T	A	A	G	C	T	G	G	C	C	A	T
Dmx2	G	C	T	A	A	G	C	T	C	C	C	G	G	A
Dmx3	G	A	T	C	A	T	C	C	A	A	A	T	C	C
Dmx4	C	T	C	A	A	T	C	C	G	C	G	A	T	C
DvA1	T	C	T	A	A	T	C	T	G	C	A	T	A	G
DvA2	T	C	T	A	A	T	C	C	A	G	T	T	T	G
DvA3	T	C	T	A	A	T	C	C	C	T	T	G	A	C
Dvx3	G	A	T	C	A	T	C	C	A	A	A	T	C	C
Mda	C	T	T	A	A	T	G	G	C	A	A	T	A	T
Mdb	A	T	T	G	A	T	C	C	C	T	T	T	T	T
Mdc	T	T	T	A	A	T	C	C	A	T	T	T	C	T
Mdd	C	T	T	A	A	C	T	T	C	G	A	A	G	C
Mde	C	T	T	A	A	C	G	G	C	A	A	C	A	C
Mdf	G	C	T	A	A	T	C	T	T	G	G	C	G	A
Mdg	T	T	T	A	A	T	C	C	A	T	T	C	T	C
Mdh	T	T	T	G	A	T	C	C	A	G	A	C	T	A
Mdi	T	C	T	A	A	T	C	T	A	C	T	C	T	G
Mdj	T	C	T	A	A	T	C	T	C	G	T	G	T	G

Could also be
described by
consensus
sequence

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

EXAMPLE

bicoid
(transcription
factor) site
in fly hunchback
genes

DmA1	C	G	T	A	A	T	C	C	C	C	A	T	A	G
DmA2	T	C	T	A	A	T	C	C	A	G	A	A	T	G
DmA3	T	C	T	A	A	T	C	C	C	T	T	G	A	C
Dmx1	G	C	T	A	A	G	C	T	G	G	C	C	A	T
Dmx2	G	C	T	A	A	G	C	T	C	C	C	G	G	A
Dmx3	G	A	T	C	A	T	C	C	A	A	A	T	C	C
Dmx4	C	T	C	A	A	T	C	C	G	C	G	A	T	C
DvA1	T	C	T	A	A	T	C	T	G	C	A	T	A	G
DvA2	T	C	T	A	A	T	C	C	A	G	T	T	T	G
DvA3	T	C	T	A	A	T	C	C	C	T	T	G	A	C
Dvx3	G	A	T	C	A	T	C	C	A	A	A	T	C	C
Mda	C	T	T	A	A	T	G	G	C	A	A	T	A	T
Mdb	A	T	T	G	A	T	C	C	C	T	T	T	T	T
Mdc	T	T	T	A	A	T	C	C	A	T	T	T	C	T
Mdd	C	T	T	A	A	C	T	T	C	G	A	A	G	C
Mde	C	T	T	A	A	C	G	G	C	A	A	C	A	C
Mdf	G	C	T	A	A	T	C	T	T	G	G	C	G	A
Mdg	T	T	T	A	A	T	C	C	A	T	T	C	T	C
Mdh	T	T	T	G	A	T	C	C	A	G	A	C	T	A
Mdi	T	C	T	A	A	T	C	T	A	C	T	C	T	G
Mdj	T	C	T	A	A	T	C	T	C	G	T	G	T	G
A	1	2	0	17	21	0	0	0	8	4	9	3	7	3
C	5	10	1	2	0	2	18	12	9	5	2	6	3	8
G	5	1	0	2	0	2	2	2	3	7	2	4	3	6
T	10	8	20	0	0	17	1	7	1	5	8	8	8	4
	T	C	T	A	A	T	C	C	C	G	A	T	T	C
Max	10	10	20	17	21	17	18	12	9	7	9	8	8	8
Total	21	21	21	21	21	21	21	21	21	21	21	21	21	21
Perc	48	48	95	81	100	81	86	57	43	33	43	38	38	38

Or describe by
position weight
matrix (PWM)

Position Weight Matrices (PWM)

Instead of using consensus sequences to detect promoter sequence, use the PWM. This allows to account for promoter variability.

Construct table of statistics $f(b,i)$

- frequency of base b at position i in known promoter region

Let $f(b)$ denote the background frequency of base b in the genome

Example: calculate likelihood for TATA-box

Likelihood for TATA-box

$$P(S|S \text{ is a TATA-box}) = \prod_{i=1}^6 f_{B_i,i}$$

$$P(S|S \text{ is not a TATA-box}) = \prod_{i=1}^6 f_{B_i}$$

$$\log \left(\frac{P(S|\text{promoter})}{P(S|\text{non-promoter})} \right) = \log \left(\frac{\prod_{i=1}^6 f_{B_i,i}}{\prod_{i=1}^6 f_{B_i}} \right) = \sum_{i=1}^6 \log \left(\frac{f_{B_i,i}}{f_{B_i}} \right)$$

$$s_{b,i} = \log \left(\frac{f_{b,i}}{f_b} \right) \quad \text{Scores}$$

Construction of PWM

- Consider the PWM for the -10 region (TATAAT) of E. Coli.
- Suppose N sequences were aligned by their promoter sequences
- For example, T occurs in column 1 of alignment with frequency 0.79

Position	A	C	G	T
1	0.02	0.09	0.10	0.79
2	0.94	0.02	0.01	0.03
3..6

Construction of PWM

Suppose that the background frequency of each base = 0.25

Now convert frequency table to log-odds score

For example

if frequency = 0.79 then odds score is $0.79/0.25=3.16$

This means that if a sequence is being examined and a T is present at position i , then the odds of the sequence representing a promoter (a win) to the sequence not representing a promoter (a lose) is 3.16:1.00

Next convert frequencies to log-odds scores by taking the logarithm (base 2; units of bits), which gives the log-likelihood

Construction of PWM

TATAAT

Position	A	C	G	T
1	-3.80	-1.49	-1.34	1.67
2	1.92	-3.81	-4.81	-3.22
3	-0.06	-0.81	-0.66	0.81
4	1.24	-1.00	-0.72	-0.89
5	1.02	-0.35	-1.00	-0.56
6	-4.81	-3.22	-4.81	1.95

Application of PWM

Consider query sequence: CATCGTATAATGTGT

log odds score

$$= -1.49 - 4.81 + 0.81 + 1.24 - 0.56 - 4.81$$

$$= -9.62 \text{ bits.}$$

This gives an odds of $2^{-9.62} = 1/786$

C					
A					
T		A	C	G	T
C	1	-3.80	-1.49	-1.34	1.67
G	2	1.92	-3.81	-4.81	-3.22
T	3	-0.06	-0.81	-0.66	0.81
A	4	1.24	-1.00	-0.72	-0.89
T	5	1.02	-0.35	-1.00	-0.56
A	6	-4.81	-3.22	-4.81	1.95
A					
T					
G					
T					

Application of PWM

Consider query sequence: CATCGTTATAATGTGT

log odds score
-9.30 bits
gives an odds of 1/630

A					
T					
C		A	C	G	T
G	1	-3.80	-1.49	-1.34	1.67
T	2	1.92	-3.81	-4.81	-3.22
A	3	-0.06	-0.81	-0.66	0.81
T	4	1.24	-1.00	-0.72	-0.89
A	5	1.02	-0.35	-1.00	-0.56
A	6	-4.81	-3.22	-4.81	1.95
T					
G					
T					
G					

Example PWM

Consider query sequence: CATCGTATAATGTGT

log odds score

+8.61,

gives an odds of 391/1

clearly indicates a TATAAT region

T					
C					
G		A	C	G	T
T	1	-3.80	-1.49	-1.34	1.67
A	2	1.92	-3.81	-4.81	-3.22
T	3	-0.06	-0.81	-0.66	0.81
A	4	1.24	-1.00	-0.72	-0.89
A	5	1.02	-0.35	-1.00	-0.56
T	6	-4.81	-3.22	-4.81	1.95
G					
T					
G					
T					