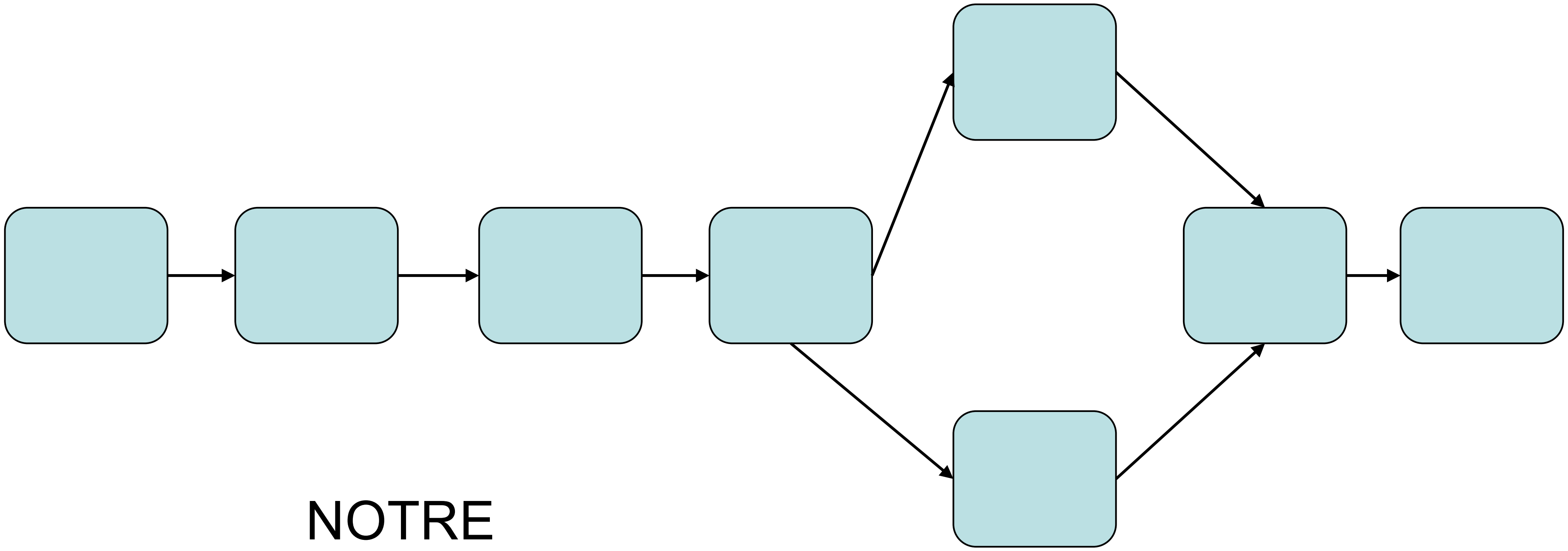
Lecture 10

Ab initio gene finding

Topology

- Some characteristics: number of nodes, alphabet, subset of edges
- We want to exploit domain knowledge
  - Limits number of states / edges
  - Still expressive enough to model relationships
- Length distributions (Durbin 3.4)

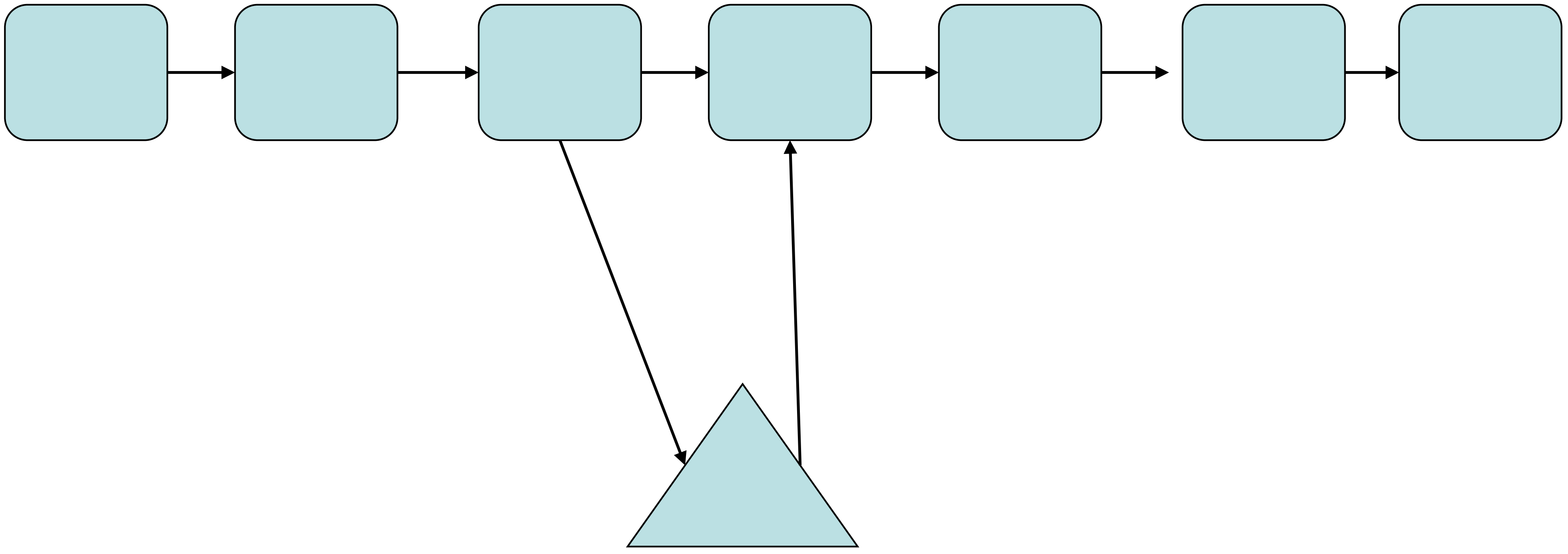
Example

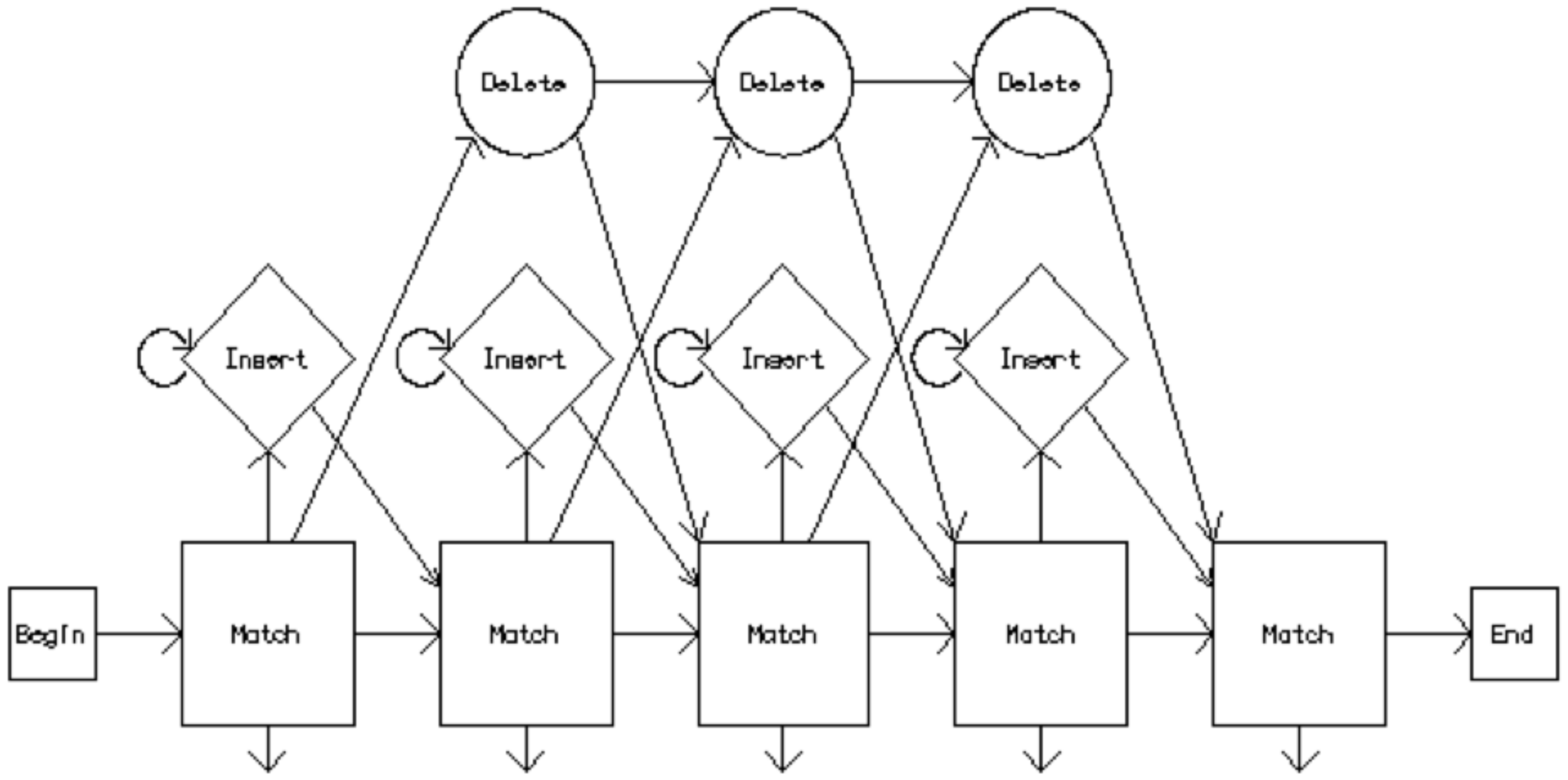


NOTRE
NOTRE
NOTMEI
NOTME

EM4(R) = 1 eM5(M) = 5

Insertions





http://genome.jouy.inra.fr/doc/genome/suite-logicielle/gcg-11.1/html/figure/hmmessay_2.gif

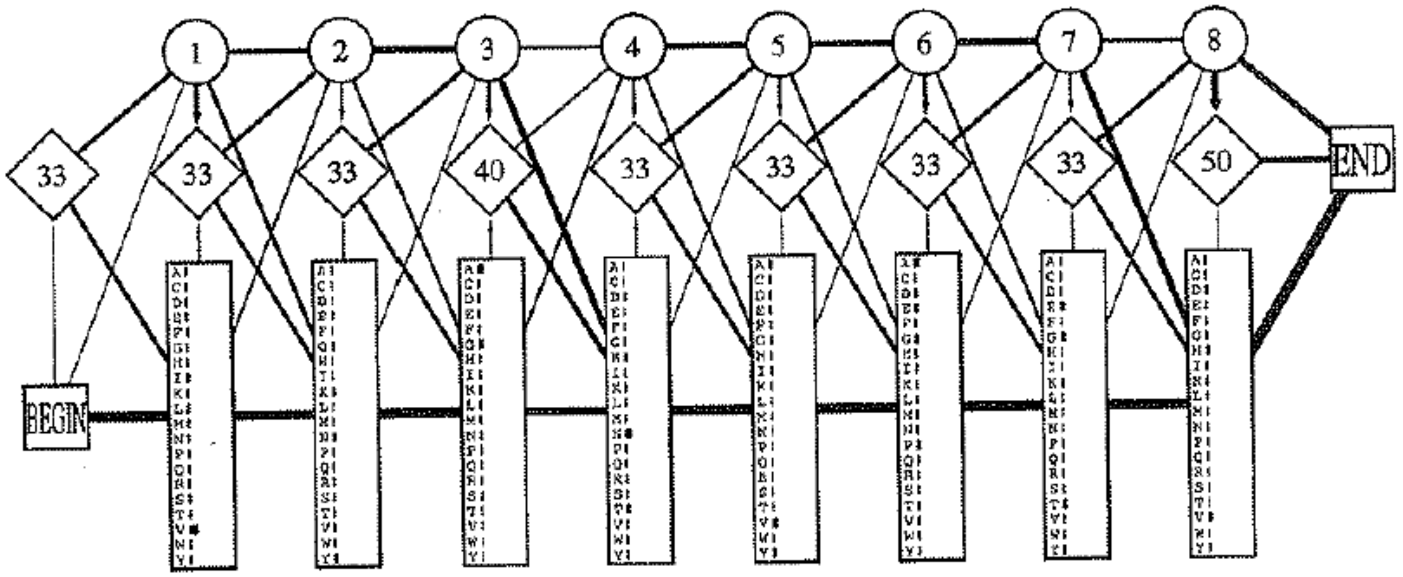


Figure 5.4 *A hidden Markov model derived from the small alignment shown in Figure 5.3 using Laplace's rule. Emission probabilities are shown as bars opposite the different amino acids for each match state, and transition probabilities are indicated by the thickness of the lines. The* I $\rightarrow$ I *transition probabilities times 100 are shown in the insert states. (Figure generated automatically using the* SAM *package.)*

Uses of probabilistic sequence models/HMMs

- Segmentation
- Multiple alignment using profile HMMs
- Prediction of sequence function (gene family models)
- ** Gene finding **

Review

- **Gene**
  - A sequence of nucleotides that are translated into proteins

- ***Gene prediction***
  - Given the model of a gene above, determine the beginning and end positions of all genes in a genome.

Central dogma of molecular biology

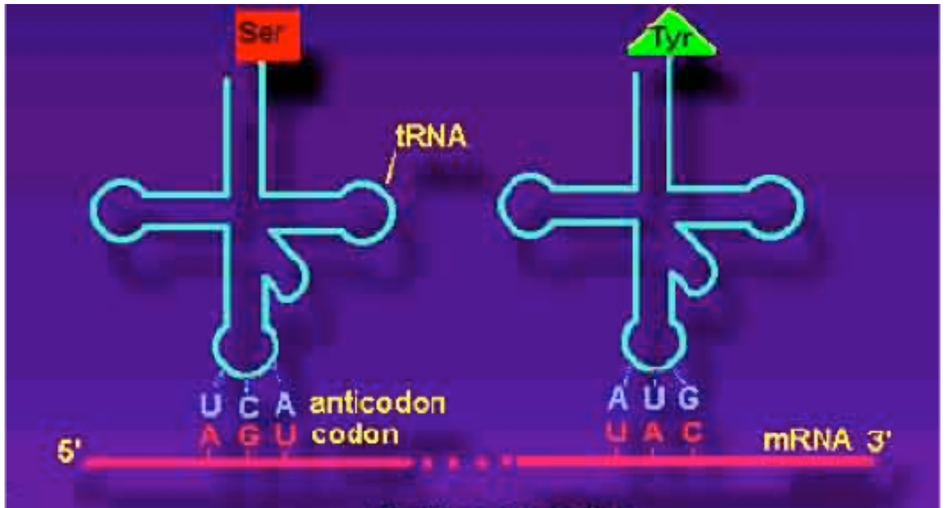
- Information is stored in DNA
- Genome is processed into messenger RNA molecules (transcription)
- RNA molecules are processed to form proteins (translation)

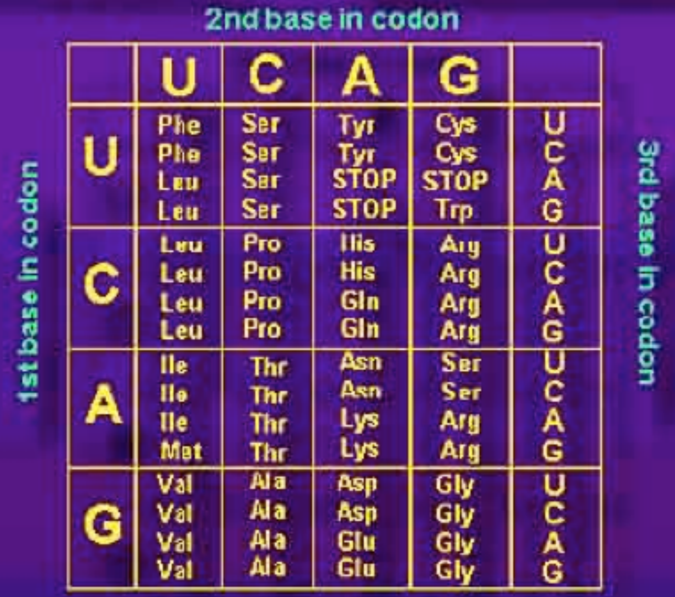
General facts about the genetic code

- There is inherent redundancy; 64 possibilities and 20 amino acids
- Most of the "flexibility" is in the third position
  - "wobble" position
- Deletions that are not multiples of 3 change the resulting amino acid

$$\begin{tikzcd} \boxed{\text{DNA}} \arrow[d, "\text{transcription}"] \\ \boxed{\text{mRNA}} \arrow[d, "\text{translation}"] \\ \boxed{\text{Protein}} \end{tikzcd}$$

$$\begin{tikzcd} \text{CCTGAGCCAACTATTGATGAA} \arrow[d] \\ \text{CCUGAGCCAACUAUUGAUGAA} \arrow[d] \\ \text{PEPTIDE} \end{tikzcd}$$





2nd base in codon

1st base in codon	U	C	A	G	3rd base in codon
U	Phe	Ser	Tyr	Cys	U
	Phe	Ser	Tyr	Cys	C
	Leu	Ser	STOP	STOP	A
	Leu	Ser	STOP	Trp	G
C	Leu	Pro	His	Arg	U
	Leu	Pro	His	Arg	C
	Leu	Pro	Gln	Arg	A
	Leu	Pro	Gln	Arg	G
A	Ile	Thr	Asn	Ser	U
	Ile	Thr	Asn	Ser	C
	Ile	Thr	Lys	Arg	A
	Met	Thr	Lys	Arg	G
G	Val	Ala	Asp	Gly	U
	Val	Ala	Asp	Gly	C
	Val	Ala	Glu	Gly	A
	Val	Ala	Glu	Gly	G

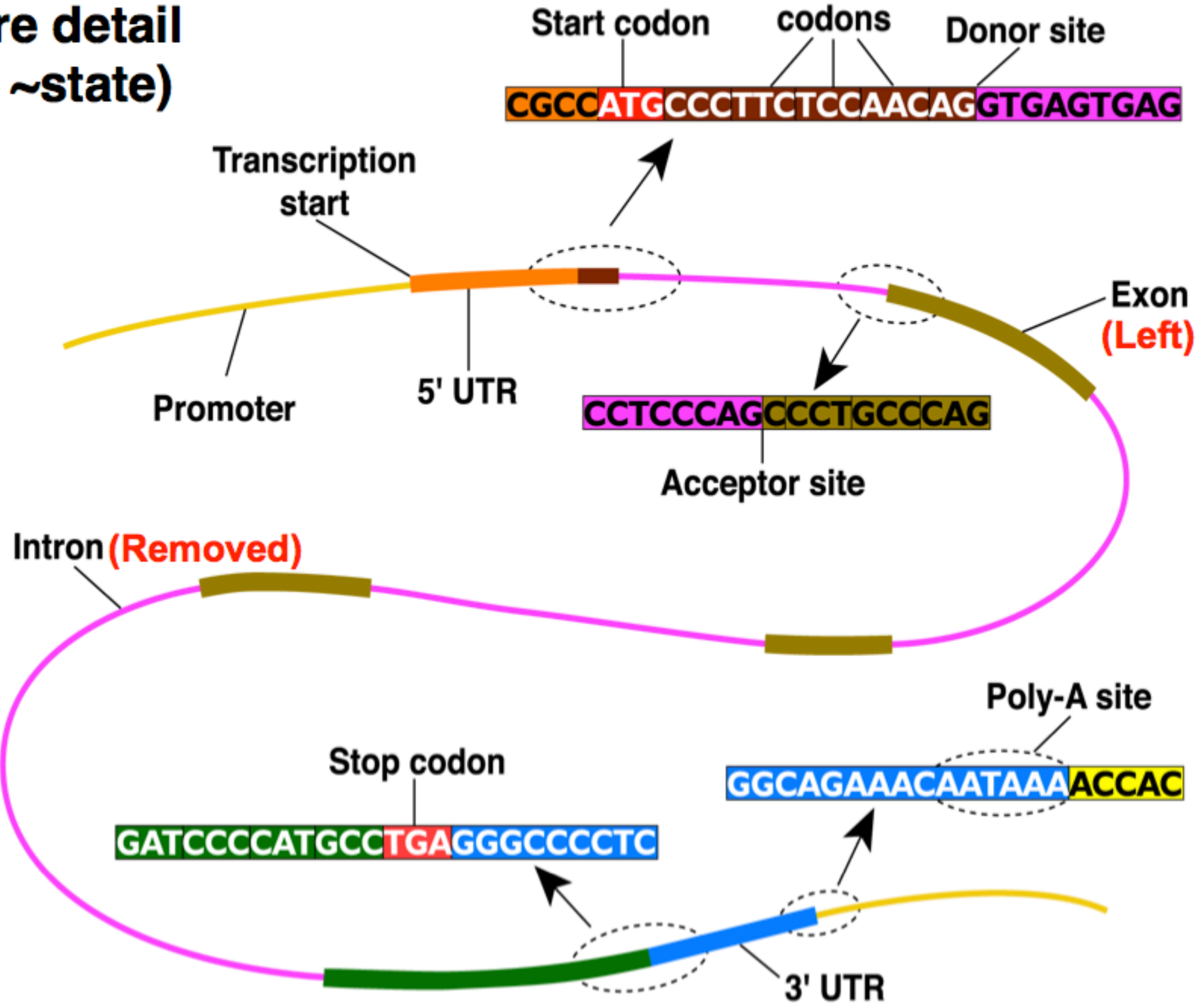
Source and interesting link:

http://www.bioscience.org/atlases/genecode/genecode.htm

Facts about gene finding

- Prediction is usually easier in prokaryotes
- Small fraction of most genomes is genes
  - 3% of the human genome
- Many false signals
- Long introns and small exons
  - Difficult for mathematical models

In more detail (color ~state)



Ab initio?

- Gene prediction is usually broken into three types:
  - *Ab initio* : making predictions based on some statistical model (GENSCAN, FGENE)
  - Knowledge-based: making predictions based on known genes (tBLASTx)
  - Comparative: making predictions based on a related genome (TwinScan)

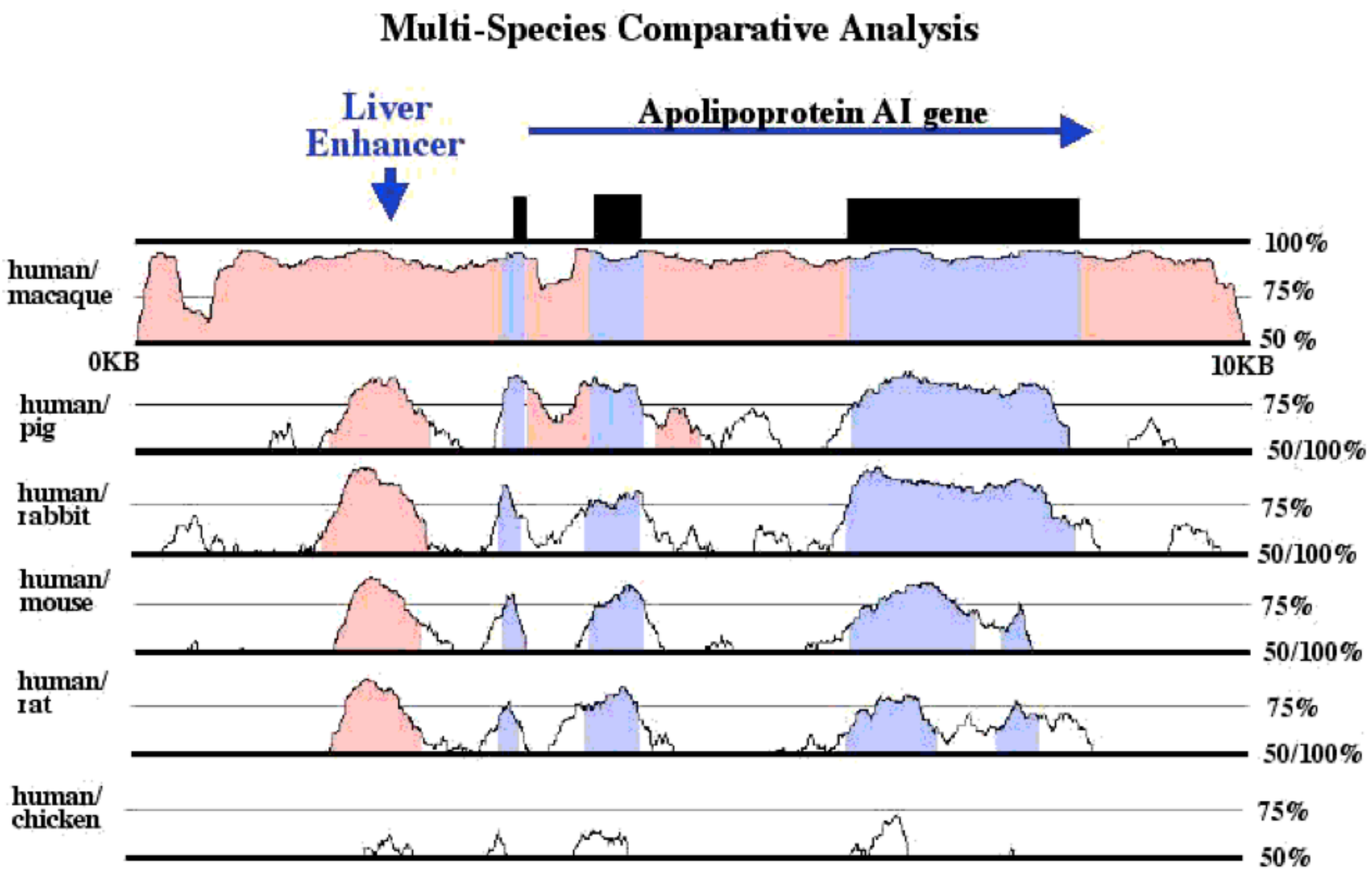
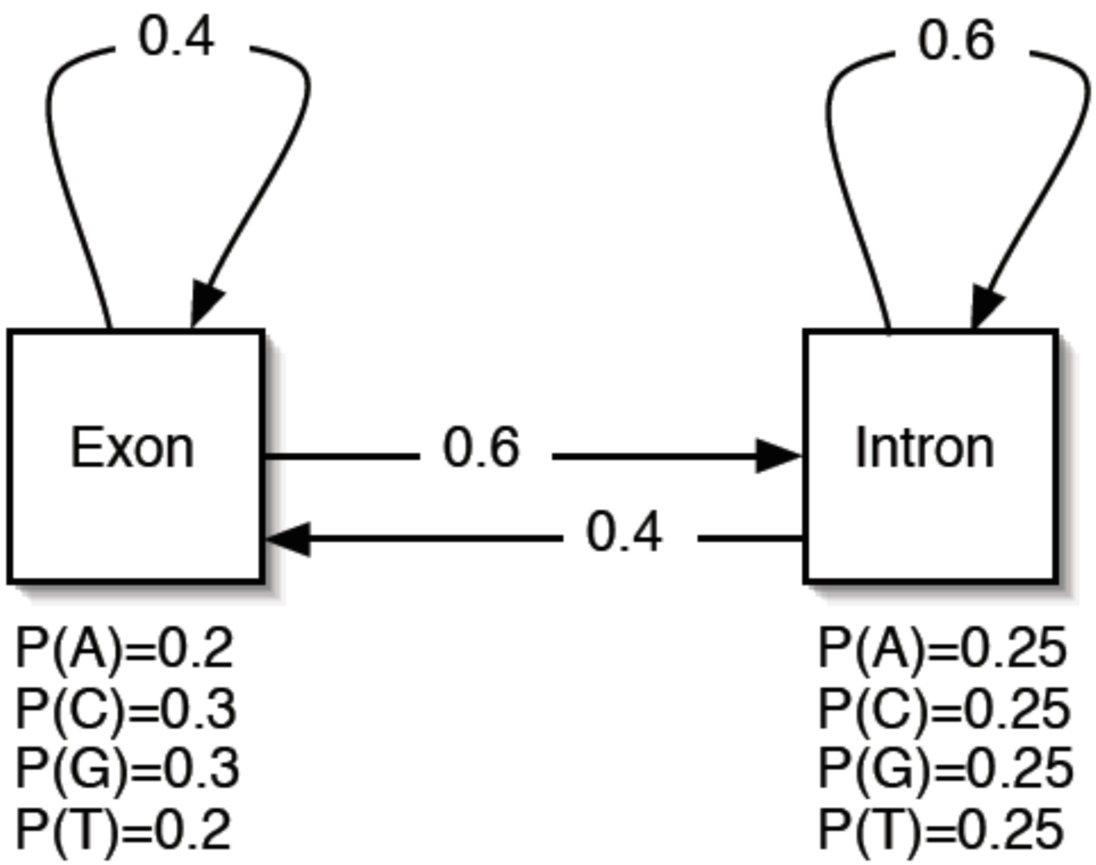
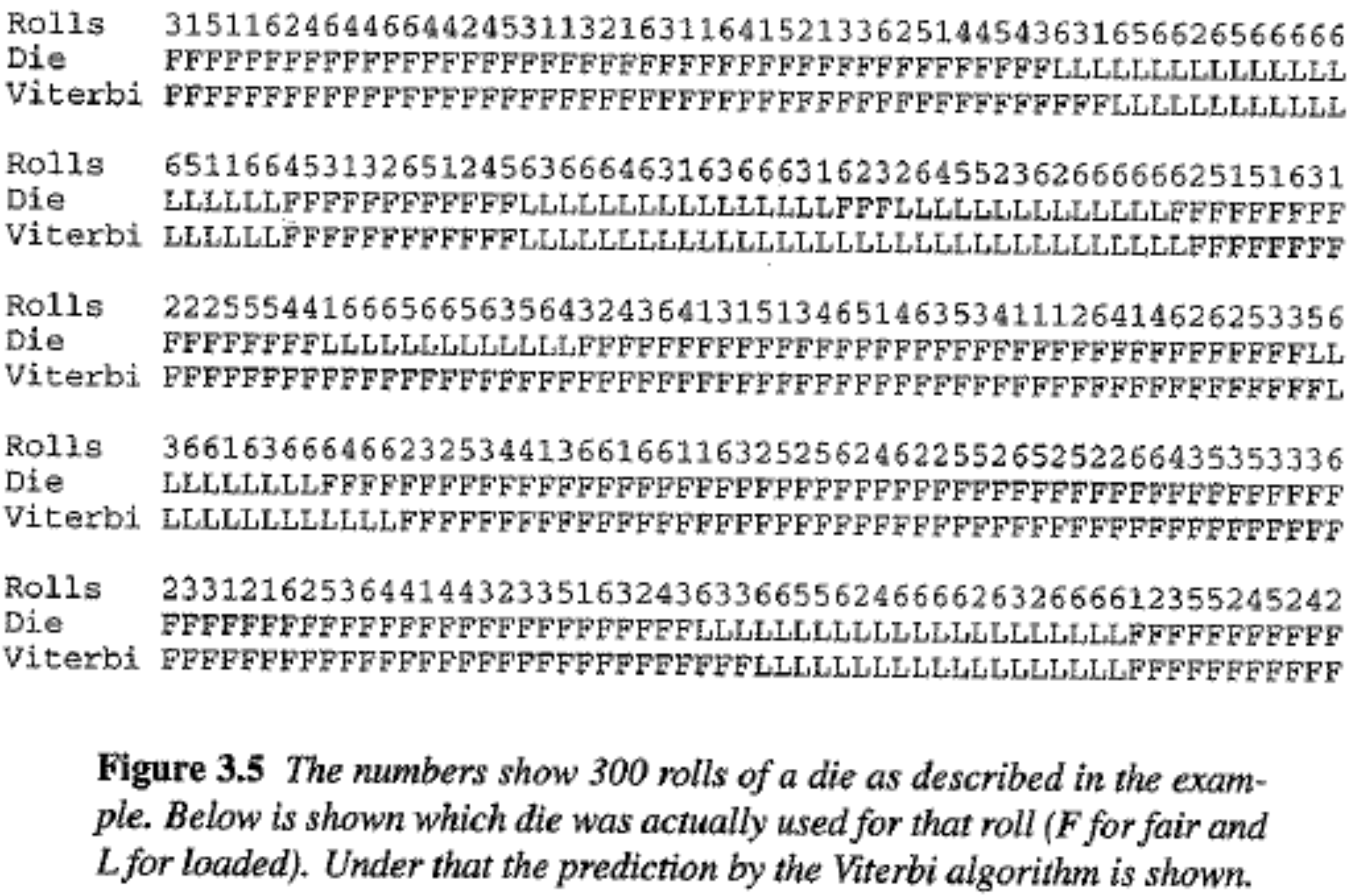


Figure 4: Multi-Species Comparative Analysis. (This picture is from Sanja Rogic's lecture slides, "Computational Gene Finding")


0.4
0.6
Exon
0.6
0.4
Intron
P(A)=0.2
P(C)=0.3
P(G)=0.3
P(T)=0.2
P(A)=0.25
P(C)=0.25
P(G)=0.25
P(T)=0.25

Observed sequence, hidden path and Viterbi path



```
Rolls   315116246446644245311321631164152133625144543631656626566666
Die     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLL
Viterbi FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLL

Rolls   651166453132651245636664631636663162326455236266666625151631
Die     LLLLLLFFFFFFFFFFFFLLLLLLLLLLLLLLLLFFFLLLLLLLLLLLLLLLFFFFFFFFF
Viterbi LLLLLLFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLFFFFFFFF

Rolls   222555441666566563564324364131513465146353411126414626253356
Die     FFFFFFFFLLLLLLLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLL
Viterbi FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFL

Rolls   366163666466232534413661661163252562462255265252266435353336
Die     LLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
Viterbi LLLLLLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

Rolls   233121625364414432335163243633665562466662632666612355245242
Die     FFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLLLLFFFFFFFFFFF
Viterbi FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLFFFFFFFFFFF
```

Figure 3.5 *The numbers show 300 rolls of a die as described in the example. Below is shown which die was actually used for that roll (F for fair and L for loaded). Under that the prediction by the Viterbi algorithm is shown.*

From Durbin

The occasionally dishonest casino: Posterior decoding

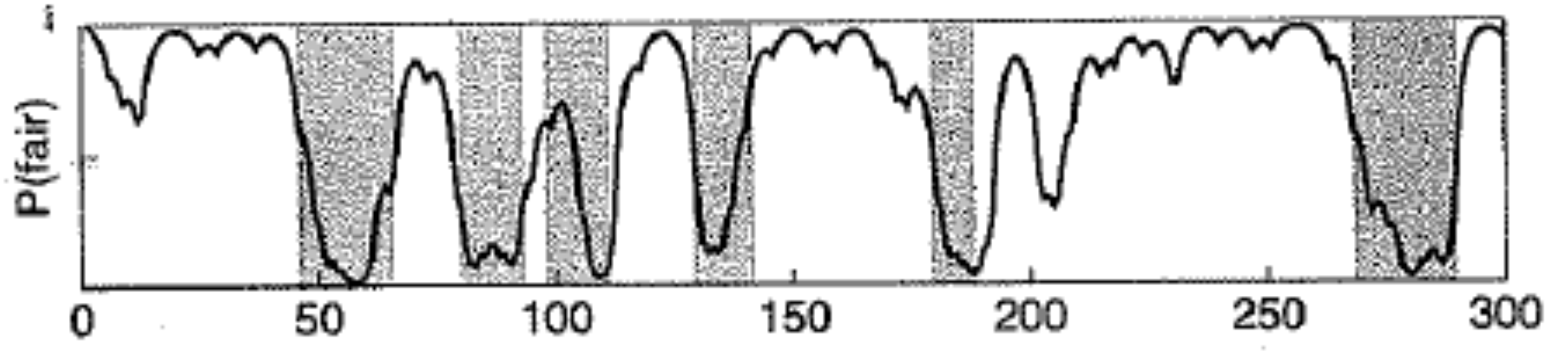


Figure 3.6 *The posterior probability of being in the state corresponding to the fair die in the casino example. The x axis shows the number of the roll. The shaded areas show when the roll was generated by the loaded die.*

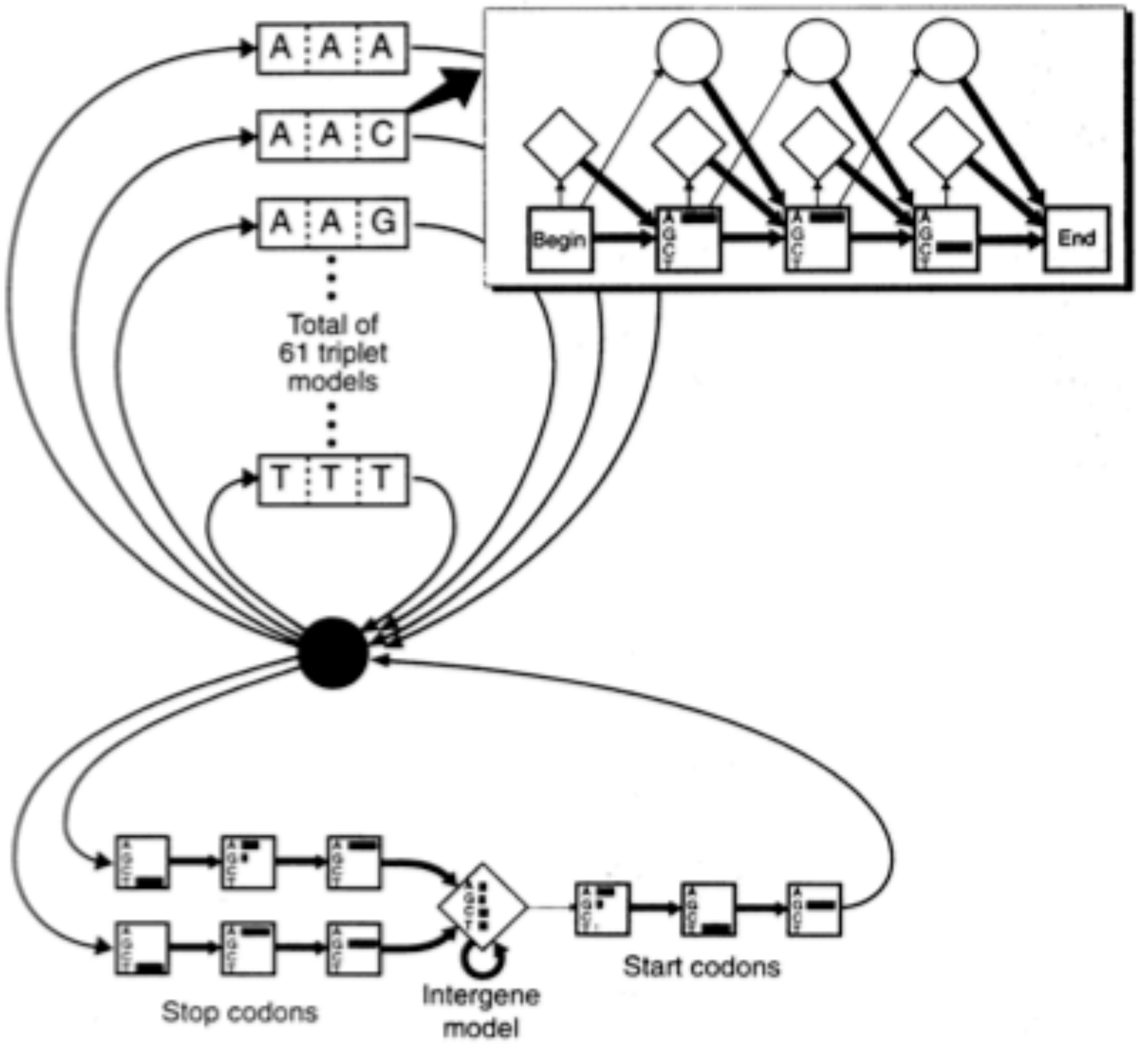
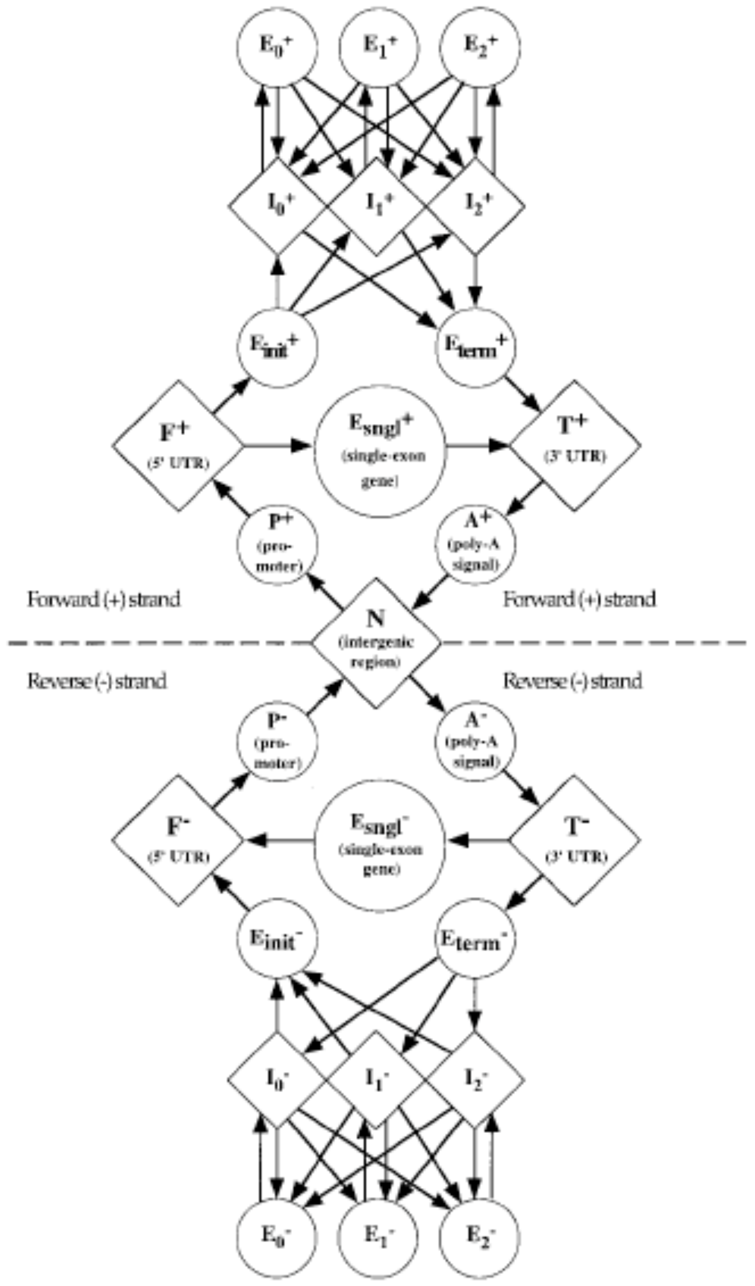
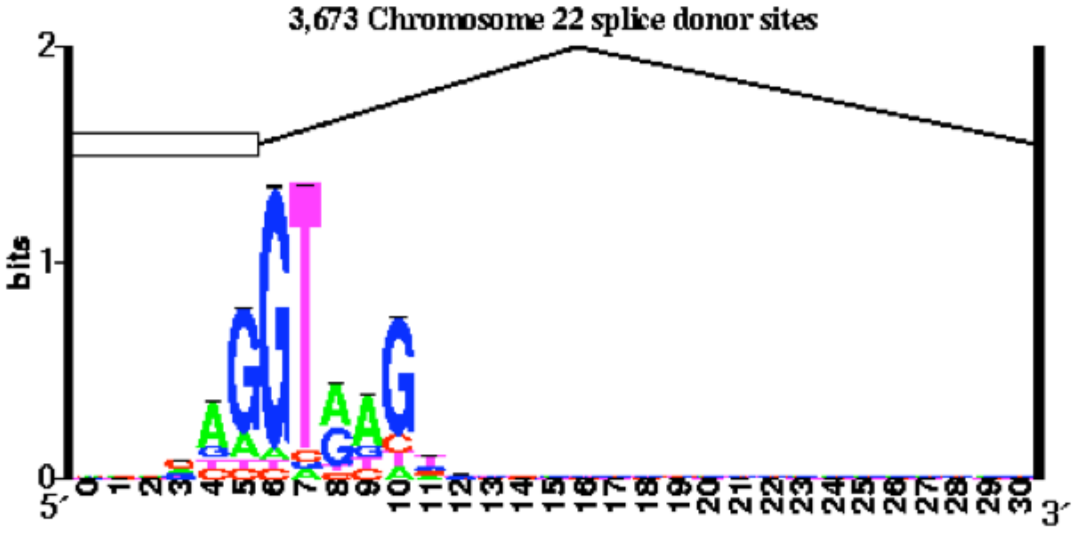


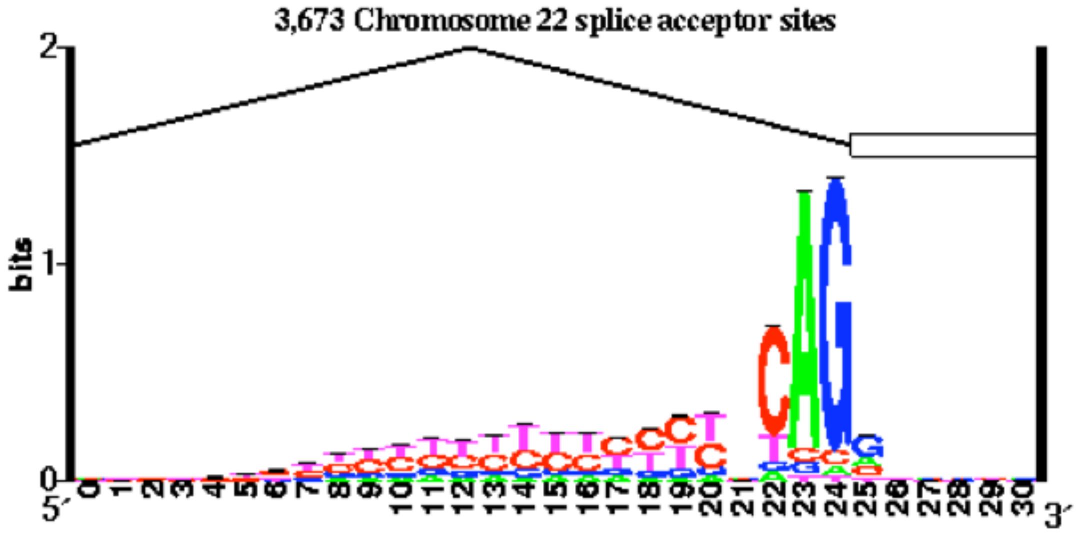
Figure 1. HMM architecture for a parser for *E.coli* DNA with a simple intergenic model. The *central state* (shaded circle), generates no nucleotides and is used to connect all the models. The 61 triplet or codon models above the central state all have identical structures, shown in detail for the codon AAC. Squares represent main states; diamonds denote a state where a nucleotide can be inserted between consecutive codon nucleotides whereas circles generate no nucleotide and can be used to delete one of the three nucleotides. The thickness of the arrows indicate the fraction of sequences making the given transistion. The insert state in the middle of the intergenic model (diamond) produces random sequences from a base distribution estimated from the actual distribution of bases in the intergenic regions of the training set. The four bases have almost the same frequency.

Krogh, Mian and Haussler (1994)



Burge and Karlin (1997)





http://www.sanger.ac.uk/HGP/Chr22/cwa_archive/splice_site_analysis.shtml

Open reading frames

- Generally defined as regions in genes between a start (ATG) and stop (eg. TGA) codon.

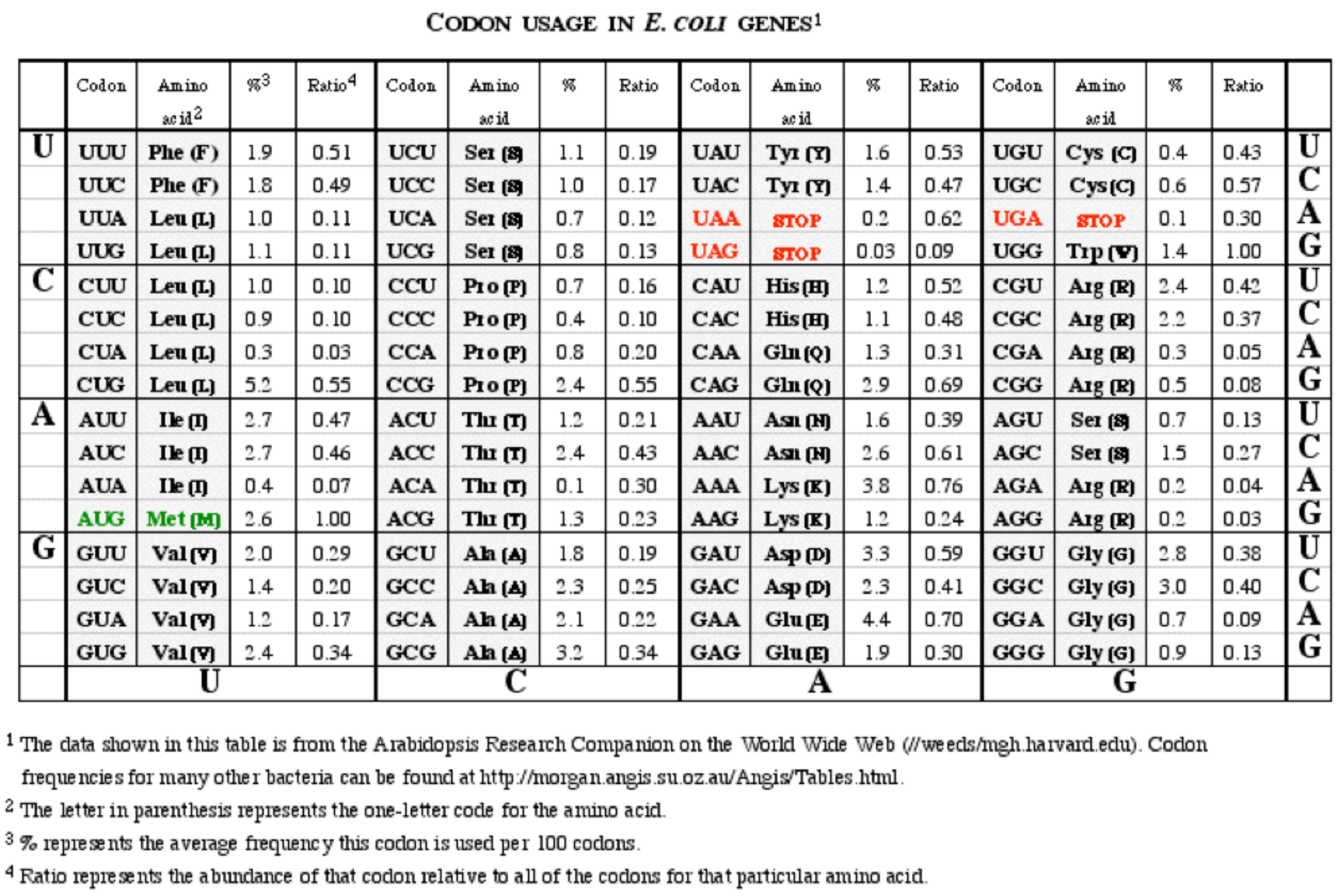
- Size is a multiple of 3

- Six possibilties given any DNA sequence
  - 0 offset, + strand; 1 offset, + strand, 2 offset, + strand
  - 0 offset, - strand; 1 offset, - strand, 2 offset, - strand

Examples from the text

- *M. genitalium* (example 2.6)
  - 402 bp is a significant value
- *H. influenzae* (2.7)
  - 573 bp is significant
  - Larger genome than *M. genitalium*, so this makes sense.

Not all codons are equal



CODON USAGE IN *E. COLI* GENES[1]

	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	A
	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	A
	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	A
	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	A
	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 (//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.

REFERENCE: Modified from Maloy, S., V. Stewart, and R. Taylor. 1996. Genetic analysis of pathogenic bacteria. Cold Spring Harbor Laboratory Press, NY.

Simple example

- Lets only consider two states: coding and non-coding (+ or -)
- Further, we will consider only four output states
  - ACGT
- At most we will have 8 states:
  - +A, +C, +G, +T, -A, -C, -G, -T

Long ORFs

- At random, we’d expect a stop codon every 64 nucleotides.
- Many bacteria genes are much longer than this.
- These can be used to train a statistical model.

Significance

- An important question often is “Are these interesting?”
- Many bioinformatics tools compute a *p*-value, or the probability of observing something in a random collection.
  - Example: e-value in BLAST

Randomization test

- There are two ways to determine significant ORF lengths:

- Permutation
  - Shuffle the original sequence in a simple or more complicated way

- Bootstrapping
  - Generate random sequences with the same statistical properties
  - Uses Markov chain models as in Hw #1

But wait!

- Finding ORFs alone isn’t sufficient. *E. coli* has up to 6500 ORFs but only 1100 “real” genes

- Complicated by the fact ORFs can overlap on different strands and be correct (figure in class)

Homework to the rescue (or at least part of the way there)

- We know codons are triplets.
- We also know from last class codons are not equal.
- A simple gene finder is no different from your Markov assignment except now we'll need to compute two probabilities

Simple gene finding

- Score every ORF using all seven models
- Normalize the scores such that they represent the probability of coding
- Choose the highest

Durbin: First order Markov (p74)

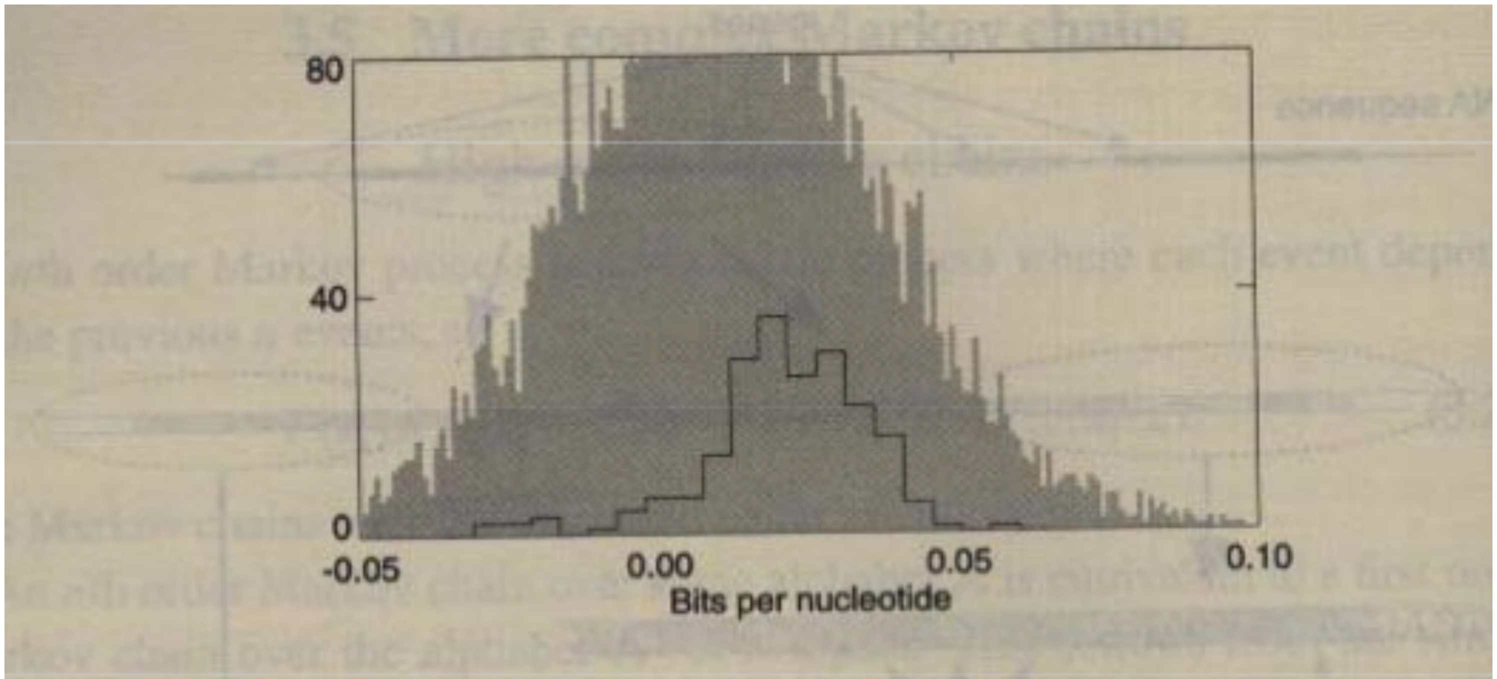


Figure 3.11 *Histograms of the log-odds per nucleotide for all NORFs (grey) and genes (black line) according to a first order Markov chain. Because of the large number of NORFs, the histogram bin size is five times smaller for the NORFs.*

Take home

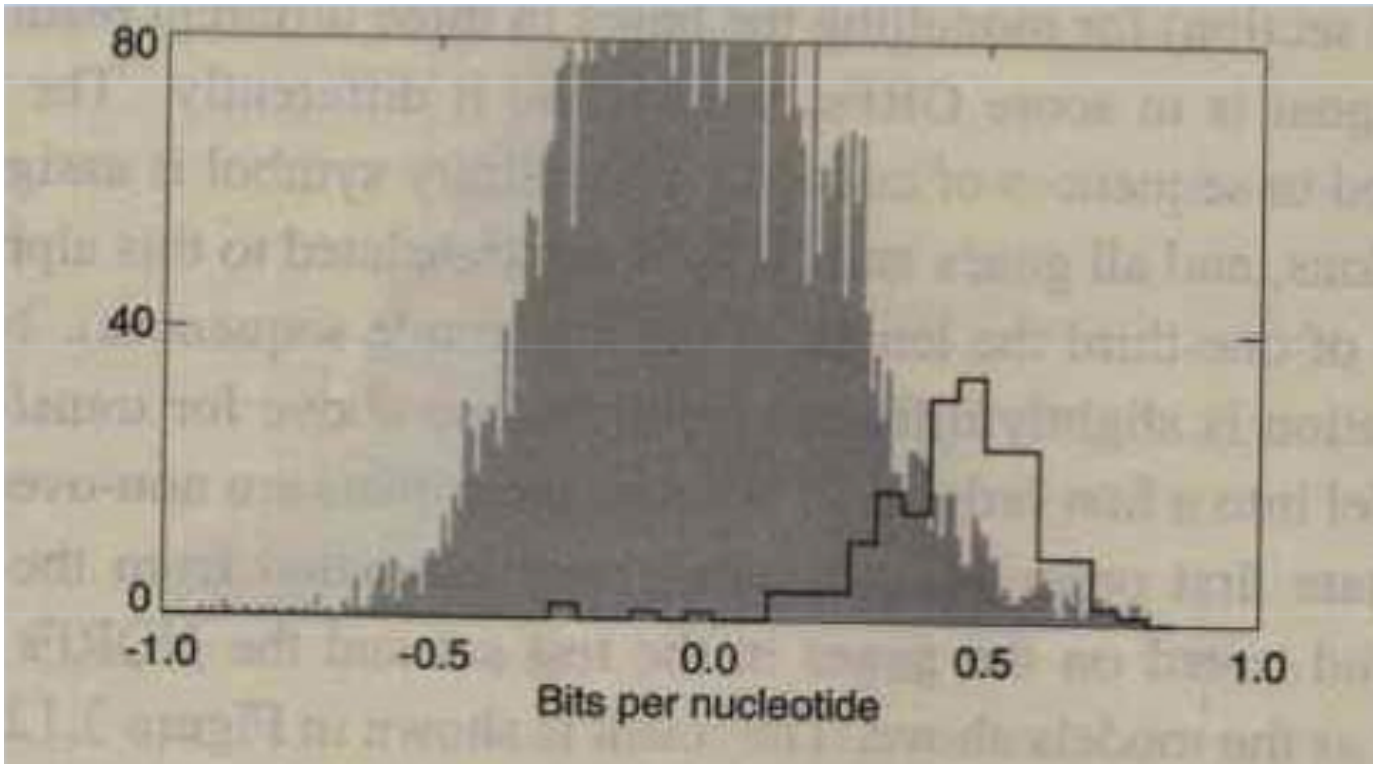
- Interestingly, there is some signal in the previous figure:
  - Average log-odds for genes: 0.018
  - Average log-ods for non-genes: 0.009

- The variance is huge, though, so it is hard to tell which is which in general.

Selecting the proper Markov order

- Higher order models remember more "history," which helps predictions
- Examples (from Colin Dewey):
  - "… you __"
  - "… can you __"
  - "… say can you__"
  - "… oh say can you __"

Durbin p. 76



One fix is to use a third order model