

Genomic sequence analysis:
gene prediction

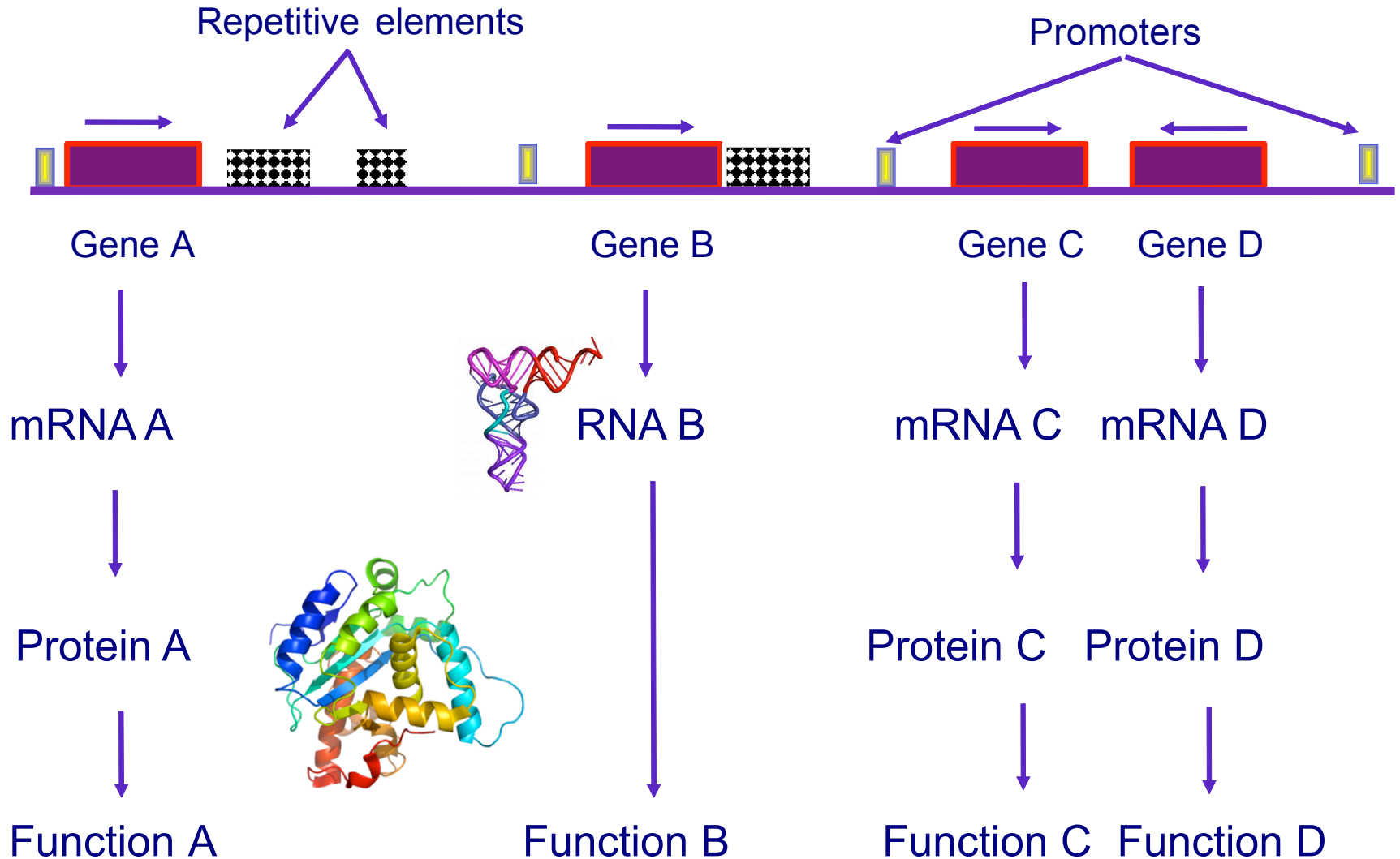
We want to know how this...

TGCATCGATCGTAGCTAGCTAGCGCATGCTAGCTAGCTAGCTAGCTACGATGCATCG
TGCATCGATCGATGCATGCTAGCTAGCTAGCTAGCATGCTAGCTAGCTAGCTATTGG
CGCTAGCTAGCATGCATGCATGCATCGATGCATCGATTATAAGCGCGATGACGTCAG
CGCGCGCATTATGCCGCGGCATGCTGCGCACACACAGTACTATAGCATTAGTAAAAA
GGCCGCGTATATTTTACACGATAGTGCGGGCGGGCGCGTAGCTAGTGCTAGCTAGTC
TCCGGTTACACAGGTAGCTAGCTAGCTGCTAGCTAGCTGCTGCATGCATGCATTAGT
AGCTAGTGTAGCTAGCTAGCATGCTGCTAGCATGCAGCATGCATCGGGCGCGATGCT
GCTAGCGCTGCTAGCTAGCTAGCTAGCTAGGCGCTAATTATTTATTTTGGGGGGTTA
AAAAAAAAAATTTTCGCTGCTTATACCCCCCCCCACATGATGATCGTTAGTAGCTACT
AGCTCTCATCGCGCGGGGGGATGCTTAGCGTGGTGTGTGTGTGTGGTGTGTGTGGTC
CTATAATTAGTGCATCGGCGCATCGATGGCTAGTCGATCGATCGATTTTATATATCT
AAAGACCCCATCTCTCTCTTTTCCCTTCTCTCGCTAGCGGGCGGTACGATTTACC
GGCCGCGTATATTTTACACGATAGTGCGGGCGGGCGCGTAGCTAGTGCTAGCTAGTC
AGCTCTCATCGCGCGGGGGGATGCTTAGCGTGGTGTGTGTGTGTGGTGTGTGTGGTC
TGCATCGATCGATGCATGCTAGCTAGCTAGCTAGCATGCTAGCTAGCTAGCTATTGG
CTATAATTAGTGCATCGGCGCATCGATGGCTAGTCGATCGATCGATTTTATATATCT
CGCTAGCTAGCATGCATGCATGCATCGATGCATCGATTATAAGCGCGATGACGTCAG
TCCGGTTACACAGGTAGCTAGCTAGCTGCTAGCTAGCTGCTGCATGCATGCATTAGT

Becomes this



What are we looking for?



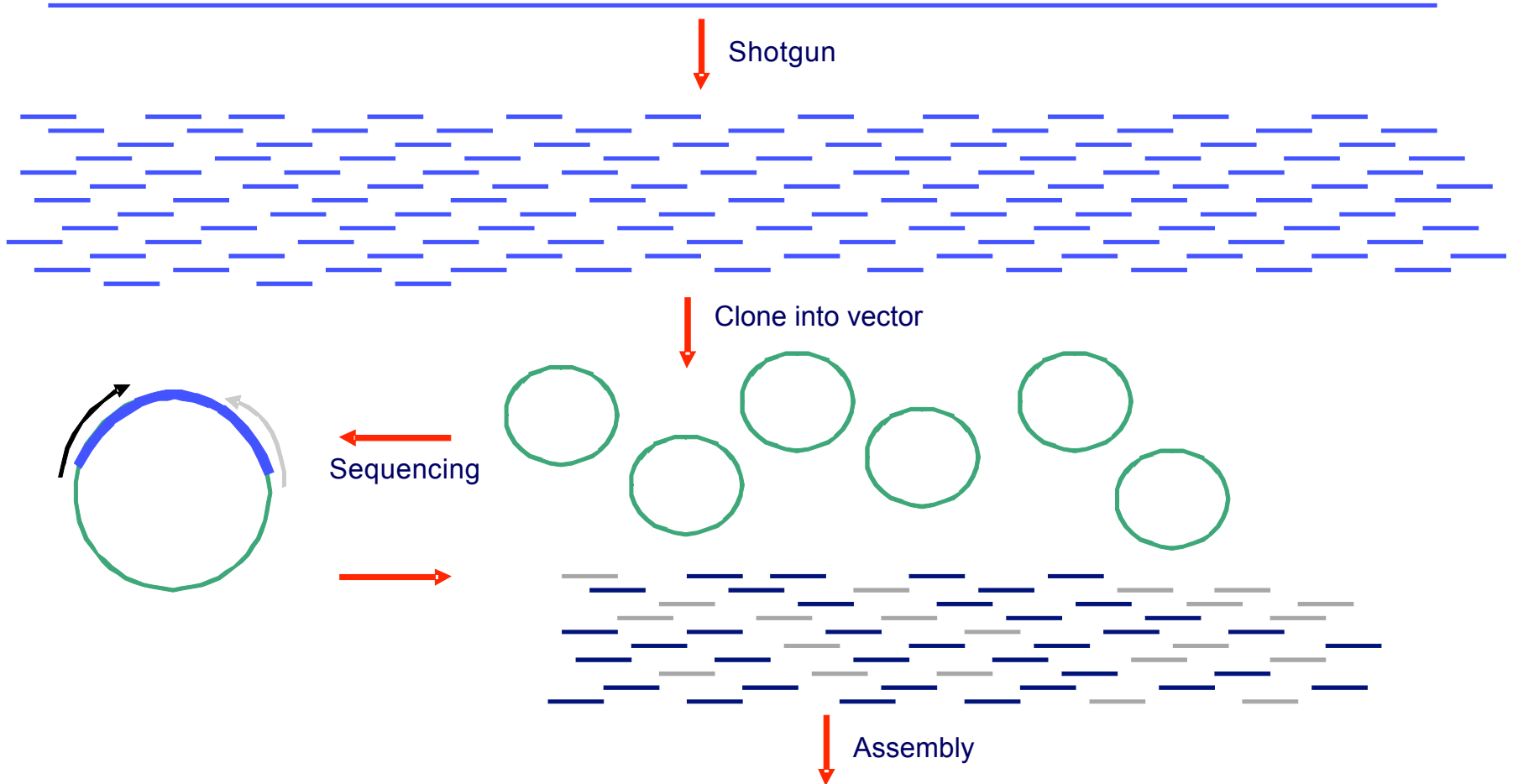
Getting all genes

- Genome sequencing
 - Access to entire genome, allows to learn more about genome organization
 - Regulatory elements
 - Only small percentage of the genome codes for genes
 - Hard to identify less typical genes
 - High rate of false positives
- EST sequencing
 - Requires less sequencing since it is focused on coding sequence only
 - Small rate of false positives, although even 10% of EST sequences could be artifacts
 - Genes with very restricted expression may never be discovered
 - In most cases gives only partial sequences

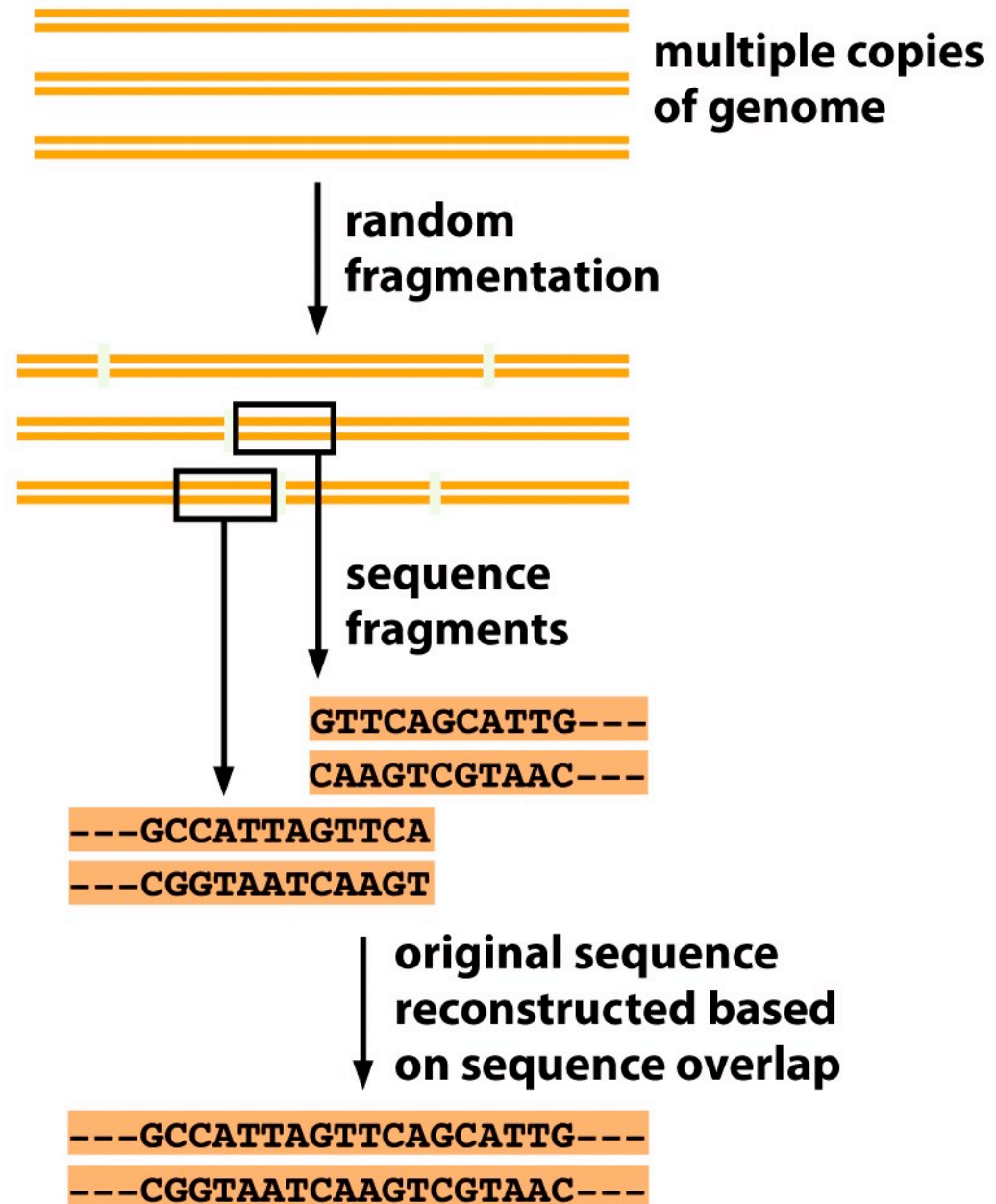
Gene identification methods

- Molecular techniques
 - Very laborious
 - Time consuming
 - Expensive
 - Low rate of false positives
- Computational methods
 - Fast
 - Relatively low cost
 - High rate of false positives
 - Poor performance on less typical genes

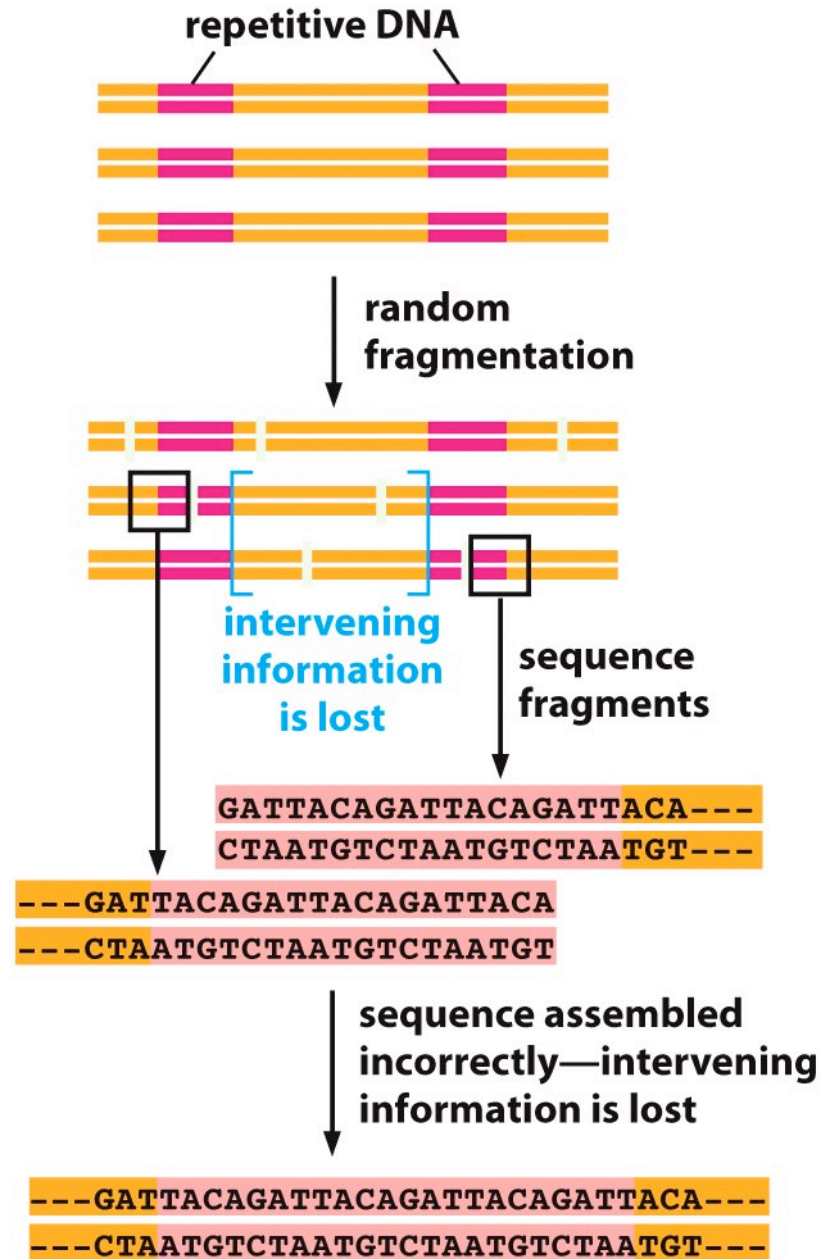
Genome sequencing



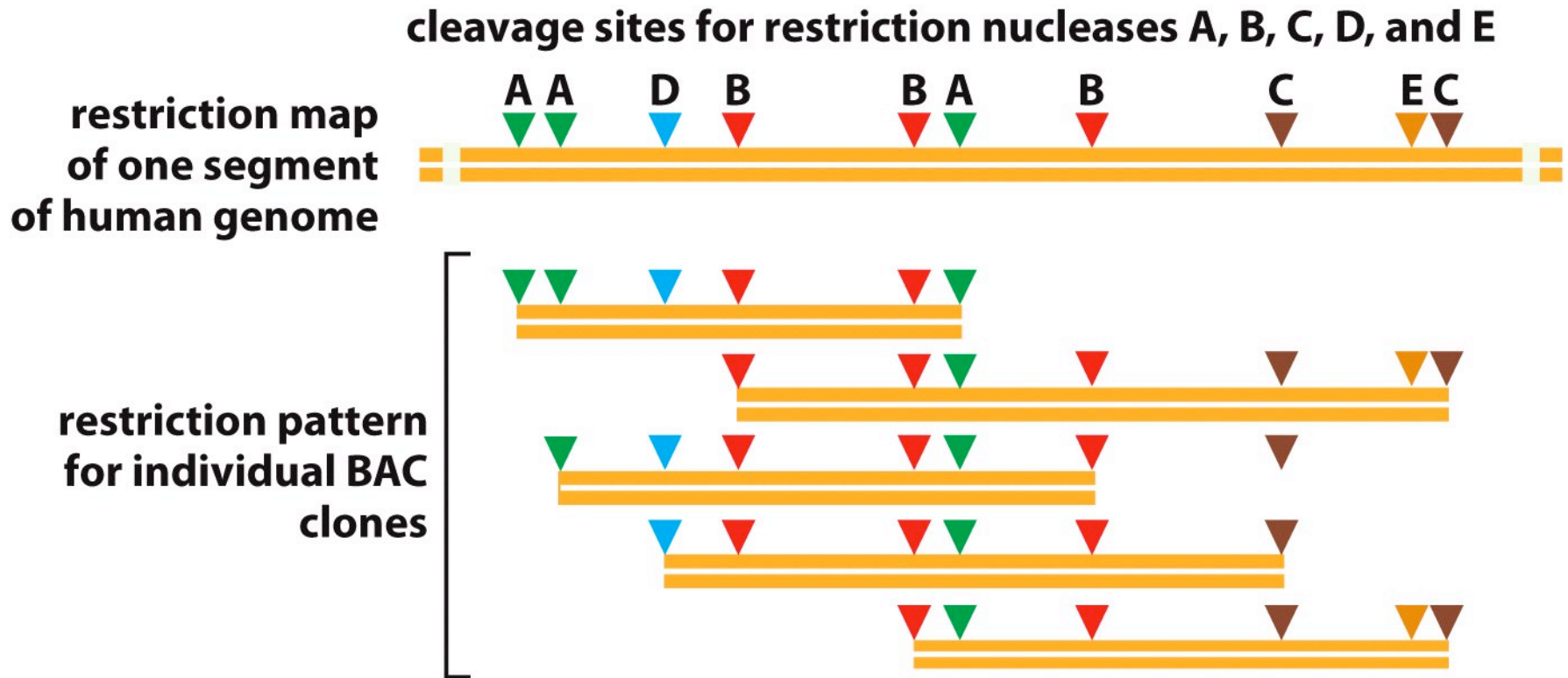
Shotgun sequencing is the method of choice for small genomes



Repetitive sequences make correct assembly difficult



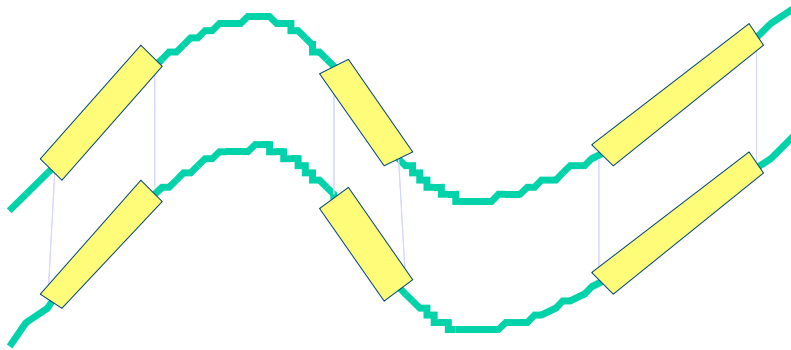
Clone-by-clone approach



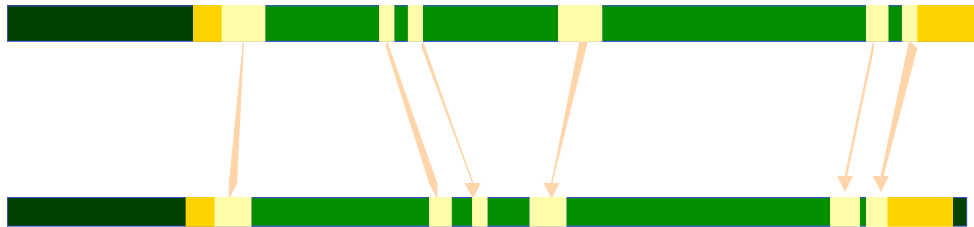
Before we start analysis...

- We have to:
 - Check sequences quality
 - Remove contamination
 - Assembly sequence reads into longer contigs
 - Close gaps (in perfect situation)

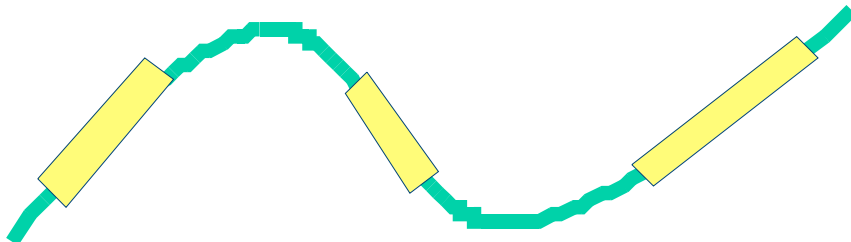
Gene finding methods classification



Similarity based predictors: make use of similarity to already known genes and proteins coded by these genes as well as expression data including sequences from cDNAs and data from hybridization experiments (tiling arrays for example)



Dual- and multi-genome predictors: rely on the fact that functional regions of a genome sequence are more conserved during evolution

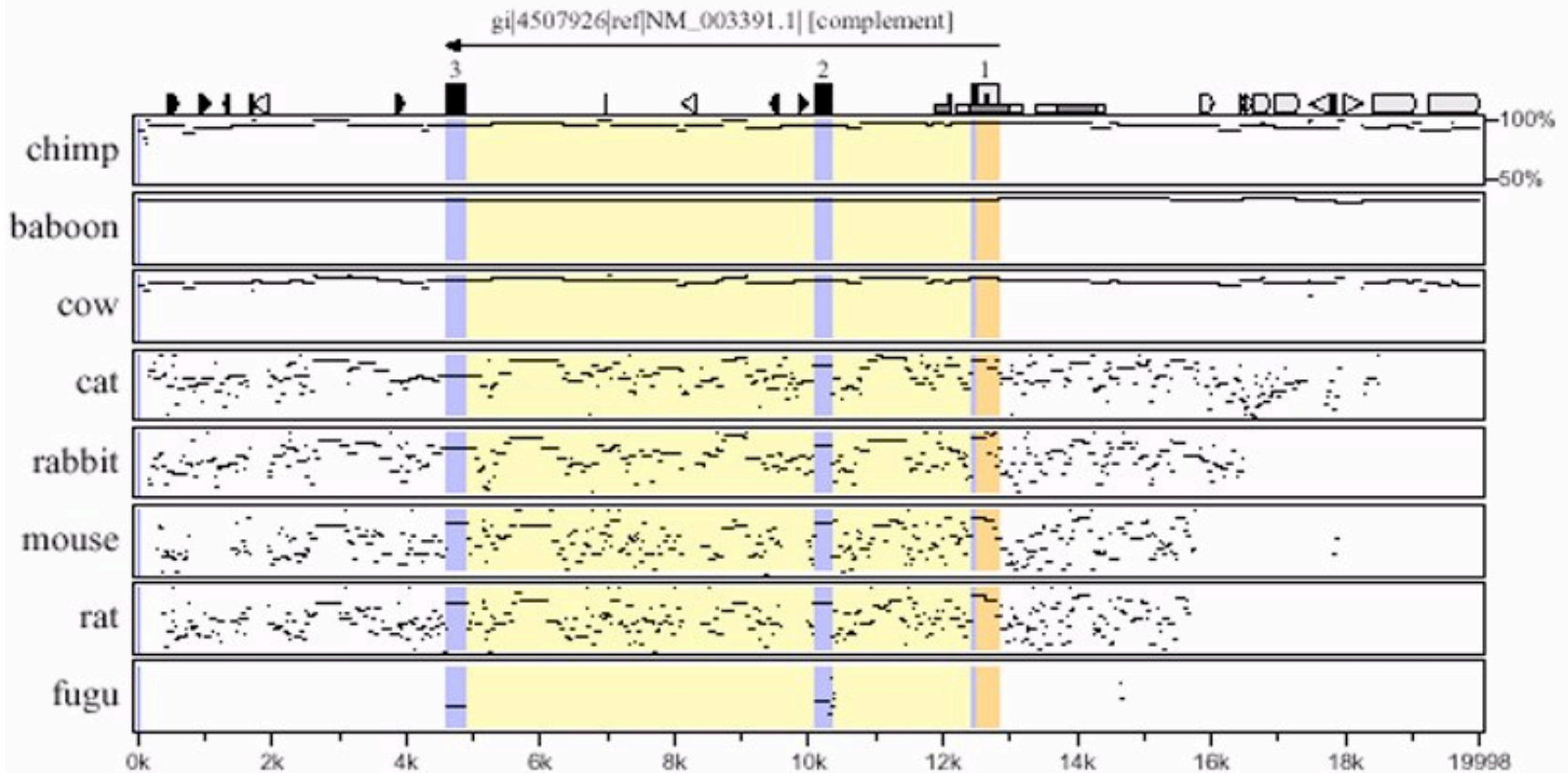


Model based predictors: use a single genome sequence and exon/intron structure is predicted based on absolute and bulk properties of the sequence

Similarity search

- We can check if any fragment of our sequence shows similarity to already known protein. We can also check if there are any mRNA sequences and ESTs which align well with the genomic sequence. Based on similarity we can deduct the gene structure and protein function

Comparative genomics - MultiPipMaker



<http://pipmaker.bx.psu.edu/pipmaker/>

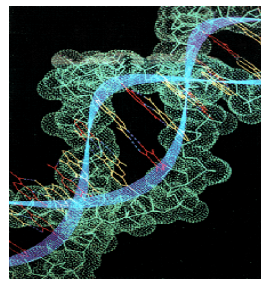
Model based methods

- We take advantage of what we already learned about gene structures and features of coding sequences. Based on this knowledge we can build theoretical model, develop an algorithm to search for important features, train it on known data and use to search for coding sequences in anonymous genomic fragments

All information is in the DNA. We just have to learn how to read the code, the program for life.

TGCATCGATCGTAGCTAGCTAGCGCATGCTAGCTAGCTAGCTAGCTACGATGCATCG
TGCATCGATCGATGCATGCTAGCTAGCTAGCTAGCATGCTAGCTAGCTAGCTATTGG
CGCTAGCTAGCATGCATGCATGCATCGATGCATCGATTATAAGCGCGATGACGTCAG
CGCGCGCATTATGCCGCGGCATGCTGCGCACACACAGTACTATAGCATTAGTAAAA
GGCCGCGTATATTTTACACGATAGTGCGGCGCGGCGCGTAGCTAGTGCTAGCTAGTC
TCCGGTTACACAGGTAGCTAGCTAGCTGCTAGCTAGCTGCTGCATGCATGCATTAGT
AGCTAGTGTAGCTAGCTAGCATGCTGCTAGCATGCAGCATGCATCGGGCGCGATGCT
GCTAGCGCTGCTAGCTAGCTAGCTAGCTAGGCGCTAATTATTTATTTTGGGGGGTTA
AAAAAAAAAATTTTCGCTGCTTATACCCCCCCCCACATGATGATCGTTAGTAGCTACT
AGCTCTCATCGCGCGGGGGGATGCTTAGCGTGGTGTGTGTGTGTGGTGTGTGTGGTC
CTATAATTAGTGCATCGGCGCATCGATGGCTAGTCGATCGATCGATTTTATATATCT
AAAGACCCCATCTCTCTCTCTTTTCCCTTCTCTCGCTAGCGGGCGGTACGATTTACC

Program for life

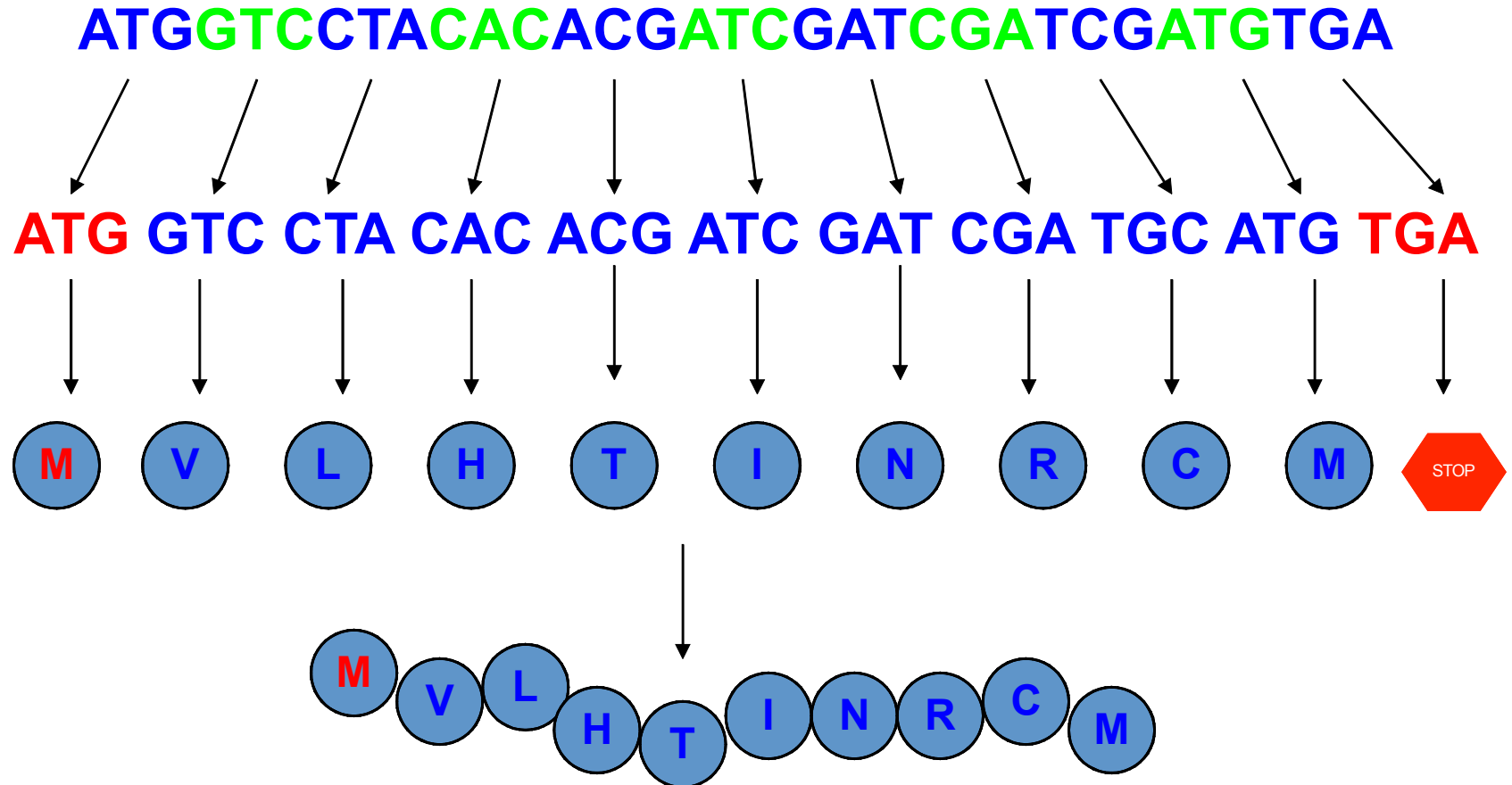


- DNA in our cells store information in a way that is very similar to the way computers do.
- Instead of being a binary memory, where everything is either 0 or 1, DNA is a 4 letter alphabet: A, C, G, T
- Using computer metaphor we can say that:
 - Plant cell do not look like a mouse cell because their “programs” are different
 - Liver cells work differently than lung cells because of different input to the program
 - Children look like parents because their program is a “revision” of parents program
 - Many diseases are caused by “bugs” in program:
 - Familial dysautonomia: A simple mistake in one line of code
 - Huntington’s disease: A “line” of code gets repeated a bunch of times by accident
- Different ways to solve the same problem:
 - Plants: photosynthesis = turn light into sugar
 - Animals: eat plant or other animals

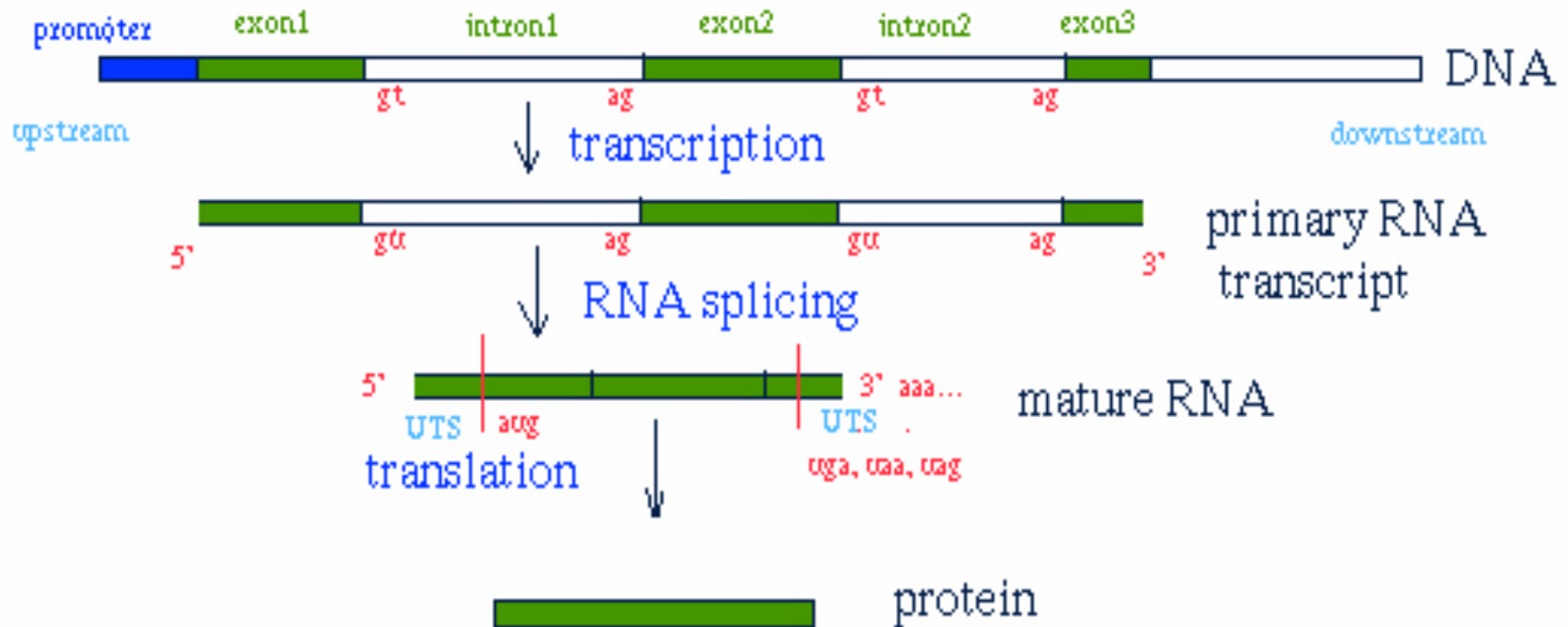
Genetic code

		SECOND BASE					
		U	C	A	G		
FIRST BASE	U	UUU } Phe	UCU } Ser	UAU } Tyr	UGU } Cys	THIRD BASE	U
		UUC } Phe	UCC } Ser	UAC } Tyr	UGC } Cys		C
		UUA } Leu	UCA } Ser	UAA } Stop	UGA } Stop		A
		UUG } Leu	UCG } Ser	UAG } Stop	UGG } Trp		G
	C	CUU } Leu	CCU } Pro	CAU } His	CGU } Arg		U
		CUC } Leu	CCC } Pro	CAC } His	CGC } Arg		C
		CUA } Leu	CCA } Pro	CAA } Gln	CGA } Arg		A
		CUG } Leu	CCG } Pro	CAG } Gln	CGG } Arg		G
	A	AUU } Ile	ACU } Thr	AAU } Asn	AGU } Ser		U
		AUC } Ile	ACC } Thr	AAC } Asn	AGC } Ser		C
		AUA } Met	ACA } Thr	AAA } Lys	AGA } Arg		A
		AUG } Met	ACG } Thr	AAG } Lys	AGG } Arg		G
G	GUU } Val	GCU } Ala	GAU } Asp	GGU } Gly	U		
	GUC } Val	GCC } Ala	GAC } Asp	GGC } Gly	C		
	GUA } Val	GCA } Ala	GAA } Glu	GGA } Gly	A		
	GUG } Val	GCG } Ala	GAG } Glu	GGG } Gly	G		

Genetic code

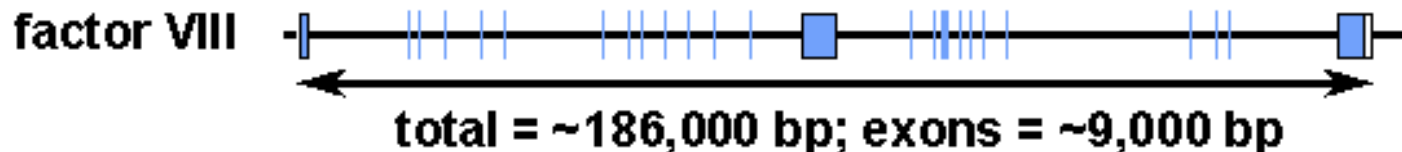
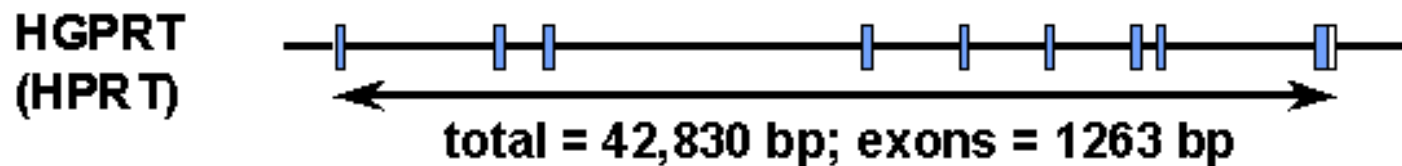
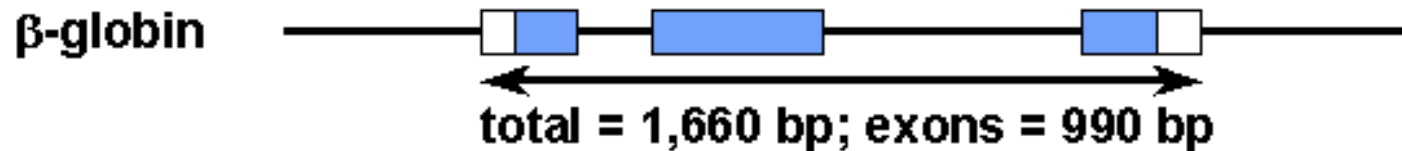
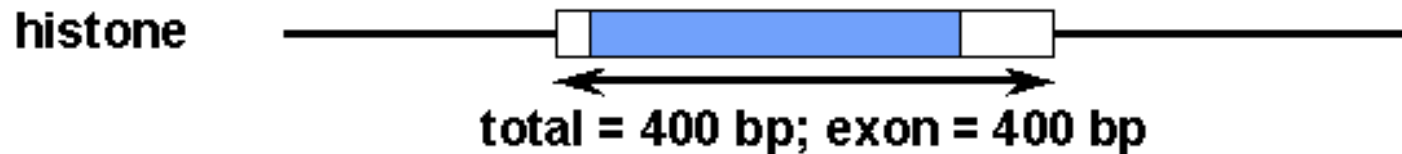


From DNA to protein



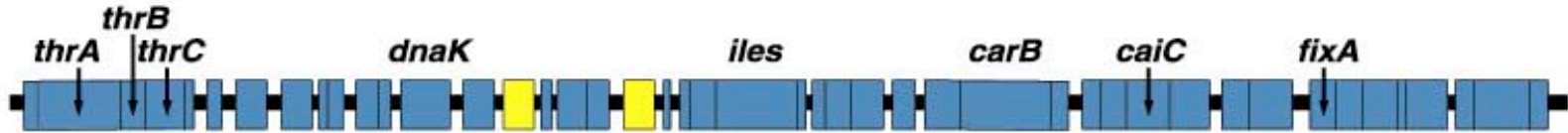
Gene structure

(exon-intron-exon)_n structure of various genes



Pseudogenes and repetitive elements

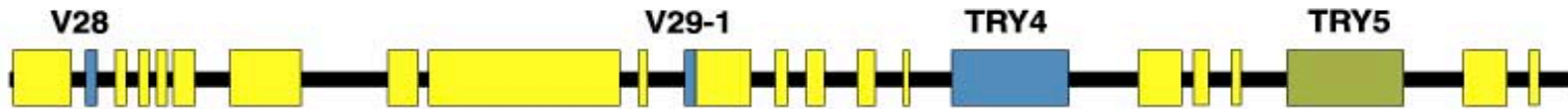
(a) *Escherichia coli*



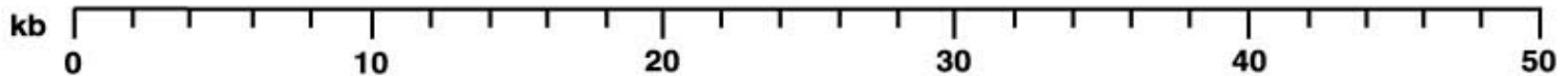
(b) *Saccharomyces cerevisiae*



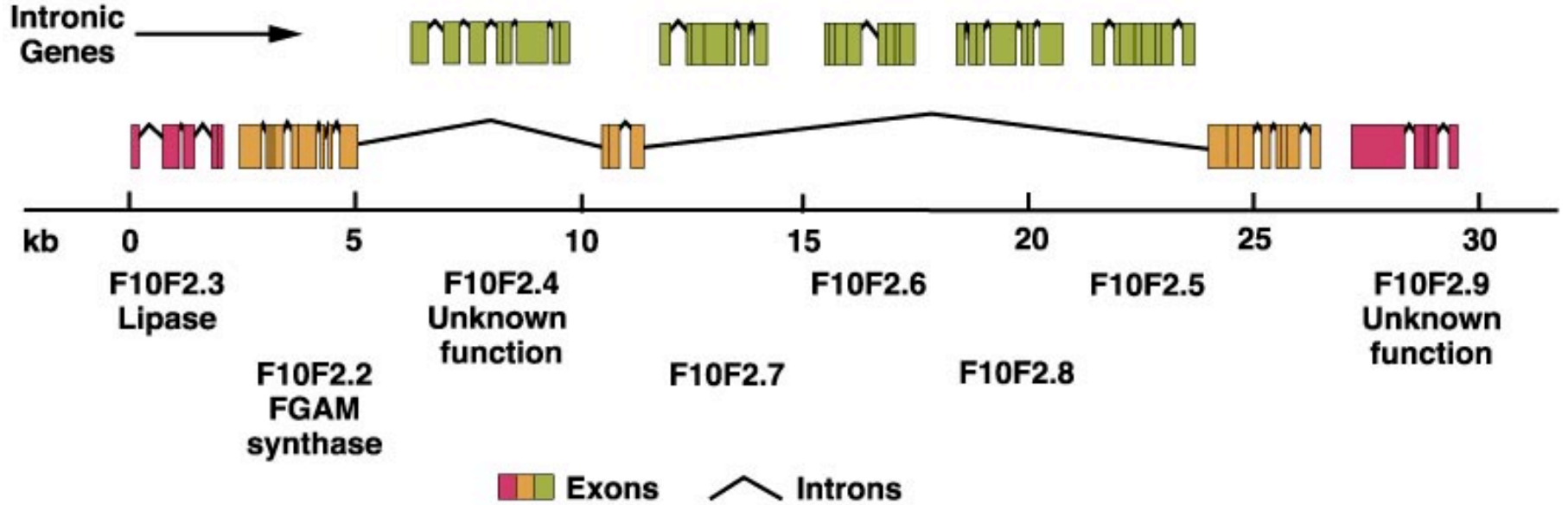
(c) Human



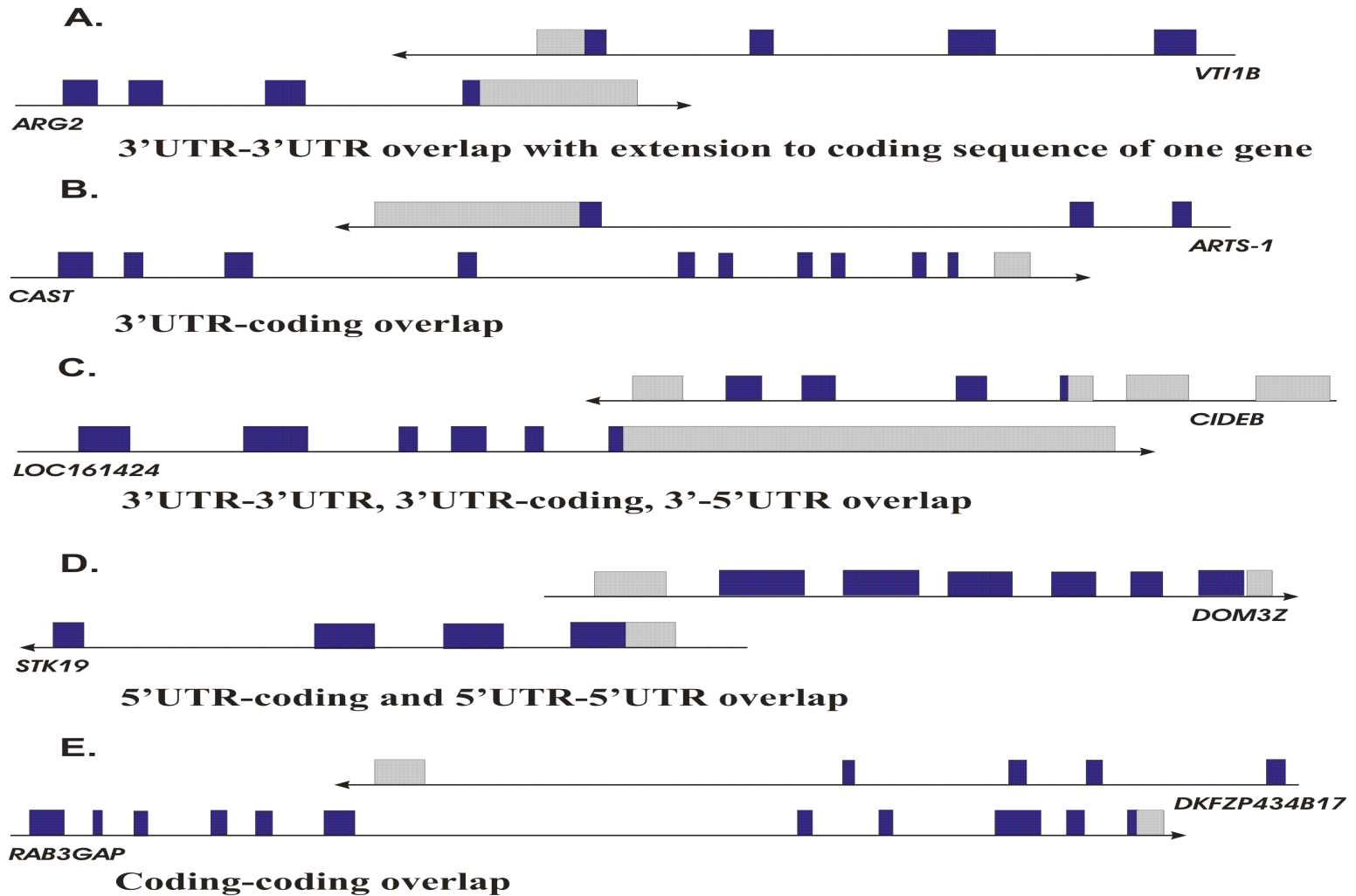
(d) Maize



Complicated gene structures



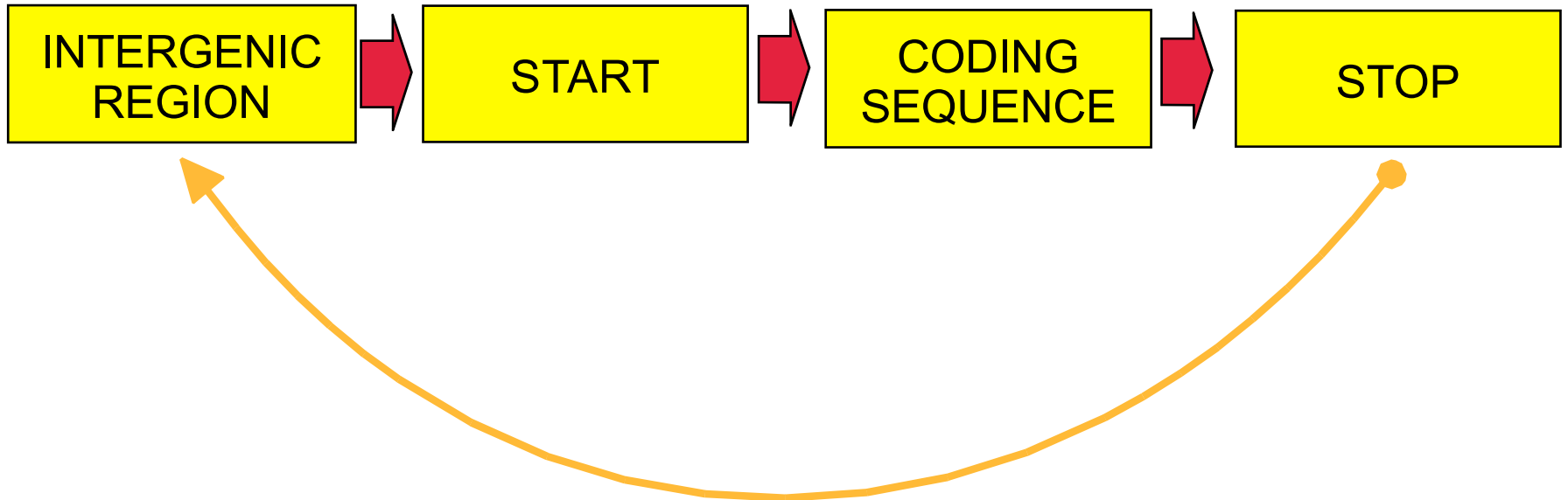
Genes may overlap



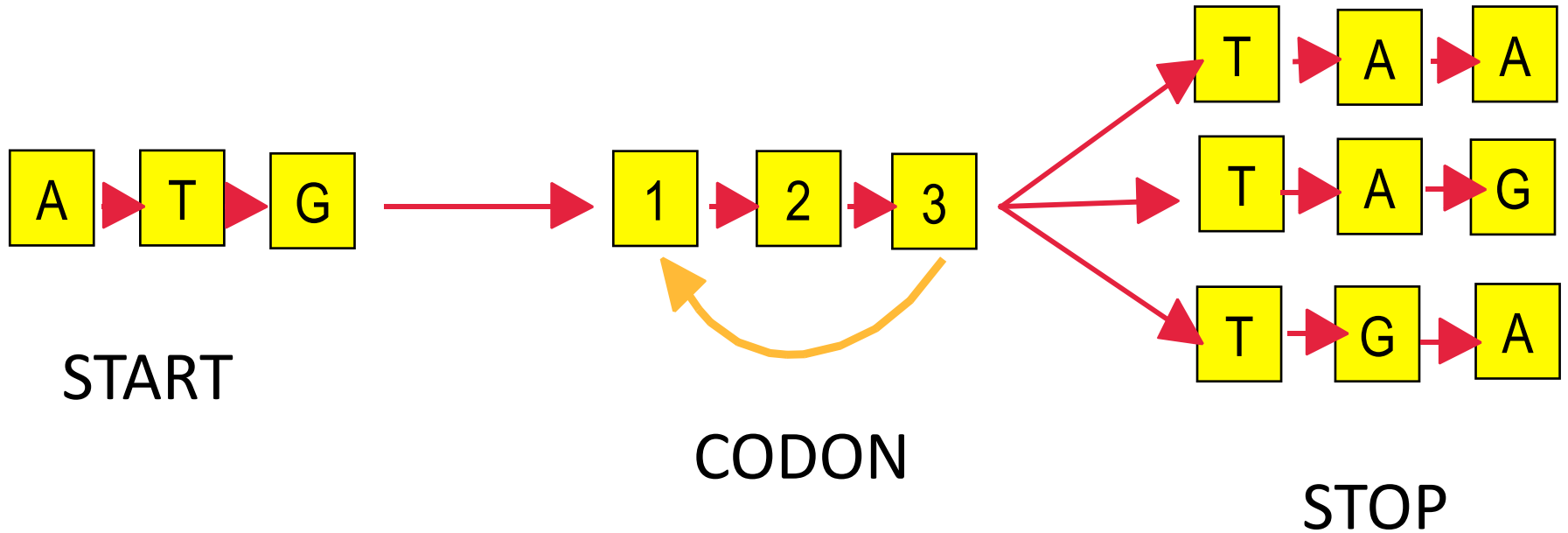
Pattern recognition and matching

- The ability of a program to compare novel and known patterns and determine the degree of similarity forms the basis of sequence analysis including gene identification. In similarity based methods we search the genome directly for nucleotide or amino acid pattern observed in one or more already known genes; in comparative genomics we look for similar sequence pattern in two or more genomes, and in method based prediction we look for patterns in sequence composition and signals.
- One of the major challenges associated with using pattern matching is in that, in most cases, we need to identify patterns that are 'similar' to a target pattern, but the concept of similarity isn't well defined from programmatic and biological sense. Also, only already known pattern may be used for searches, therefore genes with unusual patterns may not be discovered using these methods.

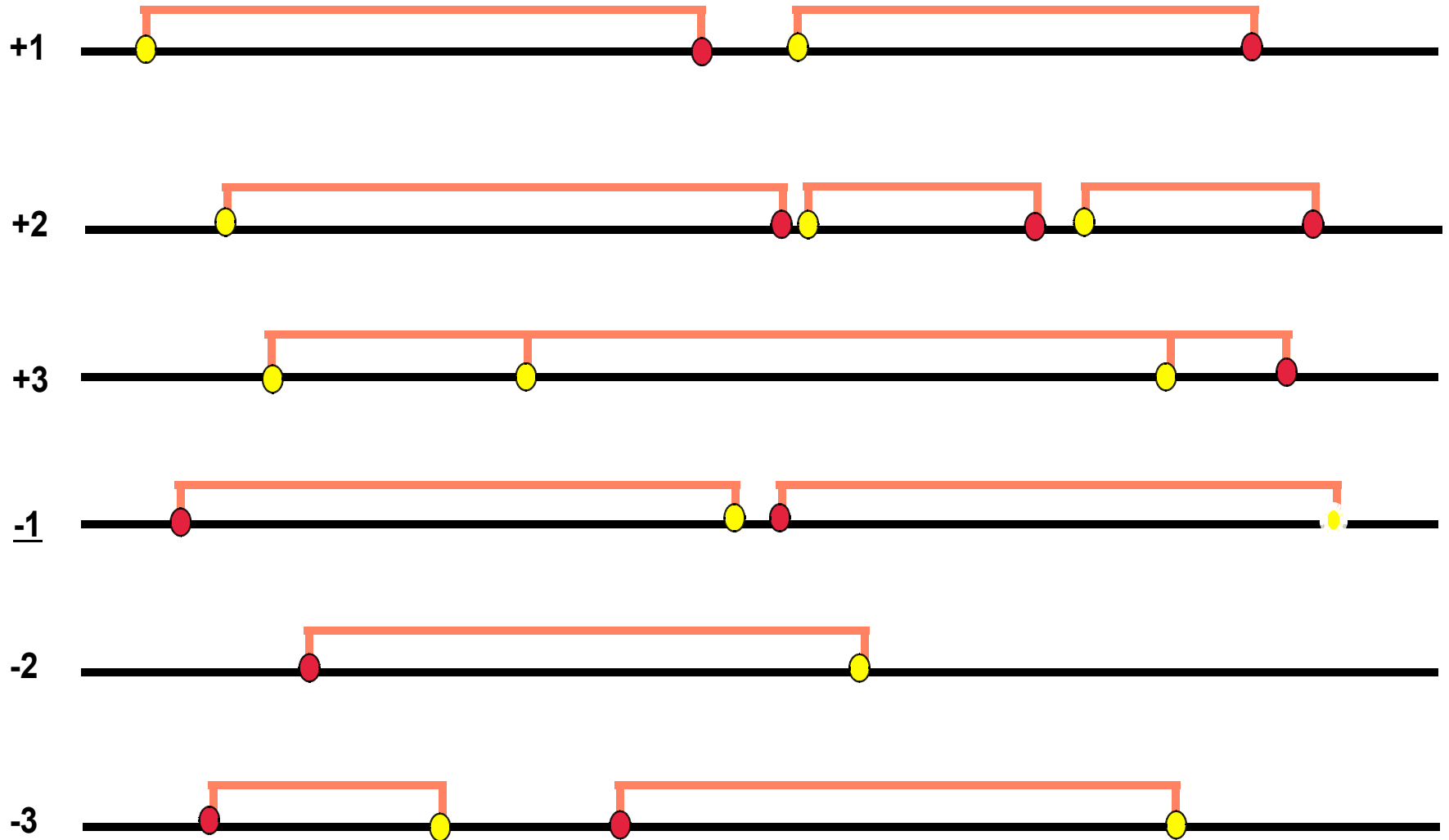
Basic model



Basic model



Open Reading Frames



Sequence features

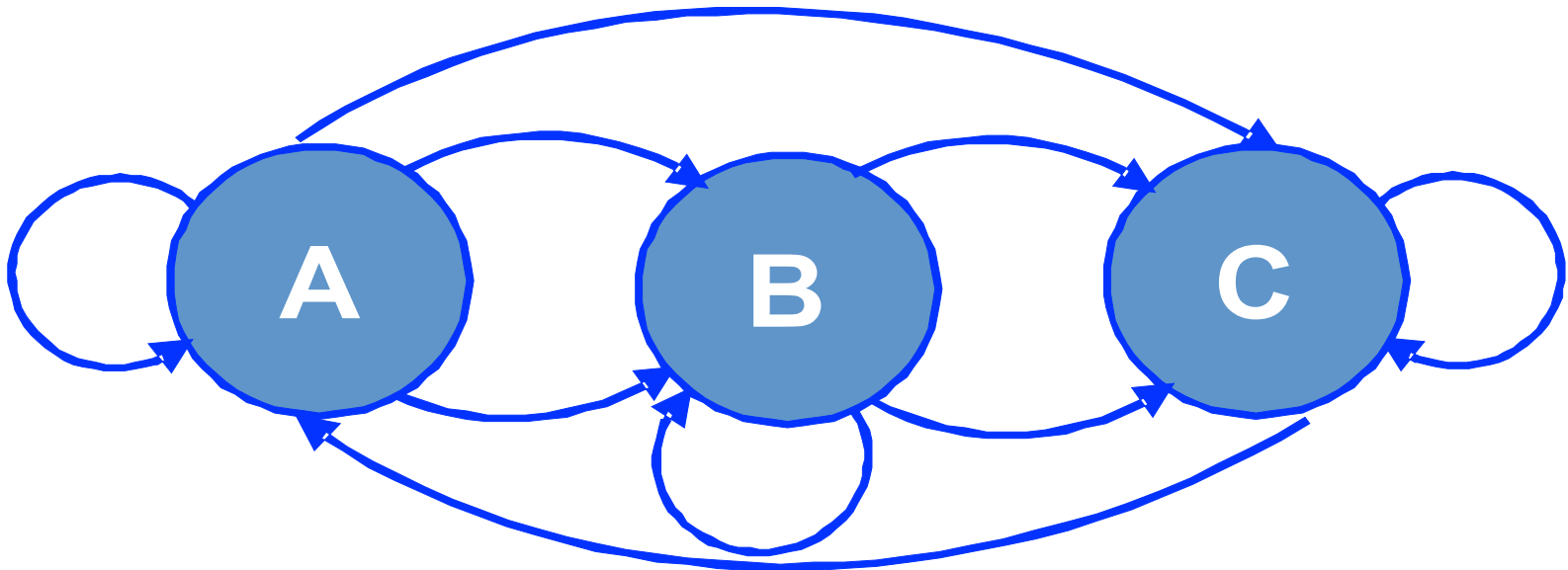
- We can check if sequence in particular ORF has some other features which could tell us if this is a putative coding sequence or the ORF is false positive. We can look at the sequence content and compare it with known coding sequence and non-coding sequence and check to which of these two the ORF sequence is more similar to.

Hidden Markov Models

- HMM is a statistical model for an ordered sequence of symbols, acting as a stochastic state machine that generates a symbol each time a transition is made from one state to the next. Transitions between states are specified by transition probabilities. A Markov process is a process that moves from state to state depending on the previous n states.
- HMM has been previously used very successfully for speech recognition.
- In biology is used to produce multiple sequence alignments, in generating sequence profiles, to analyze sequence composition and patterns, to produce a protein structure prediction, and to locate genes.
- In gene identification HMM is a model of periodic patterns in a sequence, representing, for example, patterns found in the exons of a gene. HMM provides a measure of how close the data pattern in the sequence resemble the data used to train the model.

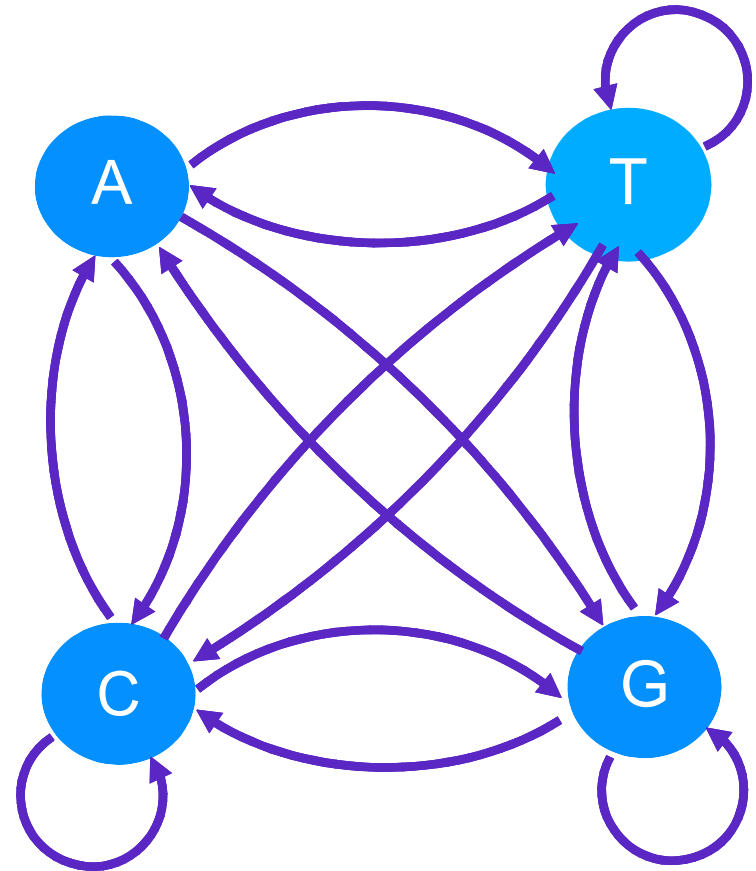
Markov Chains

- A Markov Chain is a non-deterministic system in which it is assumed that the probability of moving from one state to another doesn't vary with time. This means the current state and transition does not depend on what happened in the past. The Markov Chain is defined by probabilities for each occurring transition.



Markov Chains

In a sequence analysis we look at probabilities of transitions from one nucleotide to another. We can check, for example, if certain patterns of transition are more frequent in coding sequences than in non coding sequences.



Order of Markov Chains

GCGCTAGCGCCGATCATCTACTCG

GCGCT**AG**CGCCGATCATCTACTCG

GCGCTA**GC**GCCGATCATCTACTCG

}

First order

GCGCT**AG**CGCCGATCATCTACTCG

GCGCT**AGC**GCCGATCATCTACTCG

}

Second order

GCGCTAGCGCCGATCATCTACTCG

GC**GCTAGC**GCCGATCATCTACTCG

}

Fifth order

How far can we go?

- Order of our model will have influence on specificity and sensitivity of our program.
 - Too short sequences may not be specific enough and program may return a lot of false positives.
 - Long chains may be too specific and our program will not be sensitive enough returning false negatives.

Order of Markov Chains

GCGCTAGCGCCGATCATCTACTCG

GCGCT**AG**CGCCGATCATCTACTCG
GCGCT**AGC**GCCGATCATCTACTCG

} First order

GCGCT**AG**CGCCGATCATCTACTCG
GCGCT**AGC**GCCGATCATCTACTCG

} Second order

GCGCTAGCGCCGATCATCTACTCG
GCGCTAGCGCCGATCATCTACTCG

} Fifth order

20	G	
7	GA	7/20
1	GG	1/20
5	GT	5/20
7	GC	7/20

For non-coding sequence we assume that probability of each transition is equal. The more 'popular' in coding sequence transition, the higher probability the sequence is coding

Probability matrix

$$4^{K+1}$$

first order Markov Model - matrix of 16 probabilities

$p(A/A), p(A/T), p(A/C), p(A/G)$

$p(T/A), p(T/T), p(T/C), p(T/G)$

$p(C/A), p(C/T), p(C/C), p(C/G)$

$p(G/A), p(G/T), p(G/C), p(G/G)$

$$4^{1+1} = 4^2 = 16$$

$$4^{2+1} = 4^3 = 64 \quad 4^{3+1} = 4^4 = 256$$

GCG CTA GCG CCG ATC ATC TAC TCG
G CGC TAG CGC CGA TCA TCT ACT CG
GC GCT AGC GCC GAT CAT CTA CTC G

Frequencies of transitions may depend on in which codon position (1st, 2nd, or 3rd) is a given nucleotide (state)

Number of probabilities

Codon position 1	Codon position 2	Codon position 3
A C G T	A C G T	A C G T
A .36 .27 .35 .18	A .16 .19 .15 .07	A .22 .33 .24 .13
C .21 .23 .24 .27	C .28 .44 .41 .33	C .21 .29 .27 .21
G .19 .14 .23 .23	G .40 .12 .27 .45	G .44 .15 .37 .53
T .24 .35 .19 .31	T .16 .25 .17 .16	T .13 .22 .12 .13

$$4^{1+1} = 4^2 = 16$$

$$4^{1+1} = 4^2 = 3 \times 16 = 48$$

Calculating coding potential of a given sequence

To estimate if the sequence is coding we have to calculate probability that sequence is coding and probability the sequence is non-coding. Next we calculate logarithm from the ratio of these two probability values.

$$LP(S) = \log \frac{P^i(S)}{P_0(S)}$$

If the calculated value is > 0 the likelihood that the sequence is coding is higher than the sequence is not coding, if value is < 0 there is higher likelihood that sequence is not coding.

Coding vs. non coding sequence

A/A	C/A	G/A	T/A coding
0.36	0.21	0.19	0.24

A/A	C/A	G/A	T/A non coding
0.25	0.25	0.25	0.25

Markov Models - probabilities

$$LP(S) = \log \frac{P^i(S)}{P_0(S)}$$

S=AGGACG

	Codon position 1				Codon position 2				Codon position 3					
	A	C	G	T	A	C	G	T	A	C	G	T		
A	.36	.27	.35	.18	A	.16	.19	.15	.07	A	.22	.33	.24	.13
C	.21	.23	.24	.27	C	.28	.44	.41	.33	C	.21	.29	.27	.21
G	.19	.14	.23	.23	G	.40	.12	.27	.45	G	.44	.15	.37	.53
T	.24	.35	.19	.31	T	.16	.25	.17	.16	T	.13	.22	.12	.13

$$P(S) = f(A,1)F(G,A)F(G,G)F(A,G)F(C,A)F(G,C)$$

$$P(S) = 0.27 \times 0.19 \times 0.27 \times 0.24 \times 0.21 \times 0.12 = 0.00008377$$

$$P(S) = 0.25 \times 0.25 \times 0.25 \times 0.25 \times 0.25 \times 0.25 = 0.0002441$$

$$LP(S) = \log(0.00008377/0.0002441) = -0.4644$$

Calculating LP

$$LP(S) = \log \frac{P^i(S)}{P_0(S)}$$

	Codon position 1				Codon position 2				Codon position 3					
	A	C	G	T	A	C	G	T	A	C	G	T		
A	.36	.27	.35	.18	A	.16	.19	.15	.07	A	.22	.33	.24	.13
C	.21	.23	.24	.27	C	.28	.44	.41	.33	C	.21	.29	.27	.21
G	.19	.14	.23	.23	G	.40	.12	.27	.45	G	.44	.15	.37	.53
T	.24	.35	.19	.31	T	.16	.25	.17	.16	T	.13	.22	.12	.13

S=AGGACG

$$LP(S) = \log \frac{0.27}{0.25} + \log \frac{0.19}{0.25} + \log \frac{0.27}{0.25} + \log \frac{0.24}{0.25} + \log \frac{0.21}{0.25} + \log \frac{0.12}{0.25}$$

$$LP(S) = \log 1.08 + \log 0.76 + \log 1.08 + \log 0.96 + \log 0.84 + \log 0.48$$

$$LP(S) = 0.0334 + (-0.1191) + 0.0334 + (-0.0177) + (-0.0757) + (-0.3187)$$

$$LP(S) = -0.4644$$

GLIMMER

- Gene finding program for prokaryotes
- Saltzberg et. al, 1998
- For prediction uses:
 - Start
 - Stop
 - Sequence composition
 - Interpolated Markov Models

The GLIMMER system

- Part 1 – Program is trained for a given data set (species)
- Part 2 – Program identifies putative genes in the genomic sequence
 - Identify all ORFs longer than a threshold
 - Score each ORF in each reading frame and select these which gets the highest score in correct reading frame
 - Score overlapping genes in each frame separately to see which frame scores the highest

Running the program

- First run *build-imm* on a set of sequences to make the Markov models (long ORFs from the same or closely related species)
 - *build-imm train.seq*
- Next run *GLIMMER* to find genes in your sequence
 - *glimmer your.seq train.seq <options>*

GLIMMER options

- -g set minimum gene length
- -o set minimum overlap
- -p set minimum overlap percentage
- +r/-r independent probability score ON/OFF
- -t set threshold score for calling as gene

GLIMMER output

Minimum gene length = 180
 Minimum overlap length = 30
 Minimum overlap percent = 10.0%
 Threshold score = 90
 Use independent scores = True
 Use first start codon = True

ID#	Fr	Orf Start	Gene Start	End	Orf Length	Gene Length	Gene Score	-- Frame Scores -						Indep Score	
								F1	F2	F3	R1	R2	R3		
1	F2	302	305	616	315	312	0	-	0	-	99	-	-	0	
	R1	660	633	220	441	414	99	-	-	-	99	-	-	0	
	F2	620	650	901	282	252	0	-	0	-	-	-	99	0	
2	R3	1114	1105	638	477	468	99	-	-	-	-	-	99	0	
	F3	1119	1140	1466	348	327	0	-	-	0	-	-	99	0	
3	R3	2026	1999	1118	909	882	99	-	-	-	-	-	99	0	
4	F3	1815	1830	2054	240	225	99	-	-	99	-	-	-	0	
*** Overlaps #3		by 170		Overlap Region		Scores:		-	-	0	-	-	99	0	
Reject #4															
5	R2	2600	2597	1935	666	663	99	-	-	-	-	99	-	0	
*** Overlaps #4		by 120		Overlap Region		Scores:		-	-	0	-	99	-	0	
6	F1	2710	2719	3399	690	681	99	99	-	-	-	-	-	0	
	R3	4153	4153	3962	192	192	0	99	-	-	-	-	0	0	
7	F1	3403	3403	4230	828	828	99	99	-	-	-	-	-	0	
	R2	4700	4679	4455	246	225	0	-	-	-	-	0	-	99	
	R2	68906	68897	68670	237	228	13	-	-	-	-	-	13	86	
8	R1	101574	101544	101296	279	249	96	-	-	-	96	-	-	3	
	R3	193228	193204	193022	207	183	56	-	-	-	-	-	56	43	

List of putative genes

Putative Genes:

1	633	220	
2	1105	638	
3	1999	1118	
5	2597	1935	
6	2719	3399	
7	3403	4230	
...			
39	38472	38741	[Shorter 40 80 74]
40	38662	39450	[Bad Overlap 39 80 25]
...			
482	464206	464424	[Shadowed by 483]
...			
636	616213	615965	[Delay by 33 637 50 0]

Output description

```
39 38472 38741 [Shorter 40 80 74]  
40 38662 39450 [Bad Overlap 39 80 25]
```

[Bad Overlap *a b c*] means that gene number *a* overlapped this one and was shorter but scored higher on the overlap region. *b* is the length of the overlap region and *c* is the score of *this* gene on the overlap region. There should be a [Shorter ...] notation with gene *a* giving its score.

[Shorter *a b c*] means that gene number *a* overlapped this one and was longer but scored lower on the overlap region. *b* is the length of the overlap region and *c* is the score of *this* gene on the overlap region. There should be a [Bad overlap ...] notation with gene *a* giving its score.

Output description - 2

482 464206 464424 [Shadowed by 483]

...

636 616213 615965 [Delay by 33 637 50 0]

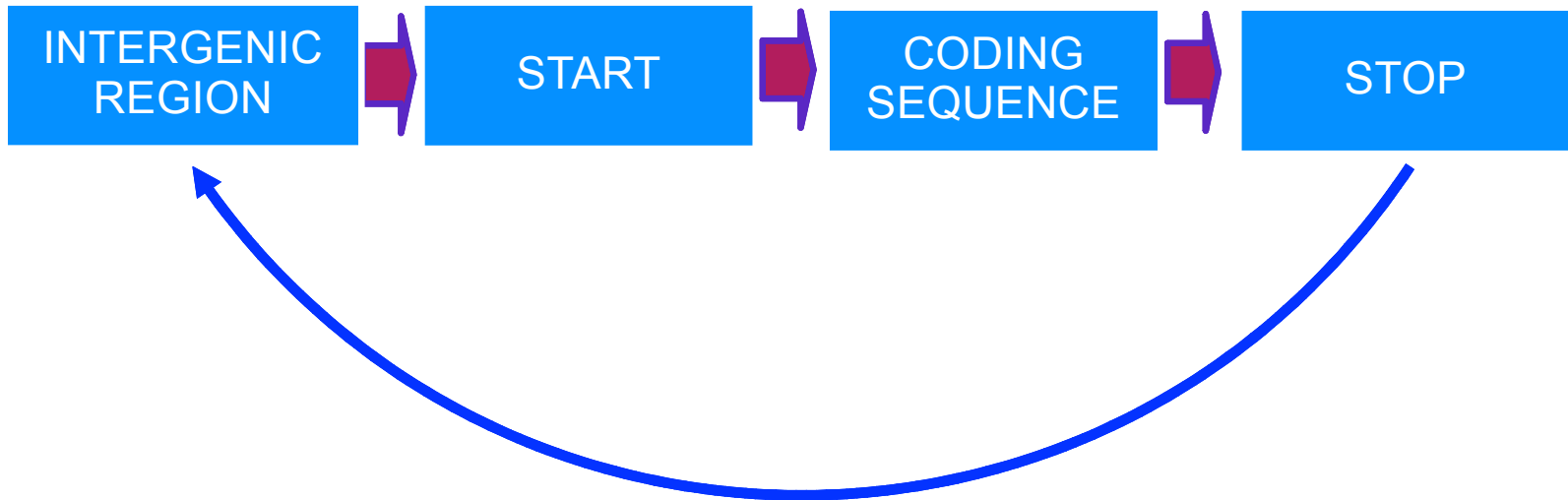
[Shadowed by *a*] means that this gene was completely contained as part of gene *a*'s region, but in another frame.

[Delay by *a b c d*] means that this gene was tentatively rejected because of an overlap with gene *b*, but if the start codon is postponed by *a* positions, then this would be a valid gene. The start position reported for this gene includes the delay. *c* is the length of the overlap region that caused the rejection and *d* is the score in this gene's frame on that overlap region.

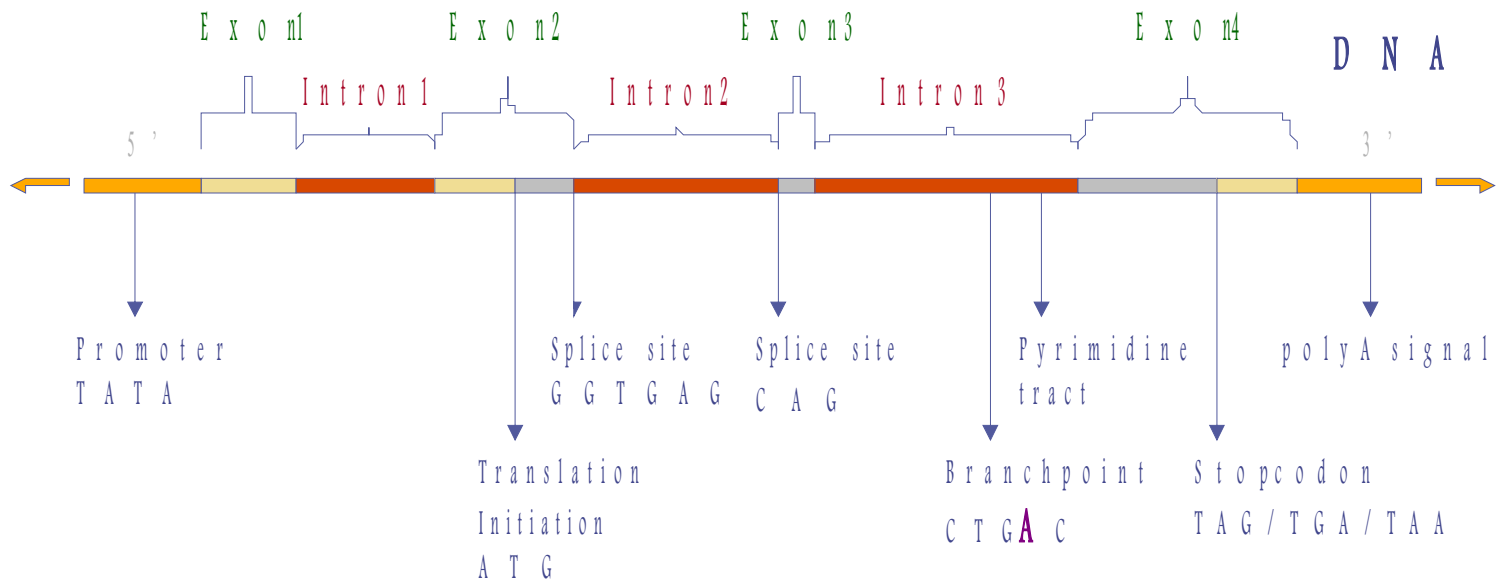
Prokaryotic vs. Eukaryotic Genes

- Prokaryotes
 - small genomes
 - high gene density
 - no introns (or splicing)
 - no RNA processing
 - similar promoters
 - terminators important
 - overlapping genes
- Eukaryotes
 - large genomes
 - low gene density
 - introns (splicing)
 - RNA processing
 - heterogeneous promoters
 - terminators not important
 - overlapping genes
 - polyadenylation

Coding regions in Prokaryotes



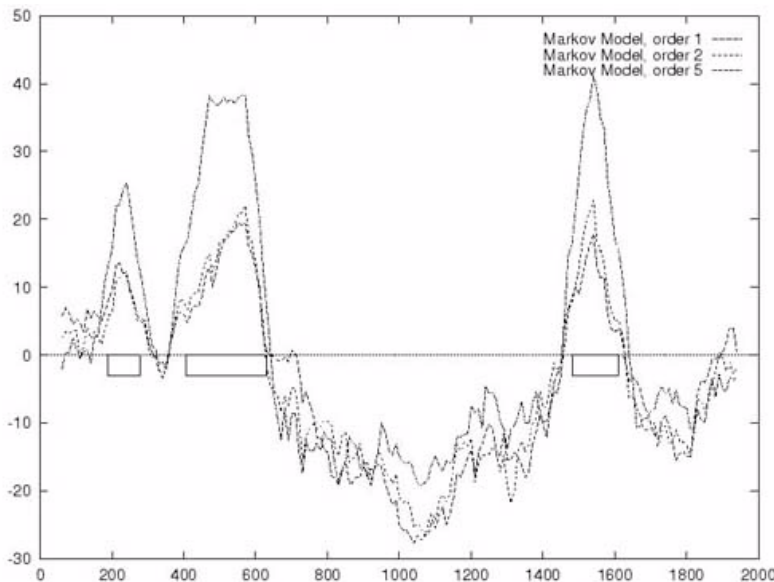
Eukaryotic gene structure



Searching for coding sequences using Markov chains

In this case we do not want check if given sequence fragment is coding or not but we rather want to identify coding fragments in a long sequence. In most cases this is done by calculating statistics in overlapping windows.

AGTACGATATTAGCGGCAATCGTATGACTACGTCTTGCTACGTCTTCTCTCGTCTGCTCTAG



This way profiles are created. This example shows a profile for a sequence analyzed using a 120-bp window and a 10-bp step.

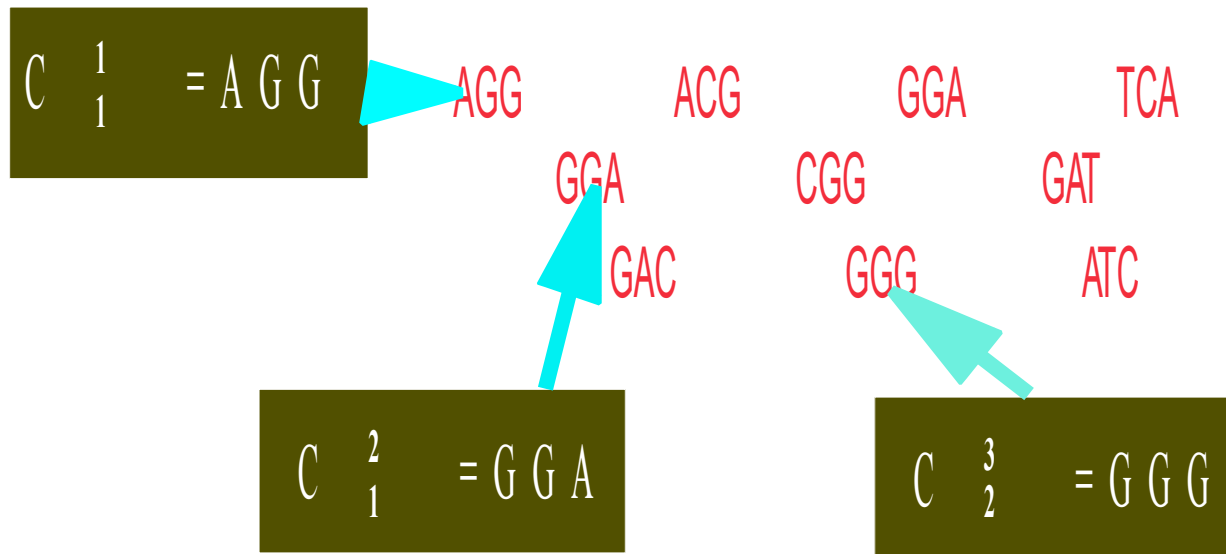
Codon usage

Gly	GGG	17.08	0.23	Arg	AGG	12.09	0.22
Gly	GGA	19.31	0.26	Arg	AGA	11.73	0.21
Gly	GGT	13.66	0.18	Ser	AGT	10.18	0.14
Gly	GGC	24.94	0.33	Ser	AGC	18.54	0.25
Glu	GAG	38.82	0.59	Lys	AAG	33.79	0.60
Glu	GAA	27.51	0.41	Lys	AAA	22.32	0.40
Asp	GAT	21.45	0.44	Asn	AAT	16.43	0.44
Asp	GAC	27.06	0.56	Asn	AAC	21.30	0.56
Val	GTG	28.60	0.48	Met	ATG	21.86	1.00
Val	GTA	6.09	0.10	Ile	ATA	6.05	0.14
Val	GTT	10.30	0.17	Ile	ATT	15.03	0.35
Val	GTC	15.01	0.25	Ile	ATC	22.47	0.52
Ala	GCG	7.27	0.10	Thr	ACG	6.80	0.12
Ala	GCA	15.50	0.22	Thr	ACA	15.04	0.27
Ala	GCT	20.23	0.28	Thr	ACT	13.24	0.23
Ala	GCC	28.43	0.40	Thr	ACC	21.52	0.38

Codon usage

DNA sequence can be divided into non-overlapping codons in three reading frames

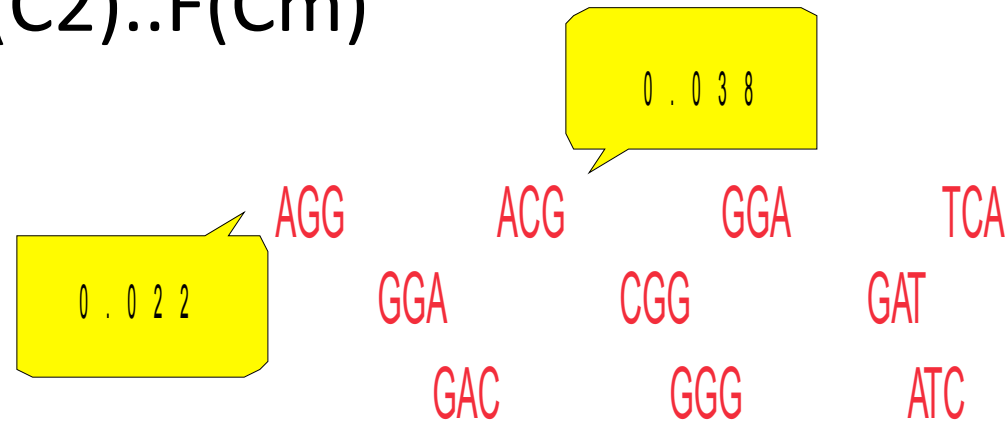
$$C = C_1 C_2 \dots C_m$$



Probability that sequence is coding

Probability that sequence is coding is equal probability that sequence of codons is coding. Assuming independence between adjacent codons the probability that sequence is coding will be equal to the product of codon frequencies.

$$P(C) = F(C1)F(C2)..F(Cm)$$



$$P(C) = F(AGG)F(ACG) = 0.022 \times 0.038 = 0.000836$$

Probability that sequence is non-coding

If the sequence is non-coding the codon frequency will be random and each codon will be equally probable. In this case frequency for each codon will be 0.0156. This is because we have 64 codons and each of them is equally possible.

Therefore probability that the sequence is non-coding will be:

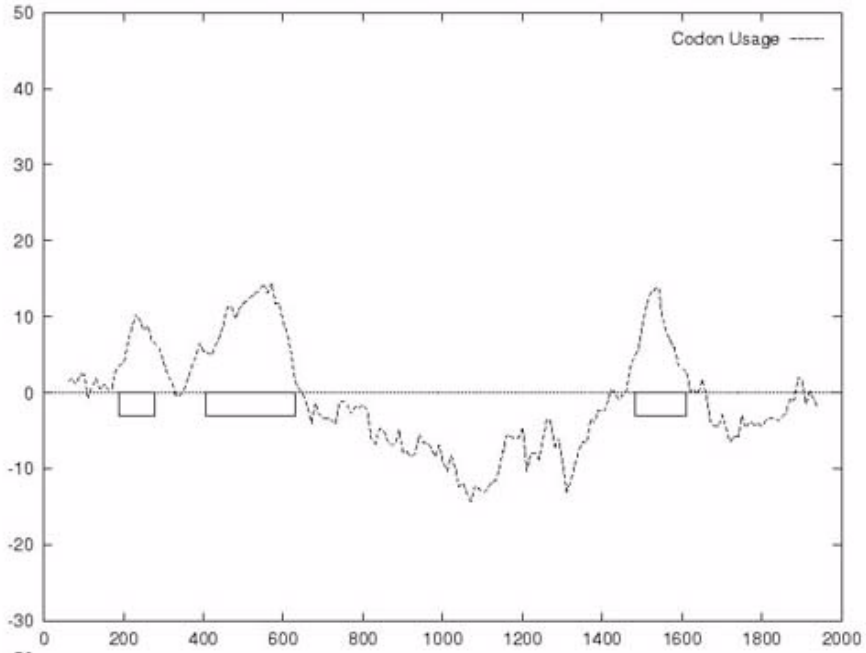
$$P(C) = F(\text{AGG})F(\text{AGC}) = 0.0156 \times 0.0156 = 0.000244$$

Log-likelihood ratio

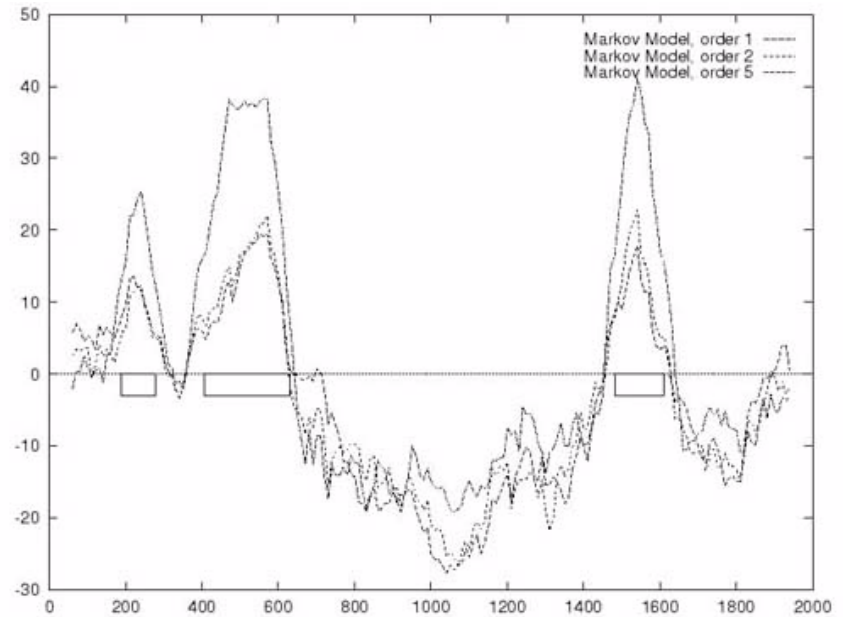
$$LP(S) = \log \frac{P_i(S)}{P_0(S)}$$

$$LP(S) = \log 1.4102 + \log 2.4358 = 0.1493 + 0.3866 = 0.53 > 0$$

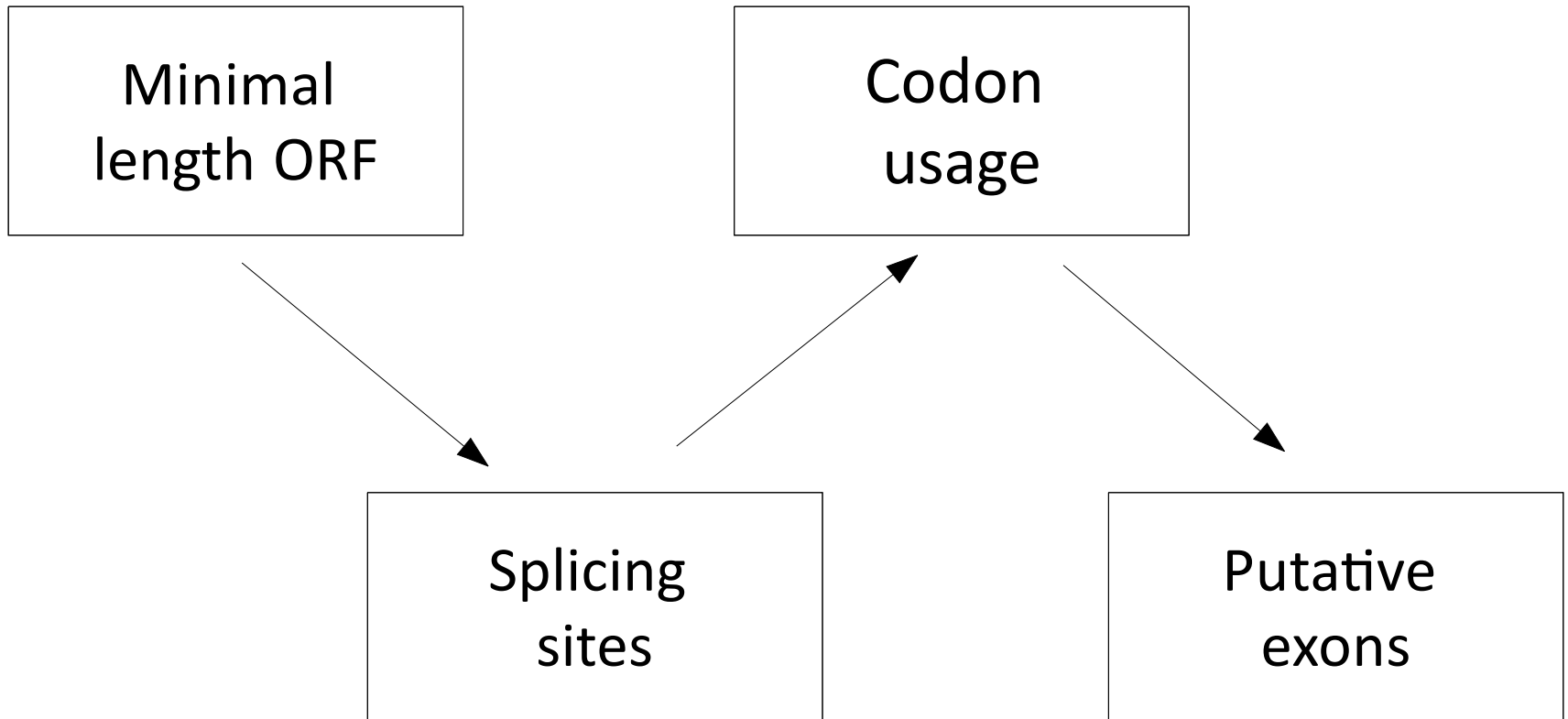
Codon usage



Markov models



Rule based methods



Gene identification programs

- The first generation of programs was designed to identify approximate locations of coding regions in genomic DNA (e.g. GRAIL). These methods could not accurately predict precise exon location.
- The second generation (e.g. MZEF, SORFIND, and Xpound) combined splice signals and coding region identification but did not attempt to assemble predicted exons into complete genes.
- Third generation (GeneID, GeneParser, GenLang, FGENES) predicted entire gene structures but their performance was rather poor. One of problems was the assumption that the input sequence contains complete genes.
- Fourth generation of programs is represented by GENSCAN or TWINSKAN. With improved accuracy and less restricted requirements (e.g. allow partial genes) these programs are considered to be the best and are widely used in large-scale genomes analysis.

Classes of gene prediction methods

- Sequence similarity based
 - BLAST can be used for aligning ESTs or proteins to the genomic sequence
 - PROCRUSTES and GenWise use global alignment of homologous protein to genomic sequence
 - The biggest limitation to this type of approaches:
 - only about half of genes being discovered have significant similarity to genes in the database
 - genes with very limited expression may never be discovered
- Model based predictors
 - GENSCAN, Genie, HMMGene, FGENES – rely on two types of sequence information: signal sensors and content sensors
 - Limitations of these approaches:
 - Newly sequenced genomes very often lack large enough samples of known genes to estimate model parameters
 - Need to be retrained as the number of available genes is growing
 - Genes of less typical structure or having rare signals may not be discovered

GRAIL

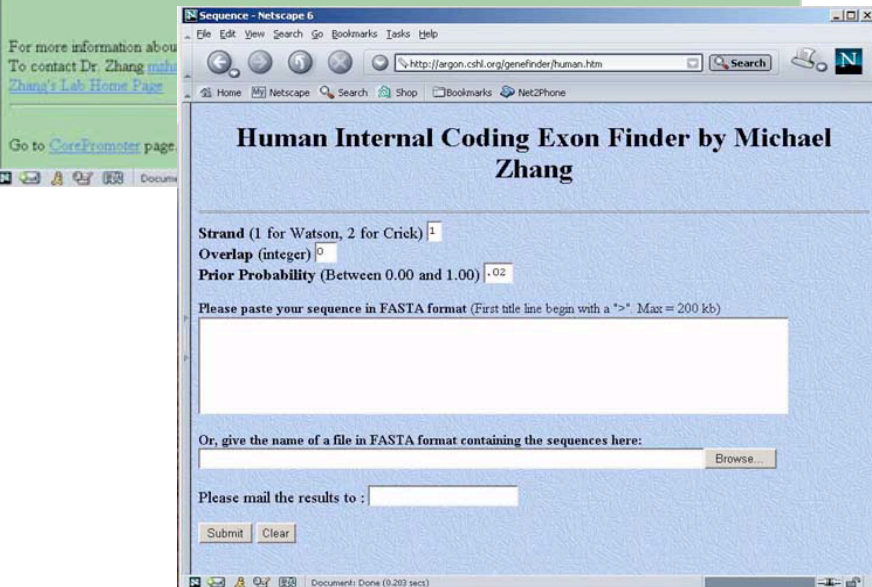
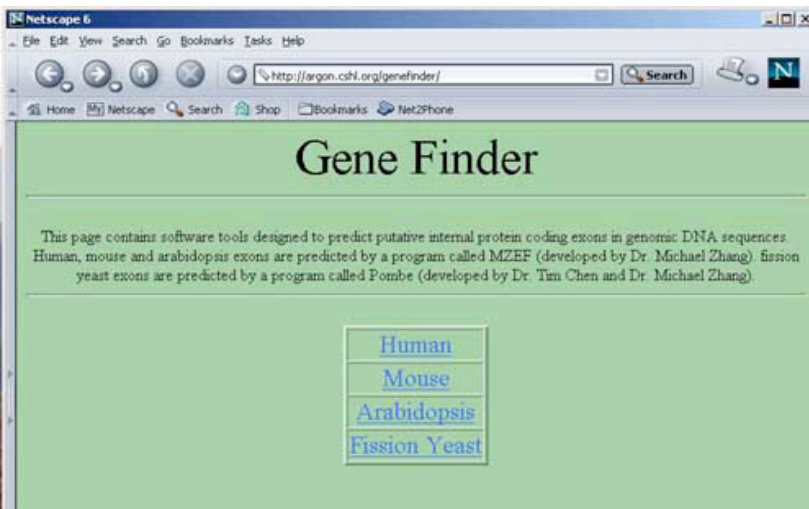
- Gene recognition and analysis
- Uberbacher and Mural, 1991
- First gene prediction program
- GRAIL1
 - Neural network recognizing coding potential within fixed-size window (100 bp)
 - Evaluates coding potential without looking for additional features (e.g. splice junctions, start and stop codons)
- GRAIL2
 - Variable size of windows
 - Incorporated genomic context information (splice sites, start and stop, polyadenylation signals)
- GrailEXP - <http://compbio.ornl.gov/grailexp/>
 - GrailEXP is a software package that predicts exons, genes, promoters, polyAs, CpG islands, EST similarities, and repetitive elements within DNA sequence.

MZEF

- M. Zhang 1997
- Predicts exons only, does not build gene structure
- Uses ‘quadratic discriminant analysis’
- Variable measures:
 - Exon length
 - Intron-exon transition
 - Branch site scores

MZEF

<http://rulai.cshl.org/tools/genefinder/>



MZEF Results - human -
GAATTCAGGGGGACTTGGAAAGGTACATCTGAGTTCATCTTCCAGGAGTCCACACACTTAAATC
Wed Jun 5 12:26:39 2002

Strand = 1
Overlap = 0
Prior Prob. = .02

Internal coding exons predicted by MZEF
Sequence_length: 140257 G+C_content: 0.524

Coordinates	P	Fr1	Fr2	Fr3	Orf	3ss	Cds	5ss
655 - 732	0.814	0.606	0.432	0.457	121	0.508	0.539	0.565
38250 - 38615	0.744	0.583	0.441	0.438	122	0.555	0.545	0.486
39390 - 39801	0.718	0.498	0.449	0.515	122	0.513	0.533	0.695
48629 - 48682	0.994	0.421	0.412	0.658	111	0.557	0.540	0.609
75440 - 75484	0.954	0.365	0.501	0.633	111	0.547	0.559	0.616
76590 - 76634	0.733	0.587	0.433	0.463	211	0.533	0.556	0.650
97315 - 97474	0.530	0.380	0.573	0.368	212	0.496	0.514	0.640
101417 - 101470	0.999	0.337	0.597	0.384	212	0.592	0.515	0.690
105937 - 105990	0.832	0.537	0.665	0.313	112	0.505	0.569	0.547
108000 - 108053	0.933	0.585	0.417	0.557	111	0.560	0.545	0.608
110558 - 110611	0.998	0.396	0.631	0.595	112	0.574	0.609	0.723
110772 - 110825	0.600	0.562	0.382	0.533	121	0.541	0.522	0.627
122931 - 122984	0.618	0.460	0.384	0.578	221	0.515	0.553	0.618
123217 - 123270	0.977	0.460	0.603	0.418	111	0.555	0.535	0.673
128879 - 128932	0.866	0.441	0.488	0.489	112	0.547	0.530	0.689
135760 - 135813	0.790	0.532	0.468	0.283	111	0.571	0.489	0.683

GENSCAN

- Burge, C. and Karlin, S. (1997) Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.* 268, 78-94.
- Search for general and specific compositional properties of distinct functional units in eukaryotic genes
- General fifth-order Markov model of coding regions
- Analyzes both DNA strands
- Sequences may contain multiple and/or partial genes
- <http://genes.mit.edu/GENSCAN>



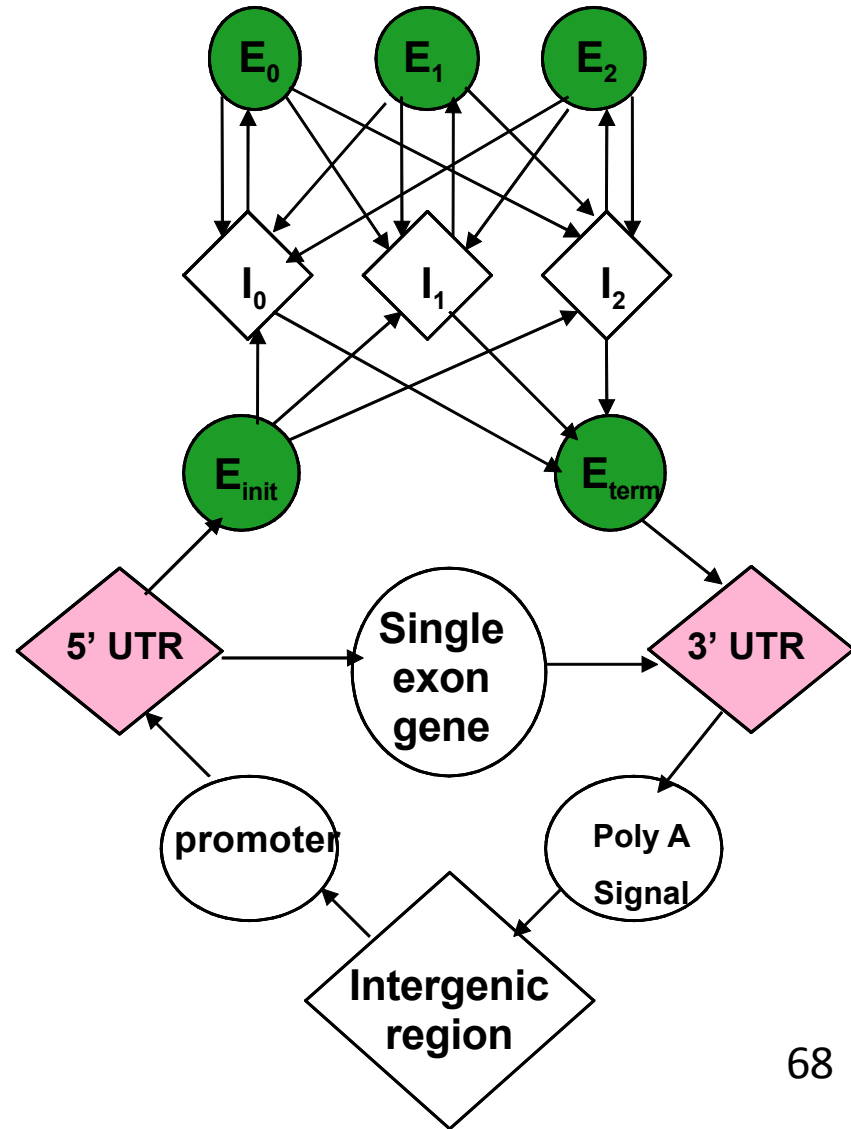
Each state corresponds to one of the seven categories with which all nucleotides are ultimately labeled- promoter, 5'UTR, exon, intron, 3'UTR, PolyA, intergenic

Three components:

Transition model – specifies probability of moving from any one state to another

Duration model – specifies the probability of staying in a given state

State specific sequence models – specifies the probability of any given nucleotide sequence being generated from any given state




GENSCAN options

New GENSCAN Web Server at MIT - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Kryzys w mojej ... AIM Mail Amazon.com: F... GrailEXP Home ... http://r...efinder/ New GENSCAN x

 [For information about Genscan, click here](#)

This server provides access to the program Genscan for predicting the locations and exon-intron structures of genes in genomic sequences from a variety of organisms.

This server can accept sequences up to 1 million base pairs (1 Mbp) in length. If you have trouble with the web server or if you have a large number of sequences to process, request a local copy of the program (see instructions at the bottom of this page) or use the [GENSCAN email server](#). If your browser (e.g., Lynx) does not support file upload or multipart forms, use the [older version](#).

Organism: Suboptimal exon cutoff (optional):

Sequence name (optional):

Print options:

Upload your DNA sequence file (one-letter code, upper or lower case, spaces/numbers ignored):

Or paste your DNA sequence here (one-letter code, upper or lower case, spaces/numbers ignored):

- Organism
 - vertebrate
 - Maize
 - Arabidopsis
- Output
 - predicted peptides only
 - predicted CDS and peptides
- Suboptimal exon cutoff
 - 1.00
 - 0.50
 - 0.25
 - ...
 - 0.01

GENSCAN output

```

GENSCANW output for sequence GBP

GENSCAN 1.0      Date run: 5-Jun-102      Time: 11:19:45

Sequence GBP : 2854 bp : 44.85% C+G : Isochore 2 (43 - 51 C+G%)

Parameter matrix: HumanIso.smat

Predicted genes/exons:

Gn.Ex Type S .Begin ...End .Len Fr Ph I/Ac Do/T CodRq P.... Tscr...
-----
1.01 Intr + 1452 1663 212 2 2 34 16 201 0.326 5.21
1.02 Intr + 1841 2084 244 2 1 34 34 245 0.690 11.20
1.03 Intr + 2172 2319 148 2 1 64 29 204 0.997 11.91
1.04 Term + 2368 2774 407 2 2 5 53 449 0.868 28.45

Click here to view a PDF image of the predicted gene(s)

Click here for a PostScript image of the predicted gene(s)

Predicted peptide sequence(s):

Predicted coding sequence(s):

>GBP|GENSCAN_predicted_peptide_1|336_aa
KFGELANTKESKALBGLYHGQVLCCKENTSGAPQKDVKHLAIPGAGENNGAGIAQVSVDRGR
KTIKLDATLTGFKHRVLKEVEAVIPDHYVFAZNTSPLPVSEIAAVSKRPERVIGRHYFSP
ADKMQLLEMITTRKTSKDTASTVAIGLEKQKREGVDPKLDLSLTSLGLPUGAATLVDEV
GVDVAKHIVAKDLGKAFGEQFGGRQSGRGFSIYQESVFNKQNLNSDNNGLASLKNPPKSEV
SSDEDIOPRLLTRFVNEAVTQPQEGILATPAEGDIGAVFGLSFPPLCGGPFVFDLYEAQ
KIVDGLKKYEAAYGKEFTPSQLLADSTHSPNKKFHQ

>GBP|GENSCAN_predicted_CDS_1|1011_bp
aaattcgggagagcttgcgaatgaccaagaatcaaggccttgatgggacttaccatggt
caggtcctgtgcgaagaagaatcacatctggagctccacagaaggatgttaagcatctggot
attcctgggtgcaggggatgatggggagcagccattgcccaagttccctggatcagggggga
aagactatacttaagatgccacactcactgggttttaagcacagagtgtcaagggaagta
gaagagatgatccagatcactaggtcttttgcctagtaagacatctcctctccagatcagt

```

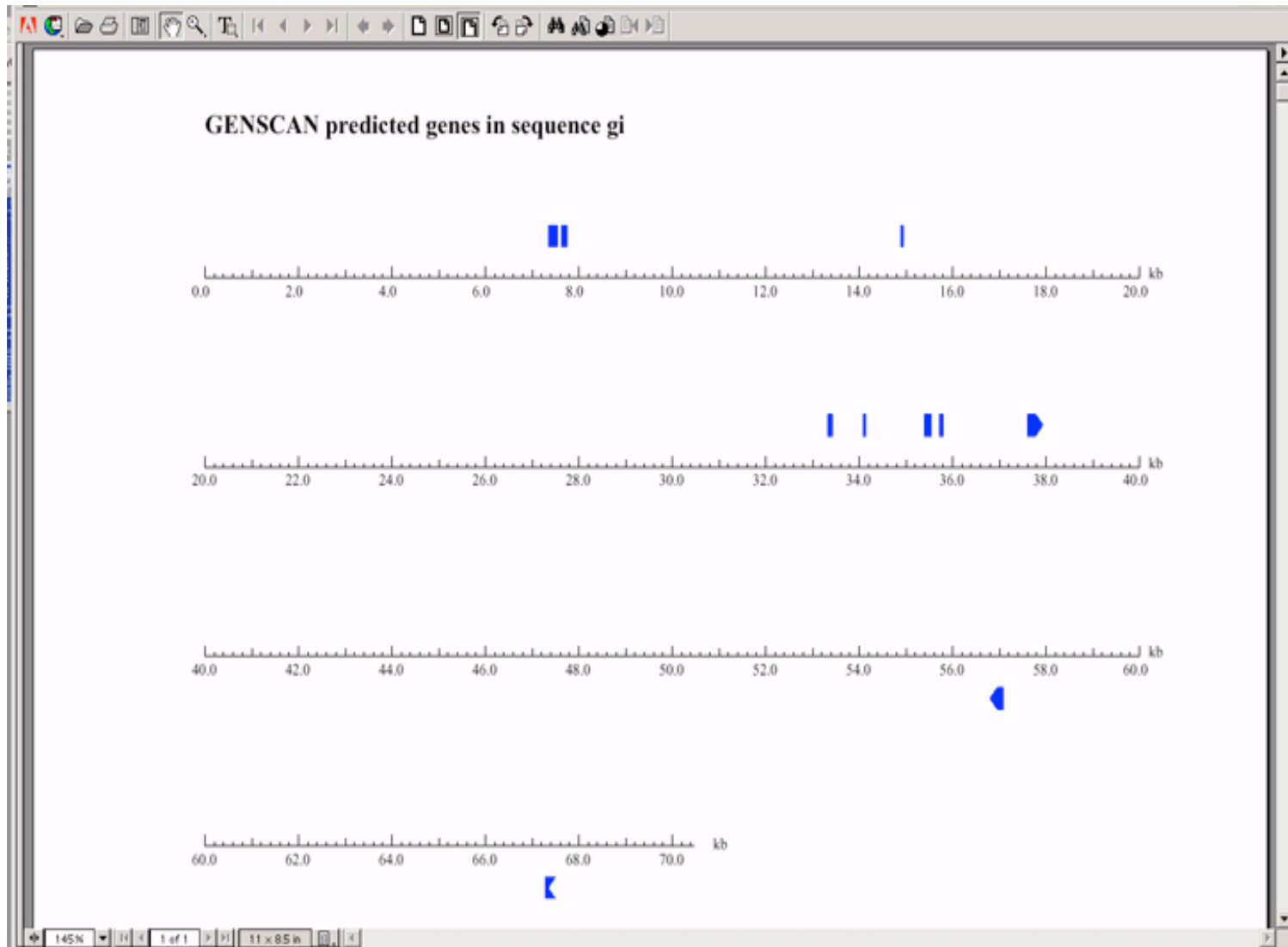
```

Prom = promoter
Init = Initial exon
Intr = Internal exon
Term = Terminal exon
Sngl = Single exon gene
PlyA = poly-A signal

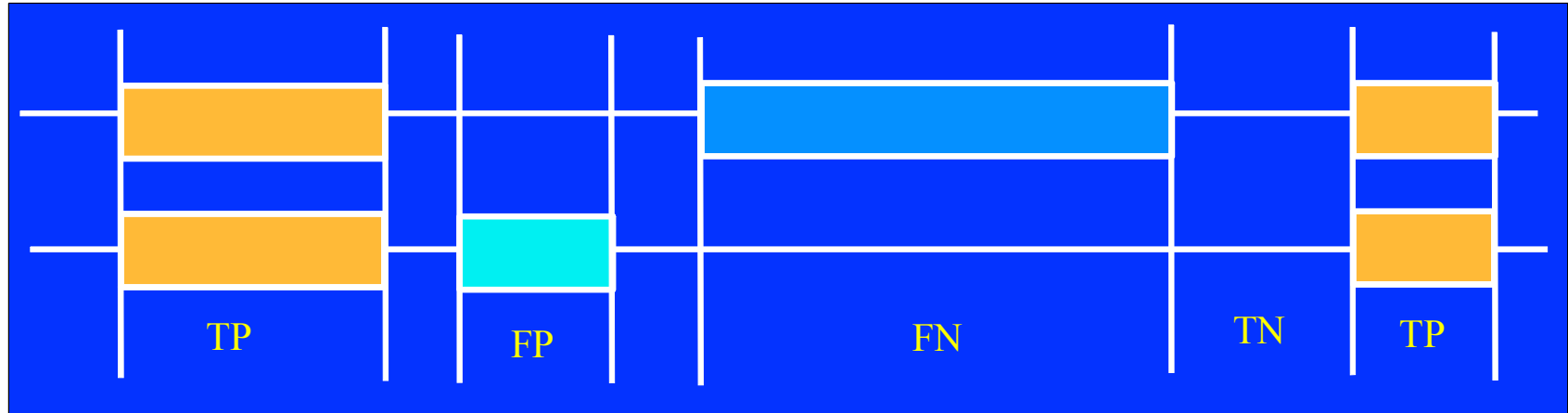
```

P - range	Accuracy
0.00 - 0.50	29.8%
0.50 - 0.75	54.1%
0.75 - 0.90	74.8%
0.95 - 0.99	92.4%
0.99 - 1.00	97.7%

Graphical output



Evaluation statistics



Sensitivity Fraction of actual coding regions that are correctly predicted as coding, ranging from 0 to 1

$$S_n = TP / (TP + FN)$$

Specificity Fraction of the prediction that is actually correct, ranging from 0 to 1

$$S_p = TP / (TP + FP)$$

Correlation Combined measure of sensitivity and specificity, ranging from -1 (always wrong) to +1 (always right)

$$CC = \frac{TP \times TN + FP \times FN}{\sqrt{(PP)(PN)(AP)(AN)}}$$

TP - true positive

FP - false positive

FN - false negative

TN - true negative

Experimental validation of predicted genes

20 not annotated human BAC clones

- 3 finished
- 17 unfinished

Genes that had at least two exons, each predicted by at least two programs

- the overlap of the predicted exons did not have to be perfect
- similarity to ESTs or known genes was used as supporting evidence but was not required
- 40 genes (number of exons 2-11)

Six single exons predicted by three or four programs

Three two-exon genes predicted by one program only but strongly supported by similarities to EST sequences

Total: 49 putative transcripts

Selection of predicted genes - II

37 genes were selected for experimental validation, all of them were potentially novel as they were not annotated nor were their mRNA sequence present in the GenBank at the time of analysis

12 genes were eliminated from further studies as they contained repetitive elements and were most likely false positives

Results published:

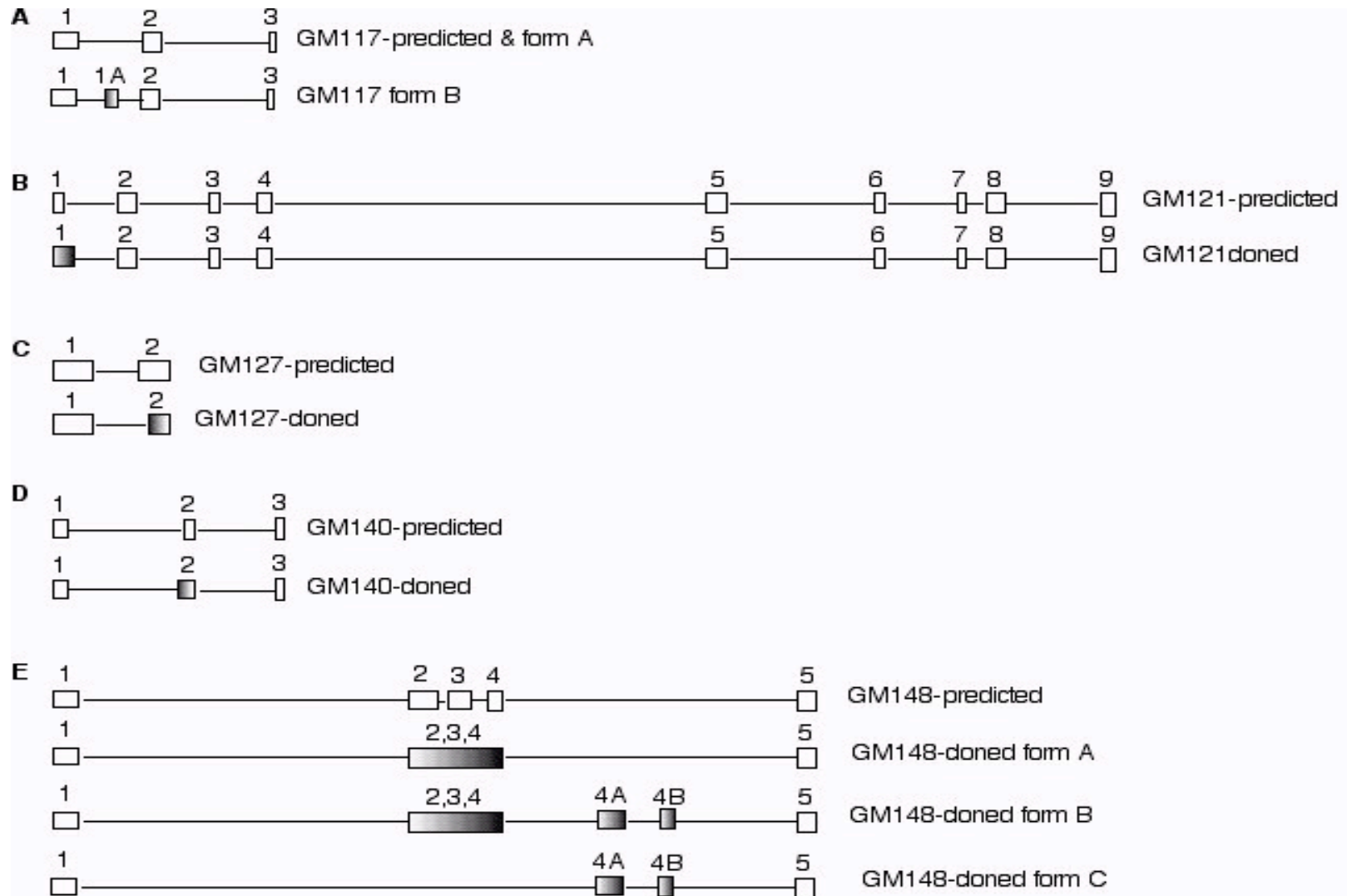
- Makalowska I, Sood R, Faruque MU, Hu P, Robbins CM, Eddings EM, Mestre JD, Baxevanis AD, Carpten JD. Identification of six novel genes by experimental validation of GeneMachine predicted genes. *Gene*. 2002 284(1-2):203-13.

Prediction programs performance

37 genes were tested, 16 of them (43%) were confirmed.
At the exon level 159 exons were predicted and 58 (36%)
were found to be real

	predicted exons	specificity	sensitivity
MZEF	34	0.51	0.56
GRAIL	11	0.48	0.19
GENSCAN	52	0.46	0.91
FGENES	45	0.37	0.75

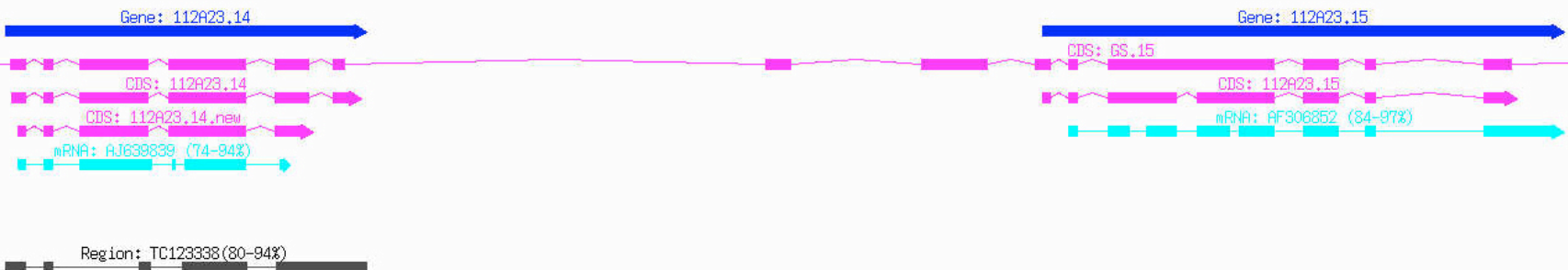
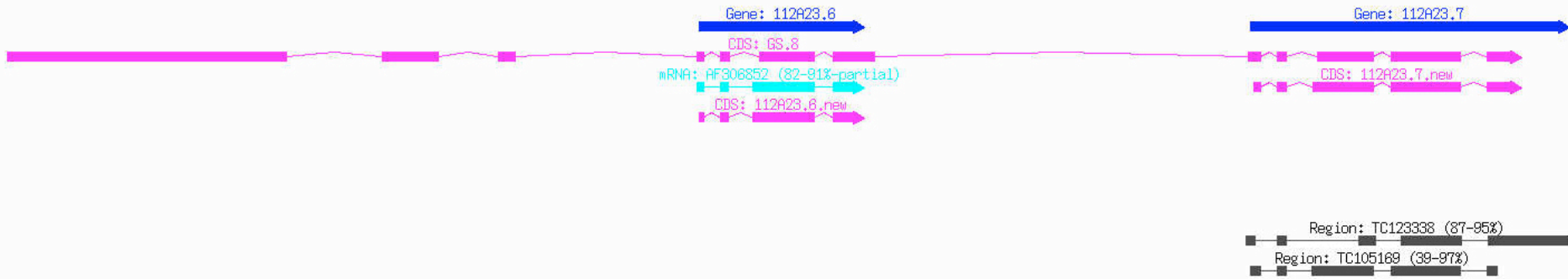
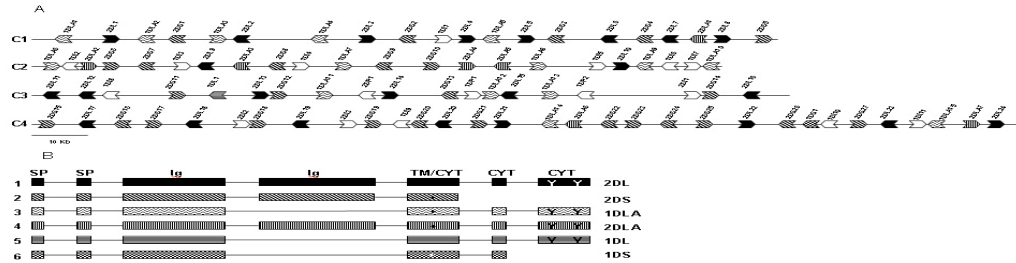
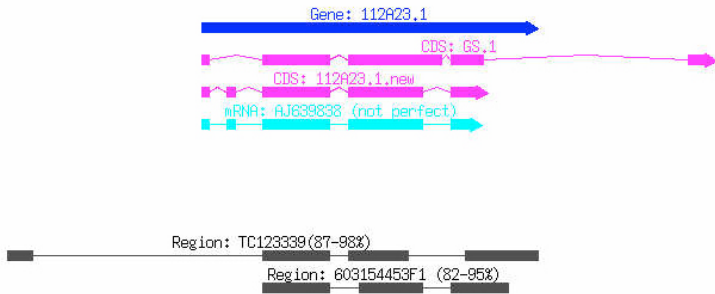
Problems related to gene prediction - gene structure and alternative splicing



Sequence analysis example

- Recently a sequence of a chicken genome was released. We were interested in immunoglobulin like receptors (CHIR) in chicken, their number, structure, and evolution. However, no immunoglobulin genes were annotated on chicken genome.

Prediction quality



Repetitive elements

50% of mammalian genome consists of repeats:

DNA transposons

retrotransposons

LINEs

SINEs

tandem repeats

masking before similarity search - helps avoid getting similarities caused by the presence of repetitive elements, not because of sequences homology

predicted gene with repetitive elements are less likely to be real, although sometimes repeats are true parts of coding sequence

RepeatMasker

Searches for Alu, MIR, LINE, LTR and other repeats by comparison to sequences in RepBase library

RepBase is a database of repetitive DNA sequence elements found in a variety of eukaryotic organisms including primates, rodents, cow, dog, chicken, Fugu, Drosophila, Arabidopsis, rice

Accepts local databases with repetitive

INSTITUTE FOR Systems Biology RepeatMasker Web Server

RepeatMasker screens DNA sequences in fasta format against a library of repetitive elements and returns a masked query sequence as well as a table annotating the masked regions.
Reference: A.F.A. Smit & P. Green, unpublished data. Current Version: open-3.0.8

[Check Current Queue Status](#)

Basic Options

[Large sequences](#) will be queued, and may take a while to process.
Enter the [file](#) to process:

Or paste the sequence(s) in [FASTA format](#):

Select [return format](#): html tar file links
Select return method: html email

Advanced Options

[Speed/Sensitivity](#): rush quick default slow
[DNA source](#):
[Contamination Check](#):
[Repeat Options](#):
[Artifact Check](#):
[Alignment Options](#):
[Masking Options](#):
[Matrix](#):
[Divergence Cutoff](#):
Only mask repeats that are less than percent diverged from a repeat consensus.

Lineage Annotation Options

If the query is mammalian, RepeatMasker can determine if a repeat instance is expected to be present in one or more other mammals can be used to annotate the repeatmasker output or control the masking process.

INSTITUTE FOR Systems Biology Repeat Annotation Request Form

Sequence Selection

Genome/Assembly: *Select the genome and assembly from one of the options in the drop down box.*

Range: *Ranges consist of three identifiers. A valid dna chromosome for the genome specified followed by a start and end position (inclusive). For example human chromosome 1 from position 10-1000 would be chr1:10-1000. Multiple ranges can be entered separated by a ",".*

Result Type:
 annotations
 raw alignments
 masked genomic sequence
 fasta
Select the result type for the range. "annotations" returns RepeatMasker style table of repeat annotations. "raw alignments" returns the alignment file used to create the RepeatMasker annotations. "masked genomic sequence" returns fasta formatted data from the assembly with interspersed repeats masked. "fasta" returns each interspersed repeat instance sequence in fasta format.

Masking Format:
 x
 n
 lower case
Specify the character to use for masking or use lower case to designate repetitive sequences.

Filtering

Score: >= *Filter out all repeats which score below this threshold.*

Divergence: < % *Filter out all repeats with a higher divergence.*

Repeat Classes: *Repeat classes you would like included in your results.*

Repeat Name: *Search for a particular repeat name ie. "AluXc". Do not include the type information in your name ie. "AluXc\$SINE/Alu". The classes filter should be set to "All" if you are using a name filter.*

Institute for Systems Biology
This server is made possible by funding from the National Human Genome Research Institute (NHGRI grant # RO1 HG002939-01) 2003.

<http://www.repeatmasker.org/>

PipMaker <http://pipmaker.bx.psu.edu/pipmaker/>

Computes alignments of similar regions in two or more DNA sequences

Resulting alignments are summarized with a "percent identity plot"

As an output PipMaker generates PDF or PostScript document

MultiPipMaker can be requested to compute true multiple alignment and return a nucleotide level view of the results

PipMaker input requirements

First sequence in FASTA format

RepeatMasker output for first sequence (Do NOT include masked sequence, PipMaker requires file with information about each repeat name and localization)

```
413 5.6 0.0 0.0 Human 1 54 C Alu SINE/Alu (238) 62 9 SINE/Alu (238) 62 9
```

Exons for the first file:

```
>100 800 gene1
```

```
100 200
```

```
500 750
```

Second sequence in FASTA format

PipMaker server

PipMaker ([instructions](#)) aligns two DNA sequences and returns a percent identity plot of that alignment, together with a traditional textual form of the alignment.

- First sequence (FASTA format):

or filename (**file must be plain text only**):

- Second sequence (FASTA format):

or filename (**file must be plain text only**):

- Your email address:

- Optional features:

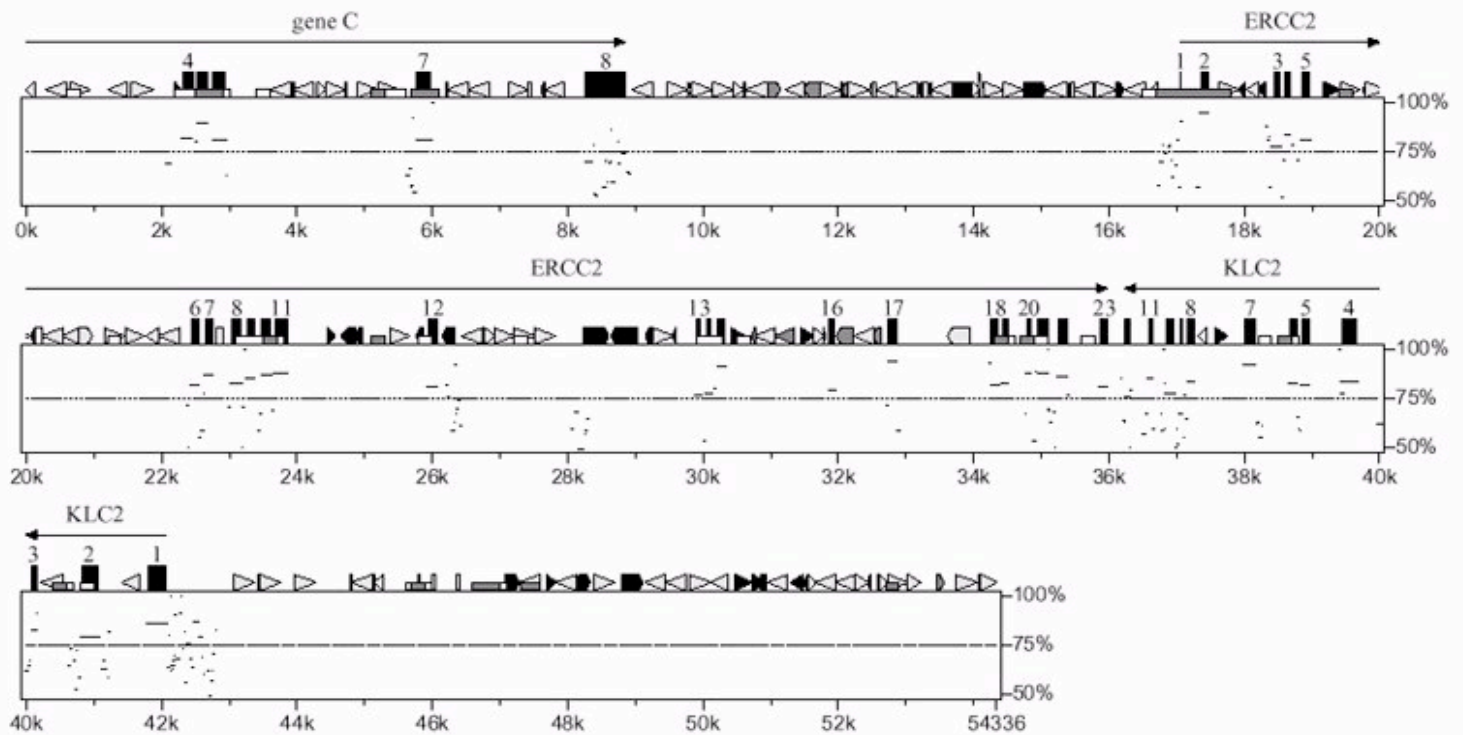
- First sequence mask:

- First sequence exons:

PipMaker output

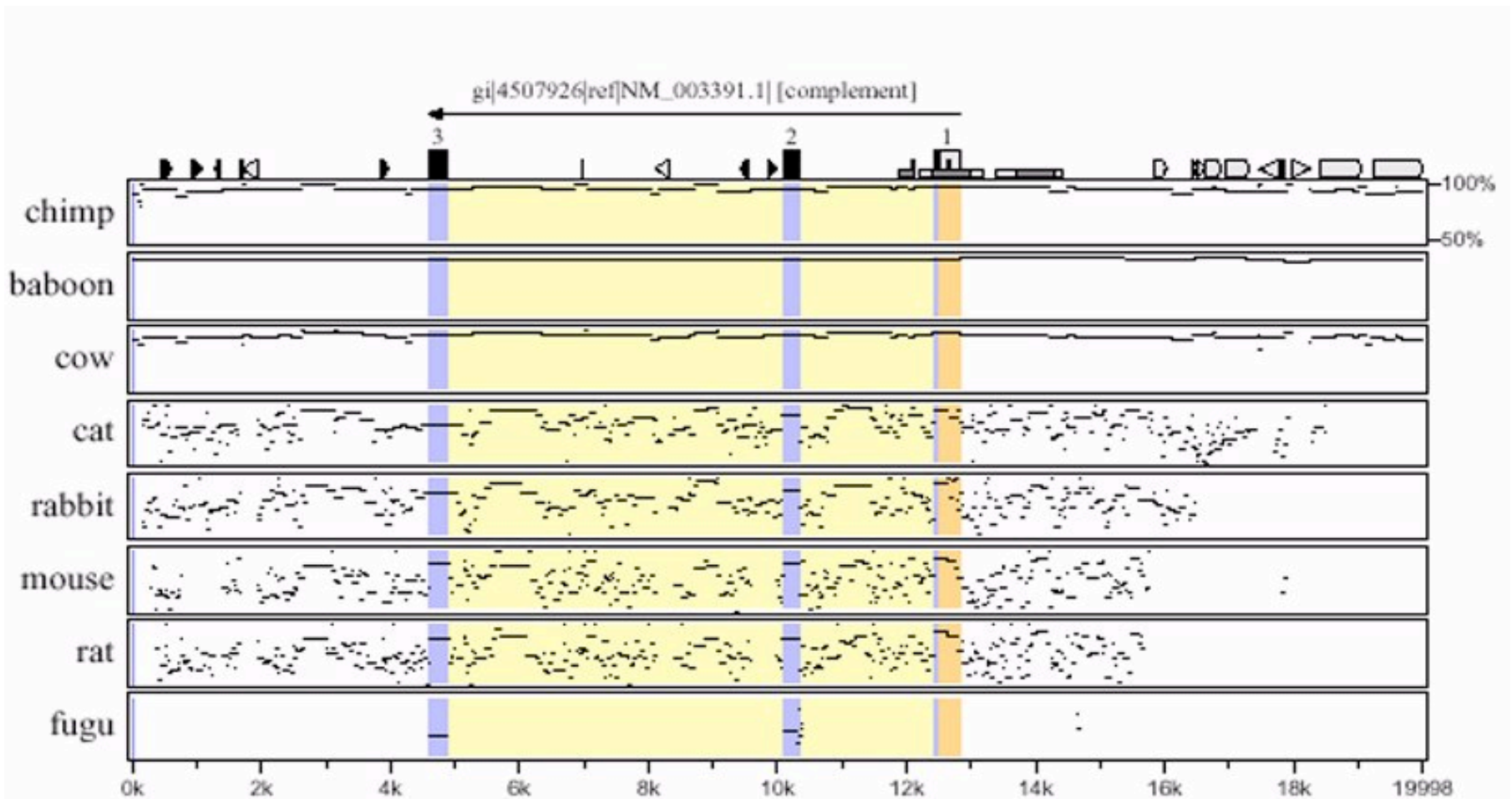
Fri Sep 3 15:28:07 EDT 1999
<http://globin.cse.psu.edu/pipmaker/>

Human ERCC2 region:



- Gene
- Exon
- UTR
- RNA
- Simple
- MIR
- Other SINE
- LINE1
- LINE2
- LTR
- Other repeat
- CpG/GpC \geq 0.60
- CpG/GpC \geq 0.75

MultiPipMaker



Gene finding strategies

Search for conserved regions

Presence of ORF

Codon usage

Splice sites

Polyadenylation signal

Similarity search

Presence of regulatory elements

Why is promoter prediction difficult?

Not a one single type of core promoter

Promoter needs additional regulatory elements

Transcription may be activated or repressed by many regulatory proteins

Transcriptional activators and repressors act very specifically both in terms of the cell type and point in the cell cycle

Not all regulatory factors have been characterized

Prokaryote promoter prediction

Most bacterial promoters contain:

The Pribnow box, at about -10bp from the start codon there is consensus sequence: 5'-TATAAT-3'

The -35 sequence, centered about -35bp from the start codon there is consensus sequence: 5'-TTGACA

NNN**TTGACA**NNNNNNNNNNNNNNNNNNNNNN**TATAAT**NNNNNN**ATG**cccccc

-35 region

-10 region



RNA start site

E.coli promoters

(b) Strong *E. coli* promoters

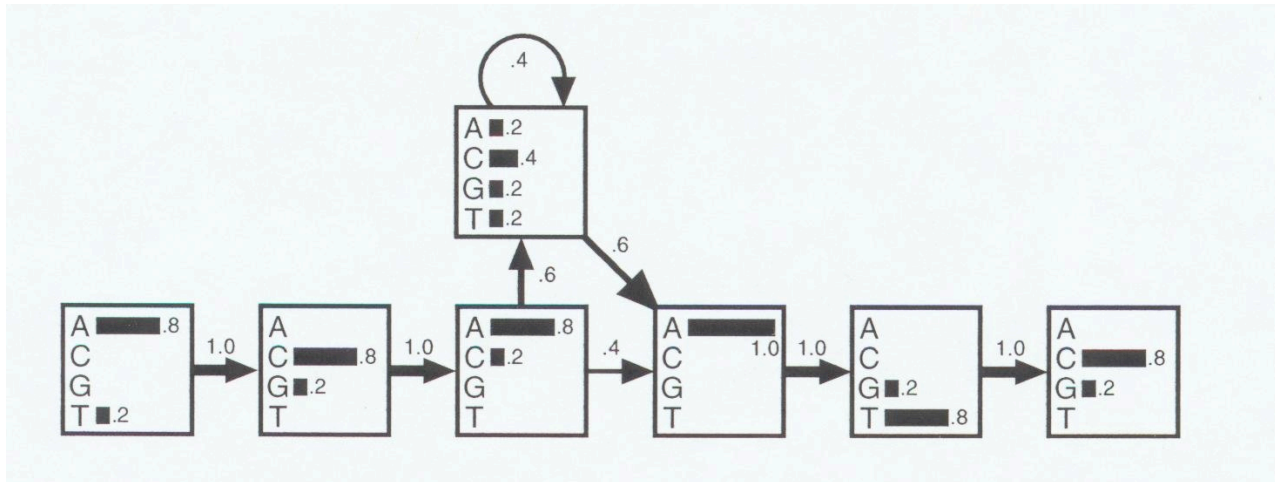
tyr tRNA	TCTCAACGTAACAC	TTTACA	GCGGCG	CGTCATTTGA	TATGATGC	GCCCCG	CTTCCCGATAAGGG
rrn D1	GATCAAAAAAATAC	TTGTGCAAAAA	TTGGGATCCC	TATAATGCGCCTCC	TTGAGACGACAACG		
rrn X1	ATGCATTTTTCCGC	TTGTCTT	CCTGA	GCCGACTCCC	TATAATGCGCCTCC	ATCGACACGGCGGAT	
rrn (DXE) ₂	CCTGAAATTCAGGG	TTGACTCTGAAA	GAGGAAAGCG	TAATATAC	GCCACC	TCGCGACAGTGAGC	
rrn E1	CTGCAATTTTTCTA	TTGCGGCCTGCG	GAGAACTCCC	TATAATGCGCCTCC	ATCGACACGGCGGAT		
rrn A1	TTTTAAATTTCTC	TTGTCA	GGCCGG	AATAACTCCC	TATAATGCGCCACC	CTGACACGGAAACA	
rrn A2	GCAAAAAATAAATGC	TTGACTCTGTAG	CGGGAAGGCG	TATTATGC	ACACC	CGCGCCGCTGAGAA	
λ P _R	TAACACCGTGCGTG	TTGACTATTTTA	CCTCTGGCGGTGATAATGG	TTGC	ATGTA	CTAAGGAGGT	
λ P _L	TATCTCTGGCGGTG	TTGACATAAATA	CCACTGGCGGTGATACTGA	GCAC	ATCAGC	AGGACGCAC	
T7 A3	GTGAAACAAAACGG	TTGACAACATGA	AGTAAACACGGTACGATGT	ACCAC	ATGAAACGACAGTGA		
T7 A1	TATCAAAAAGAGTA	TTGACTTAAAGT	CTAACCTATAGGATACTTA	CAGCC	ATCGAGAGGGACACG		
T7 A2	ACGAAAAACAGGTA	TTGACAACATGAAGTAACATGCAGTAAGATAC	AAATCG	CTAGGTAA	CACTAG		
fd VIII	GATACAAATCTCCG	TTGTACT	TTGT	TCGCGCTTGG	TATAATCG	CTGGG	GTCAAAGATGAGTG
		-35		-10		+1	→

Promoters sequences can vary tremendously.

RNA polymerase in eukaryotes recognizes hundreds of different promoters

Markov modeling - again

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



Eukaryotic promoters

Three types of RNA polymerase (I, II, III), each binding to various kinds of promoters

Polymerase II transcribes genes coding for proteins

Core Promoter - most have TATA box that is centered around position -25 and has the consensus sequence: 5'-TATAAAA-3'

Several promoters have a CAAT box around -90 with the consensus sequence: 5'-GGCCAATCT-3'

promoters for "housekeeping" genes contain multiple copies of a GC-rich element that includes the sequence 5'-GGGCGG-3'

Proximal Promoter Regions - transcription factor binding regions within ~200 bp of the Core Promoter

Enhancers - transcription factor binding regions that can act to regulate transcription from the core promoter even from many kilobases away from the core promoter

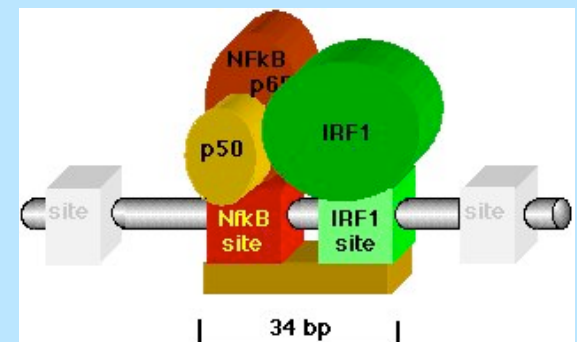
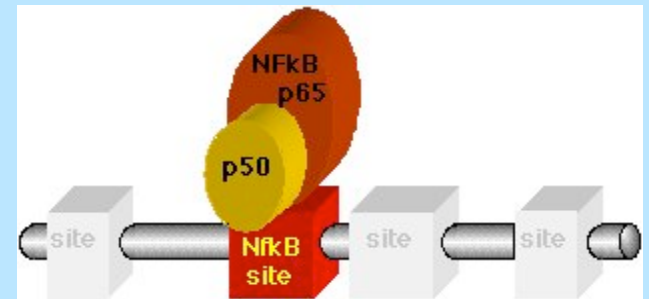
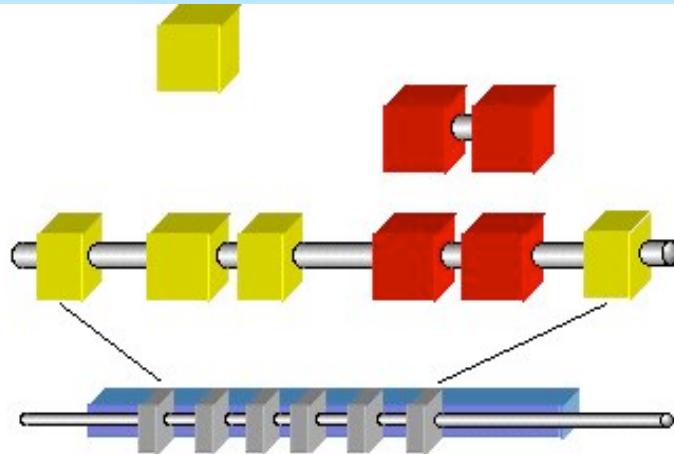
Eukaryotic promoters

Transcription factor binding site

Promoter module

Complete promoter

Promoter Context



Cister

Detects cis-elements clusters by using Hidden Markov Model

For each element uses separate matrix with frequencies of each nucleotide in each position; user can input matrix for elements not included in the basic option

User can specify:

- distance between neighboring cis-elements within a cluster

- number of cis-elements in the cluster

- distance between clusters

- half-width of the sliding window

Example of matrix

```
NA    AML-1a
XX
DE    runt-factor AML-1
XX
BF    T02256; AML1a; Species: human, Homo
      sapiens.
XX
P0      A      C      G      T
01      5      1      2      49      T
02      2      2      52     1      G
03      4     14      1     38      T
04      0      0     57      0      G
05      1      0     55      1      G
06      1      4      0     52      T
```

```
TGTGGT
TGCGGT
TGTGGT
AGTGGT
TGTGGC
```

Sequences of experimentally identified elements are aligned and frequencies in each position are calculated

Cister - <http://zlab.bu.edu/~mfrith/cister.shtml>

Cister - Netscape

File Edit View Go Bookmarks Tools Window Help

http://zlab.bu.edu/~mfrith/cister.shtml

NCBI Sequence Viewer Species Details DeCIFR (ei at CJFR) Cister

Cister : Cis-element Cluster Finder

[Instructions](#)

Paste a DNA sequence into the box or enter a [GenBank identifier](#):

OR upload a DNA sequence from a file:

(Optional) [Set subsequence](#) From: To:

Choose a bunch of cis-elements:

IATA Sp1 CRE ERE NF-1 E2F Mef-2 Myf

CCAAT AP-1 Ets Myc GATA LSF SRF Tef

AND / OR [enter your own cis-elements](#).
(Get cis-element matrices from [TRANSFAC](#) - free registration required)

AND / OR upload cis-elements from a file:

Parameters: (use the defaults if in doubt)

a average distance between motifs within a cluster

b average number of motifs in a cluster

g average distance between clusters

w half-width of sliding window for local base composition

Motif probability threshold

Pseudocount

[Return to Cis-tertia!](#)

Contact: [Martin Frith](#)
Last modified: Tuesday, 18-Mar-2003 11:13:41 EST

public - 05-01-2001 - MATRICES sorted by factor name - Netscape

Go Bookmarks Tools Window Help

http://transfac.gbf.de/TRANSFAC/lists/matrix/matrixByName.html

sequence Viewer Species Details DeCIFR (ei at CJFR) TRANSFAC 5.0 - public - 05-01-...

No.	Factor name	No.	Factor name	No.	Factor name
1	AbaA	1	Abd-B	2	ARF1
1	Adf-1	1	ADR1	1	AG
2	AGL3	1	Ahr	2	Ahr/Arnt
1	AML-1a	5	AP-1	1	AP-2
3	AP-4	1	Arnt	1	ARP-1
1	ATF	1	Athb-1		

No.	Factor name	No.	Factor name	No.	Factor name
1	Barbie Box	1	Bcd	1	BR-C Z1
1	BR-C Z2	1	BR-C Z3	1	BR-C Z4
1	Brachyury	1	Bm-2	2	BSAP
2	bZIP910	2	bZIP911		

No.	Factor name	No.	Factor name	No.	Factor name
2	c-Ets-1p54	2	c-Myb	2	c-Myc/Max
1	c-Rel	3	C/EBP	1	C/EBPalpha
2	C/EBPbeta	1	cap	1	CCAAT
1	CCAAT box	1	CDC5	2	CDP
1	CDP_CR1	1	CDP_CR3	1	CDP_CR3+HD
2	CdxA	1	ces-2	2	CF1 / USP
2	CF2-II	1	CHOP-C/EBPalpha	1	Ciox
1	COMP1	1	COUP-TF / HNF-4	1	CP2
2	CRE-BP1	1	CRE-BP1/c-Jun	4	CREB
1	Croc	1	CRP		

Document: Done (8.422 secs)