

Gene finding and gene structure prediction

Lorenzo Cerutti
Swiss Institute of Bioinformatics

EMBNNet course, March 2003

Outline

- Introduction
- Homology methods
 - Principles
 - Examples of programs using homology methods
- Ab initio methods
 - Signal detection
 - Coding statistics
 - Methods to integrate signal detection and coding statistics
- Ab initio gene predictors on the web
- Evaluating performances of gene predictors
- Gene prediction limitations

Color code: *Keywords*, *Databases*, *Software*

Introduction

- Gene finding is about detecting coding regions and infer gene structure
- Gene finding is difficult
 - DNA sequence signals have low information content (degenerated and highly unspecific)
 - It is difficult to discriminate real signals
 - Sequencing errors
- Prokaryotes
 - High gene density and simple gene structure
 - Short genes have little information
 - Overlapping genes
- Eukaryotes
 - Low gene density and complex gene structure
 - Alternative splicing
 - Pseudo-genes

Gene finding strategies

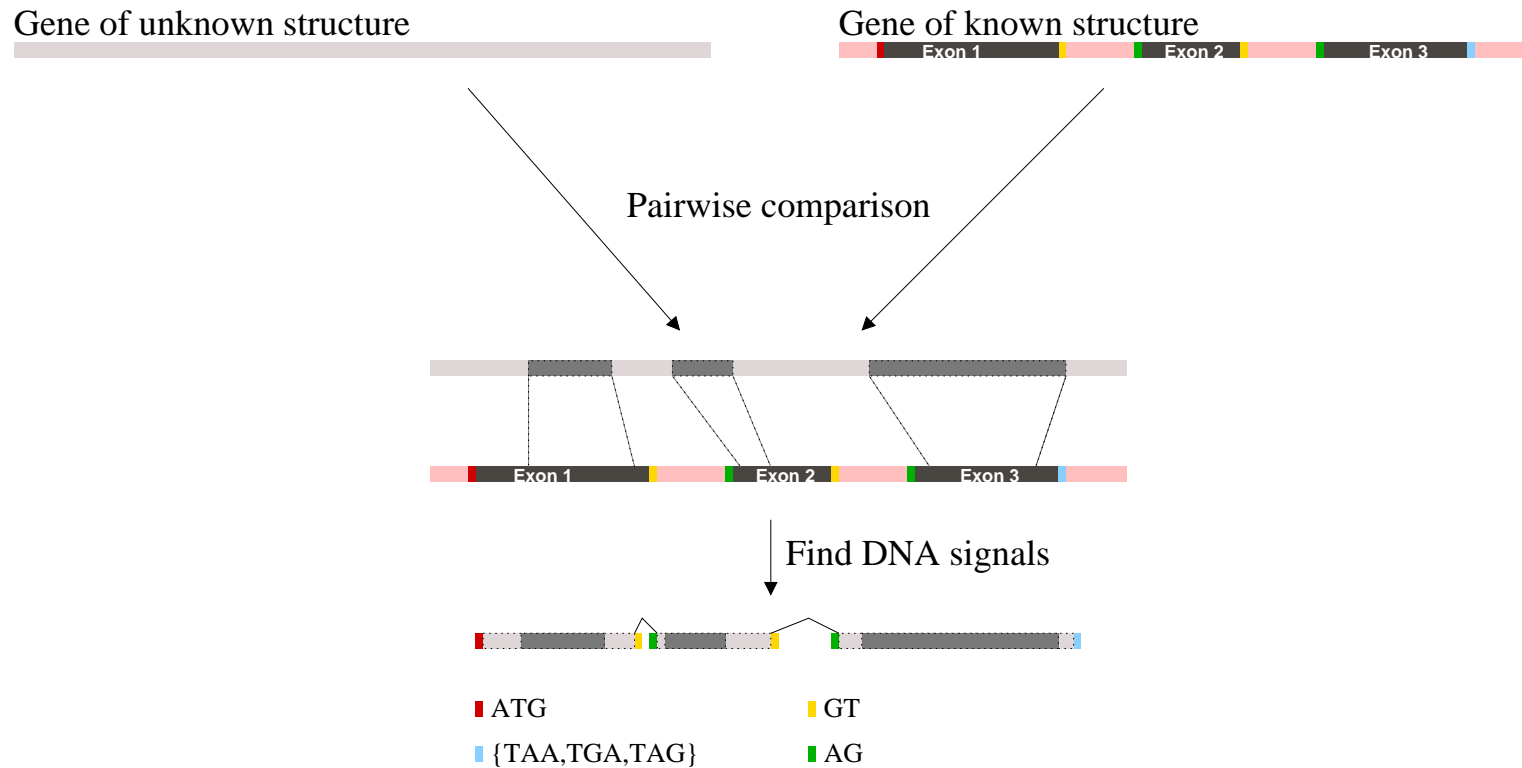
- *Homology method*
 - Gene structure can be deduced by *homology*
 - Requires a not too distant homologous sequence
- *Ab initio method*
 - Requires two types of information
 - *compositional information*
 - *signal information*

Gene finding: Homology method

Homology method

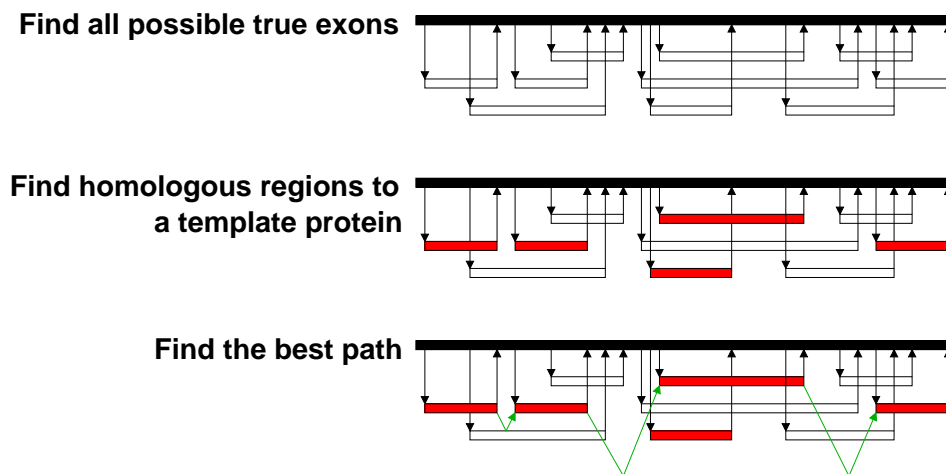
- Principles of the homology method.
 - Coding regions evolve slower than non-coding regions, i.e. local sequence similarity can be used as a gene finder.
 - Homologous sequences reflect a common evolutionary origin and possibly a common gene structure, i.e. gene structure can be solved by homology (mRNAs, ESTs, proteins, protein domains).
 - Standard pairwise comparison methods can be used (*BLAST*, *Smith-Waterman*, ...).
 - Include "gene syntax" information (start/stop codons, ...).
- Homology methods are also useful to confirm predictions inferred by other methods

Homology method: a simple view



Procrustes

- *Procrustes* is a software to predict gene structure from homology found in proteins (*Gelfand et al., 1996*)
 - *Principle of the algorithm*
 - Find all possible blocks (exons) in the query sequence (based on the acceptor/donor sites)
 - Find optimal alignments between blocks and model sequences
 - Find the best alignment between concatenation of the blocks and the target sequence



Procrustes

- Advantages of the homology method
 - Successfully recognizes short exons and exons with unusual codon usage
 - Assembles correctly complex genes (> 10 exons)
 - Available on the web <http://hto-13.usc.edu/software/procrustes/>
- Problems of the homology method
 - Genes without homologous in the databases are missed
 - Requires close homologous to deduce gene structure
 - Very sensitive to frame shift errors
- Protocol to find gene structure using protein homology
 - Do a BLASTX of your query sequence against a protein database (SWISS-PROT/TrEMBL)
 - Retrieve sequences giving the best results
 - Find gene structure using the retrieved sequences from the BLASTX search (Procrustes)
 - BLAST the predicted protein against a protein database to verify the predicted gene structure

Genewise

- *Genewise* uses *HMMs* to compare DNA sequences to protein sequences at the level of its conceptual translation, regardless of sequencing errors and introns.
- Principle
 - The gene model used in genewise is a HMM with 3 base states (match, insert, delete) with the addition of more transition between states to consider frame-shifts.
 - Intron states have been added to the base model.
 - Genewise directly compare HMM-profiles of proteins or domains to the gene structure HMM model.
- Genewise can be used with the whole *Pfam* protein domain databases (find protein domain signatures in the DNA sequence).
- Genewise is a powerful tool, but time consuming.
- Genewise is part of the *Wise2* package: <http://www.ebi.ac.uk/Wise2/>.

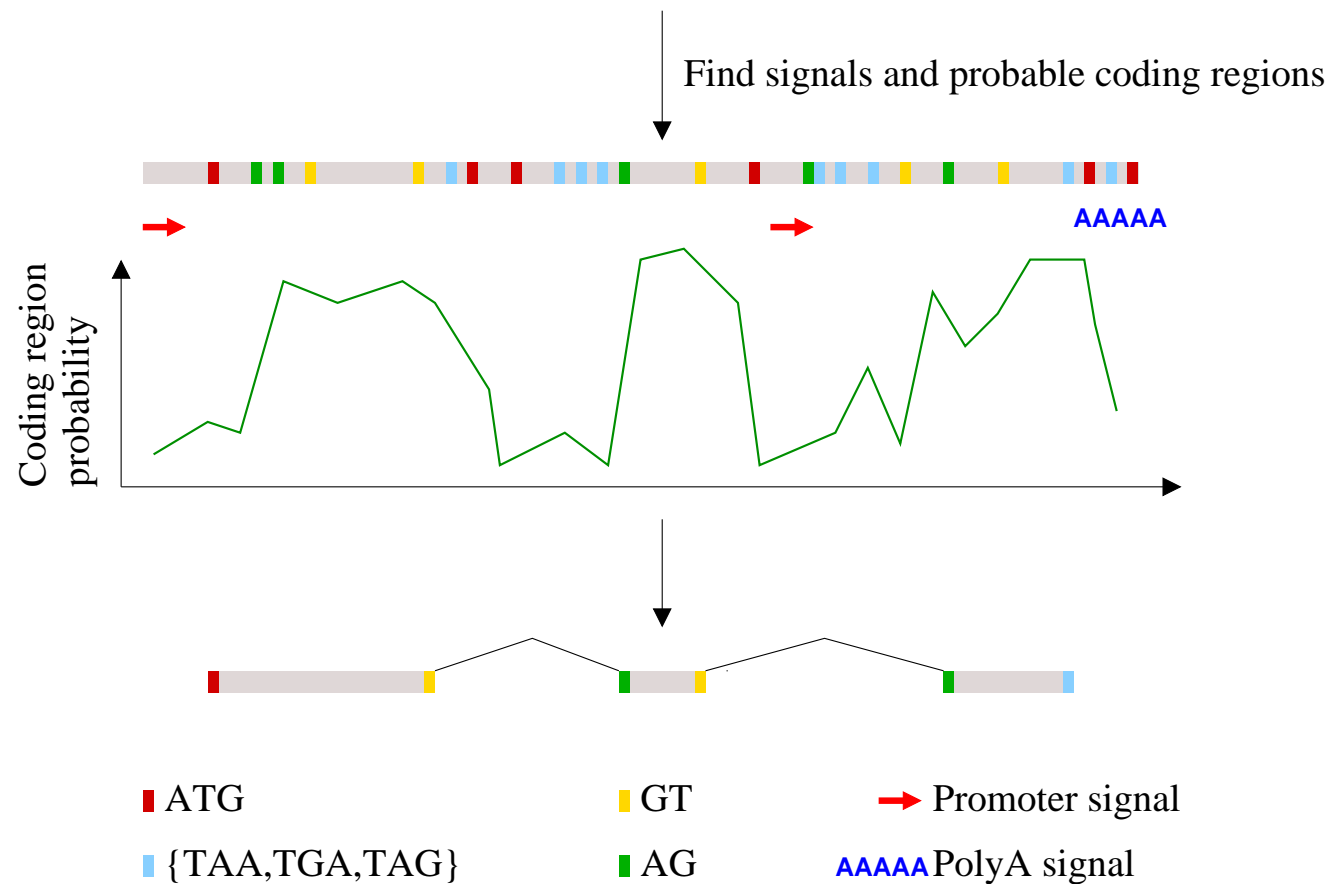
Gene finding: Ab initio method

Ab initio method

- Principles of the ab initio methods
 - Integration of *signal detection* and *coding statistics*
 - *Signal detection* and *coding statistics* are deduced from a *training set*
 - Probabilistic frameworks are used to infer a probable gene structure
 - A solid *scoring system* can be used to evaluate the predictions
 - Algorithms used for the ab initio methods derive from *machine learning* theory.

Ab initio method

Gene of unknown structure



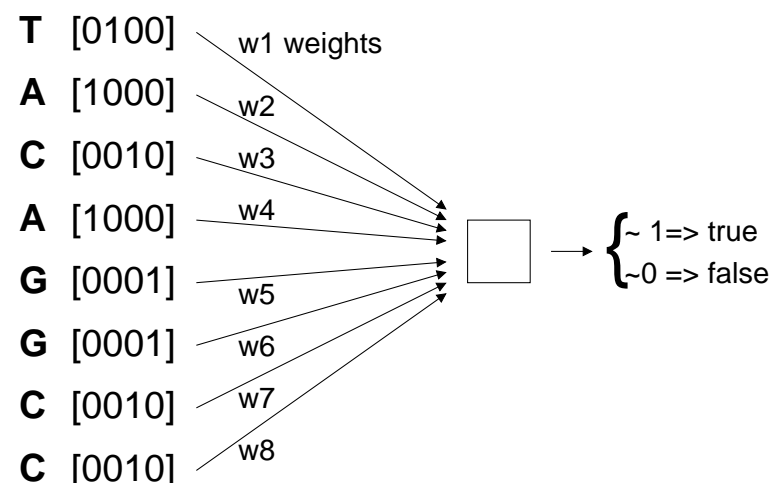
Signal detection

- Detect short DNA motifs (promoters, start/stop codons, splice sites,...).
- A number of methods are used for signal detection:
 - *Consensus string*: based on most frequently observed residues at a given position.
 - *Pattern recognition*: flexible consensus strings.
 - *Weight matrices*: based on observed frequencies of residues at a given position. Uses standard alignment algorithms. This method returns a *score*.
 - *Weight array matrices*: weight matrices based on dinucleotides frequencies. Takes into account the non-independence of adjacent positions in the sites.
 - *Maximal dependence decomposition (MDD)*: MDD generates a model which captures significant dependencies between non-adjacent as well as adjacent positions, starting from an aligned set of signals.

Signal detection

- Methods for signal detection (continuation)
 - *Hidden Markov Models* (HMM)
 - HMM uses a probabilistic framework to infer the probability that a sequence correspond to a real signal
 - *Neural Networks* (NN)
 - NN are trained with positive and negatives example and "discover" the features that distinguish the two sets.

Example: NN for acceptor sites, the *perceptron*, (*Horton and Kanehisa, 1992*):



Signal detection

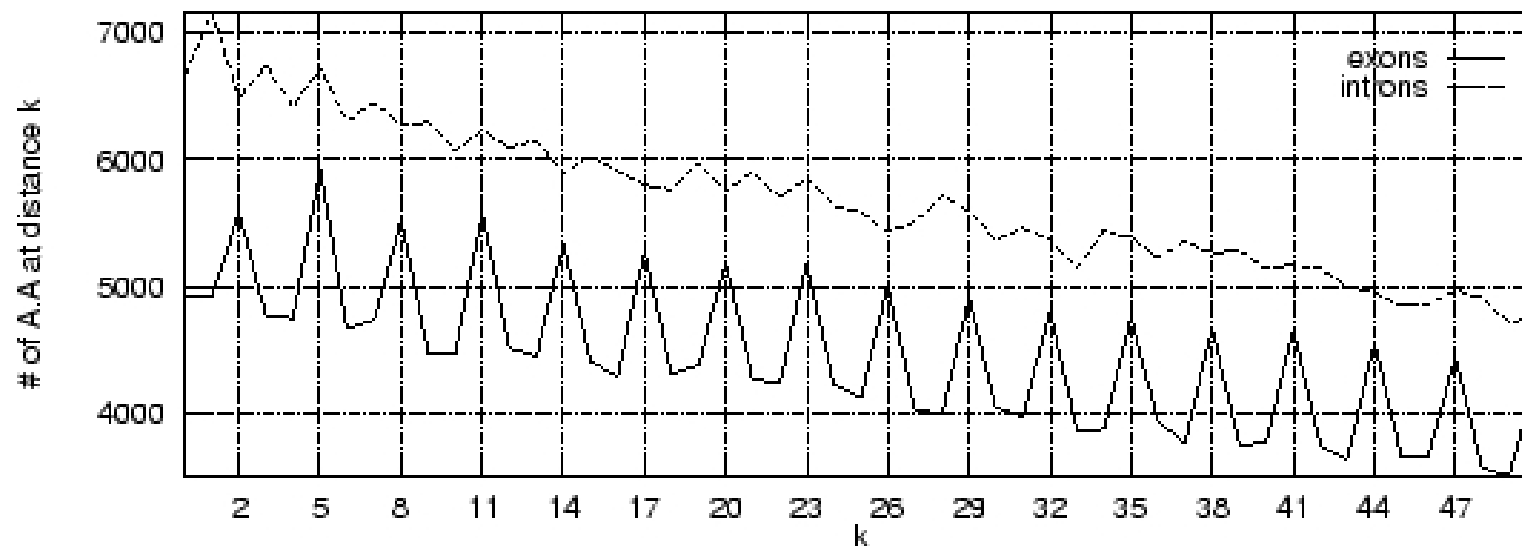
- Signal detection problem
 - DNA sequence signals have low information content
 - Signals are highly unspecific and degenerated
 - Difficult to distinguish between true and false positive
- How improve signal detection
 - Take context into consideration (ex. acceptor site must be flanked by an intron and an exon)
 - Combine with coding statistics (compositional bias)

Coding statistics

- Inter-genic regions, introns, exons, ... have different nucleotides contents
- This compositional differences can be used to infer gene structure
- Examples of coding region finding methods:
 - ORF length
 - Assuming an uniform random distribution, stop codons are present every $64/3$ codons (≈ 21 codons) in average
 - In coding regions stop codon average decrease
 - Method sensitive to frame shift errors
 - Can't detect short coding regions
 - Bias in nucleotide content in coding regions
 - Generally coding regions are G+C rich
 - There are exceptions. For example coding regions of *P. falciparum* are A+T rich

Coding statistics

- Examples of coding region finding methods (continuation):
 - Periodicity
 - Plot of the number of residues separating a pair of nucleotides is periodic in coding regions, but not in non-coding regions.



- From *Guigó*, "Genetic Databases", Academic Press, 1999.

Codon frequencies

- Codon frequencies

- Synonym codon usage is biased in a species dependent way
- 3rd codon position: 90% are A/T; 10% are G/C

- How to calculate codon frequencies

Assume $S = a_1b_1c_1, a_2b_2c_2, \dots, a_{n+1}b_{n+1}c_{n+1}$ is a coding sequence with unknown reading frame. Let f_{abc} denote the appearance frequency of codon abc in a coding sequence.

The probabilities p_1, p_2, p_3 of observing the sequence of n codons in the 1st, 2nd and 3rd frame respectively are:

$$\begin{aligned} p_1 &= f_{a_1b_1c_1} \times f_{a_2b_2c_2} \times \dots \times f_{a_nb_nc_n} \\ p_2 &= f_{b_1c_1a_2} \times f_{b_2c_2a_3} \times \dots \times f_{b_nc_na_{n+1}} \\ p_3 &= f_{c_1a_2b_2} \times f_{c_2a_3b_3} \times \dots \times f_{c_na_{n+1}b_{n+1}} \end{aligned}$$

The probability P_i of the i th reading frame for being the coding region is:

$$P_i = \frac{p_i}{p_1 + p_2 + p_3}$$

where $i \in \{1, 2, 3\}$.

Codon frequencies

- In practice we use these computations in a search algorithm as follows:
 - Select a window of size n (for example $n = 30$)
 - Slide the window along the sequence and calculate P_i for each start position of the window
- A variation of the codon frequency method is to use 6-tuple frequencies instead of 3-tuple (codon) frequencies. This method was found to be the best single property to predict whether a window of vertebrate genomic sequence was coding or non-coding (*Claverie and Bougueleret, 1986*).
- The usage of hexamers frequencies has been integrated in a number of gene predictors.

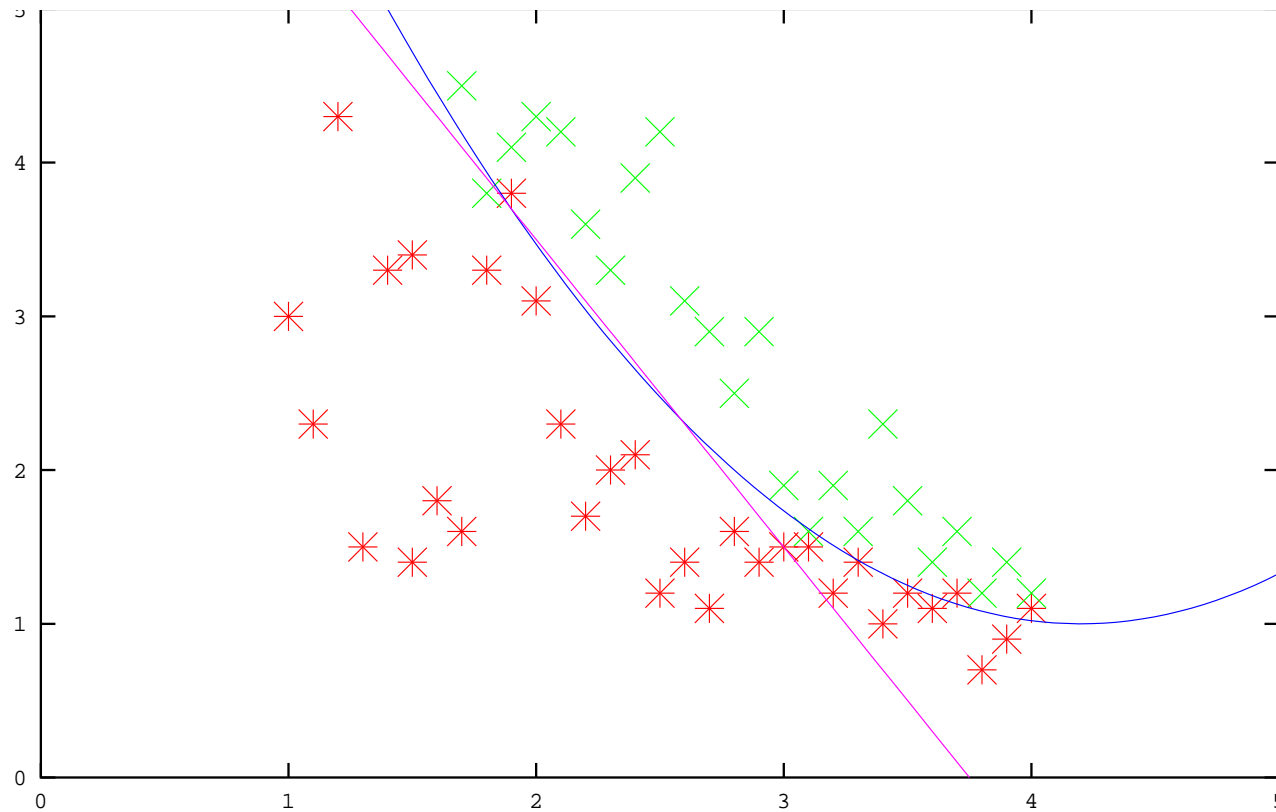
Integrating signal information and compositional information for gene structure prediction

- A number of methods exists for gene structure prediction which integrate different techniques to detect signals (splicing sites, promoters, etc.) and coding statistics.
- All these methods are *classifiers* based on *machine learning* theory.
- *Training sets* are required to train the algorithms.
- The following slides present a non-exhaustive list of these methods.

Linear and quadratic discrimination analysis

- *Linear discrimination analysis* is a standard technique in multivariate analysis.
- Linear discrimination analysis is used to *linearly combine* several measures (e.g. signals and coding statistics) in order to perform the best discrimination between coding and non-coding sequences.
- *Quadratic discriminant analysis*. Similar to linear discrimination analysis, but uses a quadratic discriminant function
- Dynamic programming is used in to combine the inferred exons

Linear and quadratic discrimination analysis

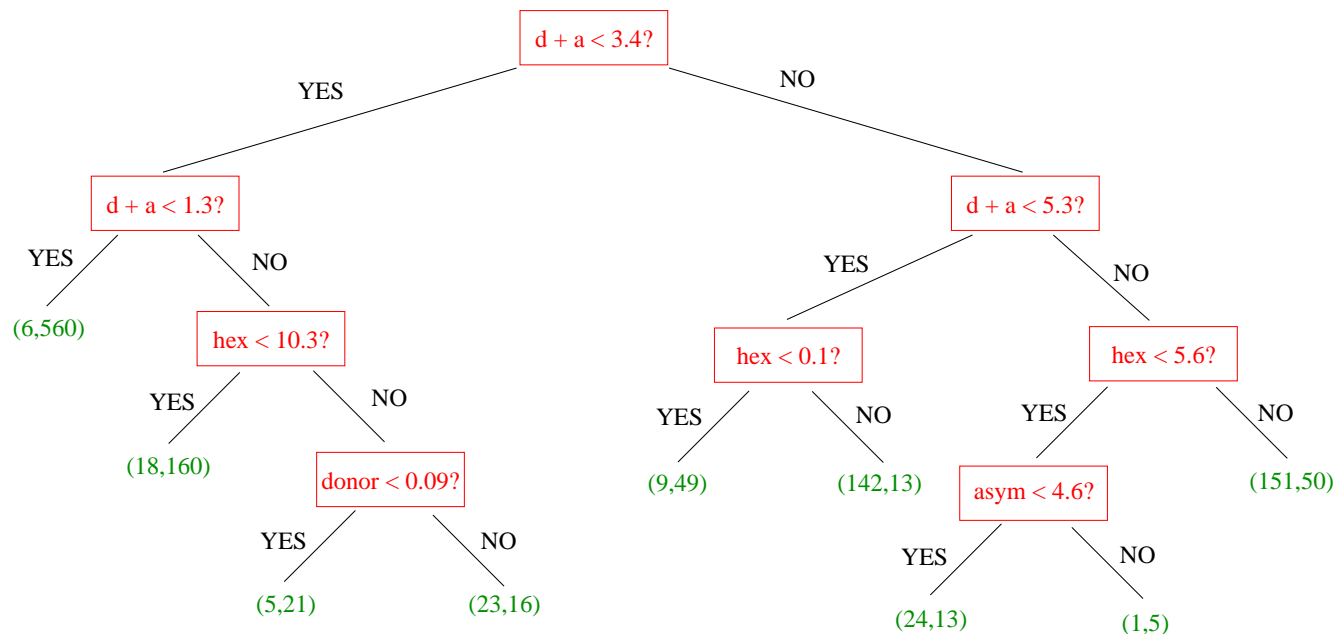


Decision trees

- *Decision trees* can be automatically build using training algorithms.
- Internal nodes of a decision tree are *property values* tested for each subsequence passed to the tree.
- Properties can be various coding statistics (e.g. hexamers frequencies), signal strengths
- Bottom nodes (*leaves*) of the tree contains *class labels* to be associated with the subsequences.
- Dynamic programming can be used to deduce the complete gene structure.

Decision trees

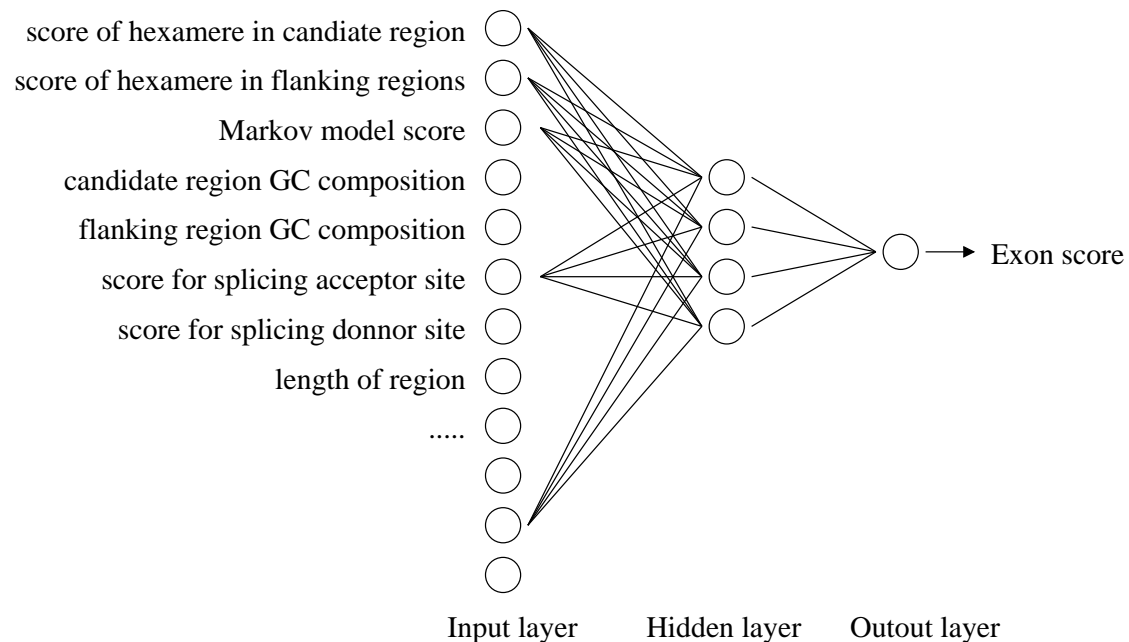
- Example: from *MORGAN* (a decision tree system for finding genes in vertebrate DNA) (*Salzberg et al. 1998*).



d: donor site score
 a: acceptor site score
 hex: in-frame hexamer frequency
 asym: Fickett's position asymmetry statistic
 donor: donor site score
 leaf nodes: exon, pseudo-exon distribution in the training set

Neural network

- A *neural network* is trained with a set of true positives and true negatives examples (set of true exons/false exons, ...).
- For each training example, the neurons are tuned to return the right answer.
- Dynamic programming can be used to deduce the complete gene structure.



(Uberbacher et al., 1996)

Markov derived models

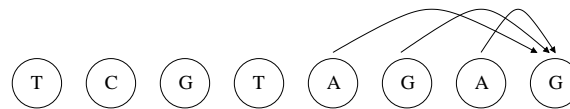
- Principle of Markov Models
 - Given a DNA string S , find the most probable path M in the model that generates S . This will be the most probable gene structure.
- Markov derived models have many desirable properties
 - Modeling: theoretically well-founded models.
 - Efficient: $O(|M| \cdot |S|)$ where $|M|$ is the number of states in the model and $|S|$ is the length of the string.
 - Scoring: theoretically well-founded scoring system.

Markov derived models

● *Markov Model* (MM)

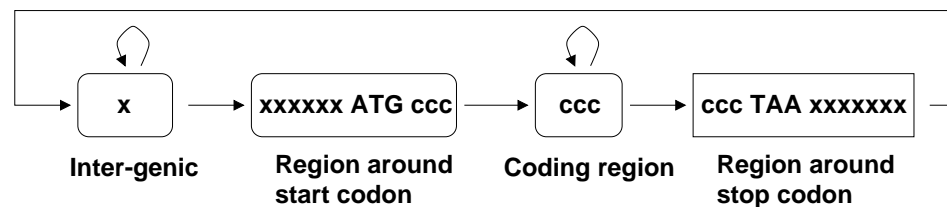
- Biological sequences can be modeled as the output of a stochastic process in which the probability for a given nucleotide to occur at position p depends on the k previous positions. This representation is called k -order Markov Model.

$$P(x_i | x_1, x_2, \dots, x_{i-1}) = P(x_i | x_{i-k}, x_{i-(k-1)}, \dots, x_{i-1})$$



● *Hidden Markov Model* (HMM)

- In a HMM the biological sequences are modeled as the output of a stochastic process that progresses through a series of discrete states. Each state model correspond to a Markov Model.



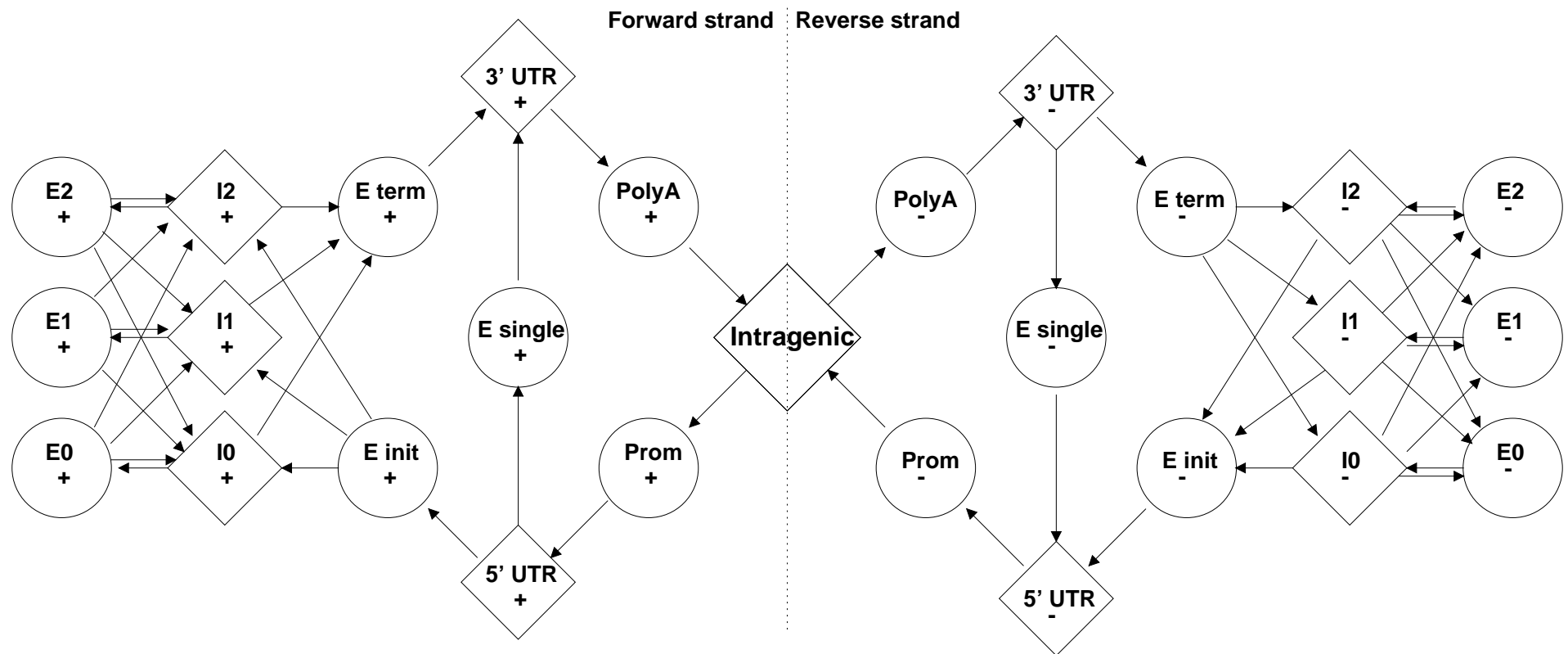
(Krog, 1998)

Markov derived models

- *Generalized Hidden Markov Model* (GHMM)
 - GHMMs are HMMs where states are arbitrary sub-models (e.g., neural networks, position weight matrices, etc.).
 - The *duration* of a particular state depends on some probability distributions.
 - Example of the GHMM in *GENSCAN* (*Burge and Karlin, 1997*)
 - Each state can correspond to a position weight matrix, a neural network, ...
 - Each state has an associated length distribution (time) → there are no transitions from the state to the state itself.

Markov derived models

- The underlying (hidden) model of GENSCAN:



Description of some gene predictors available on the web

GRAIL

- *GRAIL 1*
 - Neural network recognizing coding potential within a fixed-size window (100 bases)
 - Evaluates coding potential without looking for additional features (*e.g.*, splice junctions, start/stop codons)
 - Exon quality evaluated by a score
- *GRAIL 1a*
 - Look at regions immediately adjacent to regions with coding potential
 - Determine the "best" boundaries for the coding region
 - Performs better than GRAIL 1 in finding true exons and eliminating false positives
 - Returns a score

GRAIL

- *GRAIL 2*
 - Uses variable window length
 - Incorporates genomic context information
 - Splice junctions
 - Start and stop codons
 - PolyA signals
 - Requires regions next to an exon
 - Not appropriate for sequences without genomic context
 - Much better at estimating the true extent of an exon as compared to GRAIL 1
 - Returns scores
 - WEB server
 - *<http://compbio.ornl.gov>*
 - *Accept multiple sequence*
 - *Length 100 bases to 100 kilobases*
 - *Available for Human, Mouse, Drosophila, Arabidopsis, and E. coli*

GRAIL

- *GRAIL 2* output

- [grail2exons -> Exons]

	St	Fr	Start	End	ORFstart	ORFend	Score	Quality
1-	f	1	479	666	452	670	52.000	good
2-	f	0	5176	5290	5176	5370	82.000	excellent
3-	f	2	5395	5562	5364	5618	99.000	excellent
4-	f	0	7063	7113	7063	7113	53.000	good
5-	f	0	11827	11899	11590	11925	74.000	good
6-	f	0	12188	12424	12163	12633	88.000	excellent
7-	f	0	14288	14623	14194	14640	94.000	excellent
8-	f	0	17003	17203	16957	17235	100.000	excellent
9-	f	0	17751	17859	17659	17988	50.000	good
10-	f	1	18212	18264	18071	18268	61.000	good

[grail2exons -> Exon Translations]

11- MLRGTDASNSEVFKKAKIMFLEVRKSLTCGQGPTGSSCNGAGQRESGHA
AFGIKHTQSVDR

12- AQIPNQELKETTMCRASLRLLLLLLQLCKFSDLGT

13- AQLLAVTQGKTLVLGKEGESAEPCSSQKKITVFTWKFSQDKILGQHG
KGVLR

FGENES

- *FGENES* uses *linear discriminant analysis*.
 - Combine several measures of pattern recognition using a linear discriminant analysis
 - Donor and acceptor splice sites
 - Putative coding regions
 - 5' and 3' intronic regions of the putative exon
 - Pass the previous results to a dynamic programming algorithm to find a coherent gene model
- Each predicted exon has an associated score, but not directly useful (most of the scores < 10 , while probable good prediction scores > 10)
- WEB server
 - <http://www.softberry.com/berry.phtml>
 - Can combine homology method with ab initio results
 - Available organisms
 - Human, Drosophila, Worm, Yeasts, Plants

FGENES

● FGENES output (CDSf = first exon; CDSi = internal exon; CDSl = last exon; CDSo = only one exon; PolA = PolyA signal)

- Length of sequence: 20000 GC content: 0.48 Zone: 2
 Number of predicted genes: 2 In +chain: 2 In -chain: 0
 Number of predicted exons: 12 In +chain: 12 In -chain: 0
 Predicted genes and exons in var: 2 Max var= 15 GENE WEIGHT: 27.3

G	Str	Feature	Start	End	Weight	ORF-start	ORF-end
1	+	1 CDSf	990	- 1032	1.84	990	- 1031
1	+	2 CDSl	1576	- 1835	0.89	1578	- 1832
1	+	PolA	3106		4.64		
2	+	1 CDSf	5215	- 5266	5.25	5215	- 5265
2	+	2 CDSi	5395	- 5562	3.08	5397	- 5561
2	+	3 CDSi	11464	- 11490	0.76	11466	- 11489
2	+	4 CDSi	11738	- 11899	3.28	11740	- 11898
2	+	5 CDSi	12188	- 12424	2.48	12190	- 12423
2	+	6 CDSi	14288	- 14623	3.26	14290	- 14622
2	+	7 CDSi	17003	- 17203	2.79	17005	- 17202
2	+	8 CDSi	17741	- 17859	1.62	17741	- 17857
2	+	9 CDSi	18197	- 18264	2.53	18196	- 18264
2	+	10 CDSl	18324	- 18630	0.87	18325	- 18627

Predicted proteins:

```
>FGENES-M 1.5 >MySeq          1 Multiexon gene    990 -    1835    100 a Ch+
MSSAFSDPFKEQNPVISLITRTNLNSSSLPVRIYQPPNMFLYIAPCAVLVLSTSSTPRR
TENGPLRMALNSRFPASFYLLCRDYQYTPPQLGPHGRCS
>FGENES-M 1.5 >MySeq          2 Multiexon gene   5215 -   18630    558 a Ch+
MCRAISLRRLLLLLLQLSQQLLAVTQGKTLVLGKEGESAEPCSSQKKITVFTWKFSQQR
```

MZEF

- Designed to predict only internal coding exons
- To discriminate between coding and non-coding regions, *MZEF* uses *quadratic discriminant analysis* of different measures
 - Exon length
 - Intron-exon transition/Exon-intron transition
 - Branch-site scores
 - 5' and 3' splice sites scores
 - Exon score
 - Strand score
- Possible to set a prior probability depending on the gene density and G+C content
- Scores over-estimate of the accuracy of the prediction
- WEB server: <http://www.cshl.org/genefinder>, <http://www.ebi.ac.uk/thanaraj/MZEF-SPC.html>
 - Length up to 200 kilobases
 - Available organisms: Human, Mouse, Arabidopsis, Fission yeast

MZEF

● MZEF output

- Internal coding exons predicted by MZEF
Sequence_length: 19920 G+C_content: 0.475

Coordinates	P	Fr1	Fr2	Fr3	Orf	3ss	Cds	5ss
5315 - 5482	0.580	0.623	0.528	0.585	122	0.506	0.608	0.552
6475 - 6582	0.752	0.482	0.563	0.558	221	0.505	0.567	0.598
11658 - 11819	0.822	0.476	0.569	0.497	211	0.554	0.560	0.651
14208 - 14543	0.903	0.593	0.619	0.469	212	0.497	0.603	0.575

- Description of the symbols
 - P: Posterior probability (between .5 to 1.)
 - Fr_i : Frame preference score for the i th frame of the genomic sequence
 - Orf: ORF indicator,"011" (or "211") means 2nd and 3rd frames are open
 - 3ss: Acceptor score
 - Cds: Coding preference score
 - 5ss: Donor score

HMMgene

- *HMMgene* has been designed to predict complete gene structures.
- Uses HMMs with a criterion called *Conditional Maximum Likelihood* which maximize the probability of correct predictions
- Can return sub-optimal prediction to help identifying alternative splicing
- Regions of the sequence can be locked as coding and non-coding by the user
- Probability score for each predicted exon
- WEB server
 - <http://genome.cbs.dtu.dk/services/HMMgene>
 - Available organisms:
 - Human and worm

HMMgene

- HMMgene output

- # SEQ: Sequence 20000 (-) A:5406 C:4748 G:4754 T:5092

Sequence	HMMgene1.1a	firstex	17618	17828	0.578	-	1	bestparse:cds_1
Sequence	HMMgene1.1a	exon_1	17049	17101	0.560	-	0	bestparse:cds_1
Sequence	HMMgene1.1a	exon_2	14517	14607	0.659	-	1	bestparse:cds_1
Sequence	HMMgene1.1a	exon_3	13918	13973	0.718	-	0	bestparse:cds_1
Sequence	HMMgene1.1a	exon_4	12441	12508	0.751	-	2	bestparse:cds_1
Sequence	HMMgene1.1a	lastex	7045	7222	0.893	-	0	bestparse:cds_1
Sequence	HMMgene1.1a	CDS	7045	17828	0.180	-	.	bestparse:cds_1
Sequence	HMMgene1.1a	DON	19837	19838	0.001	-	1	
Sequence	HMMgene1.1a	START	19732	19734	0.024	-	.	
Sequence	HMMgene1.1a	ACC	19712	19713	0.001	-	0	
Sequence	HMMgene1.1a	DON	19688	19689	0.006	-	1	
Sequence	HMMgene1.1a	DON	19686	19687	0.004	-	0	
...								
			-----	-----	-----			
			position		prob		strand	and frame

- Symbols: firstex = first exon; exon_n = internal exon; lastex = last exon; singleex = single exon gene; CDS = coding region

GENSCAN

- *GENSCAN* has been designed to predict complete gene structures.
- Uses *generalized hidden Markov models* (GHMM): structure of genomic sequence is modeled by explicit state duration HMMs.
- Signals are modeled by weight matrices, weight arrays, and maximal dependence decomposition
- Probability score for each predicted exon.
- WEB server
 - <http://genes.mit.edu/GENSCAN.html>
 - Sequence length up to 200 kilobases
 - Graphical output available
 - Available organisms
 - Vertebrate, Arabidopsis, Maize

GENSCAN

● GENSCAN output

●

Gn.Ex	Type	S	.Begin	...End	.Len	Fr	Ph	I/Ac	Do/T	CodRg	P....	Tscr..
-----	-----	-	-----	-----	-----	--	--	-----	-----	-----	-----	-----
1.00	Prom	+	1653	1692	40							-1.16
1.01	Init	+	5215	5266	52	0	1	83	75	151	0.925	12.64
1.02	Intr	+	5395	5562	168	2	0	89	75	163	0.895	15.02
1.03	Intr	+	11738	11899	162	0	0	74	113	101	0.990	11.15
1.04	Intr	+	12188	12424	237	0	0	71	86	197	0.662	15.39
1.05	Intr	+	14288	14623	336	0	0	82	98	263	0.986	22.19
1.06	Intr	+	17003	17203	201	0	0	116	86	102	0.976	12.06
1.07	Intr	+	17741	17859	119	0	2	78	109	51	0.984	6.38
1.08	Intr	+	18197	18264	68	1	2	103	72	81	0.541	5.70

```
>02:36:44|GENSCAN_predicted_peptide_1|448_aa
MCRAISLRRLLLLLLQLSQLLAVTQGKTLVLGKEGESAEPCSSQKKITVFTWKFSDQR
KILGQHKGVLIRGGSPSQFDRFDSKKGAWEGKSFPLIINKLKMEDSQTYICELENRKEE
```

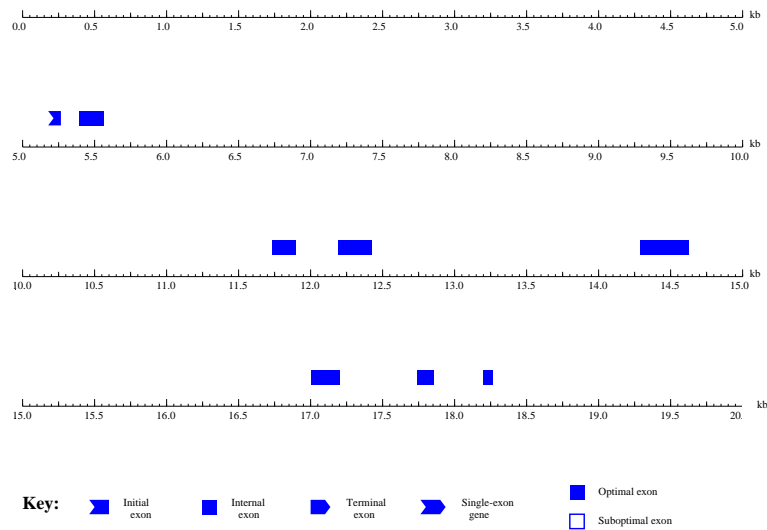
...

```
Gn.Ex : gene number, exon number (for reference)
Type  : Init = Initial exon (ATG to 5' splice site)
       : Intr = Internal exon (3' splice site to 5' splice site)
       : Term = Terminal exon (3' splice site to stop codon)
       : Sngl = Single-exon gene (ATG to stop)
       : Prom = Promoter (TATA box / initiation site)
       : PlyA = poly-A signal (consensus: AATAAA)
S      : DNA strand (+ = input strand; - = opposite strand)
Begin  : beginning of exon or signal (numbered on input strand)
End    : end point of exon or signal (numbered on input strand)
Len    : length of exon or signal (bp)
```

Fr : reading frame (a forward strand codon ending at x has frame $x \bmod 3$)
 Ph : net phase of exon (exon length modulo 3)
 I/Ac : initiation signal or 3' splice site score (tenth bit units)
 Do/T : 5' splice site or termination signal score (tenth bit units)
 CodRg : coding region score (tenth bit units)
 P : probability of exon (sum over all parses containing exon)
 Tscr : exon score (depends on length, I/Ac, Do/T and CodRg scores)

● Graphical output

GENSCAN predicted genes in sequence 02:36:44



GeneMark.hmm

- *GeneMark.hmm* was initially developed for bacterial gene finding, then modified to predict gene structure in eukaryotes.
- Uses explicit state duration HMMs.
- Optimal gene candidates, selected by HMM and dynamic programming, are further processed by a ribosomal binding site recognition algorithm.
- No scores!
- WEB server
 - <http://opal.biology.gatech.edu/GeneMark>
 - Access to both the original GeneMark and GeneMark.hmm

Geneid

- *Geneid* is one of the oldest gene structure prediction programs, recently updated to a new and faster version.
- Uses a *hierarchical search structure* (signal \rightarrow exon \rightarrow gene):
 - 1st: finds signals (splice sites, start and stop codons);
 - 2nd: from the found signals start to score regions for exon-defining signals and protein-coding potential;
 - 3rd: a dynamic programming algorithm is used to search the space of predicted exons to assemble the gene structure.
- Very fast and scale linearly with the length of the sequence (both in time and memory) \Rightarrow adapted to analyze large sequences.
- Trained with Drosophila and Human.
- Available at <http://www1.imim.es/geneid.html> ... also source code!

Geneid

● Geneid output

```

● ## date Mon Aug 26 19:14:40 2002
  ## source-version: geneid v 1.1 -- geneid@imim.es
  ## Sequence emb|U47924|HSU47924 - Length = 21000 bps
  # Optimal Gene Structure. 2 genes. Score = 60.238383
  # Gene 1 (Forward). 4 exons. 266 aa. Score = 19.588764
  Internal      419      67211.08+ 1 1 5.18 3.5027.17 0.00AA   1: 86 gene_1
  Internal      1325     1455 2.54+ 2 0 2.87 0.6013.64 0.00AA  86:129 gene_1
  Internal      3900     4036-0.01+ 0 2-0.38 2.5110.53 0.00AA 130:175 gene_1
  Terminal      6740     7013 5.99+ 1 0 0.87 0.0026.16 0.00AA 175:266 gene_1
  >emb|U47924|HSU47924|geneid_v1.1_predicted_protein_1|266_AA
  gDLTDIYLLRSYIHLRYVDISENHLTDLSPLNYLTHLLWLKADGNRLRSAQMNELPYLQI
  ASFAYNQITDTEGISHPRLETNLKGNLSIHMTGLDPEKLISLHTVELRGNQLESTLGIN
  LPKLKNLYLAQNMLKKVEGLEDSLNTTLHLRDNQIDTLSGFSREMKSLQYLNLRGNMVA
  NLGELAKLRDLPKLRALVLLDNPCTDETSYRQEALVQMPYLERLDKEFYEEEEERAEADVI
  RQRLKEEKEQEPEPQRDLEPEQSLI*
  # Gene 2 (Forward). 7 exons. 395 aa. Score = 40.649619
    First      9843      9927 1.75+ 0 1 1.80 2.6910.15 0.00AA   1: 29 gene_2
  Internal     10427     10522 3.49+ 2 1 6.38 4.53 4.86 0.00AA  29: 61 gene_2
  Internal     11591     11724 2.44+ 2 0 1.72 3.0011.50 0.00AA  61:105 gene_2
  Internal     11950     12172 14.65+ 0 1 4.72 4.5335.26 0.00AA 106:180 gene_2
  Internal     13576     13866 8.03+ 2 1 3.71 3.0522.43 0.00AA 180:277 gene_2
  Internal     15590     15791 8.94+ 2 2 3.73 1.6926.71 0.00AA 277:344 gene_2
  Internal     16256     16411 1.35+ 1 2 5.90 3.92 1.13 0.00AA 344:395 gene_2

```

Evaluating performances

	TP	FP	TN	FN	TP	FN	TN
Real							
Predicted							

- Measures (*Burset and Guigo, 1996; Snyder and Stormo, 1997*):

- Sensitivity** S_n is the proportion of coding nucleotides that are correctly predicted as coding:

$$S_n = \frac{TP}{TP+FN}$$

- Specificity** S_p is the proportion of nucleotides predicted as coding that are actually coding:

$$S_p = \frac{TP}{TP+FP}$$

Evaluating performances

- Measures (contd.):
 - *Correlation coefficient* CC is a single measure that captures both specificity and sensitivity:

$$CC = \frac{(TP \star TN) - (FN \star FP)}{\sqrt{(TP + FN) \star (TN + FP) \star (TP + FP) \star (TN + FN)}}$$

- *Approximate correlation* AC is similar to CC , but defined under any circumstances:

$$AC = (ACP - 0.5) \star 2$$

where

$$ACP = \frac{1}{4} \left(\frac{TP}{TP + FN} + \frac{TP}{TP + FP} + \frac{TN}{TN + FP} + \frac{TN}{TN + FN} \right)$$

Accuracy of the different methods

- Evaluation of the different programs (*Rogic et al., 2001*)

Programs	No. of sequences	S_n	S_p	AC	CC
FGENES	195	0.86	0.88	0.84 ± 0.19	0.83
GeneMark.hmm	195	0.87	0.89	0.84 ± 0.18	0.83
GENSCAN	195	0.95	0.90	0.91 ± 0.12	0.91
HMMgene	195	0.93	0.93	0.91 ± 0.13	0.91
MZEF	119	0.70	0.73	0.68 ± 0.21	0.66

- Overall performances are the best for HMMgene and GENSCAN.
- Some program's accuracy depends on the G+C content, except for HMMgene and GENSCAN, which use different parameters sets for different G+C contents.
- For almost all the tested programs, "medium" exons (70-200 nucleotides long), are most accurately predicted. Accuracy decrease for shorter and longer exons, except for HMMgene.
- Internal exons are much more likely to be correctly predicted (weakness of the start/stop codon detection).
- Initial and terminal exons are most likely to be missed completely.
- Only HMMgene and GENSCAN have reliable scores for exon prediction.

Accuracy of the different methods

- Recently a new benchmark has been published by Makarov (2002), with similar results, but other predictors have been included.

Programs	S_n^a	S_p^a	S_n^b	S_p^b
HMMgene	97	91	93	93
GenScan		95	90	93
Geneid	86	83		
Genie	96	92	91	90
FGENES	89	77	86	88

- ^a Adh region of Drosophila.
- ^b 195 high-quality mammalian sequences (human, mouse, and rat).

Gene prediction limits

- Existing predictors are for protein coding regions
 - Non-coding areas are not detected (5' and 3' UTR)
 - Non-coding RNA genes are missed
- Predictions are for "typical" genes
 - Partial genes are often missed
 - Training sets may be biased
 - Atypical genes use other grammars

The end