Development of Complex Curricula for Molecular Bionics and Infobionics Programs within a consortial* framework**

Consortium leader

PETER PAZMANY CATHOLIC UNIVERSITY

Consortium members

SEMMELWEIS UNIVERSITY, DIALOG CAMPUS PUBLISHER

The Project has been realised with the support of the European Union and has been co-financed by the European Social Fund ***

**Molekuláris bionika és Infobionika Szakok tananyagának komplex fejlesztése konzorciumi keretben

***A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

INTRODUCTION TO BIOINFORMATICS

(BEVEZETÉS A BIOINFORMATIKÁBA)

CHAPTER 9

Gene Prediction Algorithms

(Gén predikció)

András Budinszky

Definition of Gene

As it has been discussed in Chapter 1, the gene is a segment of the DNA molecule that encodes the information required for the synthesis of a gene product (protein or RNA).

The region that encodes amino acid sequence of a protein sometimes referred to as „protein-gene”.

In addition, there are regulatory sequences in a genome as well, that guide and control the gene expression (promoters, enhancers, operators, terminators, etc.), therefore we could also consider these sequences integral parts of a gene.

Gene Prediction Problem

Its primary task is to find stretches of a DNA sequence that is responsible for protein coding.

It may also include looking for other functional elements such as regulatory regions.

Gene finding is one of the first and most important steps in understanding the genome of a species after it has been sequenced.

While earlier days gene prediction was done by experimentation, today – thanks to powerful methods – it is considered a computational problem.

Components of “Protein-Gene”

Codon:

- triplet of nucleotides that codes exactly one amino acid in a protein
- $4^3 = 64$ possible codons
- 20 possible amino acid as final product
- redundant coding
- some amino acids are coded by more than one codon (max. six, e.g. leucine)
- includes one start (ATG) and three stop codons (TAA, TAG, TGA)

Genetic Code Table

acid	*codons*	*acid*	*codons*	*acid*	*codons*	*acid*	*codons*
A	GCA GCC GCG GCT	G	GGA GGC GGG GGT	M	ATG	S	AGC AGT TCA TCC TCG TCT
C	TGC TGT	H	CAC CAT	N	AAC AAT	T	ACA ACC ACG ACT
D	GAC GAT	I	ATA ATC ATT	P	CCA CCC CCG CCT	V	GTA GTC GTG GTT
E	GAA GAG TTC TTT	K	AAA AAG	Q	CAA CAG	W	TGG
F	TTC TTT	L	CTA CTC CTG CTT TTA	R	AGA AGG CGA CGC CGG CGT	Y	TAC TAT

Each codon encodes one kind of amino acid.

Each amino acid is encoded by one or more codons.

The 3rd position of codons is the most likely to vary, for a given amino acid.

Components of “Protein-Gene” (continued)

Exons:

- during splicing they are joined into a continuous piece of mRNA
- it is often misused to refer only to coding sequences for the final protein. This is incorrect, since non-coding exons also exist.
- they are relevant only for eukaryotic genes

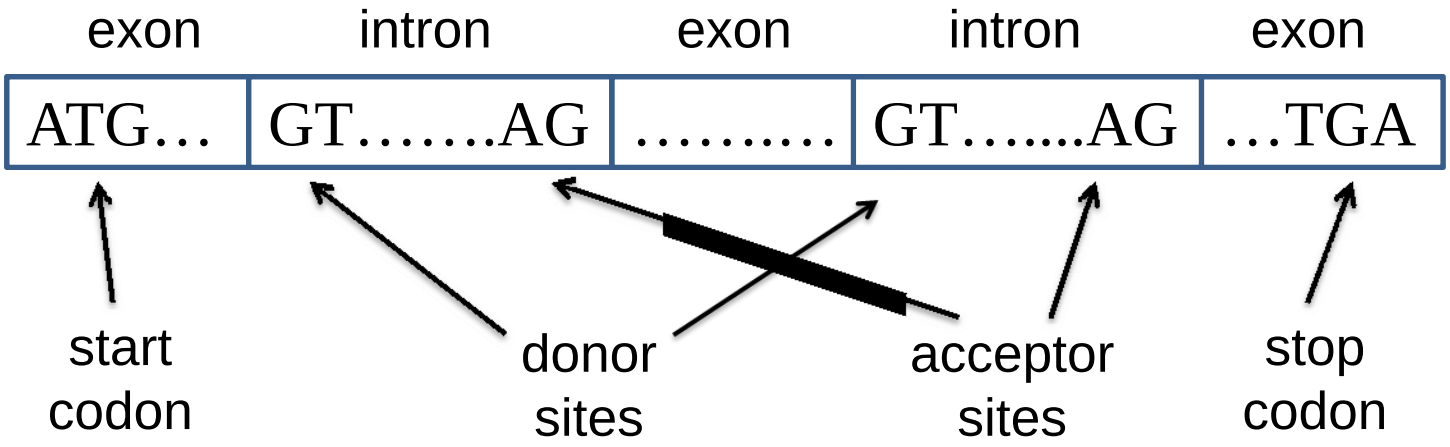
Introns:

- intervening, non-coding segments in eukaryotic genes
- could be much longer than exons
- during splicing they are cut out

Signals:

- donor sites (usually GT, beginning of introns)
- acceptor sites (AG, end of introns)

Gene Structure (in eukaryotes)



Most of the regions outside the genes (intergenic regions) are ignored by gene finders, though some of them look for important subsequences of regulatory functions (like promoters, enhancers).

Note: Prokaryotes do not have introns, only one contiguous exon.

Gene Prediction for Prokaryotes

Since prokaryotes have contiguous exons, the most common approach is using an intrinsic (ab initio) method.

That is, looking for ORFs as exon in all reading frames and then decide if the found punitive exons are really exons.

An ORF (Open Reading Frame) is a subsegment that starts from the Start codon (ATG) and ends at one of the Stop codons (TAA, TAG, TGA).

There are six possible frames: 3 upstream direction (starting at positions 0, 1, or 2) and 3 downstream direction.

ORFs in different frames may overlap.

Example for Reading Frames

Original sequence:

5' 3'

atgcccaagctgaaatgcgtagaggggttttcatcatttgaggacgatgtataa

Three reading frames in one direction:

1 atg ccc aag ctg aaa tgc gta gag ggg ttt tca tca ttt gag gac gat gta taa
M P K L K C V E G F S S F E D D V *

2 tgc cca agc tga aat gcg tag agg ggt ttt cat cat ttg agg acg atg tat
C P S * N A * R G F H H L R T M Y

3 gcc caa gct gaa atg cgt aga ggg gtt ttc atc att tga gga cga tgt ata
A Q A E # R R G V F I I * G R C I

where A, C, D, …, Y amino acids codes; # start and * stop codons

Verifying an ORF as Protein-Coding Exon

Various methods can be used to collect evidence:

- Checking the length of the ORF which should (usually) be above a threshold value.

  (Typically a protein-coding exon is at least 100 base pair long and since 3 of the 64 possible codons in the genetic code are stop codons, one would expect to see a stop codon approximately in every 20-25 codons, or 60-75 base pairs in a random sequence.)

  This may not always true because some genes (e.g. some neural and immune system genes) are relatively short.

Verifying an ORF as Protein-Coding Exon (cont)

- Applying statistical verification.

  Check nucleotide composition and especially (G+C) content (introns being more A/T-rich than exons, especially in plants).

  Check codon composition. (Amino acids are typically coded by more than one codon, but the occurrence of "synonyms" are (very) different in frequency –see Codon Usage for Human Genes on next slide. Furthermore expected codon occurrence is different in coding and non-coding regions.)

  Hexamer frequency can also be typical and reliable.

Codon Usage in Human Genes

	U		C		A		G	
U	UUU Phe	57	UCU Ser	16	UAU Tyr	58	UGU Cys	45
	UUC Phe	43	UCC Ser	15	UAC Tyr	42	UGC Cys	55
	UUA Leu	13	UCA Ser	13	UAA Stp	62	UGA Stp	30
	UUG Leu	13	UCG Ser	15	UAG Stp	8	UGG Trp	100
C	CUU Leu	11	CCU Pro	17	CAU His	57	CGU Arg	37
	CUC Leu	10	CCC Pro	17	CAC His	43	CGC Arg	38
	CUA Leu	4	CCA Pro	20	CAA Gln	45	CGA Arg	7
	CUG Leu	49	CCG Pro	51	CAG Gln	66	CGG Arg	10
A	AUU Ile	50	ACU Thr	18	AAU Asn	46	AGU Ser	15
	AUC Ile	41	ACC Thr	42	AAC Asn	54	AGC Ser	26
	AUA Ile	9	ACA Thr	15	AAA Lys	75	AGA Arg	5
	AUG Met	100	ACG Thr	26	AAG Lys	25	AGG Arg	3
G	GUU Val	27	GCU Ala	17	GAU Asp	63	GGU Gly	34
	GUC Val	21	GCC Ala	27	GAC Asp	37	GGC Gly	39
	GUA Val	16	GCA Ala	22	GAA Glu	68	GGA Gly	12
	GUG Val	36	GCG Ala	34	GAG Glu	32	GGG Gly	15

Codon Usage in Mouse Genes (fragment)

AA	codon	/1000	Prob.
Ser	TCG	4.31	0.05
Ser	TCA	11.44	0.14
Ser	TCT	15.70	0.19
Ser	TCC	17.92	0.22
Ser	AGT	12.25	0.15
Ser	AGC	19.54	0.24
Pro	CCG	6.33	0.11
Pro	CCA	17.10	0.28
Pro	CCT	18.31	0.30
Pro	CCC	18.42	0.31
. . . . .			

AA	codon	/1000	Prob.
Leu	CTG	39.95	0.40
Leu	CTA	7.89	0.08
Leu	CTT	12.97	0.13
Leu	CTC	20.04	0.20
Ala	GCG	6.72	0.10
Ala	GCA	15.80	0.23
Ala	GCT	20.12	0.29
Ala	GCC	26.51	0.38
Gln	CAG	34.18	0.75
Gln	CAA	11.51	0.25
.....			

Verifying an ORF as Protein-Coding Exon (cont)

- Looking for a typical ribosome-binding site (searching for a Shine-Dalgarno sequence in front of the putative protein coding sequence).
- Looking for a typical promoter (if consensus promoter sequences for the given organism are known, check for the presence of a similar in the upstream region).
- Checking if the ORF in question encodes a protein that is similar to previously described ones (searching the protein database for homologs of the given sequence).

Note: This last one is an integrated (intrinsic and extrinsic) method.

Verifying an ORF as Protein-Coding Exon (cont)

- Using Hidden Markov Model (HMM, Principles see next slide).

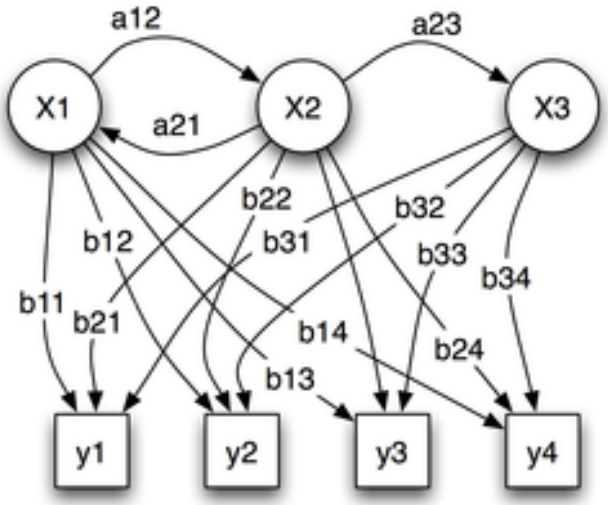
  HMM must be trained on known sequences of the same species to setup statistical rules to evaluate unknowns.

  So take known genes encoding known proteins to use as input into the program.

  These model genes will then provide statistics on codon bias, codon pairs, etc.

  Depends on accuracy of model genes (poor data can obscure important rules).

Principles of HMM



Probabilistic parameters of a HMM(example)

- x — states
- y — possible observations
- a — state transition probabilities
- b — output probabilities

Ab initio Gene Prediction for Eukaryotes

This kind of methods of gene finding for eukaryotes, especially in complex organisms like humans, is considerably more challenging for several reasons:

- The promoter and other regulatory signals are more complex and less well-understood, making them more difficult to reliably recognize. (Two classic examples of signals identified by eukaryotic gene finders are CpG islands and binding sites for a poly(A) tail.

- Eukaryotes have exons and introns, and typically exons are much shorter than introns. The length of exons (usually less than 200 and can be as short as twenty to thirty) makes it much more difficult to detect periodicities and other known content properties of protein-coding DNA.

Ab initio Gene Prediction for Eukaryotes (cont)

Finding potential exon/intron boundaries (so-called splice sites) can rely on looking for donor sites (GT) and acceptor sites (AT).

The exons can be checked with the statistical methods listed for prokaryotic exon verification.

Another type of refinement is often needed that consists of estimating several gene models according to the G+C content of the genomic sequence.

Many currently existing programs use two types of content sensors: one for coding sequences and one for non-coding sequences, i.e. introns, UTRs and intergenic regions.

Ab initio Gene Prediction for Eukaryotes (cont)

There are some non-canonical splice signals (other than GT and AT), but they are rare; unfortunately current programs cannot handle them.

The quality of eukaryotic gene prediction achieved by different programs show a rather gloomy picture of numerous errors in exon/intron recognition. Even the best tools correctly predict only ~40% of the genes.

The most serious errors come from genes with long introns, which may be predicted as intragenic sequences.

Sequencing errors in the analyzed sequence affects ORF prediction, and frameshift corrections was found to substantially improve the overall quality of gene prediction.

Extrinsic Gene Prediction for Eukaryotes

Closely-related organisms may have similar genes, therefore genome of one species may be compared to genes in some closely-related species, and a sufficient similarity between genomic sequence regions and a protein or DNA sequence present in a database can be exploited.

Basic tools for detecting sufficient similarities between sequences are local alignment methods (like the optimal Smith–Waterman algorithm, or the fast heuristic approaches such as FASTA and BLAST.

Then the best (optimal) chaining of the found local alignment subsequences (putative exons) should be produced (see later slide)

Sources for Similarity Search

1. Protein sequences that can be found in databases such as SwissProt or PIRAs. In this case – before similarity search - the program should convert each protein sequence to a family of possible coding DNA sequences by reverse translation.
2. Transcripts, sequenced as cDNAs either in the classical way for targeted individual genes with high coverage sequencing of the complete clone or as expressed sequence tags (ESTs), which are one shot sequences from a whole cDNA library or applying RNA-seq, a use of high-throughput technologies.
3. Genomic DNA (under the assumption that coding sequences are more conserved than non-coding ones). Nevertheless, in this case the similarity may not cover entire coding exons, but be limited to the most conserved part of them.

Pros/Cons for Similarity-Based Approach

Advantages:

- Predictions rely on accumulated pre-existing biological data, thus they should produce biologically relevant predictions.
- A single match is enough to detect the presence of a gene.

Disadvantages:

- Obvious weakness is that nothing will be found if the database does not contain a sufficiently similar sequence.
- Small exons are easily missed.
- Some databases may contain information of poor quality.

•Name the 4 core data types of bioinformatics! (4)

Exon Chaining Problems

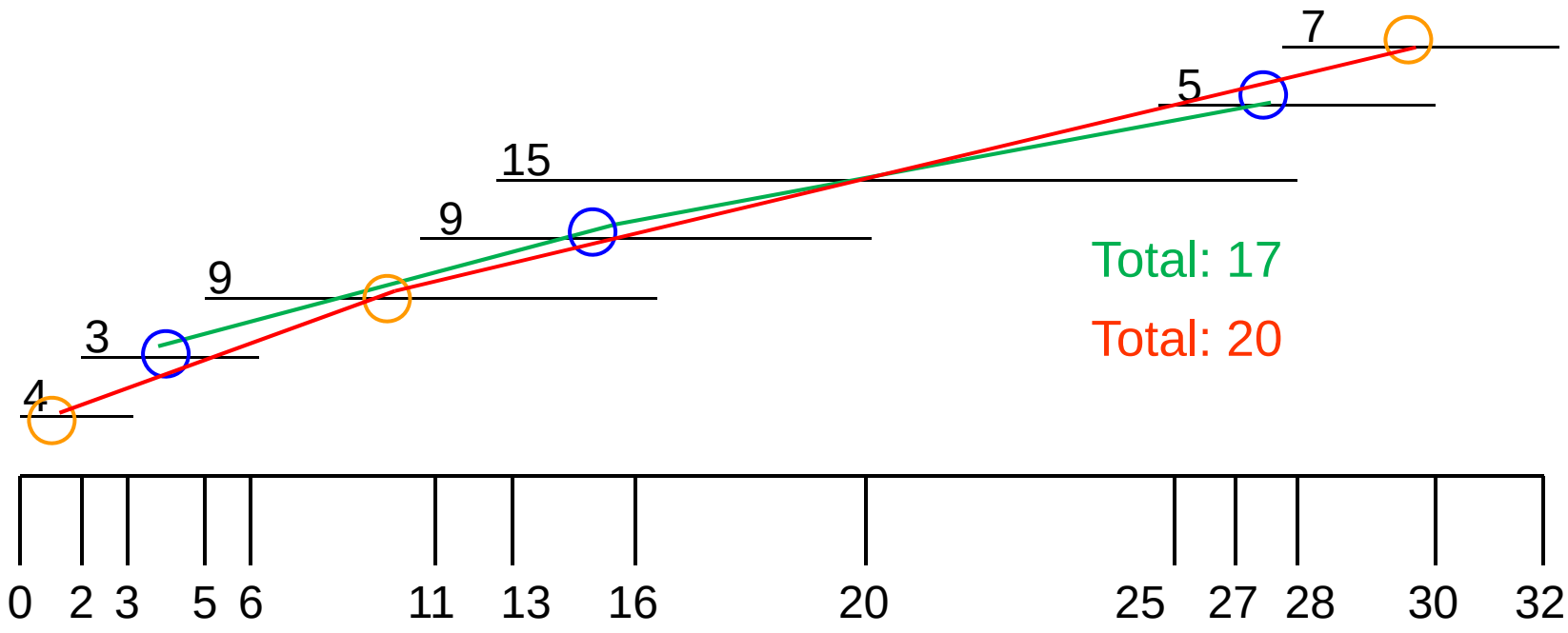
The local alignment returns a set of putative exons ("islands" in the sequence searched for a gene).

Each putative exon can be represented with a triplet (the two end positions and the similarity score returned by the alignment algorithm.

We need to find the maximum non-overlapping set of these exons (that is, the chain of exons that has the maximum weight).

This can be solved with a dynamic programming method.

An Example for Exon Chaining



DP Solution of the Exon Chaining

Given: N putative exon as (l, r, w)

where l and r are begin and end positions

w is score returned by the local alignment

Goal: Chain of non-overlapping no adjacent exons with the maximum weight.

Work data structures:

Weighted directed graph G with

vertices: one for each begin and end position

edges: one for each exon (from l to r) with weight w

one from each vertex to the one representing the next higher position with weight 0

A vector with 2*N elements (one for each vertex/position)

DP Solution of Exon Chaining (continued)

Initialize all elements of the vector to 0

For each vector elements v_i (from 2nd to last)

If an edge (**e**) ends in the vertex that v_i represents

then

j = index of the element that belongs to left end of edge **e**

w = weight of edge **e**

$v_i = \max \{v_j + w, v_{i-1}\}$

else

$v_i = v_{i-1}$

At the end, we have the total weight in v_{2N} and with backtracking we can get the optimal exon chain.

Gene Finding Programs

GeneMark

It is documented as the most accurate prokaryotic gene finder. It use the Hidden Markov models and exists in separate variants for gene prediction in prokaryotic, eukaryotic, and viral DNA sequences

http://opal.biology.gatech.edu/genemark/

Glimmer

It is a system for finding genes in microbial DNA, especially the genomes of bacteria, archaea, and viruses. It uses interpolated Markov models (IMMs) to identify the coding regions.

http://cbcb.umd.edu/software/glimmer/

Gene Finding Programs

GenScan

It uses a complex probabilistic model of the gene structure. Its high speed and accuracy make it the method of choice for the initial analysis of large stretches of eukaryotic genomic DNA.

http://genes.mit.edu/GENSCANinfo.html

GenBuilder

It performs *ab initio* gene prediction using numerous parameters, such as GC content, hexon frequencies, splicing site data, CpG islands, repetitive elements, and others.

http://www.itb.cnr.it/sun/webgene/