Genome bioinformatics 3

GENOME ANNOTATION, GENE FINDING

Finding and annotating genes

November 15, 2016

Outline

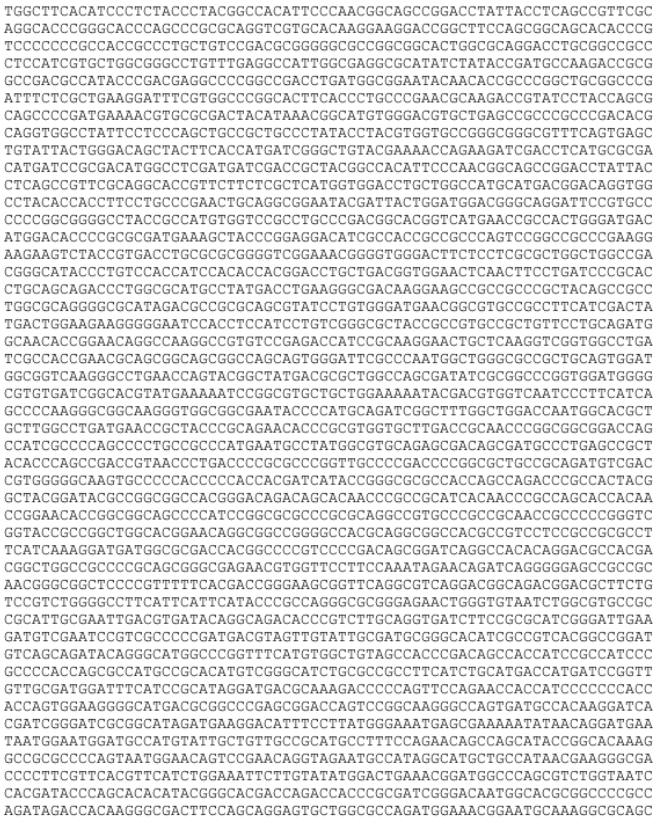
- Rapid reminder on genome assembly
- What is genome annotation?
- Gene finding
  - *Ab initio* methods
  - Homology methods
- Functional annotation
- Automatic and semi-automatic pipelines
  - Microbial genomes
  - Eukaryotic genomes

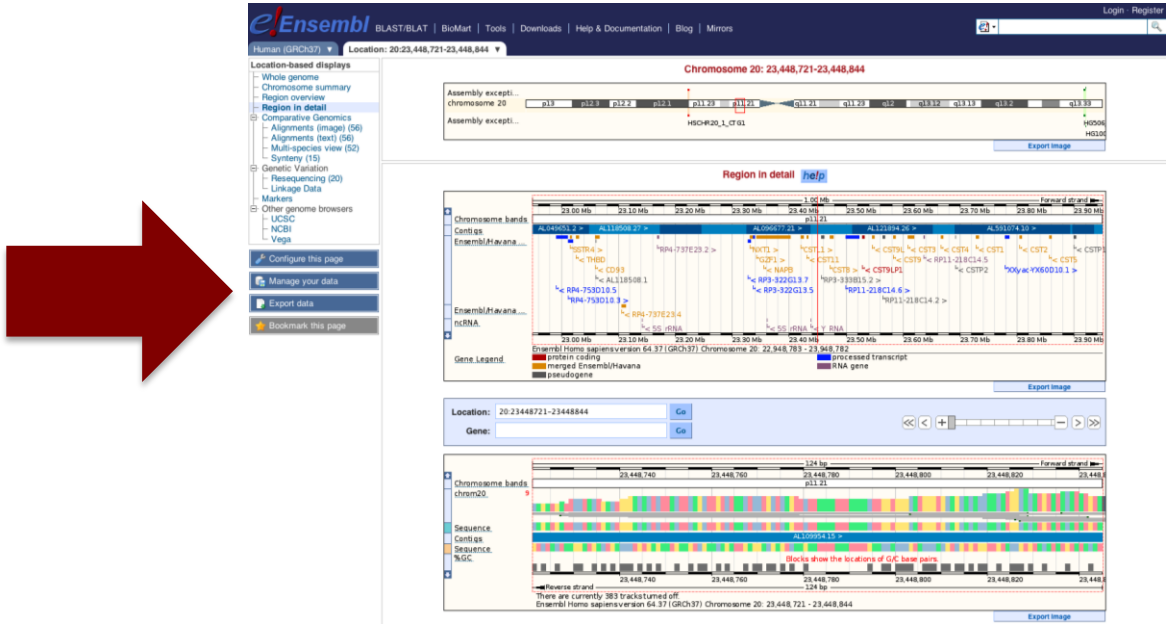
Genome assembly reminder

- The price of sequencing decreases
- The consequence of this decrease is a rush for sequencing new genomes
- However the current assembly methods only provide unfinished **draft genomes** i.e. pieces in which the genes are not all identified or annotated.

Intro

From raw DNA to annotated genome (genome viewer)





This process is called Genome Annotation

Oversimplified example: what would you do with this text?

articleoneallhumanbeingsarebornfreeandequalindignity
andrightstheyareendowedwithreasonandconscienceand
shouldacttowardsoneanotherinaspiritofbrotherhoodarticl
etwoeveryoneisentitledtoalltherightsandfreedomssetfort
hinthisdeclarationwithoutdistinctionofanykindsuchasrac
ecoloursexlanguagereligionpoliticalorotheropinionnation
alorsocialoriginpropertybirthorotherstatusfurthermoreno
distinctionshallbemadeonthebasisofthepoliticaljurisdictio
nalorinternationalstatusofthecountryorterritorytowhichap
ersonbelongswhetheritbeindependenttrustnonselfgover
ningorunderanyotherlimitationofsovereignty

(editing a text from an inefficient OCR program)

First find words (orfs)

article one all human beings are born free and equal in dignity and rights they are endowed with reason and conscience and should act towards one another in a spirit of brotherhood article two everyone is entitled to all the rights and freedoms set forth in this declaration without distinction of any kind such as race colour sex language religion political or other opinion national or social origin property birth or other status furthermore no distinction shall be made on the basis of the political jurisdictional or international status of the country or territory to which a person belongs whether it be independent trust non self governing or under any other limitation of sovereignty

(add spaces)

Intro

Then find sentences and punctuation (genes)

Article one. All human beings are born free and equal in dignity and rights. They are endowed with reason and conscience and should act towards one another in a spirit of brotherhood. Article two. Everyone is entitled to all the rights and freedoms set forth in this Declaration, without distinction of any kind, such as race, colour, sex, language, religion, political or other opinion, national or social origin, property, birth or other status. Furthermore, no distinction shall be made on the basis of the political, jurisdictional or international status of the country or territory to which a person belongs, whether it be independent, trust, non-self-governing or under any other limitation of sovereignty.

(add periods, commas,)

Finally find paragraphs and chapters (genes associated by function [pathways], co-expressing genes etc…)

Article one.

All human beings are born free and equal in dignity and rights. They are endowed with reason and conscience and should act towards one another in a spirit of brotherhood.

Article two.

Everyone is entitled to all the rights and freedoms set forth in this Declaration, without distinction of any kind, such as race, colour, sex, language, religion, political or other opinion, national or social origin, property, birth or other status. Furthermore, no distinction shall be made on the basis of the political, jurisdictional or international status of the country or territory to which a person belongs, whether it be independent, trust, non-self-governing or under any other limitation of sovereignty.

So, what did we do with the text?

- Found the syntax (characters, spaces, punctuation)
- We found semantics (the meaning of the word and sentences)
- Humans do this in parallel and in a recursive manner: Character recognition is helped by recognition of the words, misrecognized words are corrected based on the meaning of the entire phrase.
- This is NOT a trivial story (hermeneutics, semiotics, etc...).

Genome annotation definition (wikipedia)

- **Genome annotation is the process of attaching biological information to sequences**. It consists of two main steps:
  - 1. identifying elements in the genome, a process called gene prediction (syntax)
  - 2. attaching biological information to these elements (semantics)

- Automatic annotation tools try to perform all this by computer analysis, as opposed to manual annotation (a.k.a. curation) which involves human expertise. Ideally, these approaches co-exist and complement each other in the same annotation pipeline.
- The basic level of annotation is using BLAST for finding similarities, and then annotating genomes based on that. However, nowadays more and more additional information is added to the annotation platform. (…) Other databases (e.g., ENSEMBL) rely on both curated data sources as well as a range of different software tools in their automated genome annotation pipeline.

Genome annotation definition
(wikipedia, explained)

- ***Structural annotation*** consists of the identification of genomic elements. (SYNTAX, using only the sequence)
  - ORFs (open reading frames) and their localization
  - gene structure (intron-exon)
  - coding regions
  - location of regulatory motifs
- ***Functional annotation*** consists of attaching biological information to genomic elements. (SEMANTICS, using dbases)
  - biochemical function
  - biological function
  - regulation and interactions involved
  - expression
  - variants
- These steps may involve both biological experiments and *in silico* analysis.

In bioinformatics we separate the recognition of syntax and semantics. Khm, to some extent….

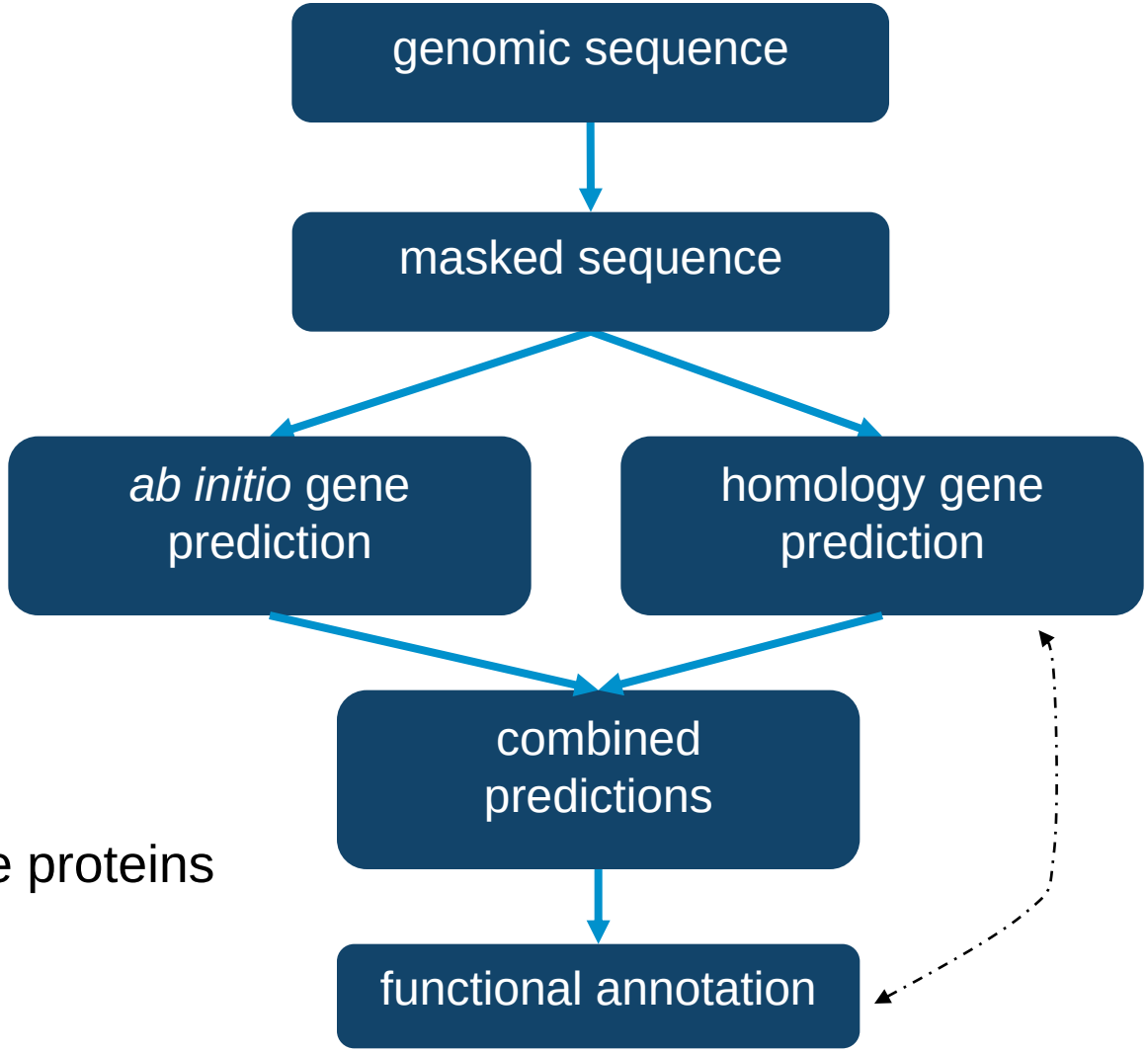
Step by step

Mask repetitive regions
(e.g., repeat masker)

Find genes *ab initio* and
if possible by homology
(many tools)

Combine both predictions
(e.g., MAKER)

Functional annotation of the proteins
(many tools)



Repeats and low complexity regions are bad…

- Repeats may confuse *ab initio* gene finders
  - they may call exons or even complete genes in repeat regions.
  - they may fragment gene predictions.
- Repeats may confound sequence alignment
  - especially in searches for synteny or segmental duplications.

Intro

Some repetitive elements are found in the human genome (and most genomes)

	Length	Copy number	Fraction of the genome
LINEs (long interspersed elements)	6-8 kb	850,000	21%
SINEs (short interspersed elements)	100-300 bp	1,500,000	13%
LTR (autonomous)	6-11 kb	450,000	8%
LTR (non-autonomous)	1.5-3 kb		
DNA transposons (autonomous)	2-3 kb	300,000	3%
DNA transposons (non-autonomous)	80-3000 bp		
SSRs (simple sequence repeats or microsatellite and minisatellites)			3%

- **These repeated elements should be masked (replaced by "N"s)**
- Repeat Masker can do it for you

Some prokaryotes (e.g. bacteria) have less repeats

What is gene finding (or gene prediction)?

- From a genomic DNA sequence we want to predict the regions that will encode for a protein: the coding genes.
- Gene finding is about detecting these coding regions and infer the gene structure starting from genomic DNA sequences.
- We need to distinguish coding from non-coding regions using properties specific to each type of DNA region.
- Gene finding is not an easy task!
  - DNA sequence signals have low information content (small alphabet and short sequences);
  - It is difficult to discriminate real signals from noise (degenerated and highly unspecific signals);
  - Gene structure can be complex (sparse exons, alternative splicing, ...);
  - DNA signals may vary in different organisms;
  - Sequencing errors (draft genome, frame shifts, ...).

GENE FINDING IN PROKARYOTES

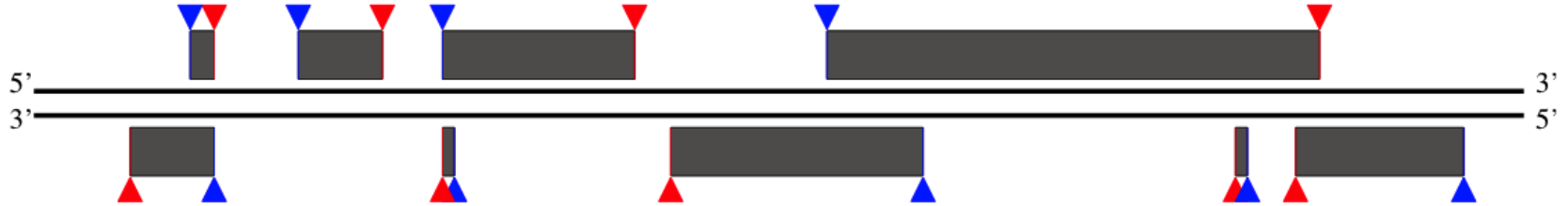
The simpler case

November 15, 2016

proks

Gene finding in prokaryotes

- High gene density and simple gene structure (mostly ORF).
- Short genes have little information.
- Overlapping genes.



Syntax is simpler than in eukaryotes

proks

Gene finding in prokaryotes

- Example of tools suitable for gene prediction in prokaryotes:
  - Glimmer 3: http://cbcb.umd.edu/software/glimmer/
  - GeneMark: http://opal.biology.gatech.edu/GeneMark/
  - MED2.0: http://ctb.pku.edu.cn/main/SheGroup/Software/MED2.htm
  - many more…
- We will mostly use GLIMMER to illustrate the principles.

proks

Step One

- Find open reading frames (ORFs).

We use a *presumed* syntax, the codon table.

proks

Step One

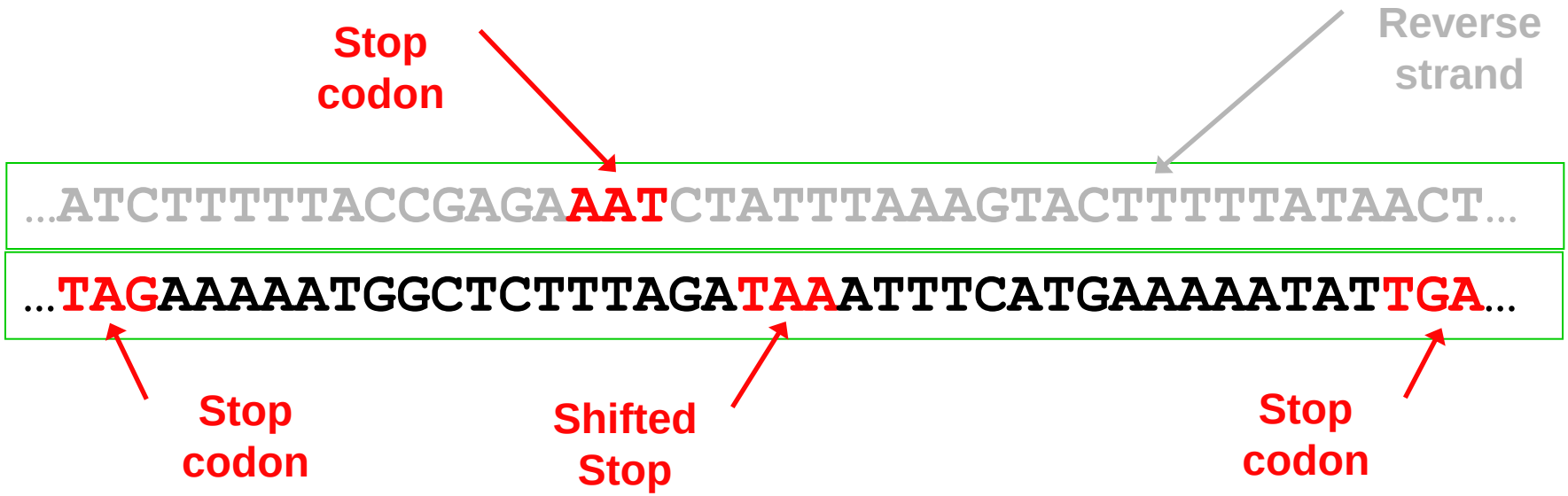
- Find open reading frames (ORFs).

We use a *presumed* syntax, the codon table.

proks

Step One

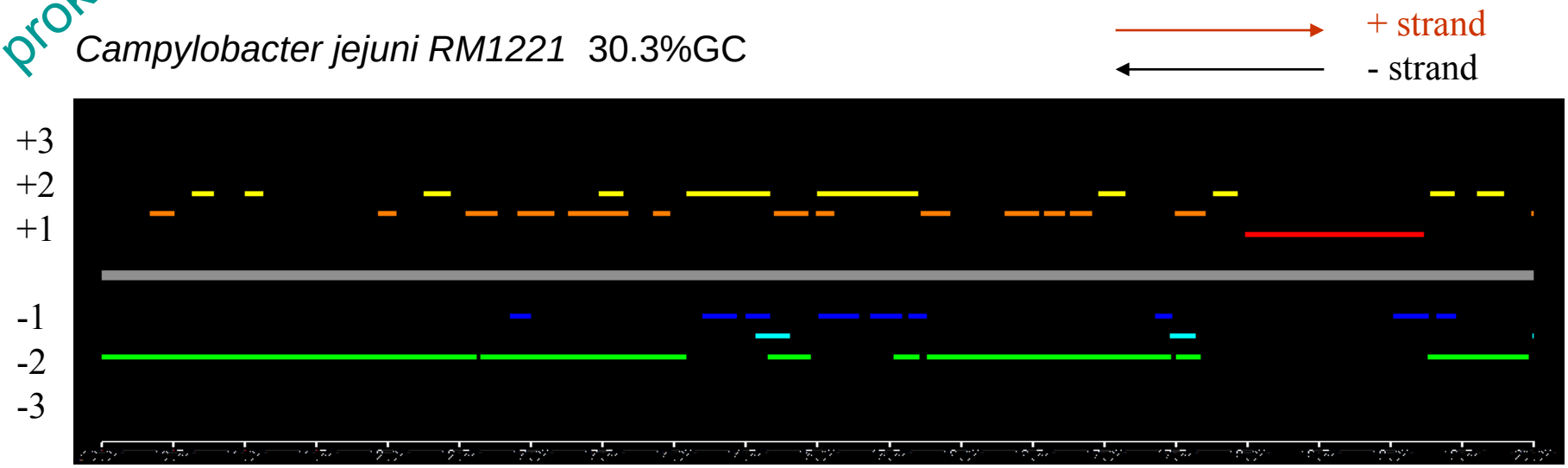
- Find open reading frames (ORFs).



- Translation programs give you 6 reading frames
- But ORFs generally overlap …

proks

Campylobacter jejuni RM1221 30.3%GC



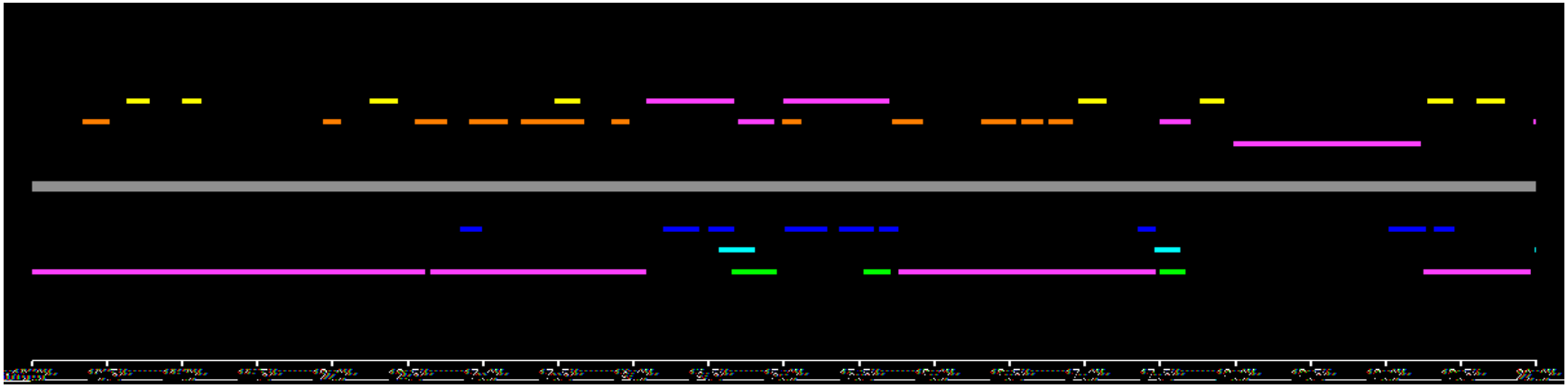
All ORFs on both strands shown
- color indicates reading frame

Longest ORFs likely to be protein-coding genes

We get this picture at normal or low GC content bacterial genomes

proks

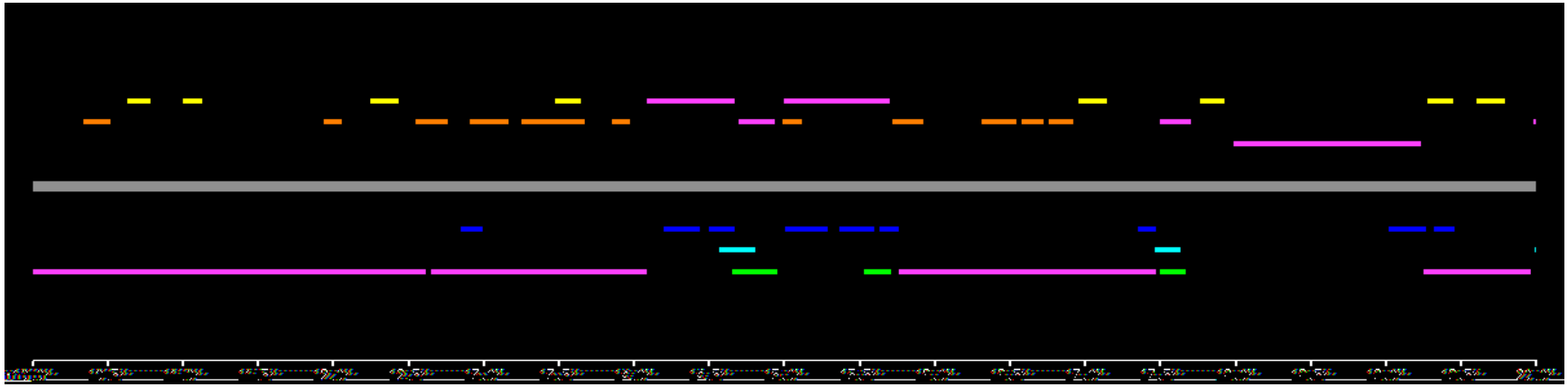
Campylobacter jejuni RM1221 30.3%GC



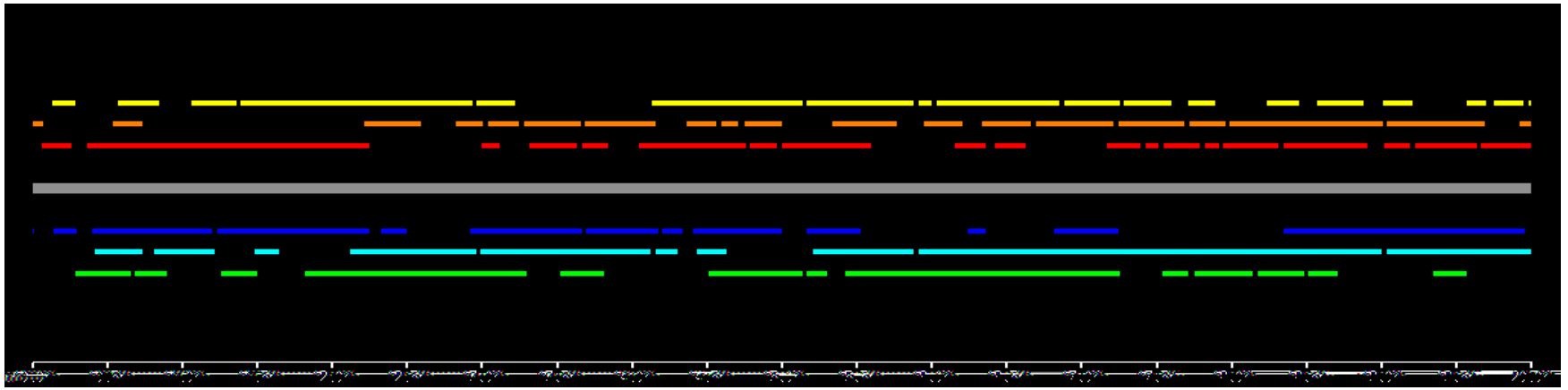
Purple ORFs show annotated (“true”) genes

proks

Campylobacter jejuni RM1221 30.3%GC



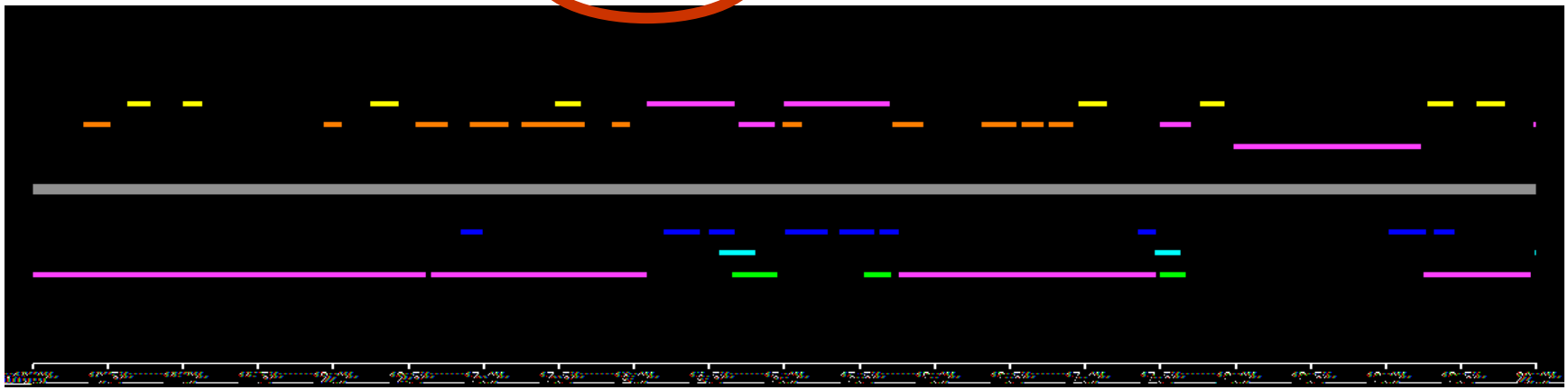
Mycobacterium smegmatis MC2 67.4%GC



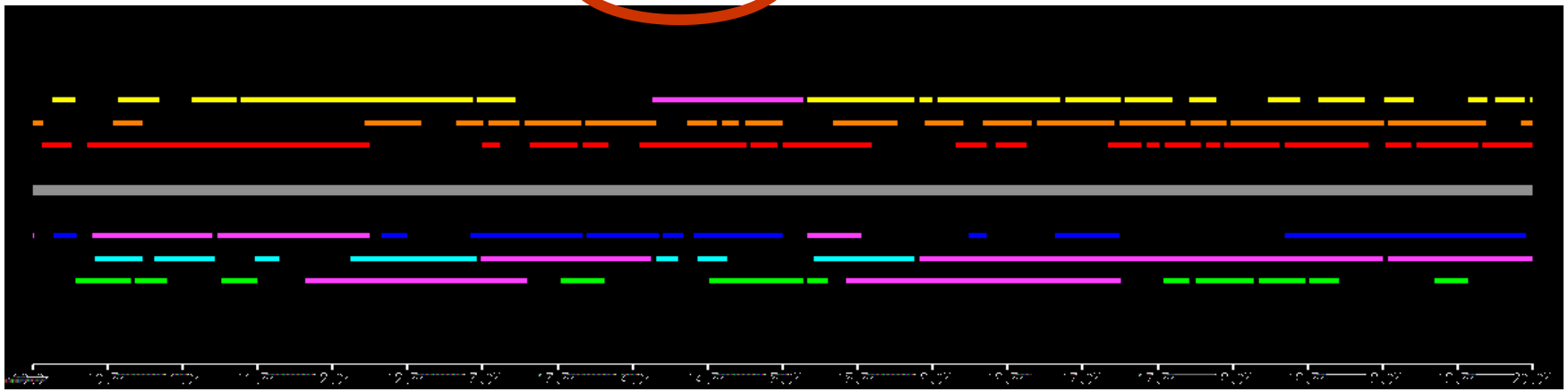
Note what happens in a high-GC genome

proks

Campylobacter jejuni RM1221 30.3%GC



Mycobacterium smegmatis MC2 67.4%GC



Purple lines show annotated genes

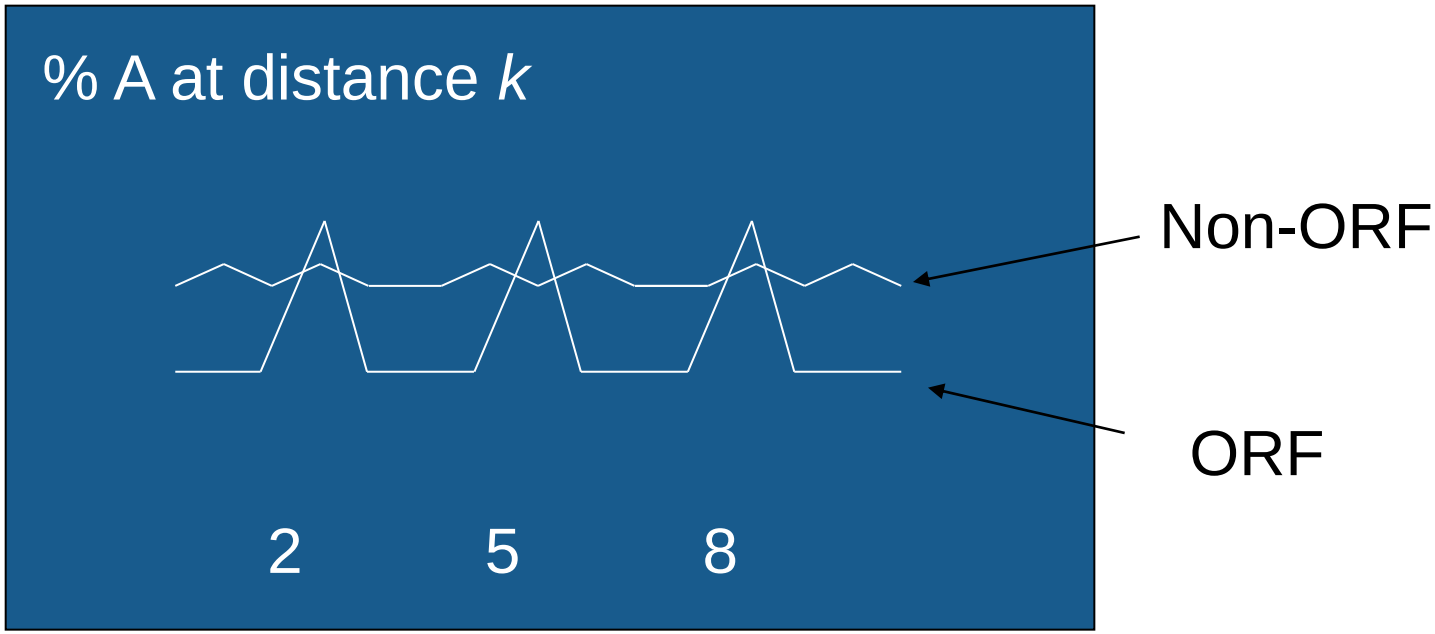
proks

The Problem

- Need to decide which orfs are genes.
  - Then figure out the coding start sites
- Can do homology searches but that won't find novel genes
  - Besides, there are errors in the databases
- Generally can assume that there are some known genes to use as training set.
  - Or just find the obvious ones

proks

Early programs used codon composition



- Codons frequently have A in the third position…
- It is easy to write periodicity recogrning programs…

proks

Codon Composition

Nucleotide variation at codon positions 1-3

Campylobacter jejuni

	Codon Position		
	1	2	3
a	36%	36%	36%
c	13%	17%	9%
g	30%	14%	10%
t	21%	33%	44%

Mycobacterium smegmatis

	Codon Position		
	1	2	3
a	19%	23%	6%
c	27%	28%	48%
g	42%	20%	39%
t	12%	28%	7%

Some bacteria have different compositions but the periodicity may still be there….

proks

Codon-Composition Gene Finders

- ZCURVE
  - Guo, Ou & Zhang, NAR 31, 2003
  - Based on nucleotide and di-nucleotide frequency in codons
  - Uses Z-transform and Fisher linear discriminant
- MED
  - Ouyang, Zhu, Wang & She, JBCB 2(2) 2004
  - Based on amino-acid frequencies
  - Uses nearest-neighbor classification on entropies

In practice:

1) Predict ORFs first, then characterize them by periodicity, length, etc.. Or

2) Predict high periodicity regions first and then look for start and stop codons within/around them.

Probabilistic Methods

- Create models that have a probability of generating any given sequence. Current programs use Markov style (HMM-like) models.
- Train the models using examples of the types of desired sequenc tyhpes (like *orf*, non-*orf*).
- The “score” of an *orf* is the probability of the model generating it.
  - Can also use a negative model (*i.e.*, a model of non-orfs) and make the score be the ratio of the probabilities (*i.e.*, the odds) of the two models.
  - Generally, use logs to avoid underflow

Quick recap of Markov models

- If we know the frequencies of A,C,G,T and the frequencies of all dinucleotides in a genome sequence, we can predict the probability of T following G as

$$p(T/G) = \frac{f(TG)}{f(G)}$$

  where f(GT) is the frequency of GT within the genome.

- This is a first order Markov model. Given a sequence, we can calculate the probability of the sequence by multiplying the probabilities of all dinucleotides. For instance, the probability score of AGGT will be

$$p(AGGT) = p(G/A) \times p(G/G) \times p(T/G)$$

- In practice we use logs, in order to avoid underflow.

Quick recap of Markov models 2

- We can easily extend first order Markov models to second order to calculate the probability of say „T” following not only „A” but „AG”:

$$p(T / AG) = \frac{f(AGT)}{f(AG)}$$

- For second order models we need to know the dinucleotide and trinucleotide frequencies. For k^{th} order models we need up to $k+1$ mer frequencies.
- Markov models allow us to predict the presence of a character (“state”) at position *i*. DNA is model with 4 observable states.
- We can assign further (“hidden”) states to the sequence, like gene and non-gene. We can calculate separate Markov models for the two hidden states. These are the Hidden Markov Models...

Quick recap of Markov models 3

- Say we have one model for “genes” and one for “non-genes”.

- The theoretically correct method is to travel along the sequence from left (5’) to right (3’) and calculate the probability of both models at every position, from the nucleotide found at that position. This is a simple dynamic programming procedure, called the Viterbi algorithm.
  - Advantage: exhaustive
  - Disadvantage: compute intensive

Quick recap of Markov models 4

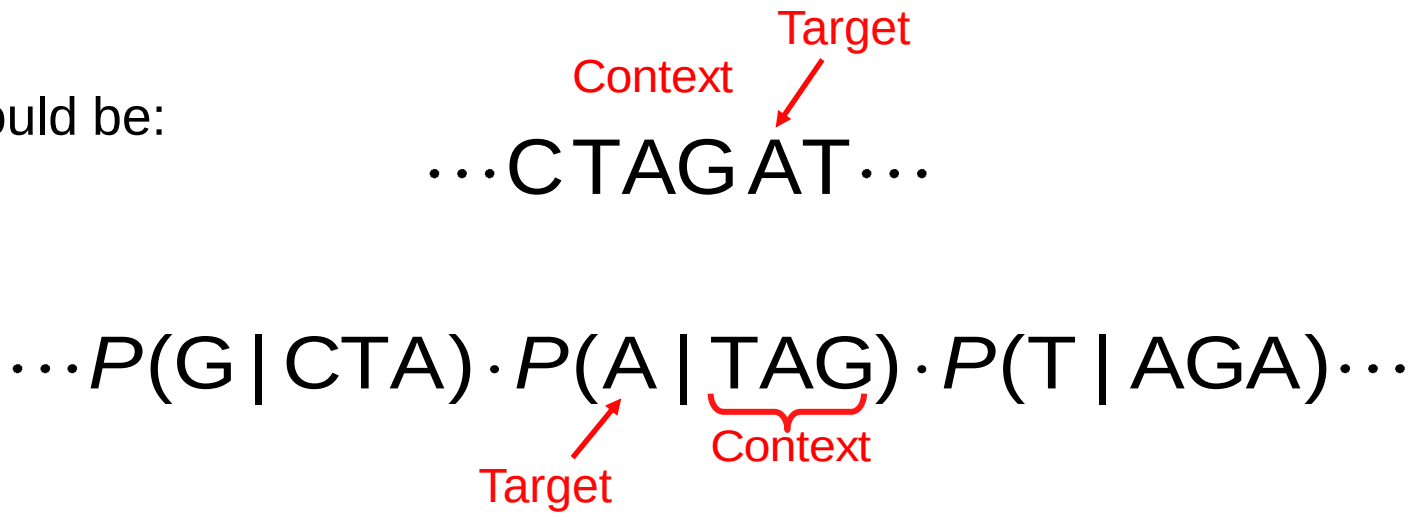
- Say we have one model for “genes” and one for “non-genes”.

- A simpler hack is to cut the sequence into ORFs in advance and score them according to the two models. Then accept those ORFs that are long enough and score higher with the model of “genes”.
  - Advantage: very fast
  - Disadvantage: fails at sequencing errors, so you need perfect data.

Early methods use fixed-order Markov models

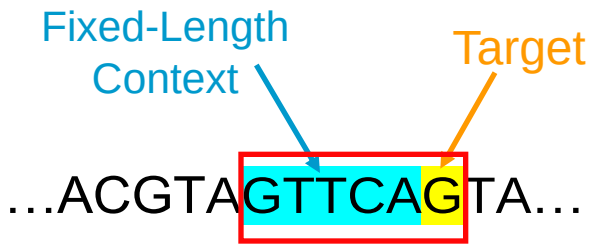
- *k* th-order Markov model bases the probability of an event on the preceding *k* events.
- Example: With a 3rd-order model the probability of this sequence:

would be:



Fixed-Order Markov Models

- Advantages:
  - Easy to train. Count frequencies of (*k*+1)-mers in training data.
  - Easy to compute probability of sequence.
- Disadvantages:
  - Many (*k*+1)-mers may be undersampled in training data.
  - Models data as fixed-length chunks.



proks

GeneMark

- Borodovsky & McIninch, *Comp. Chem* 17, 1993.
- Uses 5th-order Markov model.
- Model is 3-periodic, *i.e.*, a separate model for each nucleotide position in the codon.
- DNA region gets 7 scores: 6 reading frames & non-coding—high score wins.
- Lukashin & Borodovsky, *Nucl. Acids Res.* 26, 1998 is the HMM version.

proks

Multiple Markov Models can help if there are not enough data

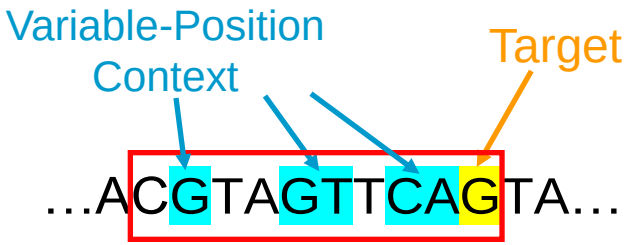
- Simply put, say we have enough trinucleotides, but the number of tetranucleotides is not sufficient to have a good estimate of the probabilities. So we will mix 4th order and 5th order Markov models.
- *E.g.*, for ggtta**x** the probability of **x** might depend on previous 3 bases tta .

  But for context cattax all 5 bases might be used.
- Glimmer 1.0 introduced this as “interpolated Markov models”, IMMs, which are a weighted average of the two largest values with statistical support.
  - Salzberg, Delcher, Kasif & White, *NAR* 26, 1998,

proks

Further improvements possible by using non-contiguous contexts

- In a classical Markov first, second, etc. models, the probability in position *i* is calculated from the previous, one, two etc. adjacent positions. (Time series analogy)
- But with sequences we can use non-neighboring positions as well….
- So choose set of positions that are most informative about the target position (have the best statistical support)



Introduced in Glimmer 2.0, Delcher, Harmon, *et al.*, *Nucl. Acids Res*. 27, 1999.

proks

Microbial gene prediction by GLIMMER 1

- Current versions of GLIMMER build separate models for all 6 reading frames, both for „genes” and for „non-genes”. These are available as precomputed files, but one might compute them for one’s favourite genome or genomes.

- GLIMMER first identifies all ORFs (hack) then scores them with all models and choses those ORFs that score the highest with the „gene” model. Separate heurisctics are used for sorting out overlapping genes.

proks

Microbial gene prediction by GLIMMER 2

- Practically 100% accuracy for microbial genes. Used by NCBI for genome annotation.
- Fails is the ORFs are incorrect (i.e. broken by point mutations, indels, etc...) This is not a problem with high coverage sequencing efforts.

proks

Functional prediction of microbial genomes is peanuts...

- We have high quality ORFs that can be translated to protein sequences.
- Protein sequences can be compared with well-annotated protein databases that gives you a function if there is very high similarity (identiy>90%. E-value $<10^{-4}$).
- The rest can be compared to functional database like COGs, or descriptions of structural groups like PFAM.
- Still, about 30% of microbial genomes remain unannatoted.
- Lesson: Peanuts are not worth a lot, after all…

GENE FINDING IN EUKARYOTES

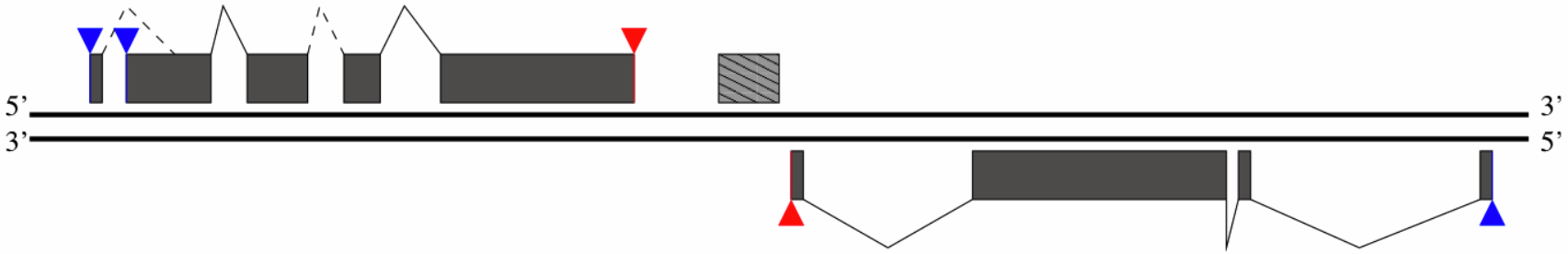
The simpler case

November 15, 2016

euks

Gene finding in eukaryotes

- Low gene density and complex gene structure.
- Alternative splicing.
- Alternative start and stop.
- Pseudo-genes.



Gene-finding is similar to, but ***more complicated*** than in prokaryotes

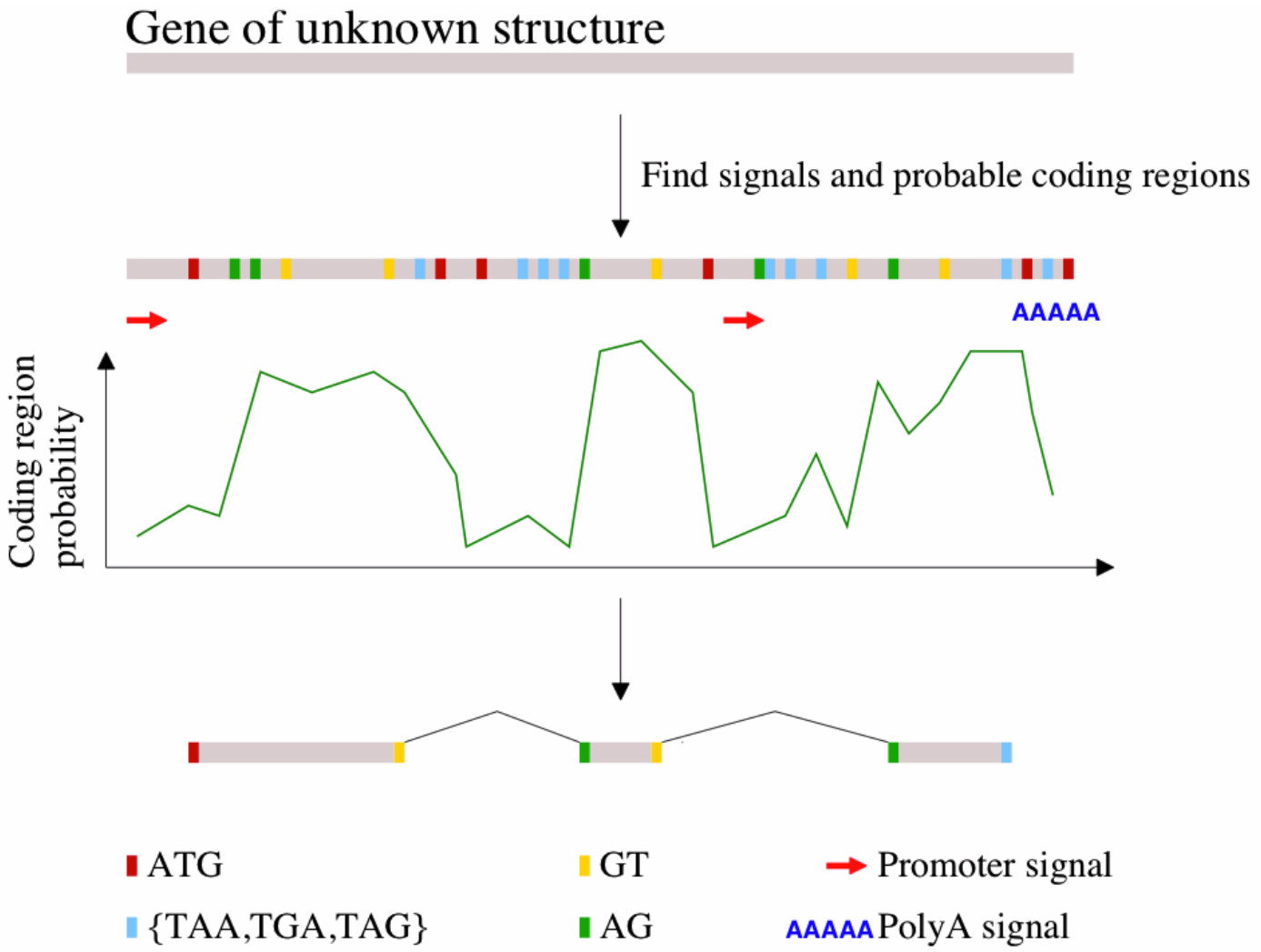
euks

Gene finding strategies: *ab initio*

- *Ab initio* methods:
  - Based on statistical signals within the DNA:
    - Signals: short DNA motifs (promoters, start/stop codons, splice sites, ...)
    - Coding statistics: nucleotide compositional bias in coding and non-coding regions
  - Strengths:
    - easy to run and fast execution time
    - only require the DNA sequence as input
  - Weaknesses:
    - prior knowledge is required (training sets)
    - high number of mispredicted gene structures

euks

Ab initio methods: a simple view



euks

Methods for signal detection

- Detect short DNA motifs (promoters, start/stop codons, splice sites, intron branching point, ...).
- 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.
  - 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 adjacent positions, starting from an aligned set of signals.

Like in GLIMMER

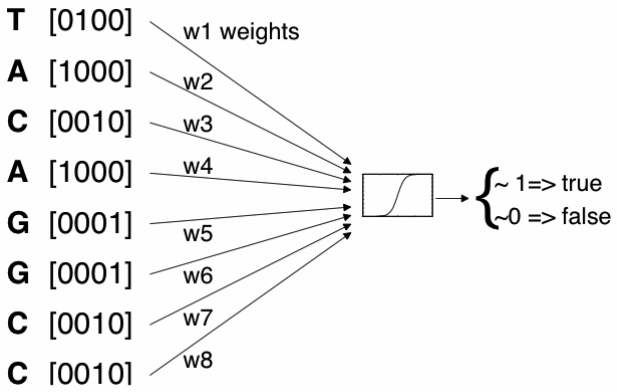
euks

Methods for signal detection

- Methods for signal detection:
  - Hidden Markov Models (HMMs):
    - HMMs use a probabilistic framework to infer the probability that a sequence correspond to a real signal.
  - Neural Networks (NNs):
    - NNs are trained with positive and negative examples. NNs ”discover” the features that distinguish the two sets.

Like in GLIMMER →

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



euks

Signal detection limitations

- Problems with signal detection:
  - DNA sequence signals have low information content.
  - Signals are highly unspecific and degenerated.
  - Difficult to distinguish between true and false positive.
- How to 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).

euks

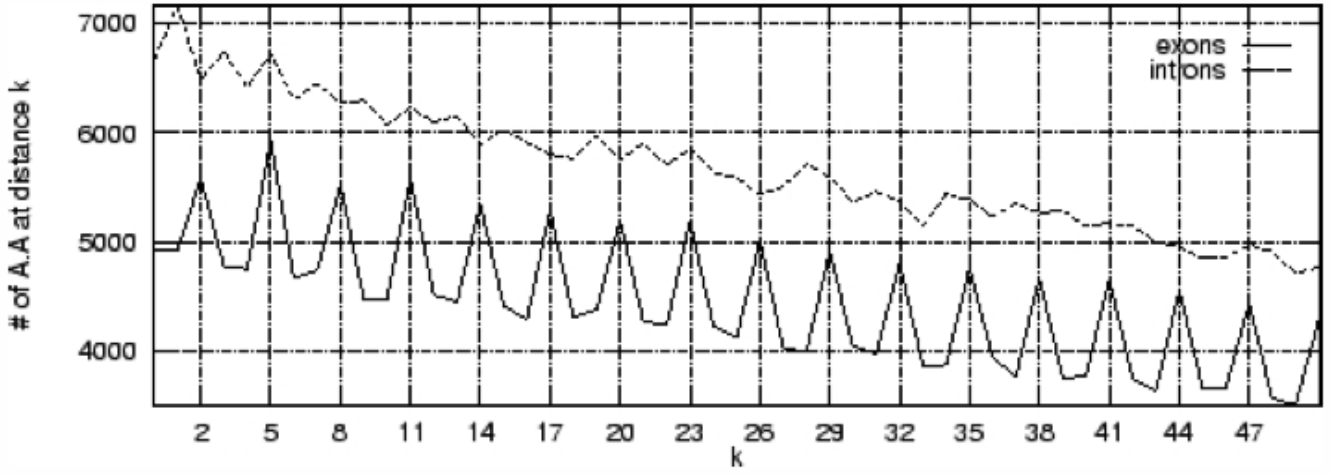
Types of coding statistics

- Inter-genic regions, introns, and exons have different nucleotides contents.
- This compositional differences can be used to infer gene structure.
- Examples of coding statistics:
  - 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.
    - This measure is 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.

euks

Types of coding statistics

- Examples of coding statistics:
  - Periodicity: The number of residues separating a pair of adenines (A) shows a periodicity in coding regions, but not in non-coding regions. This arise because of the asymmetry in base composition at the third codon position (3rd codon position: 90% are A/T; 10% are G/C).



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

euks

Coding statistics: codon frequencies

- In practice we use these computations in a search algorithm with a **sliding window**:
  - 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 (*n*=6) instead of 3-tuple (codon) frequencies. This method was found to be the best single property to predict whether a region 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.

euks

Integrating signal 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.

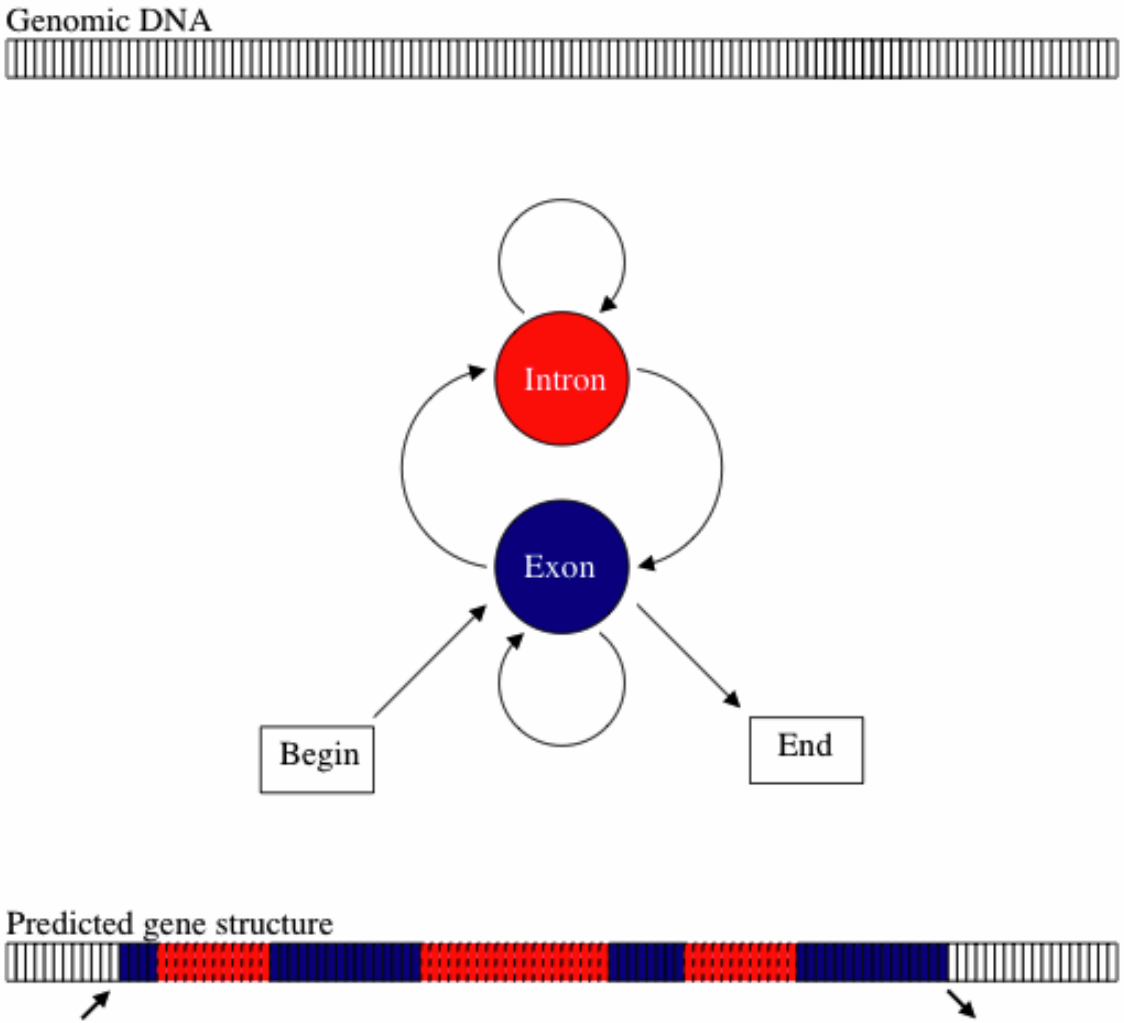
euks

Ab initio methods classification and examples

- Generalized HMM (Hidden Markov Models)
  - GenScan
  - HMMgene
  - Augustus
  - SNAP
- Linear and Quadratic discriminant analysis
  - FGENES and derived versions
  - MZEF
- Decision trees
  - MORGAN
- Neural Networks (NN)
  - GRAIL
- Support Vector Machines (SVM)
  - CONTRAST
  - mGene

euks

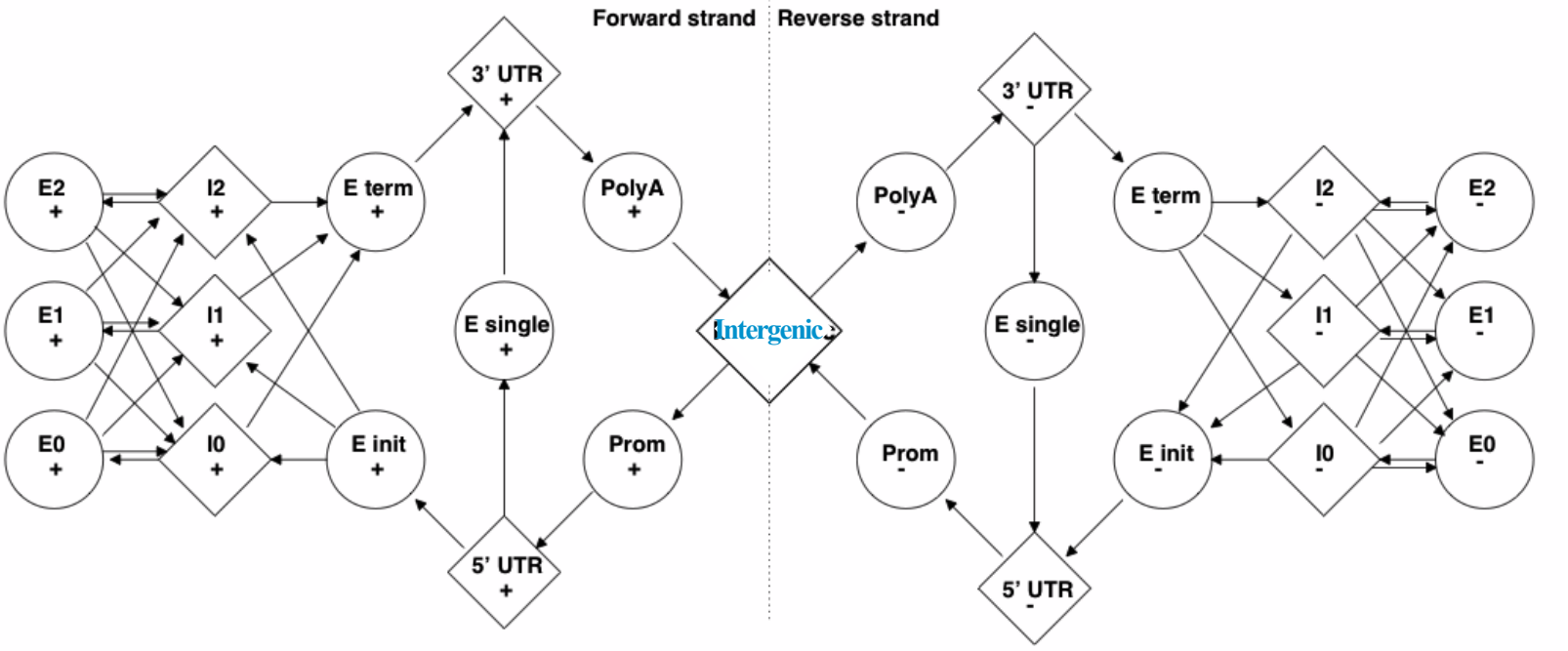
Ab initio methods: Generalized HMMs



euks

Ab initio methods: GENSCAN

- The underlying (hidden) model of GENSCAN:



euks

GENSCAN output

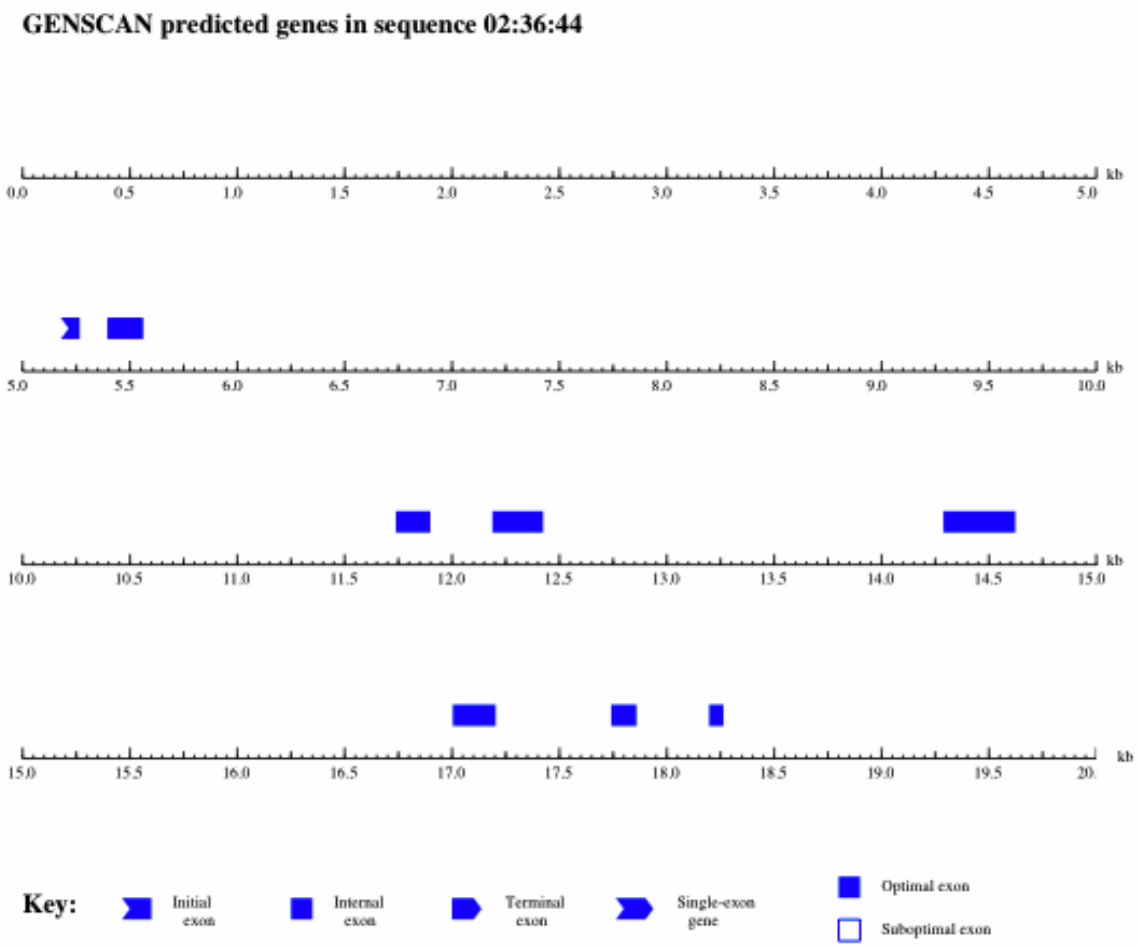
- WEB server: http://genes.mit.edu/GENSCAN.html
- Training sets: Vertebrate, Arabidopsis, Maize

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
MCRAISLRRLLLLLLQLSQLLAVTQGKTLVLGKEGESAELPCESSQKKITVFTWKFSDQR
KILGQHGKGVLIRGGSPSQFDRFDSKKGAWEKGSFPLIINKLKMEDSQTYICELENRKEE
...
```

euks

GENSCAN output



euks

Ab initio methods: HMMgene

- 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
- Web server:

http://genome.cbs.dtu.dk/services/HMMgene

- Training sets: human and worm

euks

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

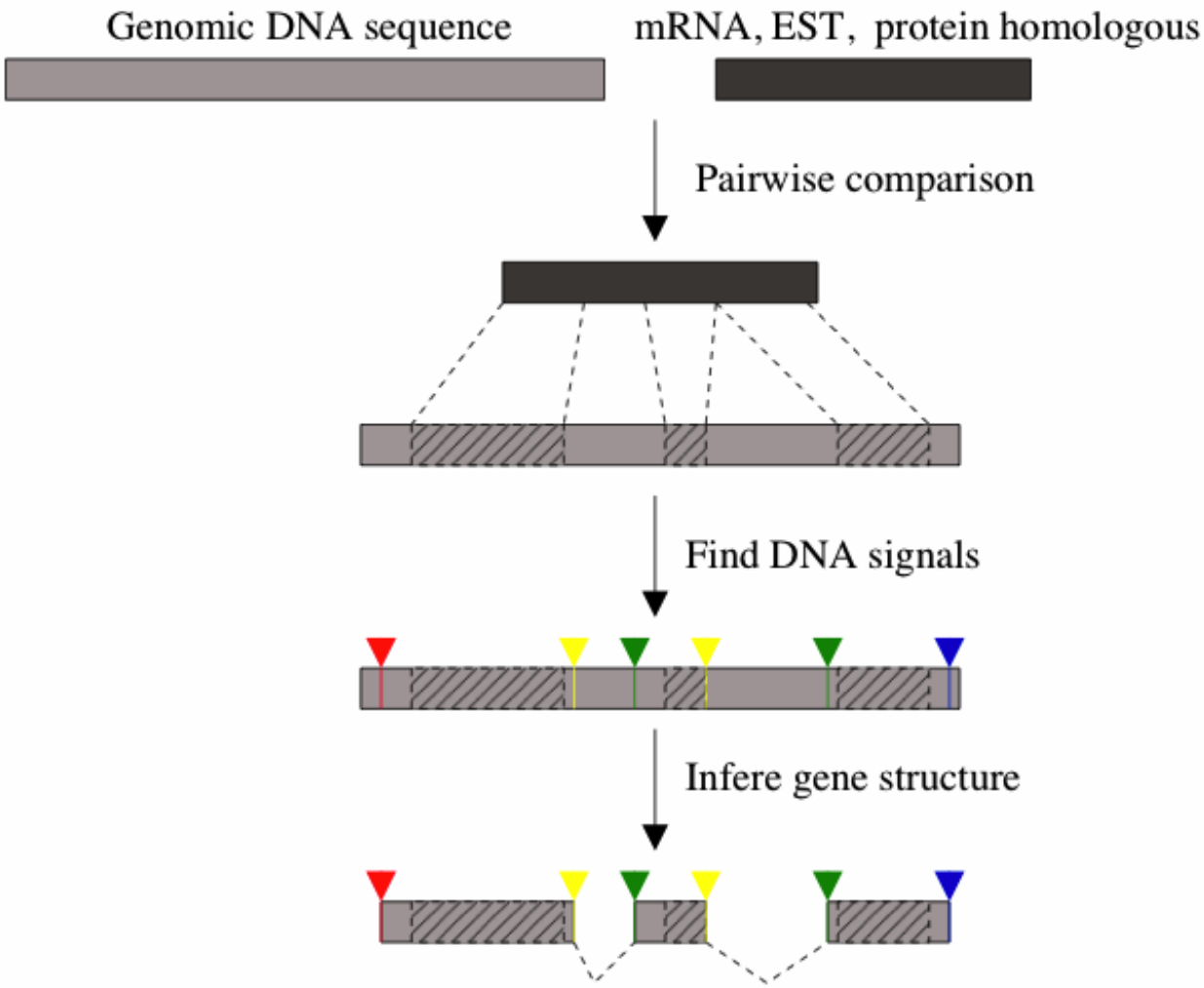
euks

Gene finding strategies: homology

- Homology methods:
  - Gene structure is deduced using known homologous sequences (RNAseq, EST, mRNA, protein).
  - Very accurate if there are homologous genes with high sequence similarity.
  - Strengths:
    - accurate
  - Weaknesses:
    - need of good homologous sequences
    - execution is slow

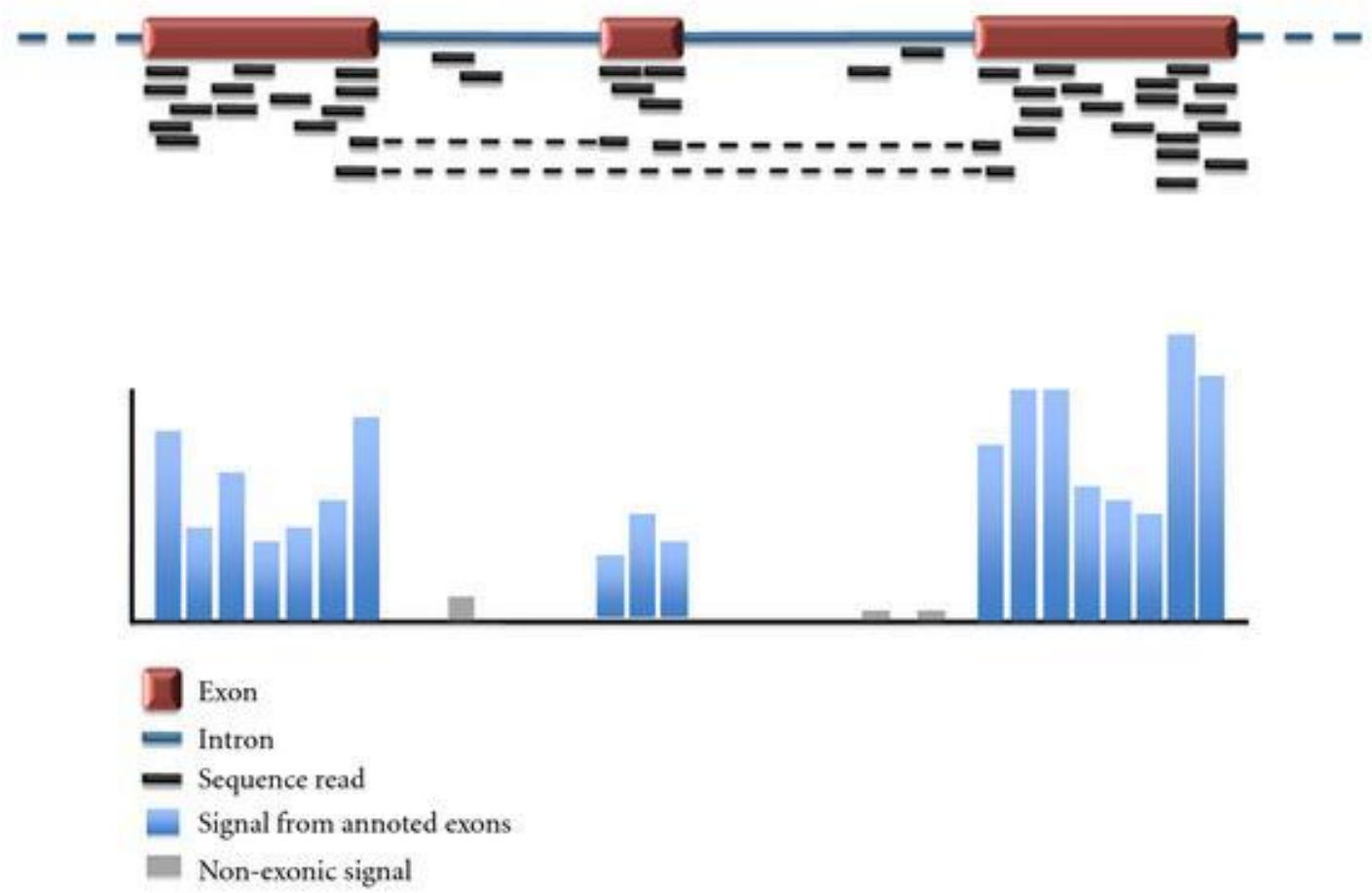
euks

Homology methods: a simple view



euks

Example of RNA-Seq exon identification



euks

Gene prediction limits

- Existing predictors are designed 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 (beware of draft genomes)
  - Training sets may be biased
  - Atypical genes use other grammars
- The best predictor is highly dependent on the genome (e.g., nGASP results are valid for *C.elegans*)

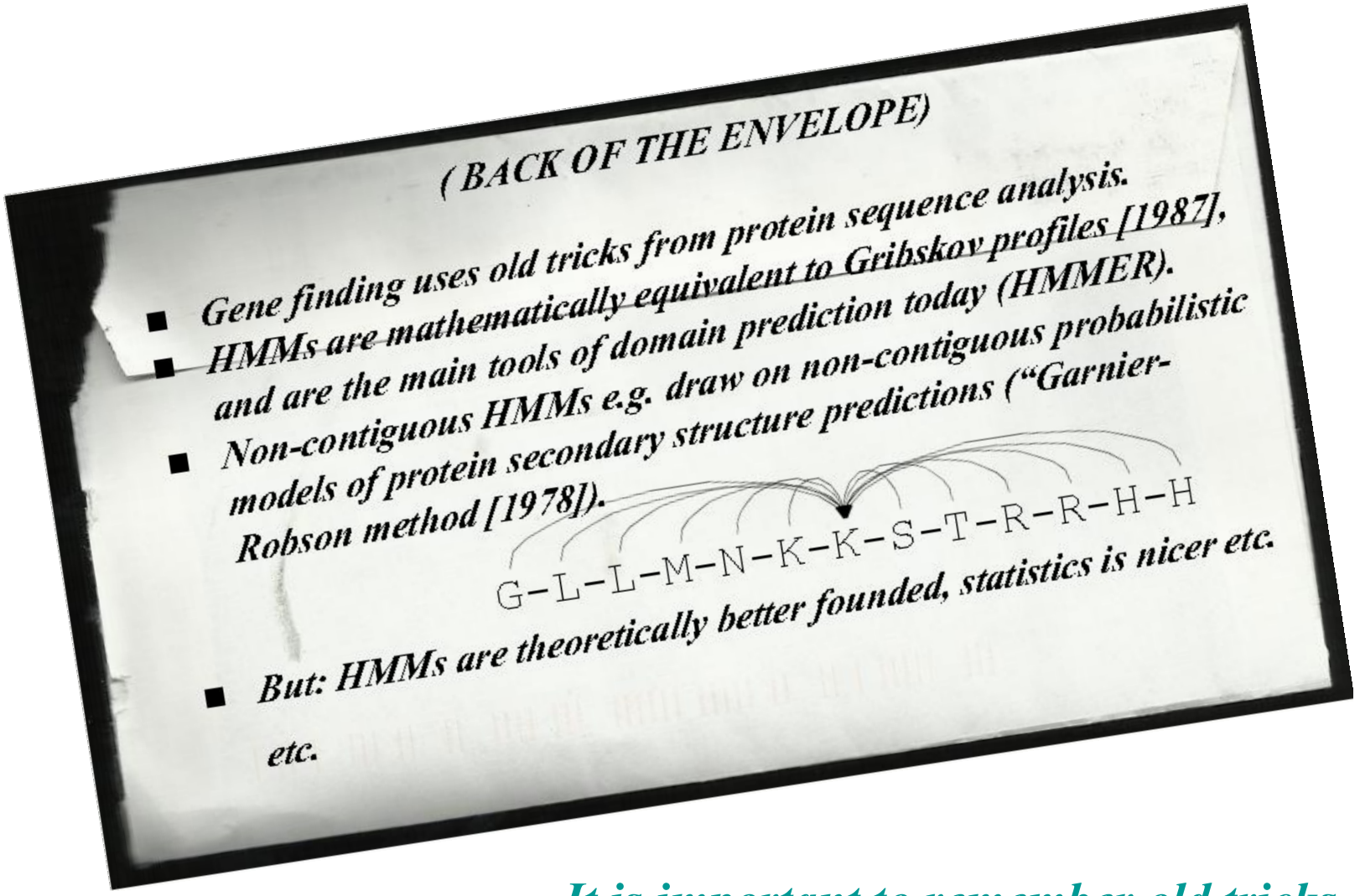
euks

Tools for ncRNA

- Genes are not only protein coding…
- Other kinds of RNA
  - ribosomal RNA
  - transfer RNA
  - small nucleolar RNA (snoRNAbase)
  - telomerase RNA
  - miRNA, siRNA (miRBase)

- Blastn 5S,16S, 23S, …
- Infernal (Rfam database) (HMMs)
- tRNAscan-SE

A note



It is important to remember old tricks. You can use them…

What have we learnt?

- Gene finding is the location of genes in raw DNA.
- Structural annotation uses only the sequence, functional annotation yses homology to known genes.
- Prokaryotes are simple, classification of ORFs with HMM-style methods is sufficient (GLIMMER)
- In eukaryotes, short signals are more important, exon/intron structure givew problems.
- Euk methods are similar to those in proks but there are no single best methods. The use of several gene finders and combination of the results is recommended.

Partly based on slides from

- Laurent Falquet, University of Fribourg, Switzerland
- Lorenzo Cerrutti, University of Basel, Switzerland
- Arthur Delcher, University of Maryland, USA
- The Trieste Bioinformatics Courses 2010-2014