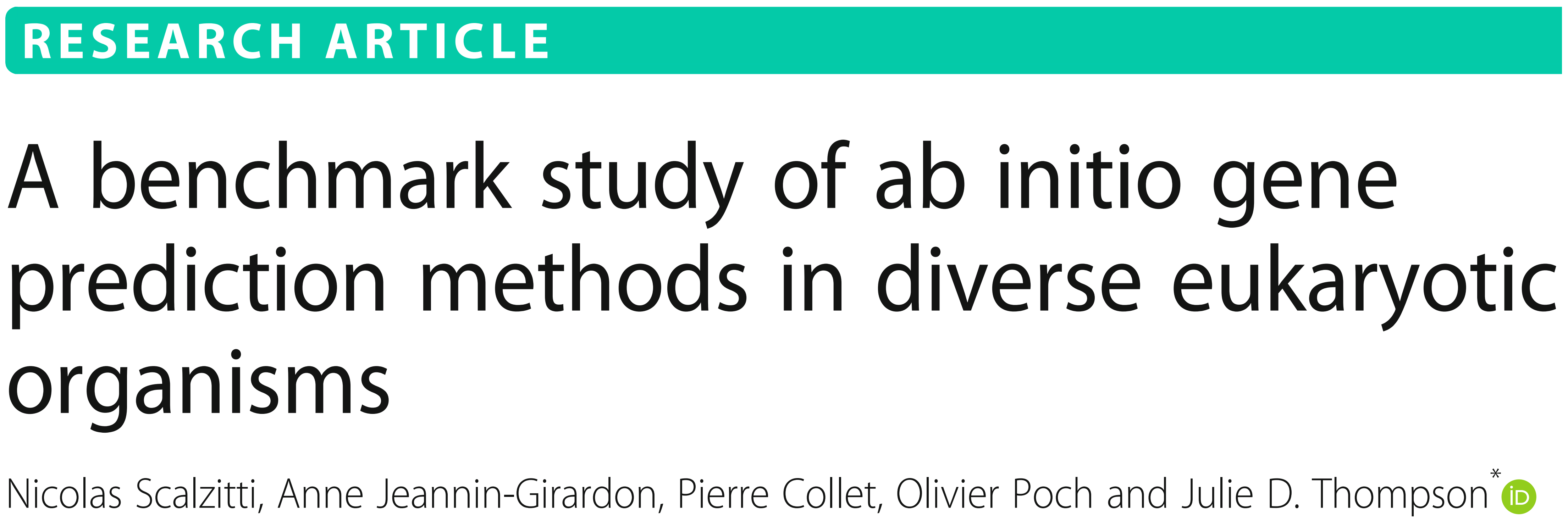


RESEARCH ARTICLE

A benchmark study of ab initio gene prediction methods in diverse eukaryotic organisms

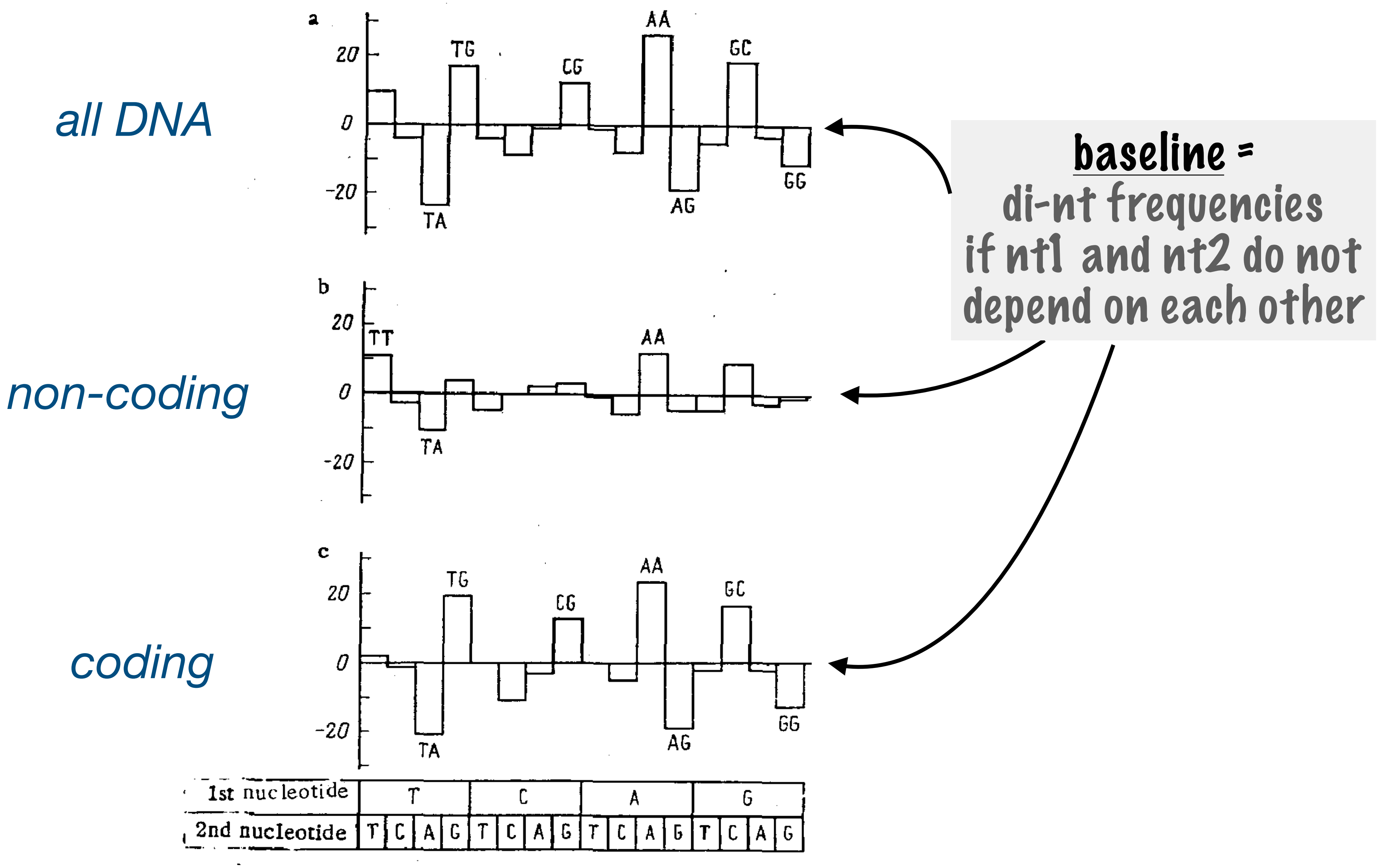
Nicolas Scalzitti, Anne Jeannin-Girardon, Pierre Collet, Olivier Poch and Julie D. Thompson*

Journal Club 2024-04-11

A little history on *ab initio* gene predictors...

1986 - Mark Borodovsky observes that in *E. coli*, the state of a nucleotide in the genome ***depends*** on the previous nucleotide, and that ***the nature of this dependency is different for coding and non-coding regions***.

He suggests that you can in principle recognise **coding** and **non-coding** regions in a DNA sequence using different kinds of **Markov chain** models


all DNA

non-coding

coding

baseline =
di-nt frequencies
if nt1 and nt2 do not
depend on each other

1st nucleotide	T				C				A				G			
2nd nucleotide	T	C	A	G	T	C	A	G	T	C	A	G	T	C	A	G

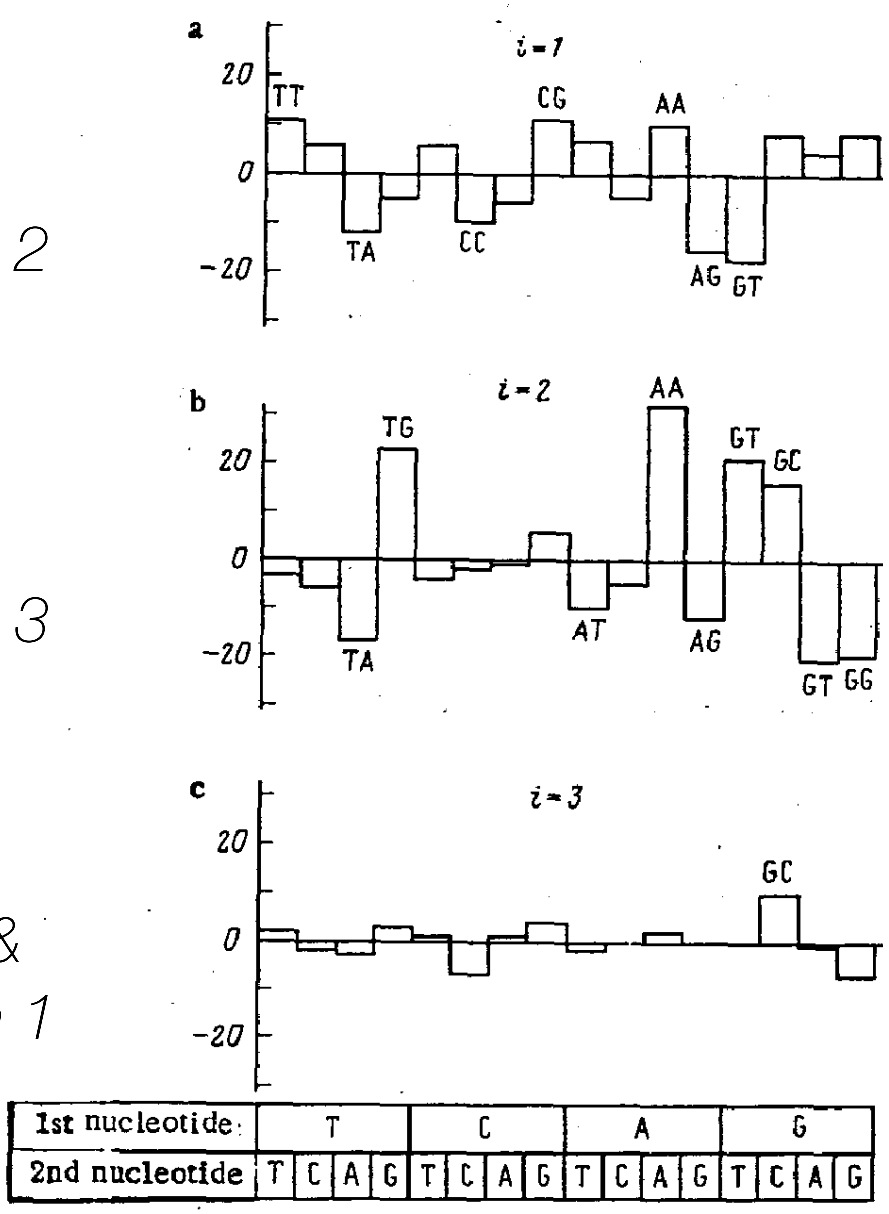
The nature of this dependency is also different ***within coding*** regions.
Different codon positions display different dependencies.

Each codon position can thus be modelled with a **distinct Markov chain**

frame 1
codon position 1 & 2

frame 2
codon position 2 & 3

frame 3
codon position 3 &
next codon position 1



baseline =
di-nt frequencies specific
for a given frame
if nt1 and nt2 do not
depend on each other

dependency for frame 3
is weak but statistically
significant

1993 - Mark Borodovsky & James McIninch explain the idea to mere mortals

- The ***state of a nucleotide*** at genome position ***i*** depends on the nucleotides that precede it (*i-1, i-2, i-3* etc)

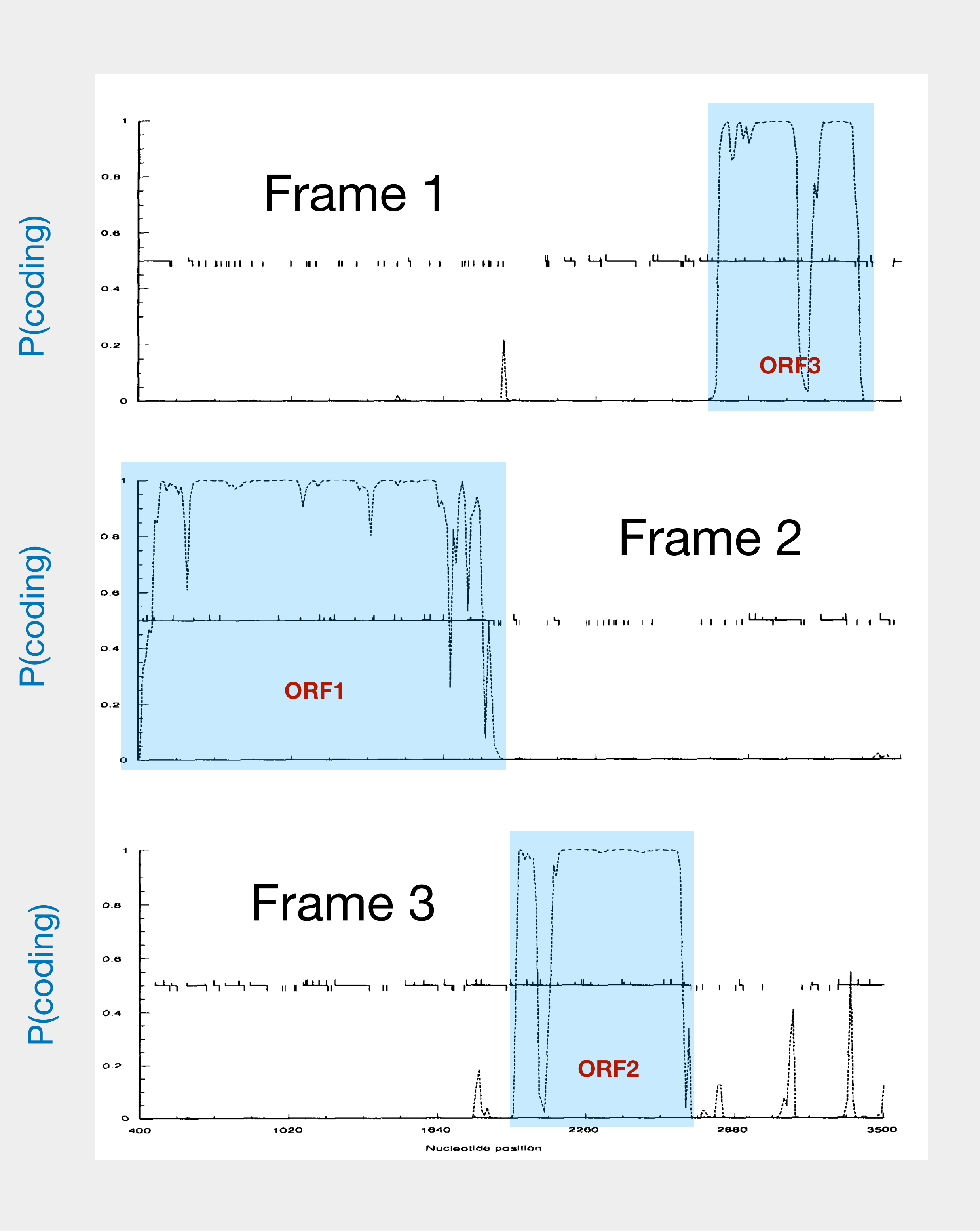
- **Protein-coding regions** can be modelled by **3-periodic non-homogeneous Markov chains**

- **Non-coding regions** can be modelled by **homogeneous Markov chains**

- You can estimate the **values of the parameters** of the Markov chains by **training** it on known coding and non-coding regions

- The trained "machine" can then applied to a new DNA sequence:
  What is the probability that this stretch of DNA was generated by this Markov model describing a protein-coding region?

- Every different species needs a differently trained "machine"



A trained "machine" is applied to a stretch of E. coli DNA, and identifies 3 genes

So why do we have these nucleotide dependencies in the genome? Why do they work so well to differentiate protein-coding regions from non-coding regions?
What is the biological origin of these signals?

Most of the papers describing the gene predictors do not really discuss why Markov models work, just that they do

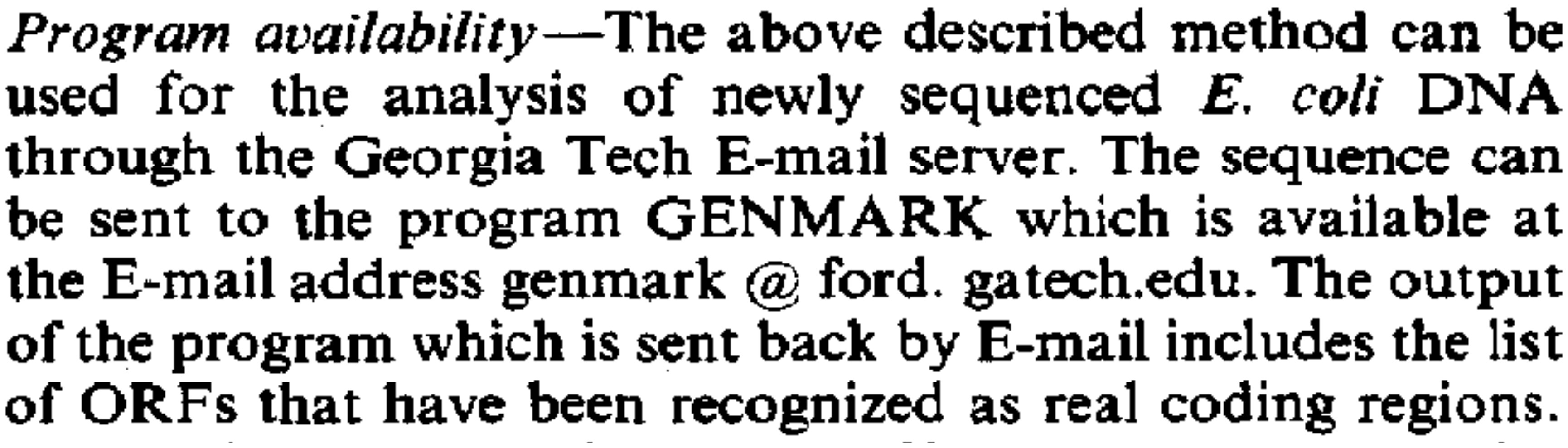
Borodovsky *et al* 1987
"... *the physical meaning of the correlation parameters of neighbouring nucleotides is not yet completely clear.*"

Stanke & Waack, 2003 (AUGUSTUS paper)
"*In theory, a perfect program should consider the biological signals for prediction instead of statistical features of the coding and non-coding sequences because - probably-* ***most of the sequence has no function for the transcription and translation process***"

The statistical dependencies are perhaps a natural consequence combining the nature of **the genetic code**, **natural selection** and **codon usage**?

1993 - The algorithm is improved:

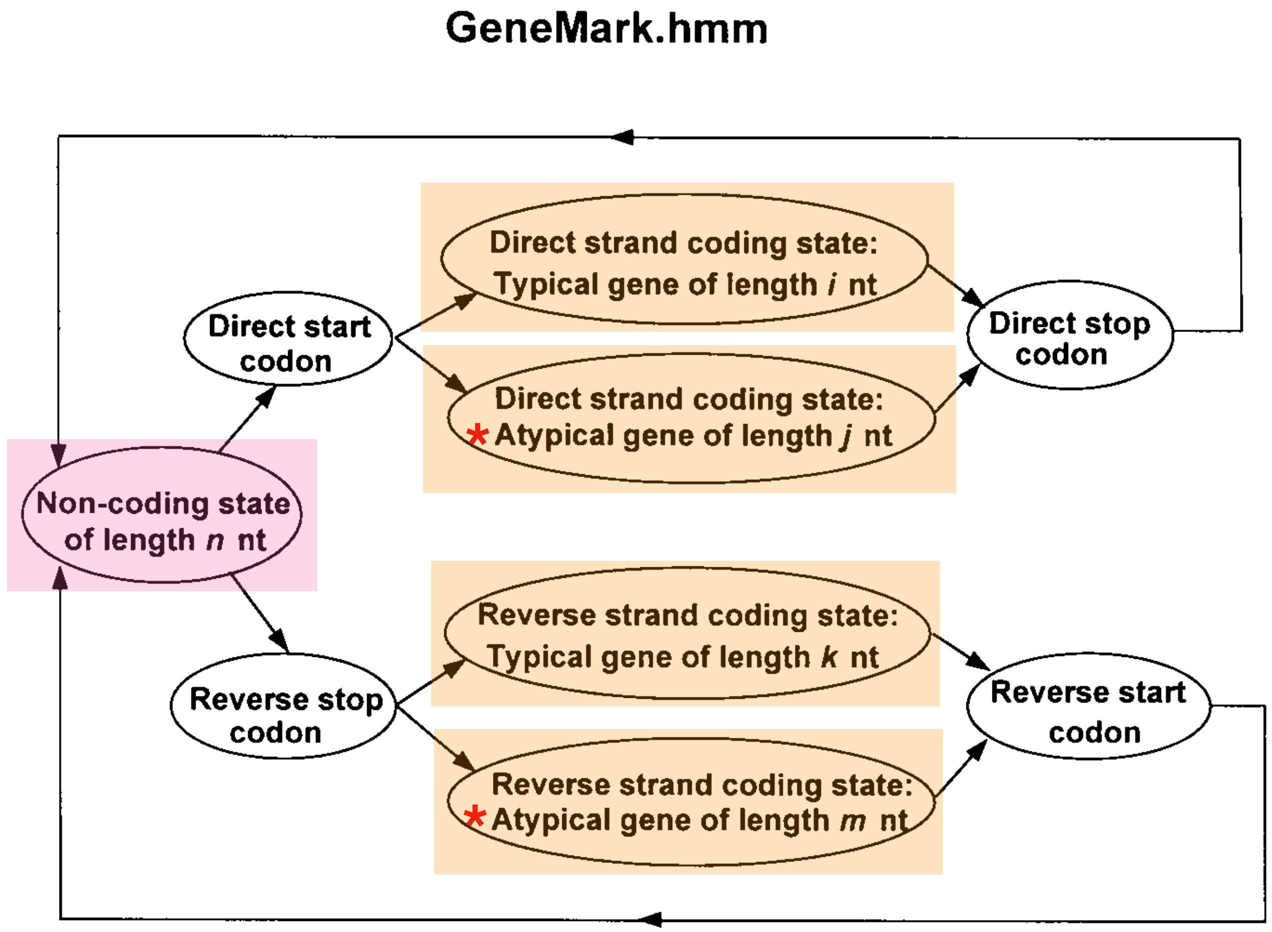
- It can now **process both strand simultaneously**
- It **no longer detects false genes on the complementary strand** of the real gene.
- It is found that accuracy is highest when using a **5th order Markov chain**, i.e. the state of nucleotide ***i*** depends on the 5-mer preceding ***i***
- The algorithm gets a name: **GENMARK**



Program availability—The above described method can be used for the analysis of newly sequenced *E. coli* DNA through the Georgia Tech E-mail server. The sequence can be sent to the program GENMARK which is available at the E-mail address genmark @ ford. gatech.edu. The output of the program which is sent back by E-mail includes the list of ORFs that have been recognized as real coding regions.

Oh the 90s …

1998 - The GeneMark algorithm is now encapsulated into a **General Hidden Markov Model framework (GHMM).** This allows for higher accuracy in predicting exact gene starts and ends. Dubbed **GeneMark.hmm**



GeneMark.hmm
Direct strand coding state: Typical gene of length *i* nt
Direct strand coding state: * Atypical gene of length *j* nt
Direct start codon
Direct stop codon
Non-coding state of length *n* nt
Reverse strand coding state: Typical gene of length *k* nt
Reverse strand coding state: * Atypical gene of length *m* nt
Reverse stop codon
Reverse start codon

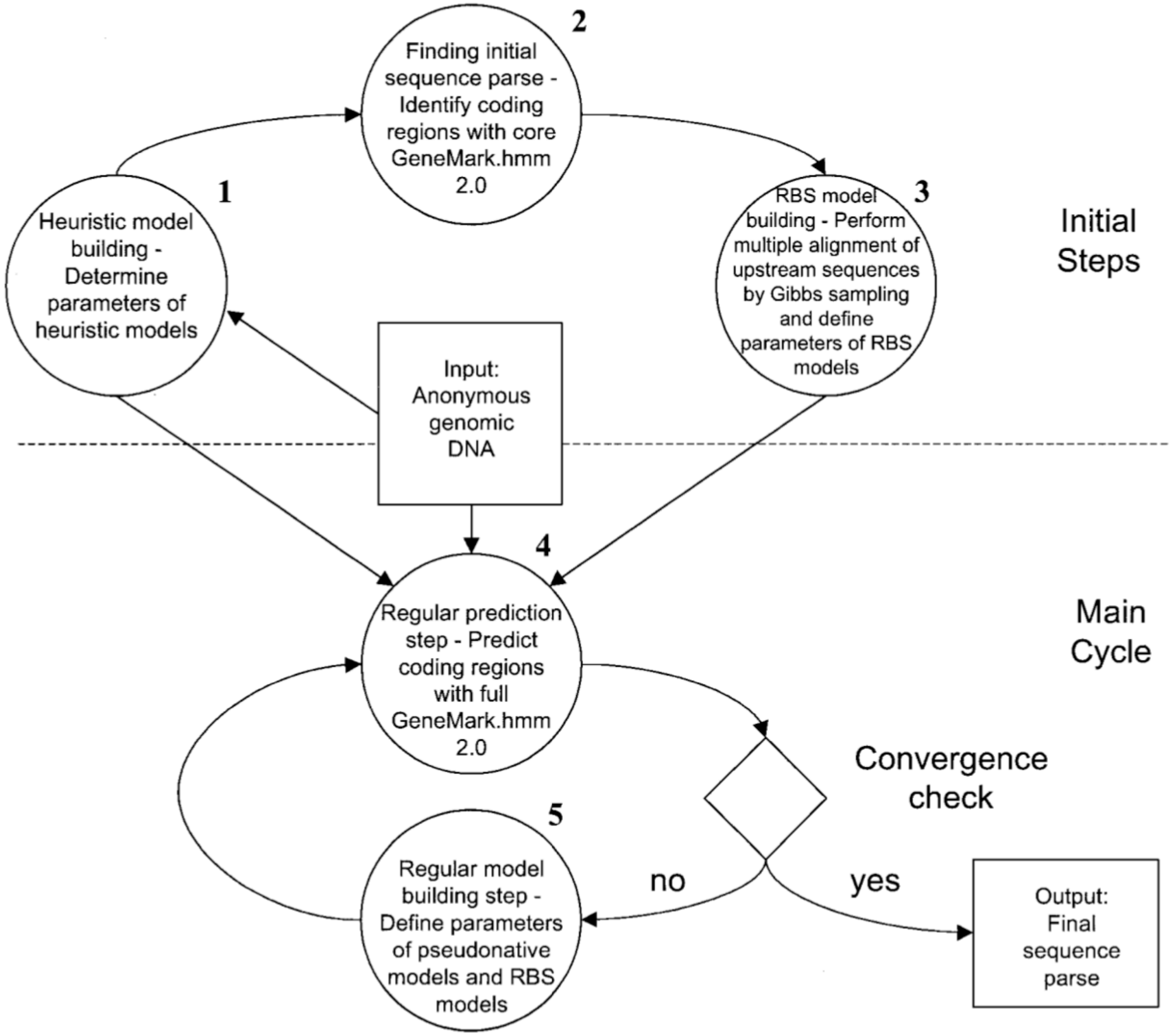
Figure 1. Hidden Markov model of a prokaryotic nucleotide sequence used in the GeneMark.hmm algorithm. The hidden states of the model are represented as ovals in the figure, and arrows correspond to allowed transitions between the states.

** "Atypical" meaning HGT genes*

protein-coding states
three periodic inhomogeneous
Markov chains

non-coding states
homogeneous
Markov chains

2001 - **GeneMarkS** is released, which improves accuracy of gene starts, and introduces **self-training**. Training the model with known genes is no longer necessary. It uses an iterative algorithm that stops once there are no more significant changes compared to the previous iteration



2
Finding initial sequence parse - Identify coding regions with core GeneMark.hmm 2.0
1
Heuristic model building - Determine parameters of heuristic models
3
RBS model building - Perform multiple alignment of upstream sequences by Gibbs sampling and define parameters of RBS models
Initial Steps
Input: Anonymous genomic DNA
4
Regular prediction step - Predict coding regions with full GeneMark.hmm 2.0
Main Cycle
Convergence check
5
Regular model building step - Define parameters of pseudonative models and RBS models
no
yes
Output: Final sequence parse

2005 - **GeneMark-ES** is released which expands the GHMM "machine" to work with **eukaryotic genomes**, and incorporates it into the self-training algorithm

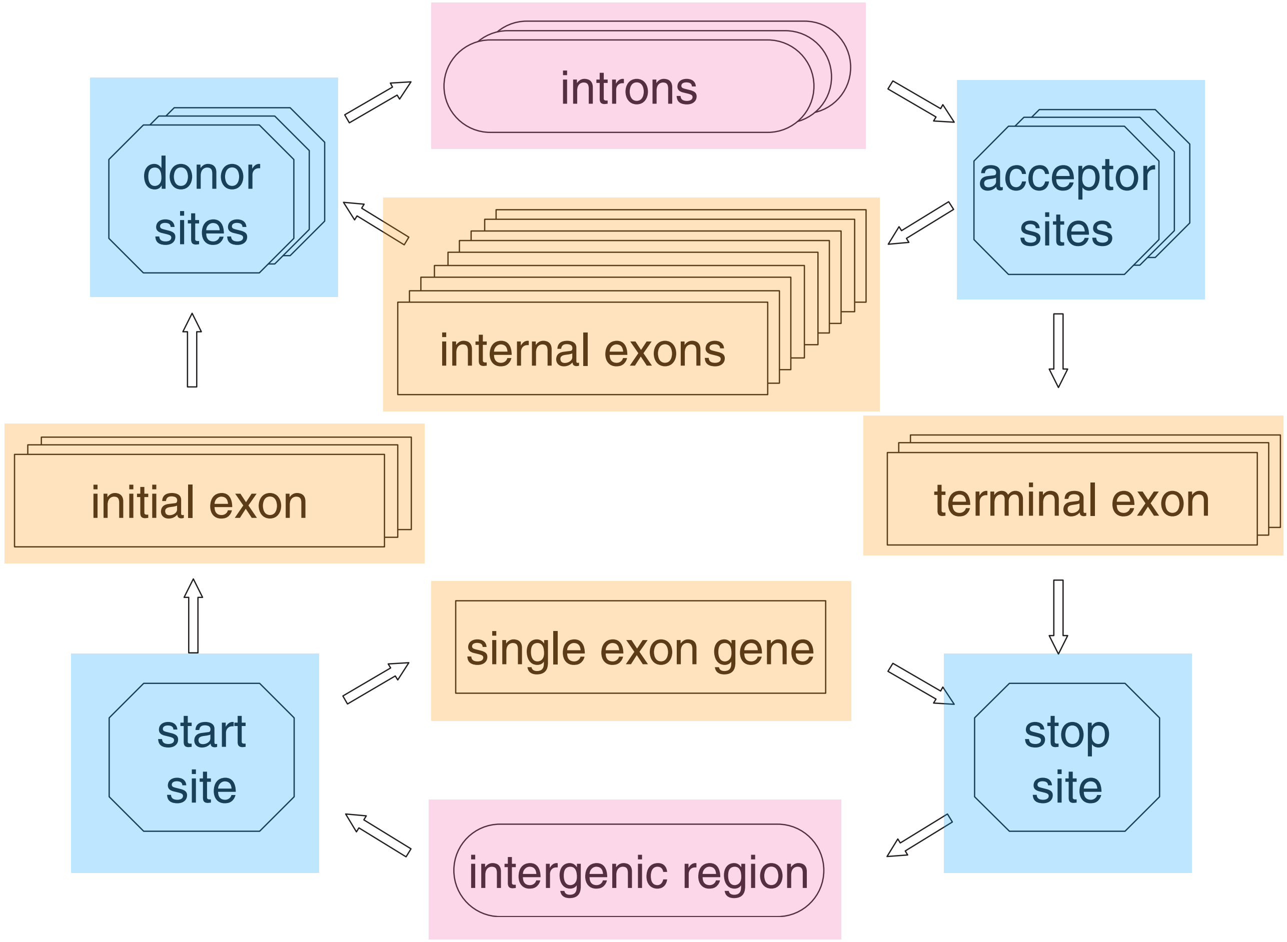


Figure 1. Diagram of hidden states of the HSMM employed in the eukaryotic GeneMark.hmm (E-3.0); only states emitting sequence of the direct DNA strand are shown, while the states generating sequence of the complementary strand (the mirror symmetrical part of the diagram with reversed arrows and horizontal symmetry line crossing 'intergenic region' state) are omitted.

site states
positional inhomogeneous
Markov chains

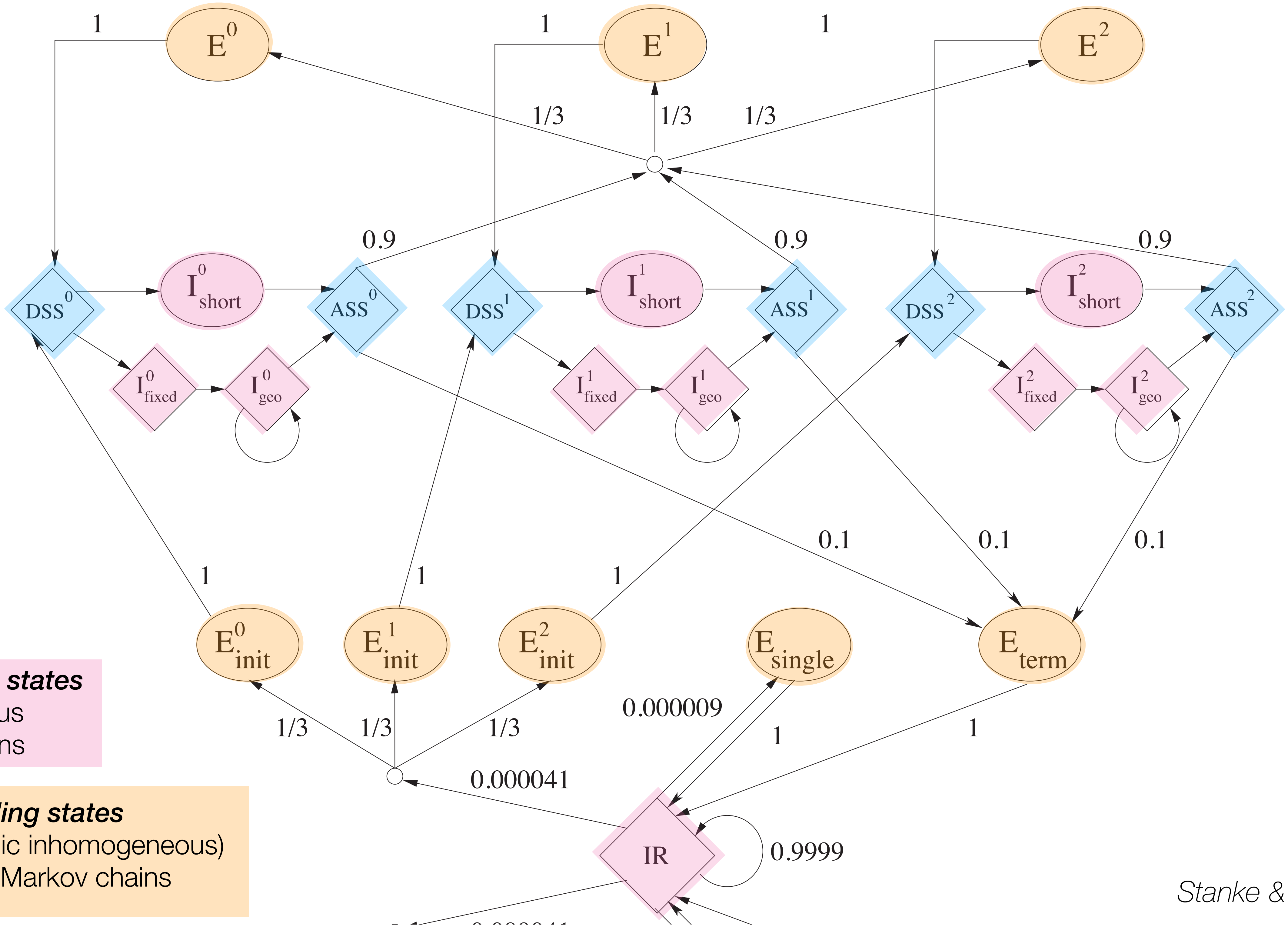
protein-coding states
three periodic inhomogeneous
Markov chains

non-coding states
homogeneous
Markov chains

AUGUSTUS	2003	***content sensors*** • Three periodic 4th order Interpolated Markov Model *(exons)* • 4th order Markov models *(intergenic regions, introns)* ***signal sensors*** • Similarity-based sequence weighting *(donor splice sites)* • Simple positional base frequencies *(acceptor splice sites)* • Windowed 3rd order Weight Array Method model *(translation initiation motifs, intron branch points)*	Assumes all introns are GT-AG type Assumes all introns have branch points No self-training! Offers 109 species specific models
GENSCAN	1997	***content sensors*** • Three periodic (inhomogeneous) 5th order Markov model *(exons)* • Homogeneous 5th order Markov model *(intergenic regions, UTRs, introns)* ***signal sensors*** • Weight Matrix Method *(polyA signals, translation initiations, TATA-box promoters, donor splice sites)* • 1st order Weight Array Method model *(acceptor splice sites)* • Windowed 2nd order Weight Array Method model *(intron branch points)*	Initially developed specially for the human genome No self training! Offers 3 pre-trained models

GeneID	2000	• 5th order Markov model
GlimmerHMM	2004	***content sensors*** • Three periodic (inhomogeneous) 0-8th order Interpolated Markov Model *(exons)* • homogeneous 0-8th order Interpolated Markov Model? *(introns, intergenic regions)* ***signal sensors*** • Maximum Dependence Composition & 1st order Markov chain *(donor splice sites, acceptor splice sites)* *or* • 2nd order Weight Array Matrices *(donor splice sites, acceptor splice sites)*
Snap		

The GHMM of AUGUSTUS



non-coding states
homogeneous
Markov chains

protein-coding states
(three periodic inhomogeneous)
Interpolated Markov chains

splice sites
DSS: similarity based sequence weighting
ASS: simple base frequencies

Stanke & Waack, 2003

These are the submodels of the forward strand.
The reverse strand is an exact mirror

ENCODE **G**enome **A**nnotation A**s**sessment **P**roject
(EGASP)

2006 - How accurate are the *ab initio* gene predictors actually?

- **Benchmark:** A highly curated but hidden set of genes in 44 separate regions (500 kbp - 2 Mpb each) of **the human genome**

- Gene annotations for 13 out of 44 regions were released to public - *training set*

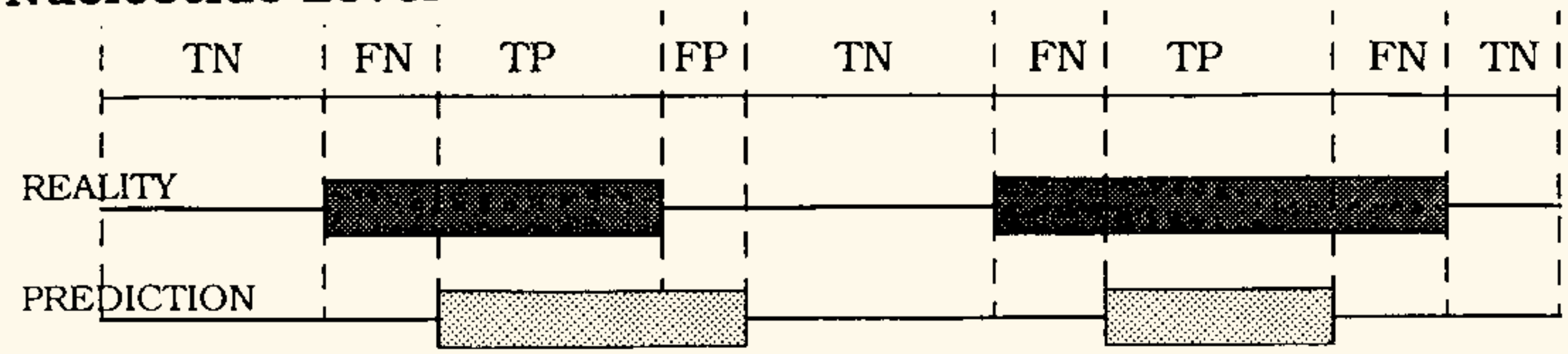
- Gene annotations for 31 out of 44 regions remained hidden - *test set*

- 3 groups (GeneMark, Augustus, GeneZilla) submitted *ab initio* predictions.

- They were then evaluated at the

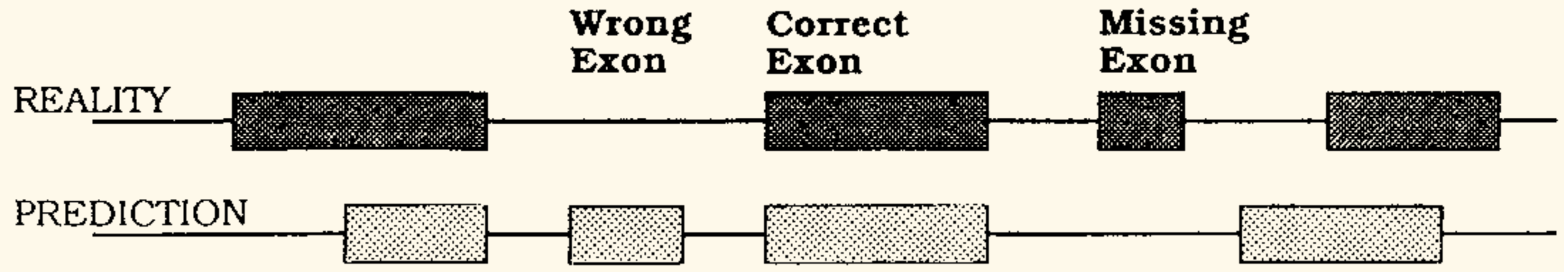
  nucleotide level - % of true coding nucleotides that were predicted
  exon level - % of true exons that were predicted exactly
  transcript level - % of true transcripts that were predicted exactly
  gene level - % of true genes that had at least 1 transcript predicted exactly

Nucleotide Level



Sensitivity: What fraction of all nucleotides that are truly coding are predicted to be coding?
Specificity: What fraction of all nucleotides predicted to be coding are actually truly coding?

Exon Level



Sensitivity: What fraction of all true exons are predicted exactly?
Specificity: What fraction of all predicted exons match true exons exactly?

Missing exons: What fraction of all true exons is entirely missed? *(i.e. not even overlapping with a predicted exon)*
Wrong exons: What fraction of all predicted exons is entirely wrong? *(i.e. not even overlapping with a true exon)*

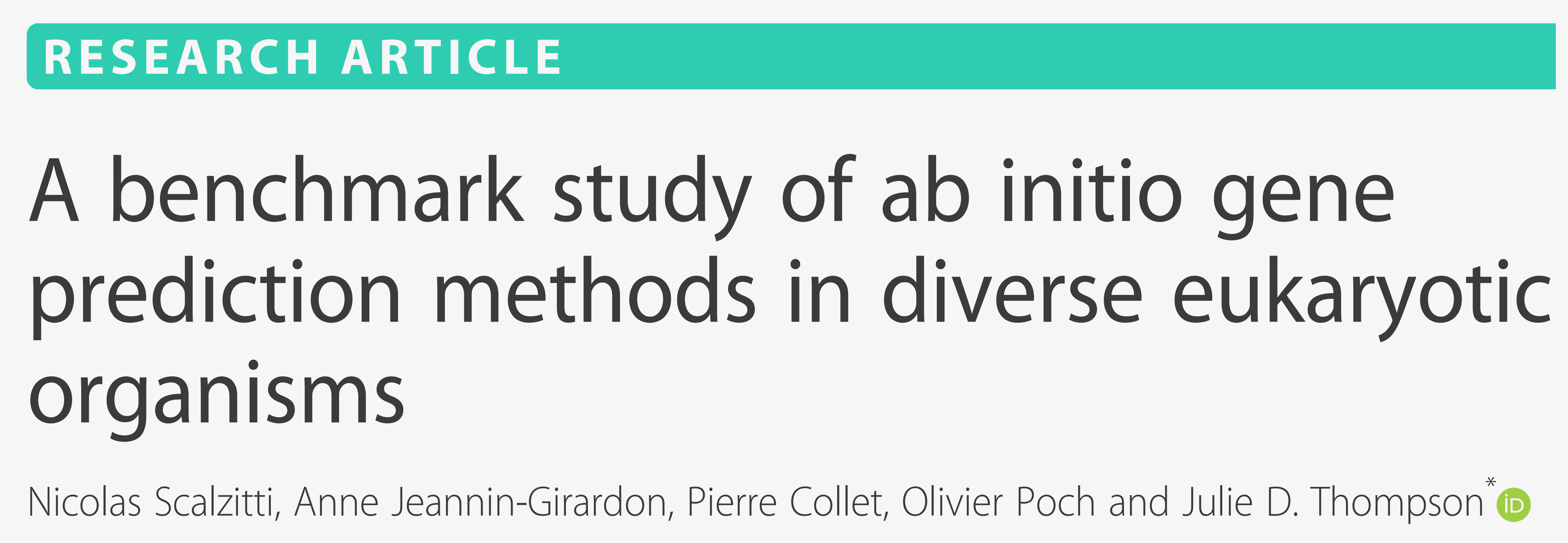
Burset & Guigo, 1996

	Nucleotide			Exon			
	NS*n*	NS*p*	N *CC*	ES*n*	ES*p*	ME	WE
AUGUSTUS-abinit	78.65%	75.29%	0.76	52.39%	62.93%	29.09%	24.82%
* GENEMARK.hmm-A	78.43%	37.97%	0.53	50.58%	29.01%	27.86%	63.27%
GENEMARK.hmm-B	76.09%	62.94%	0.69	48.15%	47.25%	31.77%	40.68%
* GENEZILLA	87.56%	50.93%	0.66	62.08%	50.25%	19.14%	41.93%

	Transcript		Gene		
	TS*n*	TS*p*	GS*n*	GS*p*	Ratio CDS/UTR
AUGUSTUS-abinit	11.09%	17.22%	24.32%	17.22%	100.00%
* GENEMARK.hmm-A	6.93%	3.24%	15.20%	3.24%	100.00%
GENEMARK.hmm-B	7.70%	7.91%	16.89%	7.91%	100.00%
* GENEZILLA	9.09%	8.84%	19.59%	8.84%	100.00%

* *used unmasked genome as input, which leads to more false positive genes, and hence lower specificity*

- Other benchmark studies have mainly looked at datasets of other mammals, and eukaryotic pathogens. In other words, their accuracy in atypical eukaryotes is unclear



RESEARCH ARTICLE

A benchmark study of ab initio gene prediction methods in diverse eukaryotic organisms

Nicolas Scalzitti, Anne Jeannin-Girardon, Pierre Collet, Olivier Poch and Julie D. Thompson* benchmark for **G**ene and **P**rotein **P**rediction **P**r**O**grams

The **G3PO** Benchmark

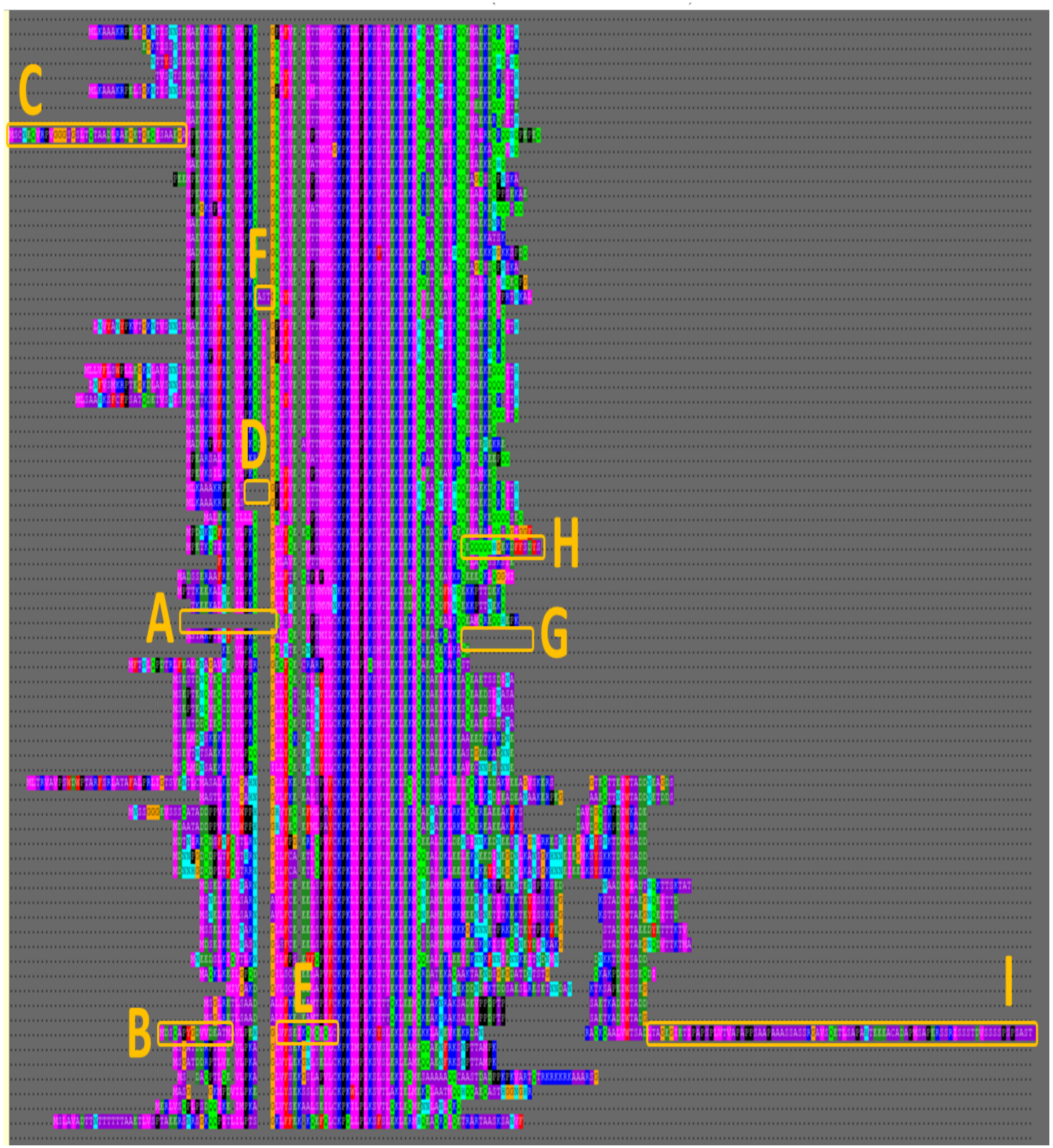
- 1793 individual gene sequences in total
- 20 Bardet-Biedl Syndrome (BBS) proteins
  BBS1-21, excluding BBS14
- Orthologs in 147 eukaryotes

102	Opisthokonta
12	Stramenopiles
9	Euglenozoa
4	Viridiplantae
11	Alveolata
2	Rhizaria
7	Others

Apusozoa, Cryptophyta, Diplomonads,
Haptophytes, Heterolobosea, Parabasalia

- Includes range from 1-exon to 40-exon genes

"This is preposterous R2! How could someone tarnish my good name so carelessly? Padmé always spoke highly of me, and now this... We must act swiftly to clear my name"



Most benchmark sequences are not curated and **could contain errors**

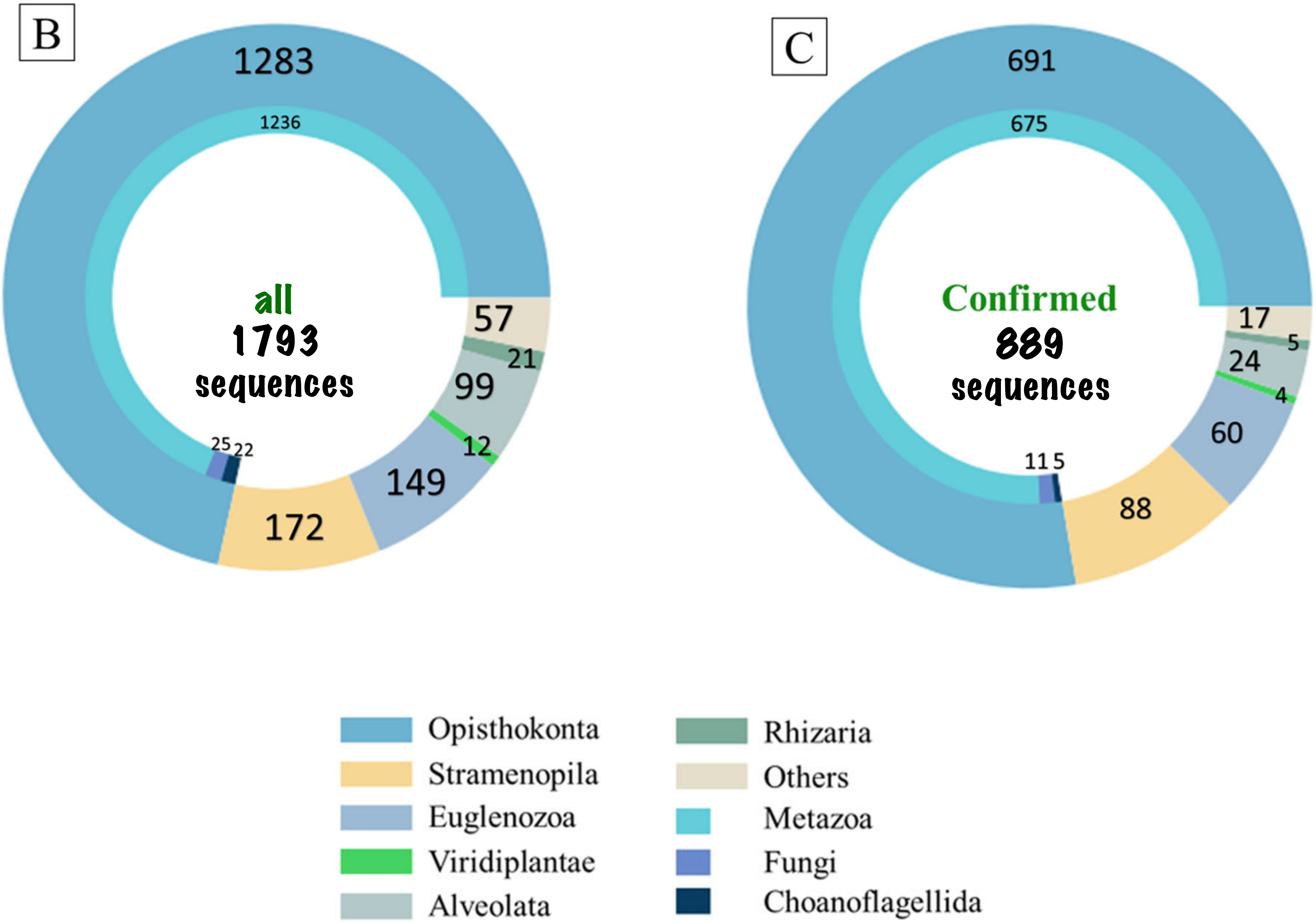
Remove those that are suspicious of containing errors because ...

A: lack N-terminus
B: dissimilar N-terminus
C: extra N-term segment

D: miss internal segment
E: dissimilar internal segment
F: extra internal segment

G: lack C-terminus
H: dissimilar C-terminus
I: extra C-term segment

1793 -> 889 "confirmed" sequences

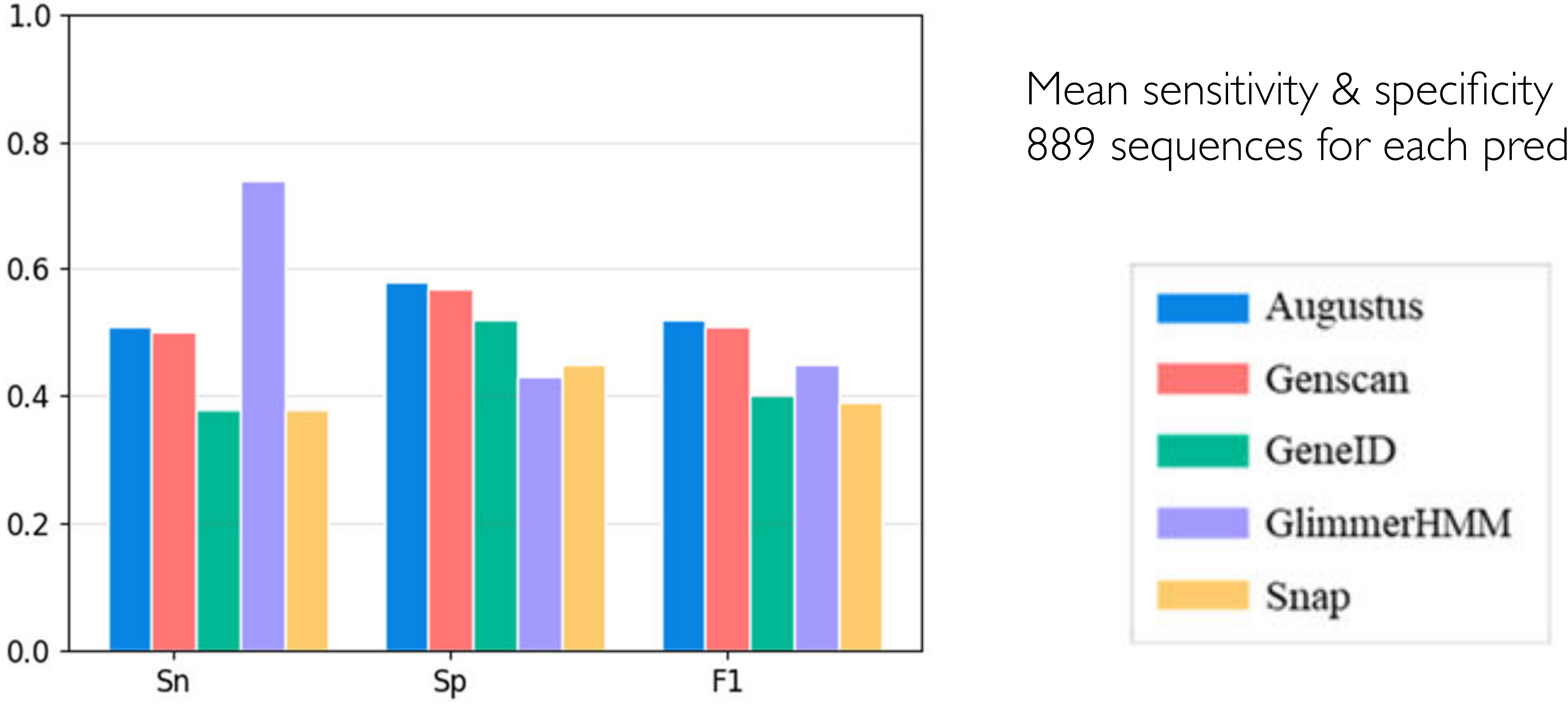

B
1283
1236
all
1793
sequences
57
21
99
12
149
172
25
22
C
691
675
Confirmed
889
sequences
17
5
24
4
60
88
11
5
Opisthokonta
Stramenopila
Euglenozoa
Viridiplantae
Alveolata
Rhizaria
Others
Metazoa
Fungi
Choanoflagellida

The Test

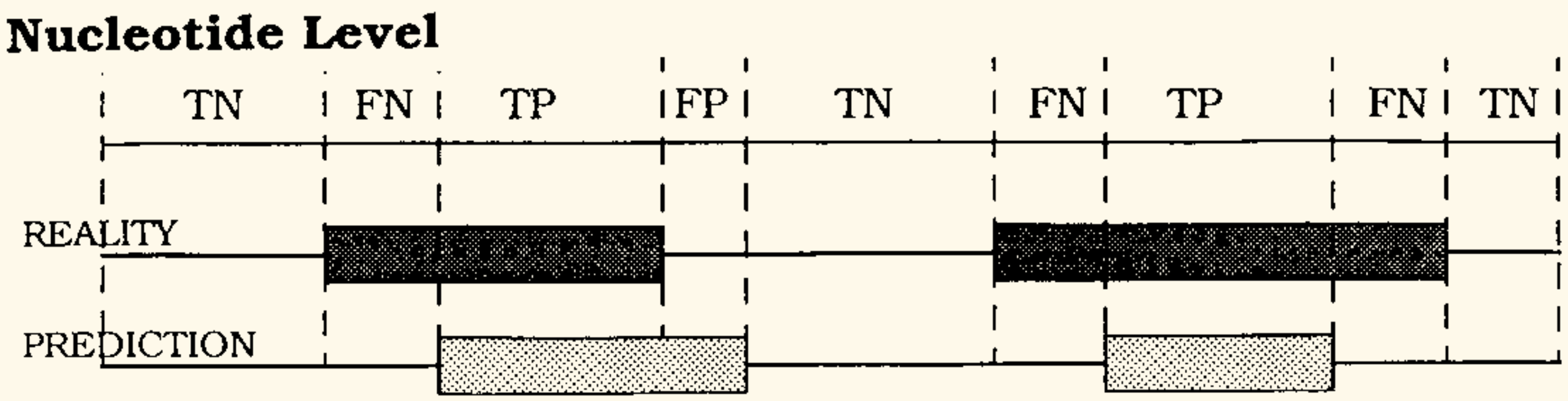
```
for each protein in 889 confirmed sequences:
    extract genomic region
    add corresponding 150 bp flanking regions
    for each predictor in [GENSCAN, GLIMMERHMM, GENEID, SNAP, AUGUSTUS]:
        choose most appropriate pre-trained model based on taxonomy
        predict gene
        compare predicted gene vs confirmed gene
```

joypixels
joypixels

Nucleotide level



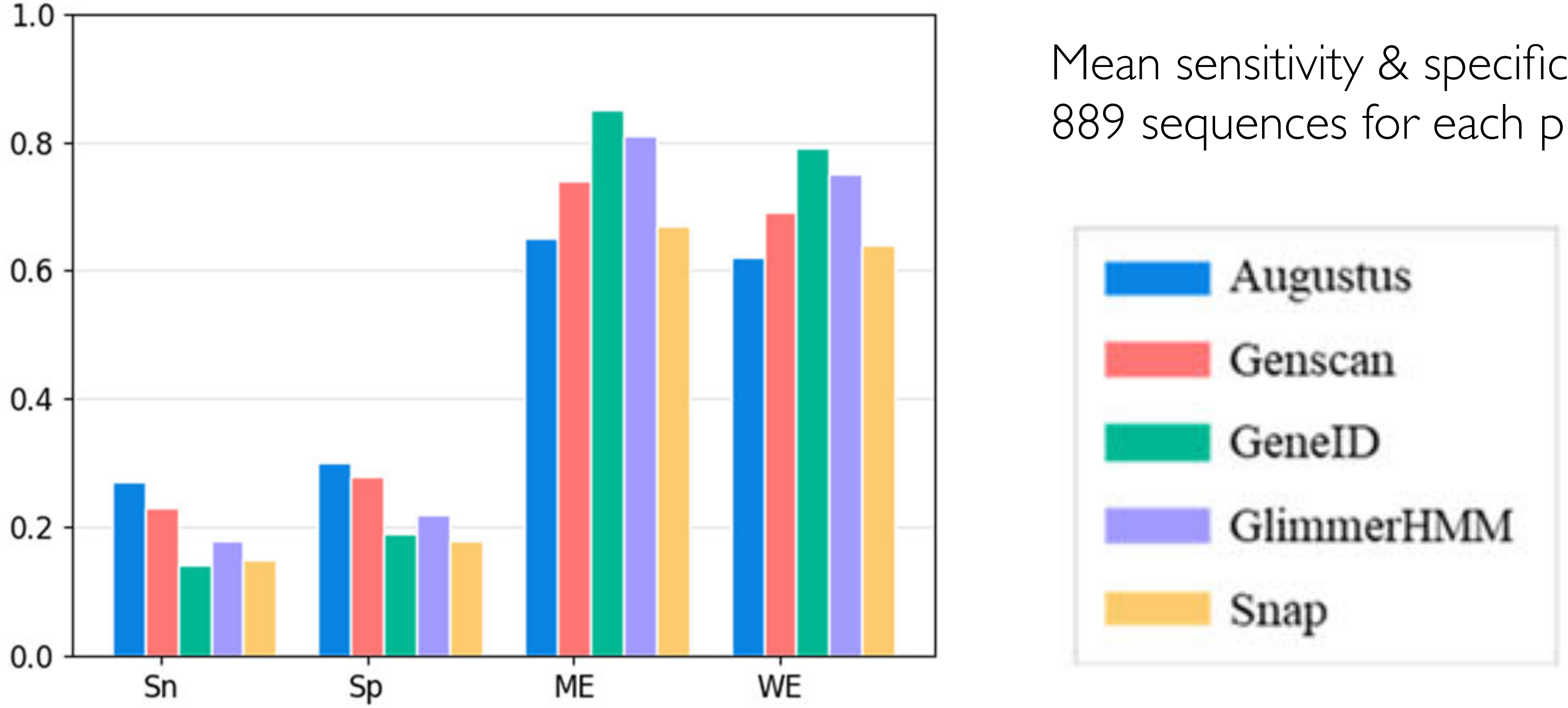
Mean sensitivity & specificity over all 889 sequences for each predictor



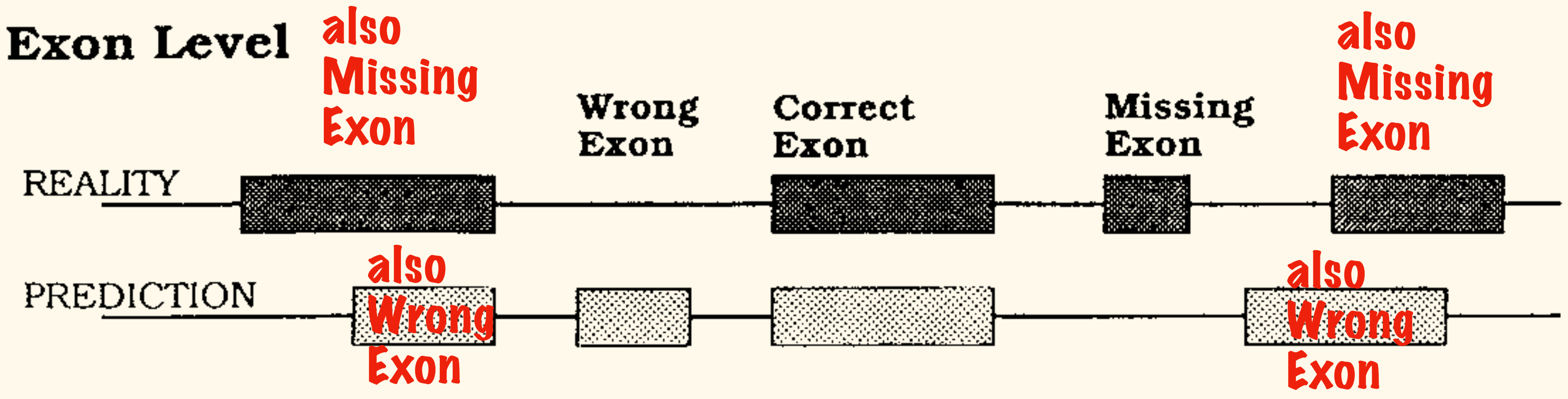
Sensitivity: What fraction of all nucleotides that are truly coding are predicted to be coding?

Specificity: What fraction of all nucleotides predicted to be coding are actually truly coding?

Exon level



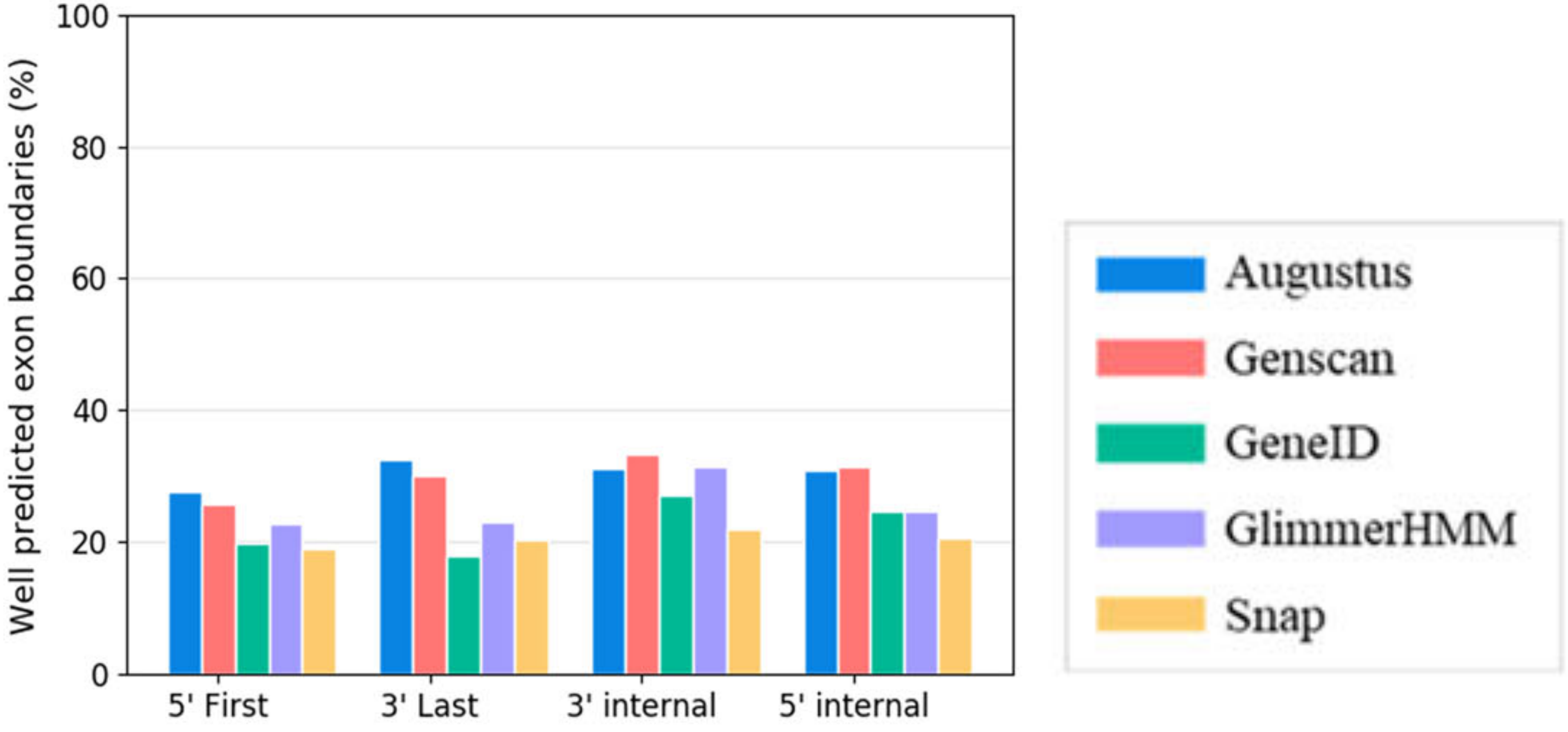
Mean sensitivity & specificity over all 889 sequences for each predictor



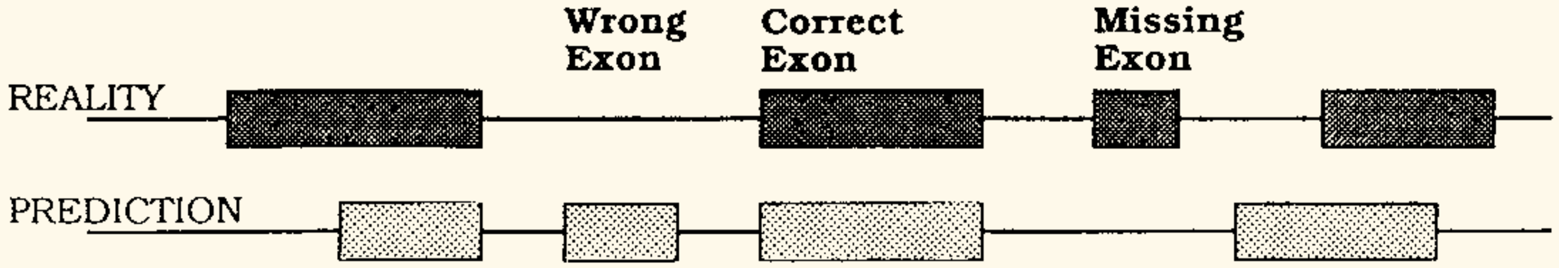
Sensitivity: What fraction of all true exons are predicted exactly?
Specificity: What fraction of all predicted exons match true exons exactly?

Missing exons: What fraction of all true exons is missed? *(**including true exons that overlap** with predicted exon)*
Wrong exons: What fraction of all predicted exons is entirely wrong? *(**including predicted exons that overlap** with a true exon)*

Exon level



Exon Level

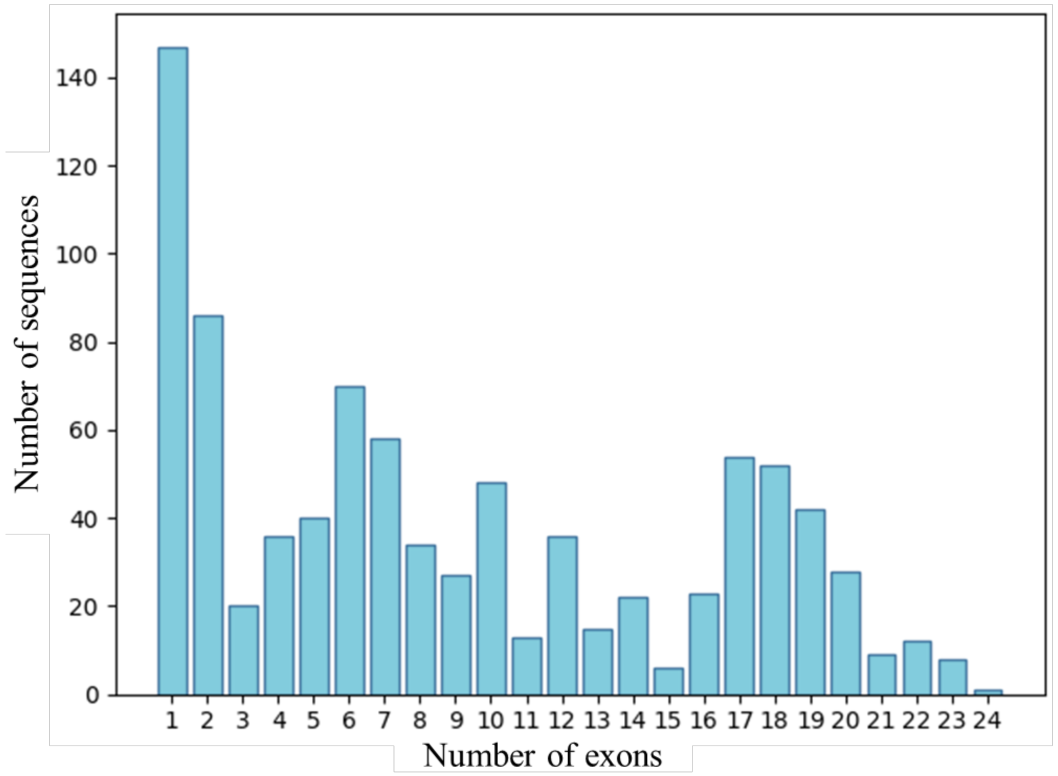


5' First: What fraction of all *exons predicted to be the first exon* match the 5' end of the true *first* exon?
3' Last: What fraction of all *exons predicted to be the last exon* match the 3' end of a true *last* exon?

5' Internal: What fraction of all *exons predicted to be internal* match the 5' end of the true *internal* exon?
3' Internal: What fraction of all *exons predicted to be internal* match the 3' end of the true *internal* exon?

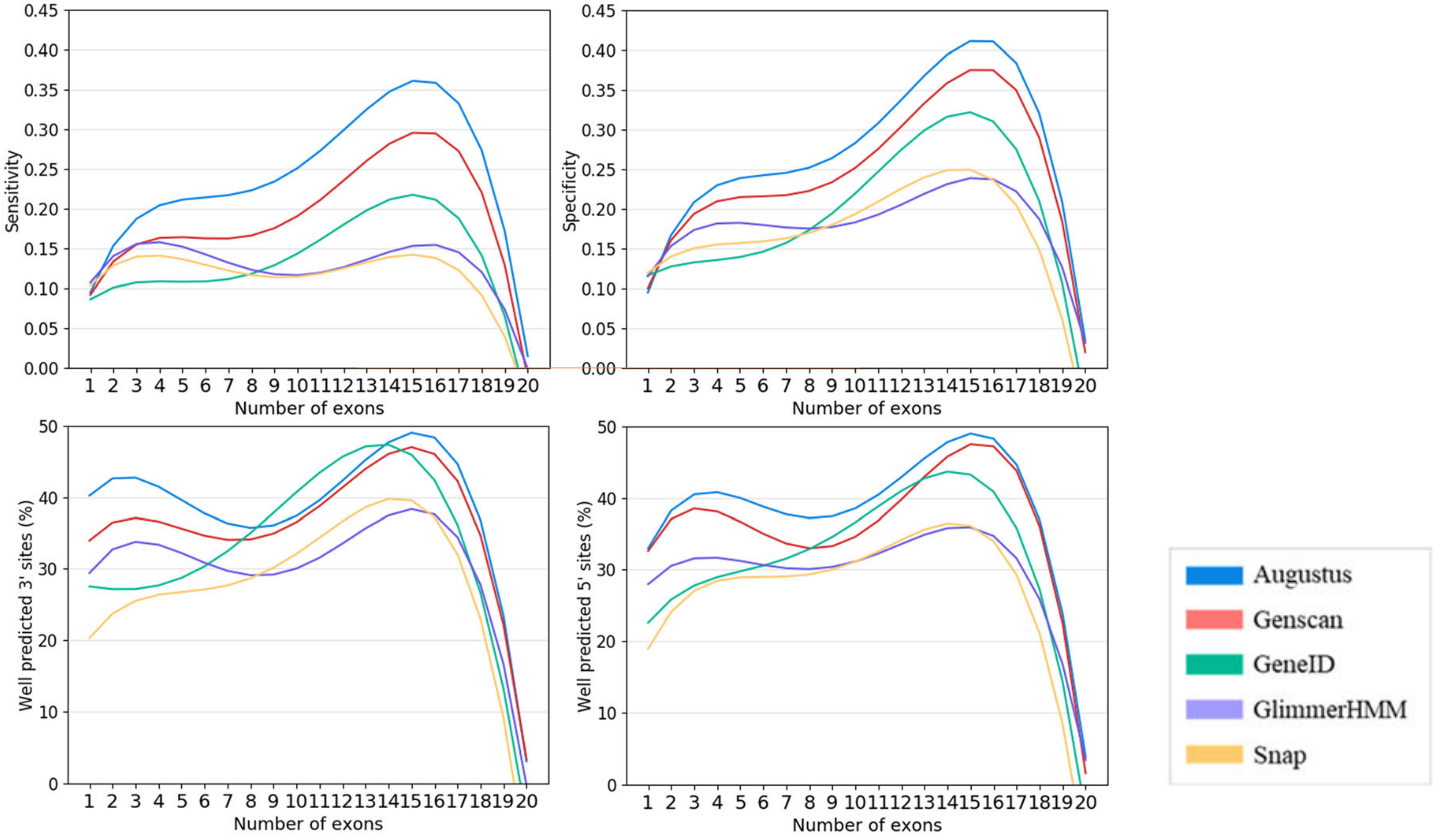
Does **exon count** influence gene prediction accuracy?

exon count in Confirmed sequences



A

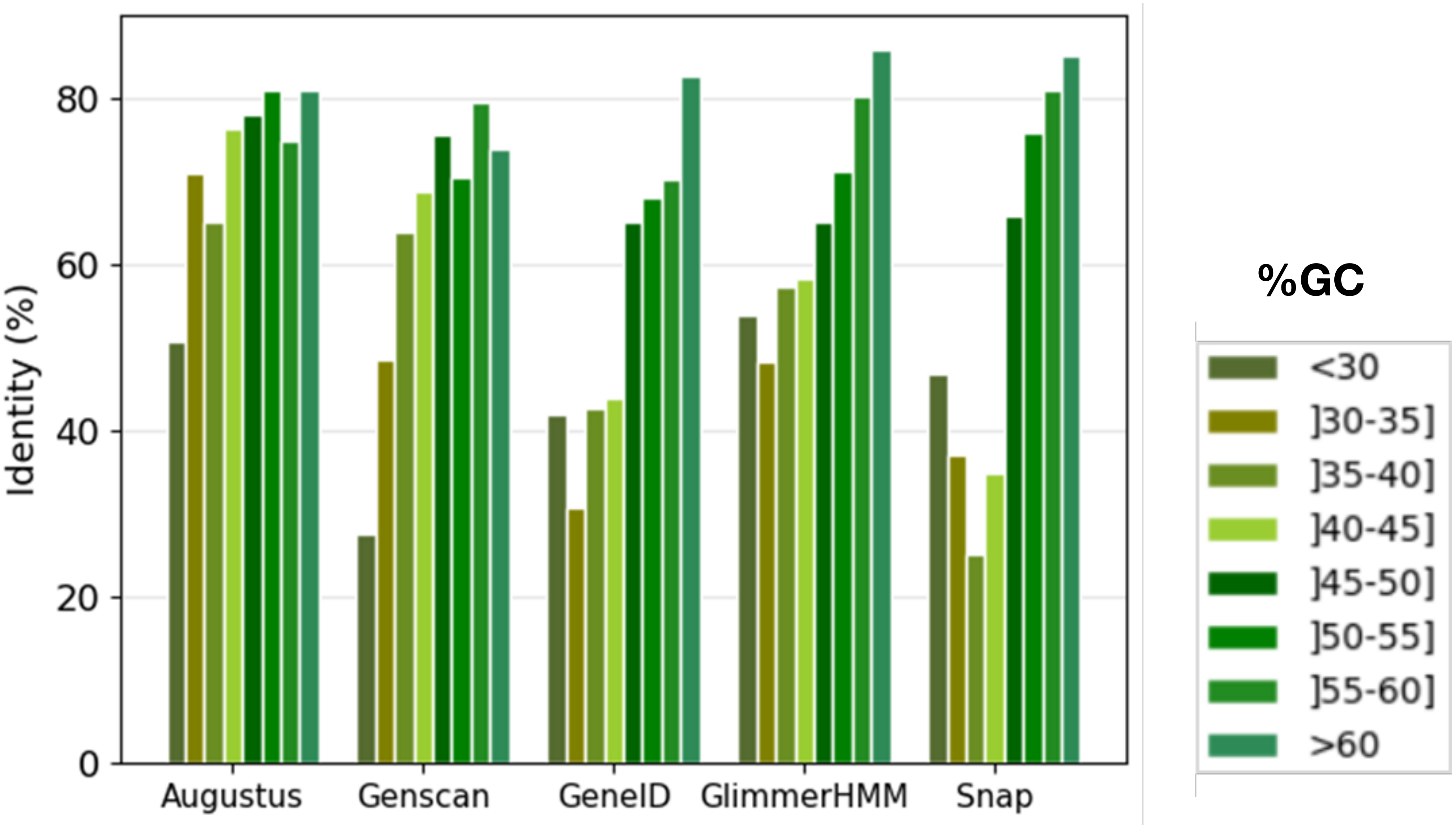
Exon level



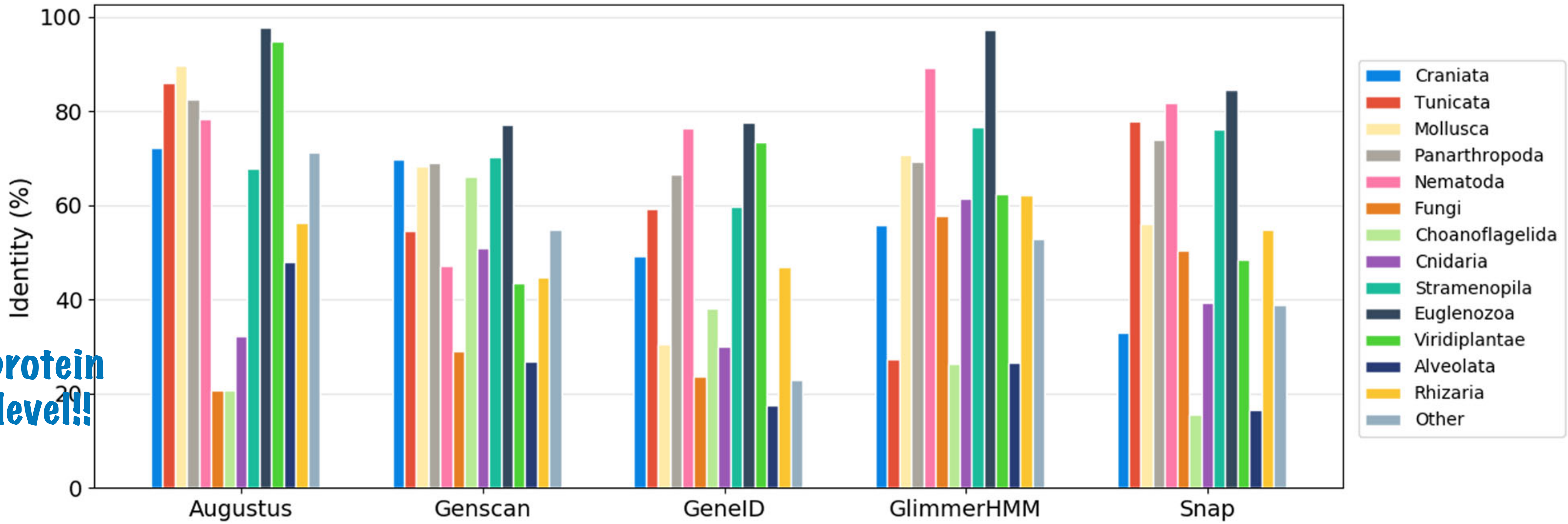
Sensitivity
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
Specificity
Well predicted 3' sites (%)
Well predicted 5' sites (%)
50
40
30
20
10
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Number of exons
Augustus
Genscan
GeneID
GlimmerHMM
Snap

Does **%GC** influence gene prediction accuracy?

protein level!!



Are gene predictors more accurate for certain lineages?



- AUGUSTUS ++ *Euglenozoa, Viridiplantae, Mollusca, Tunicata, Panarthropoda*
  - - *Fungi, Choanoflagelida**, *Cnidaria**
- GENSCAN - - *Fungi, Alveolata*
- GENEID - - *Alveolata, "Other", Fungi, Cnidaria**, *Choanoflagelida**
- GLIMMERHMM ++ *Euglenozoa, Nematoda*
  - - *Tunicata, Choanoflagelida**, *Alveolata*
- SNAP ++ *Euglenozoa, Nematoda*
  - - *Choanoflagelida**, *Alveolata, Craniata*

** only 5 Choanoflagelida and 6 Cnidaria sequences in dataset*

Conclusions

- **AUGUSTUS, GENSCAN, GENEID, GLIMMERHMM** and **SNAP** are still (*or at least in 2020*) unable to accurately predict gene structures for a large fraction of true genes

- They are relatively OKish at the **nucleotide level (~40-60% sensitivity and specificity)**, but outright awful at the **exon level (~15-25% sensitivity and specificity)**.

  In other words, identifying coding regions is alright-ish, but identifying the exact boundaries is still difficult

- They appear to be best at genes with 12-17ish exons

- They appear to **work very well (~80-95% accuracy at the protein level) for certain lineages (Euglenozoa, Nematodes)**, but are **particularly bad for others (Fungi, Choanoflagelida, Alveolata)**

  It is probably these extremely poor predictions for these lineages that pull the mean accuracy visible in most figures down so much

Thoughts

- The results seem a little too bad...
- Perhaps most striking is the difference in accuracy between different lineages

  The **available pre-trained models may be too taxonomically distant** from the genome being gene predicted (available models are typically few and biased towards metazoa, plants, fungi, pathogens)

  Also, the predictors may
  - model sequence **features that do not exist** in certain lineages or
  - **do not model sequence features** that do exist in certain lineages

  Fungi typically have gene dense genomes, with short introns, and more frequently have UTRs of adjacent genes overlap, either on the same strand or on opposite strands (hence CodingQuarry)

  For example, Augustus v1 only allowed GT-AG intron boundaries and assumes introns have a branch point