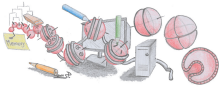
Gene Prediction and Gene Annotation

Sonja J. Prohaska

Computational EvoDevo, University of Leipzig
Santa Fe Institute, Santa Fe, NM, U.S.



WS 2026/27

Slides developed with the help of Claude (Anthropic), an AI assistant; content checked by S. J. Prohaska (2026).

Gene prediction and gene annotation

Today you will learn

- what a gene annotation is and how it is written down (GFF3/GTF)
- why finding genes in bacteria is easy and in eukaryotes is hard
- which information tells us where a gene is: *intrinsic* (the genome sequence) and *extrinsic* (transcripts, proteins)
- how a gene finder combines this information – and where its parameters come from
- which tool to use for which data
- how to judge an annotation: completeness, accuracy, function

From genome sequence to gene annotation

Input

- an assembled genome
  (lecture 1, 3–4)
- evidence: proteins of related species, transcripts (RNA-seq, Iso-seq)
- the rules of what a gene looks like (lecture 2): the gene grammar

Output: gene annotation

- annotated gene structures: coordinates of genes, transcripts, exons, CDS, UTRs
- links to the gene products and their functions

Gene prediction $\neq$ gene annotation $\neq$ genome annotation

Prediction is the computational step that proposes gene structures. **Gene annotation** is the curated result: predictions plus evidence, names and functions. **Genome annotation** covers any functional element or sequence feature – regulatory elements linked to their target genes, transposable elements, simple repeats – and may or may not include the genes.

A gene structure in a file: GFF3

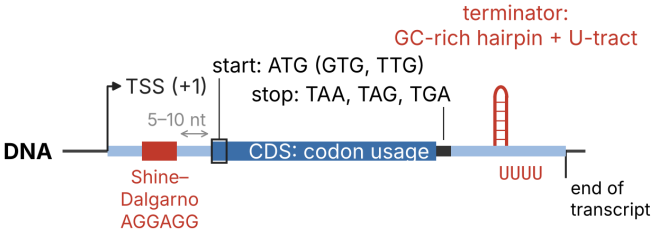
```
##gff-version 3
#seqid  source  type             start  end   score  strand  phase  attributes
chr1    .       gene             1001   4800  .      +       .      ID=gene1
chr1    .       mRNA             1001   4800  .      +       .      ID=tx1;Parent=gene1
chr1    .       exon             1001   1300  .      +       .      Parent=tx1
chr1    .       exon             2001   2400  .      +       .      Parent=tx1
chr1    .       exon             3001   3250  .      +       .      Parent=tx1
chr1    .       exon             4001   4800  .      +       .      Parent=tx1
chr1    .       five_prime_UTR   1001   1300  .      +       .      Parent=tx1
chr1    .       five_prime_UTR   2001   2150  .      +       .      Parent=tx1
chr1    .       CDS              2151   2400  .      +       0      ID=cds1;Parent=tx1
chr1    .       CDS              3001   3250  .      +       2      ID=cds1;Parent=tx1
chr1    .       CDS              4001   4151  .      +       1      ID=cds1;Parent=tx1
chr1    .       three_prime_UTR  4152   4800  .      +       .      Parent=tx1
```

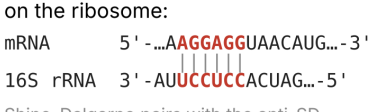
- the same 4-exon gene as in lecture 2: exon 1 non-coding, exons 2 and 4 partially coding; in English-language databases and papers a gene structure is often called a *gene model*
- coordinates are 1-based and inclusive; one line per feature

  `Parent` builds the hierarchy gene → mRNA → exon/`five_prime_UTR`/`three_prime_UTR`/CDS
- **phase**: bases to skip at the start of a CDS line to reach the next full codon

  e.g.: 250 nt = 83 codons + 1 nt ⇒ the next CDS line has phase 2
- the CDS here includes the stop codon (GFF3); in GTF it would end at 4148 (backup slide)

The easy case: what marks a bacterial gene





- dense (about 88% of *E. coli* is coding), no spliceosomal introns: a protein-coding gene is one ORF, often in operons
- **Shine–Dalgarno (SD)**: a consensus pairing with the 3' end of the ribosomal 16S rRNA; real sites match only 3–5 nt, strength varies (tunes translation), spacing 5–10 nt (optimum about 7–8)
- not universal: leaderless mRNAs (archaea, actinobacteria such as *Mycobacterium*); Bacteroidetes rarely use SD

How long is a random stop-free stretch?

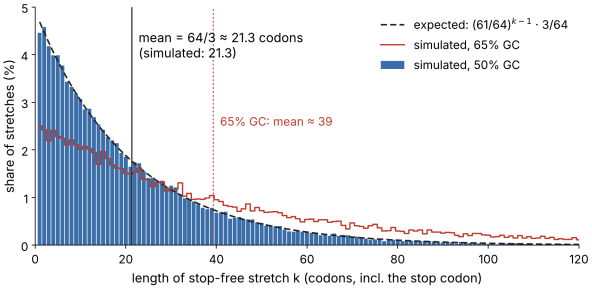
In a random sequence (50% GC), 3 of 64 codons are stop codons. Length L in codons, counted up to and including the next stop:

$$P(L = k) = \left(\frac{61}{64}\right)^{k-1} \frac{3}{64}$$

$$\text{mean } L = \frac{64}{3} \approx 21.3 \text{ codons}$$

$$P(L \geq 100) = \left(\frac{61}{64}\right)^{99} \approx 0.9\%$$

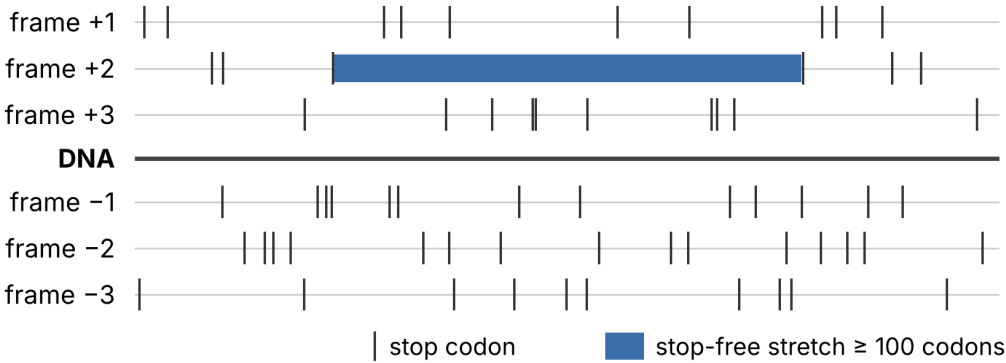
$\Rightarrow$ long stop-free stretches are rarely there by chance.



simulated: 10^6 random codons, `orf_length_sim.pl`

But: stop codons are AT-rich, so GC-rich genomes have fewer of them: at 65% GC the mean is about 39 codons.

ORFs in six reading frames



Three frames per strand. Between two stop codons lies a stop-free stretch; an ORF begins at an ATG inside it and ends at the next stop. One frame carries a long stretch: a gene candidate.

Which start? And the tools

- a long stop-free stretch often contains several in-frame ATGs: which one is the start?
- **ribosome binding site**: an SD-like motif 5–10 nt upstream supports a candidate start
- alternative start codons: ATG (about 80%), GTG, TTG
- **codon usage**: real genes use codons with species-specific frequencies

Tools

- `Prodigal`, `GeneMarkS-2`: learn their statistics from the genome itself (no known genes needed)
- pipelines: `Bakta`, NCBI `PGAP` (genes, rRNA, tRNA, functional annotation)

Prokaryotic gene prediction is largely solved; the start codon is the most frequent error.

Why eukaryotes are hard

- genes are sparse: about 1.2% of the human genome is coding sequence
- introns split the CDS; introns are often much longer than exons
- the signals are short and weak: `GT` occurs about every 16 nt by chance, yet only a tiny fraction are splice sites
- alternative splicing: several transcripts per gene
- about half of the genome are repeats, many with their own ORFs (transposons)

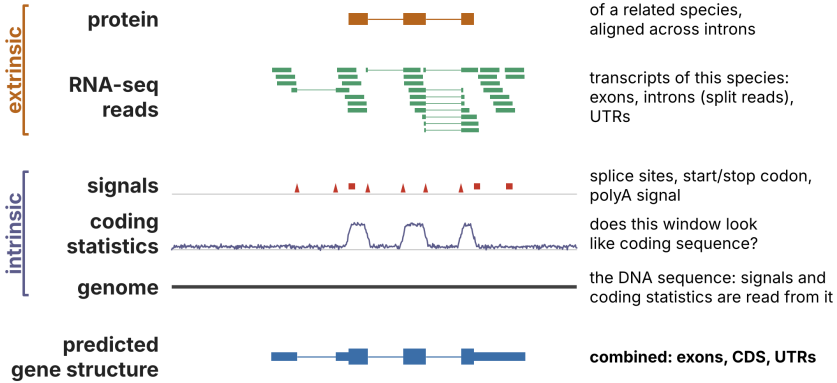
Idea

Combine many weak signals with the statistics of coding sequence – and the rules of how they must follow each other; add transcripts and proteins where they are available.

First: mask the repeats

- about half of a mammalian genome are repeats; transposable elements carry their own ORFs (e.g. reverse transcriptase)
- unmasked, they are predicted as thousands of false genes
- build a species repeat library (`RepeatModeler`), then mask (`RepeatMasker`)
- **soft-masking**: repeats in lowercase letters – gene finders avoid them but can still place exons there if evidence demands it

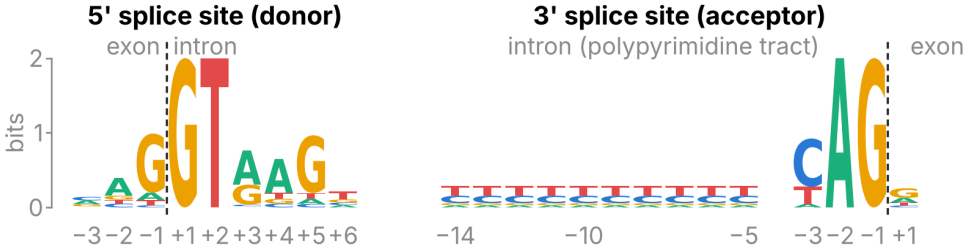
What tells us where a gene is?



intrinsic: read from the genome sequence itself – available for every gene, but the signals are weak.

extrinsic: brought in from outside and mapped onto the genome – strong, but only for genes that are expressed (RNA-seq) or conserved (proteins).

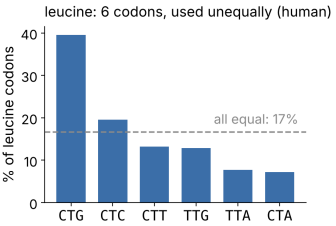
Signal sensors: what the spliceosome reads

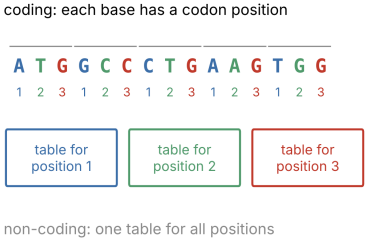


schematic, approximate human frequencies

- sequence logo: height of a column = information content (bits), letters scaled by frequency
- only `GT...AG` is (almost) invariant – the rest is a preference, scored position by position with log-odds (as in a PSSM)
- further signals: start codon in Kozak context (`gccRccAUGG`), stop codons, polyA signal (`AAUAAA`)

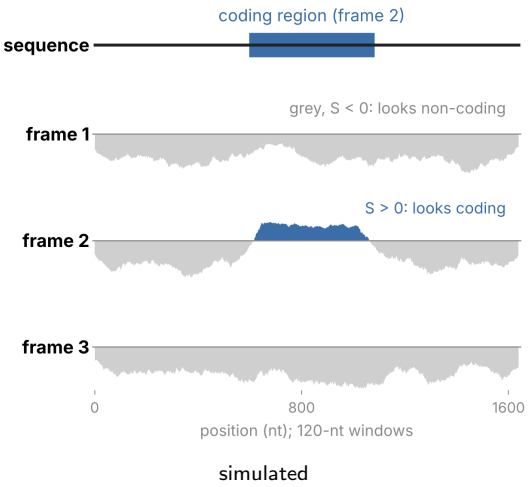
Where do codon statistics come from?





- redundant code: 61 sense codons for 20 amino acids; amino acids are used unequally (human: Leu about 10%, Trp about 1%)
- synonymous codons are used unequally (**codon bias**): tRNA supply, GC content, selection for efficient translation
- the **reading frame**: statistics depend on the codon position (third positions vary most); no stop codon in frame
- counted in known coding sequence of the species and compared with non-coding sequence

Content sensors: does this window look like coding sequence?



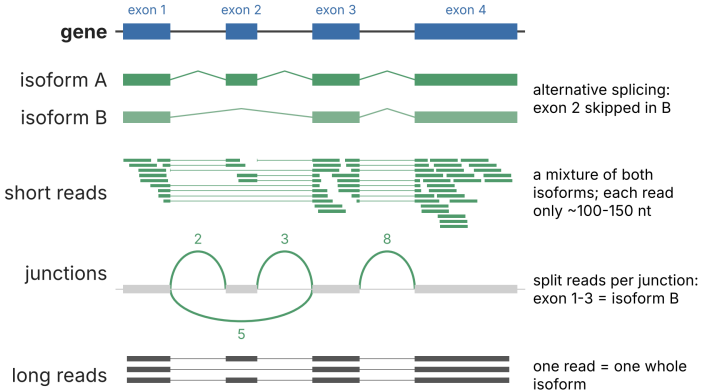
Two models of the same window x:

- **coding**: three tables, one per codon position
- **non-coding**: one table

$$S(x) = \log \frac{P(x \mid \text{coding})}{P(x \mid \text{non-coding})}$$

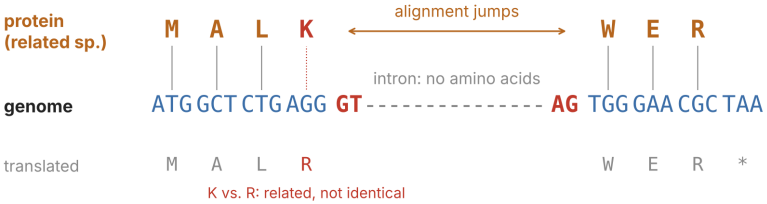
- computed in all three frames (and on both strands): only the frame of the gene scores $S > 0$
- each table is a Markov chain of order 5 (a base given the 5 bases before), learned from known genes – see *training*

Transcript evidence: RNA-seq and alternative splicing



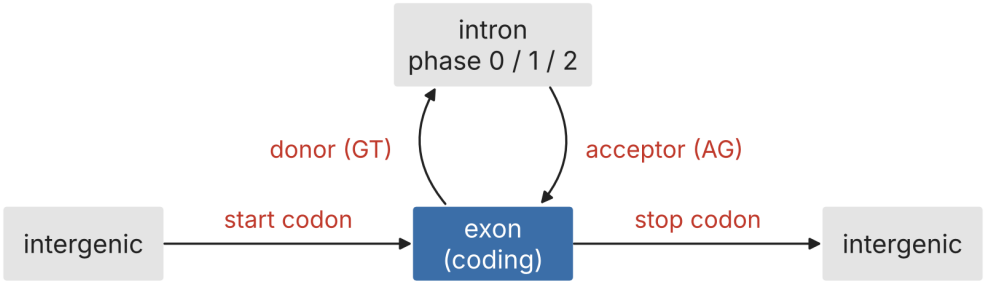
One gene, several transcripts: the spliceosome can skip exons, use alternative splice sites or retain introns. Short reads come from a mixture of isoforms; split reads mark exon–exon junctions exactly, and assembly (`StringTie`) reconstructs the isoforms. Long reads (Iso-Seq, nanopore) show whole isoforms, TSS to polyA.

Protein evidence: alignment across introns



- a protein of a related species is compared with the *translated* genome: amino acids change more slowly than DNA
- the alignment must jump over introns at `GT`...`AG` and keep the reading frame: **spliced alignment** (`miniprot`, `ProSplign`, `GeMoMa`)
- shows the CDS only (no UTRs); finds conserved genes, misses young, species-specific ones
- shortcut for a very close relative: transfer its whole annotation (`Liftoff`)

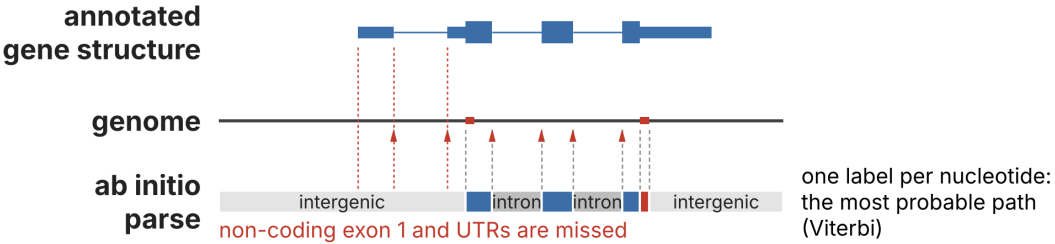
A gene as a sentence in a grammar



the intron phase (0, 1, 2) remembers where in the codon the exon was split

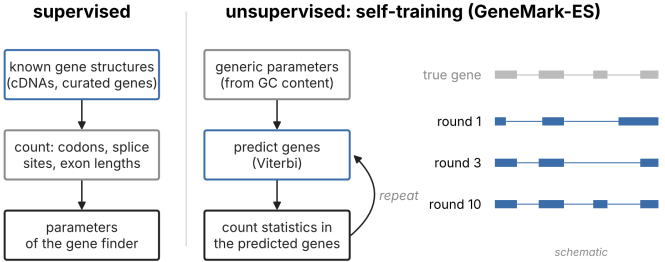
The rules of gene structure become states and transitions of a hidden Markov model (HMM): each state emits sequence (scored by content sensors), each transition needs its signal (signal sensors).

Decoding: the most probable parse



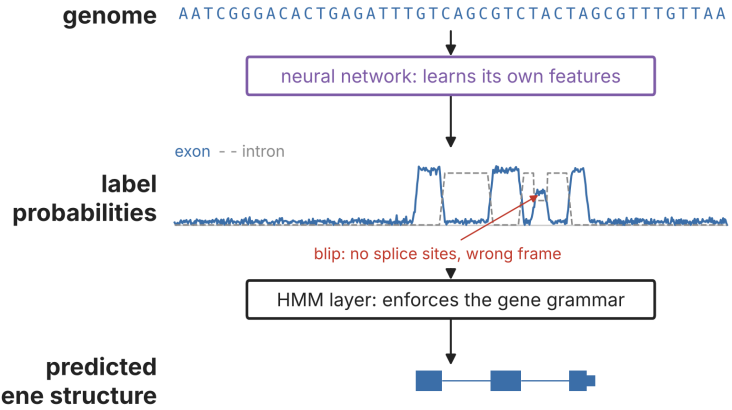
- the Viterbi algorithm (dynamic programming) finds the most probable label for every nucleotide
- `GENSCAN` (Burge & Karlin 1997), `AUGUSTUS` (Stanke & Waack 2003), `GeneMark-ES` (Lomsadze et al. 2005)
- typical weakness: UTRs and non-coding exons have no coding signal $\Rightarrow$ missed

Where do the parameters come from?



- **parameters** = the HMM's emission probabilities (codon tables, splice-site tables) and transition probabilities (incl. lengths) – thousands of numbers, different in every species
- **supervised**: count them in about 1000 known gene structures of the species – a new genome has none
- **self-training**: the gene grammar is fixed by design, only the numbers are learned – start rough, predict, recount in the predictions, repeat. Works because many first-round exons are already right; best in compact genomes (fungi, protists)

Deep learning: Tiberius



sequence only, no evidence

trained once on many annotated genomes of a clade

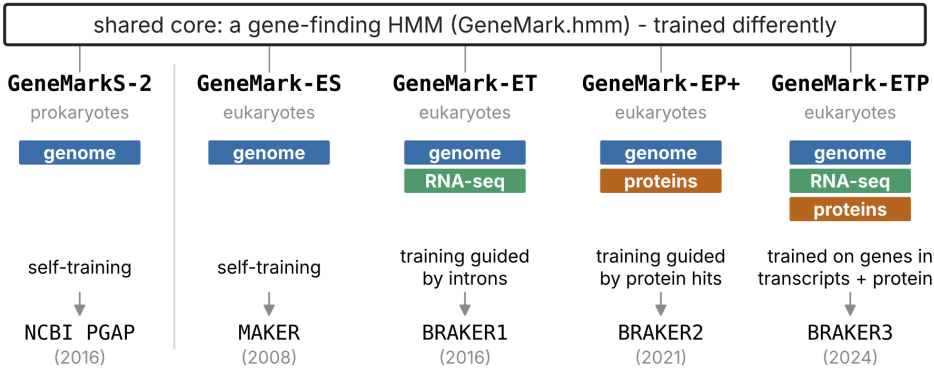
for every position: how likely exon, intron, intergenic?

start ... GT-AG ... stop, reading frame across introns

the most probable valid parse: the blip is dropped

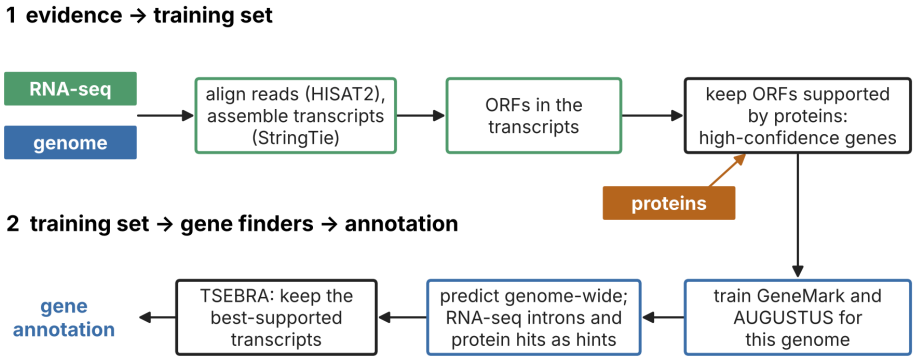
The network replaces hand-made signal and content sensors; the HMM layer keeps the gene grammar. No evidence needed: on the human genome as accurate as BRAKER3 with RNA-seq and proteins.

One core, many tools: the GeneMark family



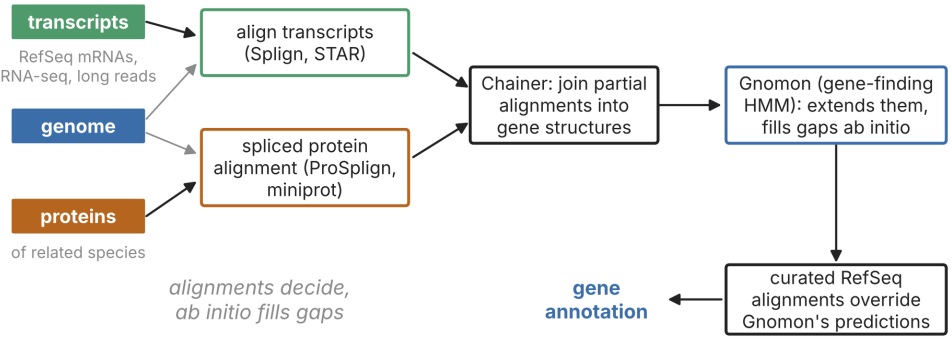
All members share one gene-finding HMM. They differ in what trains its parameters: the genome alone (self-training) or, in addition, introns from RNA-seq and/or hits of related proteins. S = self-training, E = eukaryotic, T = transcripts, P = proteins.

BRAKER3: evidence trains and guides the gene finders



Evidence covers only parts of genes and only expressed or conserved genes; the trained gene finders fill in the rest. One program, three modes: RNA-seq only (BRAKER1), proteins only (BRAKER2), both (BRAKER3).

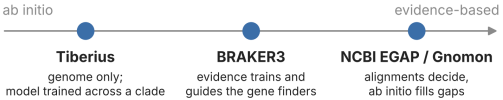
NCBI EGAP: alignments decide, *ab initio* fills gaps



Transcripts of the species and proteins of related species are aligned first; `Gnomon` only extends the gene structures and fills the gaps. Compare `BRAKER3`: there, the evidence trains the gene finders.

Homology shortcut: `Liftoff` (annotation of a close assembly)

Which tool for which data?



available data	typical choice	how it works	limitation
prokaryotic genome	`Prodigal`, `PGAP`, `Bakta`	ORFs + statistics learned from the genome (+ protein families)	start codons
eukaryotic genome only; a clade model exists	`Tiberius`, `Helixer`	network trained on the clade + HMM layer	genomes far from the training clades
eukaryotic genome only; no clade model	`GeneMark-ES`	self-training	large, repeat-rich genomes
+ proteins of related species	`BRAKER2`	protein hits guide the training	genes without homologs
+ RNA-seq + proteins	`BRAKER3`	evidence → training set → gene finders	unsupported genes less reliable
annotated close relative	`Liftoff`, `miniprot`	transfer / align its genes	species-specific genes
long reads (Iso-Seq, nanopore)	the transcripts	whole isoforms, incl. UTRs	only expressed genes

Mask repeats first. Typical choices as of 2026 – benchmarks change quickly.

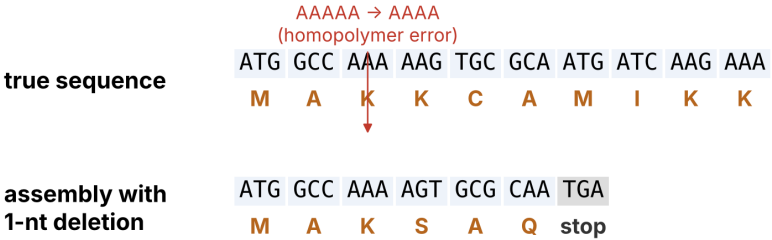
And what about ncRNAs?

No ORF, no codon statistics: the signals are **secondary structure**, conservation and expression.

ncRNA	**what identifies it**	**tools**
tRNAs	cloverleaf structure, anticodon	`tRNAscan-SE`
rRNAs	highly conserved sequence and structure	`barrnap`, `Infernal` + `Rfam`
snoRNAs	box C/D or box H/ACA motifs, guide sequence pairing with its target RNA	`snoReport`, `snoscan`, `Infernal` + `Rfam`
miRNAs	hairpin precursor; mature miRNA in small-RNA-seq	`miRDeep2`, `Infernal` + `Rfam`
lncRNAs	transcribed (RNA-seq), long, no ORF; little sequence conservation	RNA-seq assembly + coding-potential filter (`CPC2`); `RNAz` for conserved structure

`Infernal` + `Rfam`: covariance models – profiles of sequence *and* base pairing for known RNA families. `RNAz`: conserved, thermodynamically stable secondary structure in multiple alignments – also finds new ncRNAs.

When sequencing errors break genes



- long-read assemblies carry indel errors, mostly in homopolymers (lectures 3–4)
- one missing base shifts the reading frame ⇒ premature stop codon
- the gene finder splits the gene or calls it a pseudogene; polishing with accurate reads fixes it
- **pseudogene**: a gene copy that has lost its function – frameshifts and premature stops, or, for a processed pseudogene (a reverse-transcribed mRNA), no introns and no promoter

How complete is an annotation? BUSCO

- **BUSCO**: Benchmarking Universal Single-Copy Orthologs
- gene sets that are present in a single copy in nearly all species of a lineage (e.g. vertebrates, insects, fungi)
- check how many of them are found in the annotation (or the assembly):

C complete (S single copy, D duplicated)
F fragmented
M missing

Many duplicated BUSCOs hint at assembly errors (haplotypes kept twice); many fragmented ones at broken gene structures (e.g. frameshifts).

How complete is an annotation? BUSCO

- **BUSCO**: Benchmarking Universal Single-Copy Orthologs
- gene sets that are present in a single copy in nearly all species of a lineage (e.g. vertebrates, insects, fungi)
- check how many of them are found in the annotation (or the assembly):

97.2%	C	complete (95.8% S single copy, 1.4% D duplicated)
1.3%	F	fragmented
1.5%	M	missing

`C:97.2% [S:95.8%, D:1.4%], F:1.3%, M:1.5%, n:13780`

example: proteins of an annotation, n = number of BUSCO genes searched

Many duplicated BUSCOs hint at assembly errors (haplotypes kept twice); many fragmented ones at broken gene structures (e.g. frameshifts).

How accurate is an annotation?



$$\text{sensitivity Sn} = \frac{TP}{TP + FN} \qquad \text{precision Pr} = \frac{TP}{TP + FP}$$

- TP = true positive (in reference and prediction), FN = false negative (in the reference only, missed), FP = false positive (in the prediction only)
- at the exon level, an exon counts only if *both* borders are exact
- the same prediction looks good at nucleotide level and poor at exon (or gene) level

Functional annotation

- protein domains and families: `InterProScan` (Pfam, PROSITE, ...)
- orthology-based transfer of names and functions: `eggNOG-mapper`
- Gene Ontology (GO) terms: molecular function, biological process, cellular component

Transitive annotation

Function is mostly transferred from similar sequences – which got their function the same way. One wrong annotation spreads through the databases.

Predicted, transcribed, functional – not the same

- a **predicted** gene is a hypothesis about a gene structure
- RNA-seq reads on it show that it is **transcribed**
- conservation hints at a **function** (selection keeps it)

But

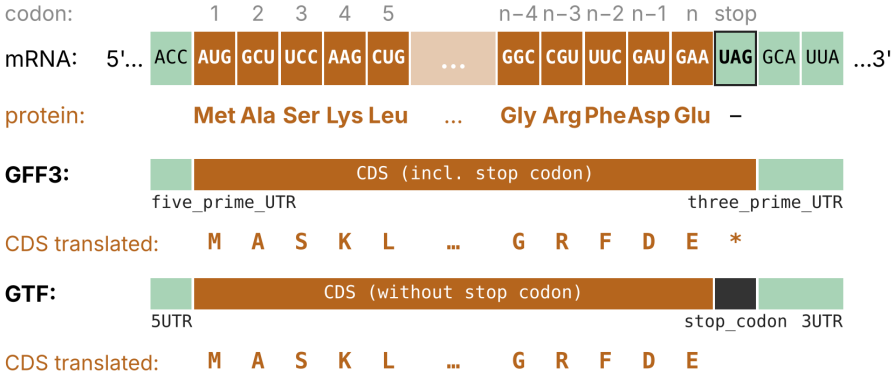
Only experiments – knockout, knock-down, reporter and protein assays – show what a gene does, and whether the predicted structure is right.

Summary

- a gene annotation = gene structures as coordinates (GFF3/GTF) + links to products and functions
- prokaryotes: ORFs, start codon and RBS – largely solved
- eukaryotes: the intrinsic information (signals, codon statistics) is weak; a gene grammar (HMM, Viterbi) combines it
- the parameters are learned: from known genes, by self-training, or across a clade (deep learning)
- extrinsic evidence (transcripts, proteins) fixes what *ab initio* misses – and trains the gene finders
- choose the tool by your data; mask repeats first; ncRNA genes need their own tools
- judge completeness (BUSCO), accuracy (Sn, Pr) – and remember: predicted $\neq$ functional

These slides were made with an AI assistant (Claude). Every figure, number and tool description was checked by hand: do the same with AI answers.

Backup: start codon, stop codon and the CDS



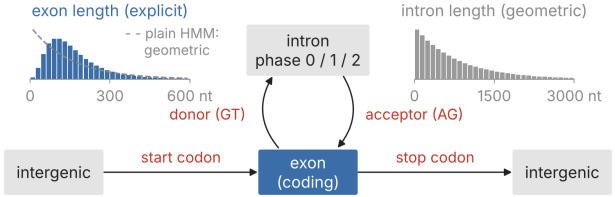
- **GFF3** (Sequence Ontology): CDS includes the stop codon ⇒ its translation ends with "*". **GTF** (GTF2.2): CDS excludes it; the stop codon is a separate `stop_codon` feature (in some files it is counted to the 3'-UTR instead) ⇒ no "*".
- CDS lengths differ by 3 nt between the formats – check the convention before computing.

Backup: other pipelines

Pipeline	Approach
`MAKER`	aligns transcripts and proteins, runs gene finders, scores every gene structure by its evidence support (AED)
Ensembl / GENCODE	automatic evidence-based annotation; for human and mouse, manual curation (HAVANA)

AED (annotation edit distance) $= 1 - \frac{SN+SP}{2}$, nucleotide by nucleotide: SN = share of the evidence covered by the gene structure, SP = share of the gene structure supported by evidence. 0 = perfect agreement with the evidence, 1 = no support.

Backup: generalised HMM – why explicit lengths matter



- **standard HMM**: one state emits one base; staying in a state is a self-loop with probability $p \Rightarrow$ lengths are geometric, $P(L = d) = p^{d-1}(1 - p)$: the shortest length is always the most likely
- exon lengths peak around 100–200 nt: a geometric model favours tiny spurious exons
- **generalised HMM** (`GENSCAN`, `AUGUSTUS`, `GeneMark.hmm`): a state first draws a segment length d from an explicit distribution, then emits d bases with its content model; signals are scored at the borders
- price: Viterbi must try every segment length – time grows with genome length $\times$ maximum segment length; introns and intergenic regions keep geometric lengths

Backup: two kinds of HMM in gene annotation

Gene-finding HMM

`GENSCAN`, `AUGUSTUS`, `GeneMark`, `Gnomon`
(gene grammar, decoding)

- **states**: the parts of a gene structure – intergenic, exon (three phases), intron, start, stop, splice sites
- **input**: genomic DNA, nucleotide by nucleotide
- **output**: Viterbi labels every position ⇒ gene structures; the HMM *is* the gene finder

Profile HMM

`Pfam`, `TIGRFAM`, `HMMER`, `BUSCO`

- **states**: match, insert, delete for each column of a multiple alignment of one protein family
- **input**: a protein or a translated ORF (genome: six-frame translation)
- **output**: a score for how well it fits the family: homology *evidence*: a family member is here, roughly – but no exon-intron structure (no intron states)

Prokaryotes: a profile-HMM hit in an ORF is nearly the whole gene (NCBI PGAP: ORFs vs. protein HMMs, `GeneMarkS-2` for starts and regions without hits). Eukaryotes: hits fall apart into exon-sized pieces; the exact structure needs a splicing model (`GeneWise`) or spliced alignment (`miniprot`).