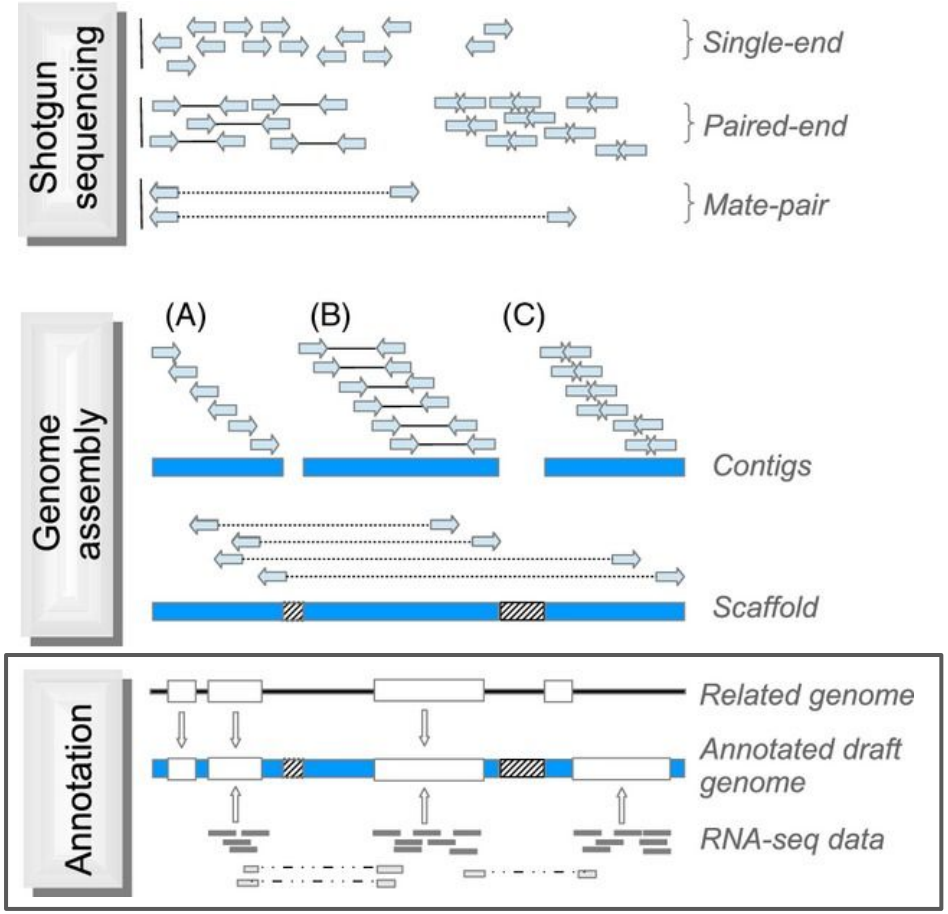
Genome assembly & annotation

- Genome annotation
  - Hidden Markov Models

Why sequence the genome?

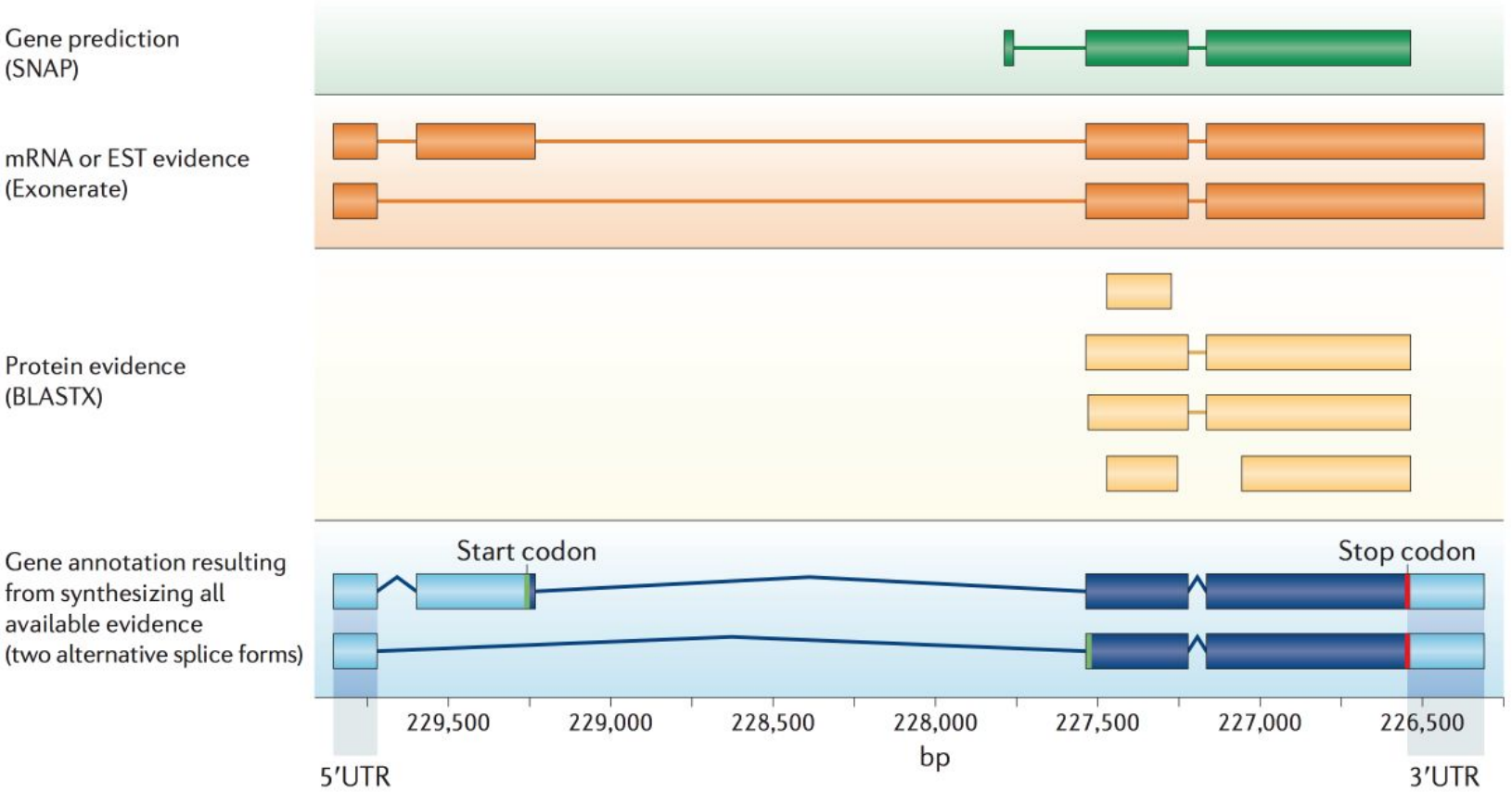
- Determine the "complete" sequence of a haploid genome.
  - Previously "snippets" of the genome were available.
- Identify the sequence and location of every gene.
- Use as a "map" to then track the location and frequency of genetic variation.
- Unravel the genetic architecture of inherited and somatic traits/diseases.
- To understand genome and species evolution.

Genome assembly & annotation – Overview



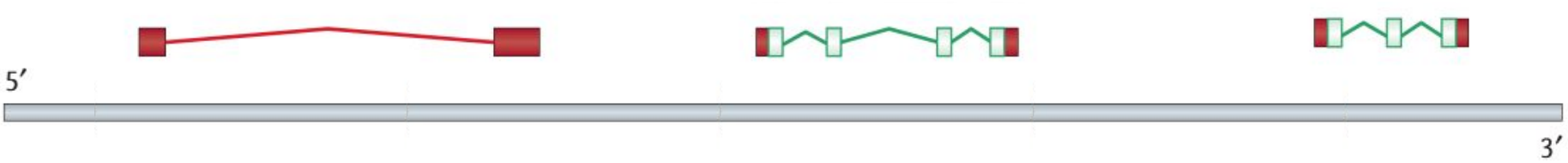
Ekblom & Wolf (2014)

Genome annotation



Yandell & Ence (2012)

Genome annotation – Gene finding

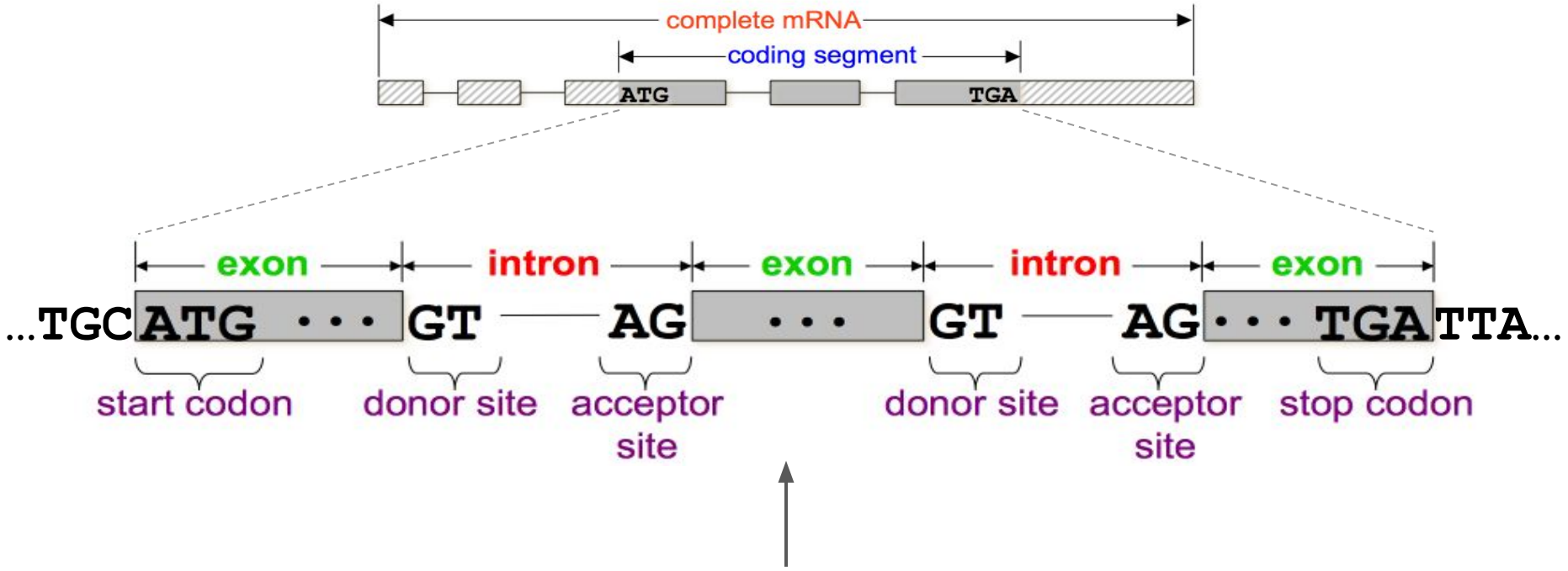


Problem: **Given** the entire genome, **label** the nucleotides as exons, introns, UTRs, or intergenic.

Requirements:

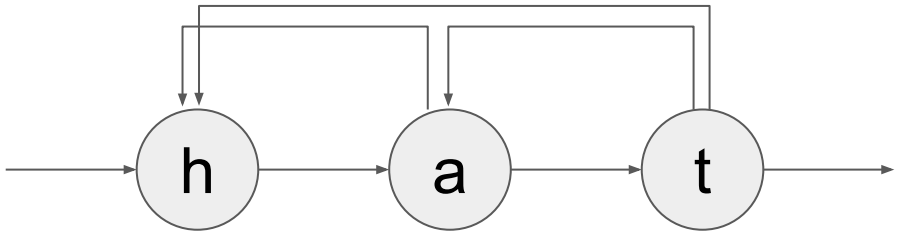
- Combine splice-site consensus, codon bias, exon/intron length preferences, and open reading frame analysis into one scoring system.
- Provide results that can be interpreted probabilistically. (How confident are we that the best scoring answer is correct?)
- Should be extensible, capable of modeling additional genomic features like translational initiation consensus, alternative splicing, and a polyadenylation signal.

Genome annotation – Gene finding



...TGCATGCGTAGGTAAGTCAGTTGCGACAGTGAGGAGCTCTTGATTA...

Markov models



Current state depends only on previous state and transition probability.

- $Pr(\text{‘at’}) = Pr(\text{‘a’}) \cdot Pr(\text{‘t’} | \text{‘a’})$
- $Pr(x_1 \ldots x_n) = Pr(x_1) \prod Pr(x_i | x_{i-1})$

Hidden Markov Models (HMMs)

HMM for probabilistic sequence classification:

- HMMs are a way of relating a **sequence of observations** to a **sequence of hidden states** (or classes) that explain the observations.

An HMM is a **full probabilistic model**:

- The model parameters and the overall sequence 'scores' are all probabilities.
    - Bayesian probability theory can be used to manipulate these numbers in standard, powerful ways, including optimizing parameters and interpreting the significance of scores).
    - Biological data is noisy.
    - Quantify uncertainty & degrees of belief.
    - Learn things we already don't know.

Brent (2007)

Hidden Markov Models (HMMs)

HMM for probabilistic sequence classification:

- HMMs are a way of relating a **sequence of observations** to a **sequence of hidden states** (or classes) that explain the observations.

- The process of discovering the sequence of hidden states, given the sequence of observations, is known as **decoding** or inference. The **Viterbi algorithm** is commonly used for decoding.

- The **parameters** of an HMM are:
    - the transition probability matrix and
    - the observation likelihood (emission probability) matrix
    - Both can be trained with the Baum-Welch or forward-backward algorithm.

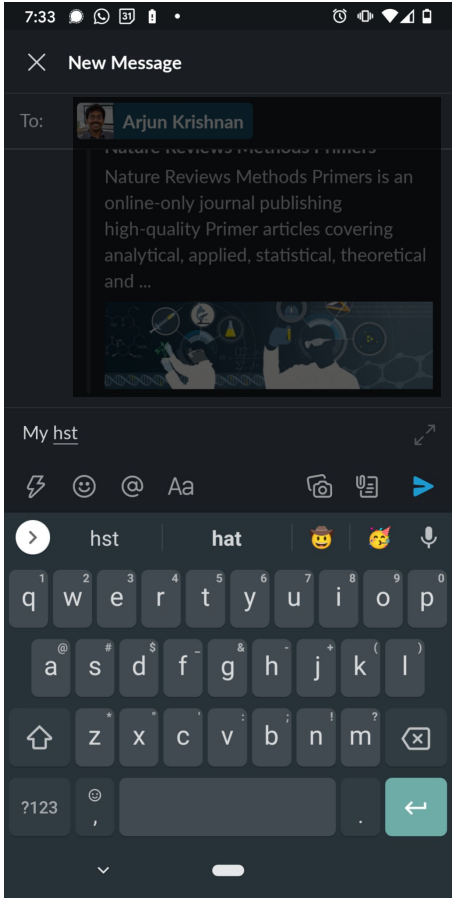
Brent (2007)

Hidden Markov Models (HMMs)

When composing a message on your phone, say you type a word:

`hst`

Observed



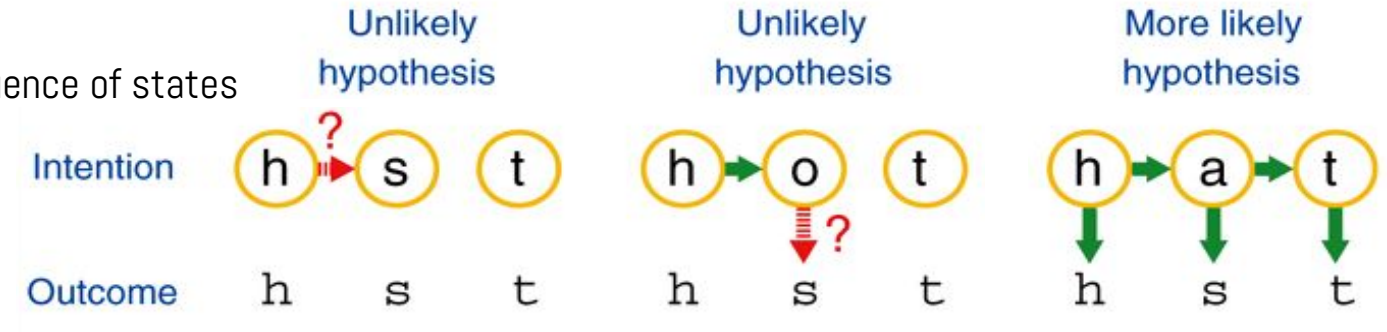
How does the phone "infer" that you probably mean:

`hat`

Hidden

Hidden Markov Models (HMMs)

Hidden sequence of states



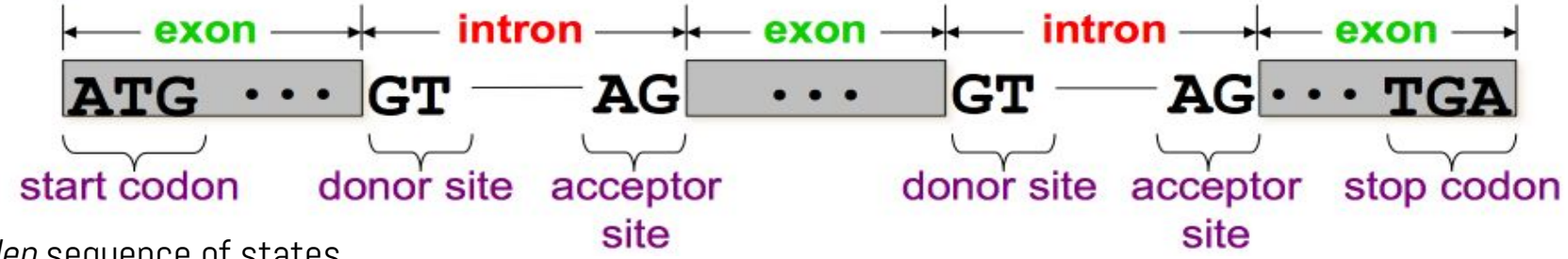
Observed sequence

Transition probabilities model letter sequences in correctly spelled words.

Emission probabilities model the probability of each possible typographical error.

Brent (2007)

An HMM for modeling eukaryotic genes

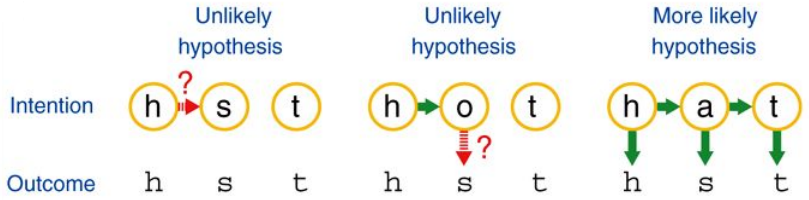


Hidden sequence of states

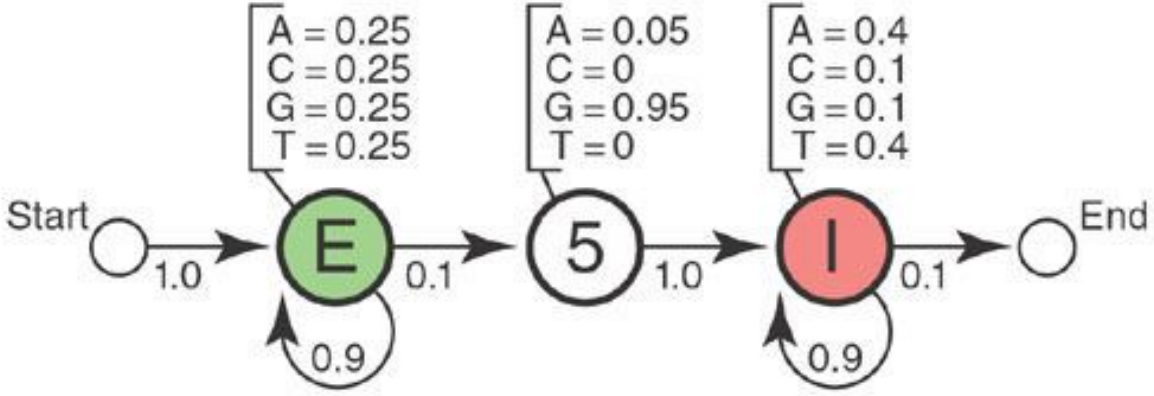
Observed sequence of nucleotides

. . .ATGCGTAGGTAAGTCAGTTGCGACAGTGAGGAGCTCTTGA. . .

A simple HMM for modeling eukaryotic genes



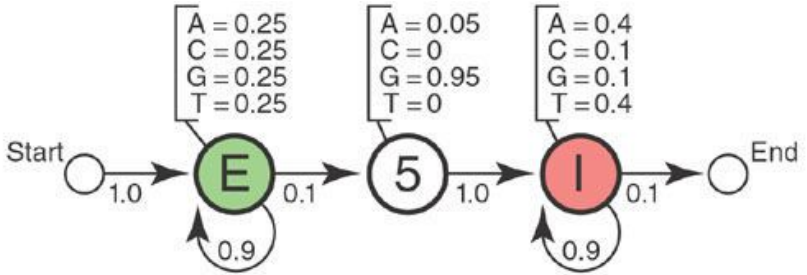
A toy HMM for 5′ splice site recognition



Eddy (2004)

A simple HMM for modeling eukaryotic genes

A toy HMM for 5′ splice site recognition



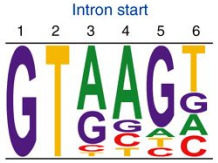
Come up with the HMM (structure λ & parameters θ; *like the one above*) based on prior knowledge + assumptions.

Then, given a DNA sequence (observations)…

`CTTCATGTGAAAGCAGACGTAAGTCA`

… we can get the probability of a sequence of hidden states that may explain the observed DNA sequence.

`EEEEEEEEEEEEEEEEEEEE5IIIIIIII`



Sequence logo representing the weight matrix for the first six bases of an intron.

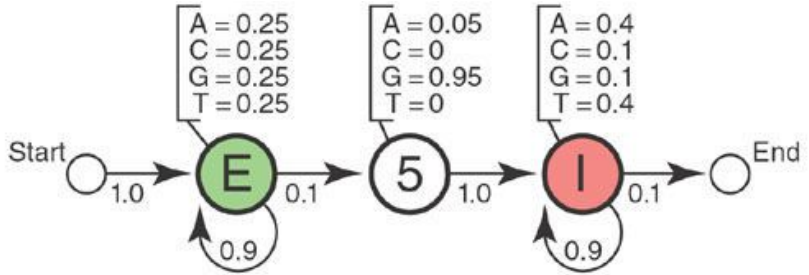
Designing the HMM λ:

1. Alphabet with M **symbols**.
2. No. of **states** in the model K w/ initial probabilities p_i for each state i; $\sum_i p_i = 1$.
3. **Emission** probabilities $e_i(x)$ for each state i, that sum to one over the M symbols x, $\sum_x e_i(x) = 1$.
4. **Transition** probabilities for each state i going to any other state j (including itself) that sum to one over the K states j, $\sum_j t_i(j) = 1$.

Eddy (2004)

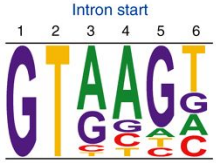
A simple HMM for modeling eukaryotic genes

A toy HMM for 5′ splice site recognition



Sequence: CTTCATGTGAAAGCAGACGTAAGTCA
State path: EEEEEEEEEEEEEEEEEEEE5IIIIIIII

log *P* = −41.22



Sequence logo representing the weight matrix for the first six bases of an intron.

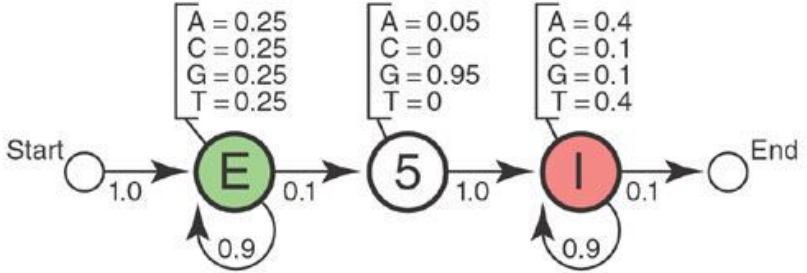
Designing the HMM **λ**:

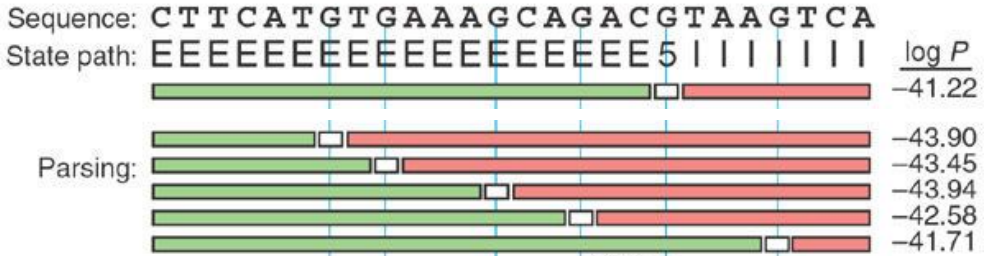
1. Alphabet with M **symbols**.
2. No. of **states** in the model K w/ initial probabilities p_i for each state i; $\sum_i p_i = 1$.
3. **Emission** probabilities $e_i(x)$ for each state i, that sum to one over the M symbols x, $\sum_x e_i(x) = 1$.
4. **Transition** probabilities for each state i going to any other state j (including itself) that sum to one over the K states j, $\sum_j t_i(j) = 1$.

Eddy (2004)

A simple HMM for modeling eukaryotic genes

A toy HMM for 5′ splice site recognition





S: observed sequence

π: state path, a Markov chain that's hidden.

θ: parameters of the model

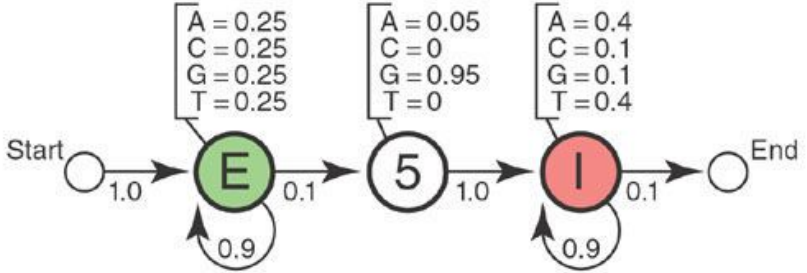
Pr(S, π | λ, θ): product of all emission & transition probabilities.

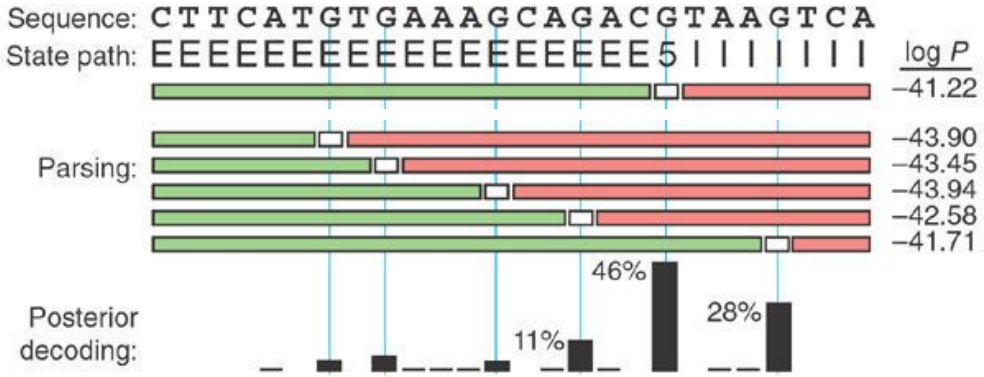
- How many emissions & transitions?
- How many possible paths?
- Which path to pick? The Viterbi algorithm (dynamic programming) is guaranteed to find the most probable path given seq & HMM λ.

Eddy (2004)

A simple HMM for modeling eukaryotic genes

A toy HMM for 5′ splice site recognition





S: observed sequence

π: state path, a Markov chain that's hidden.

θ: parameters of the model

Pr(S, π | λ, θ): product of all emission & transition probabilities.

- How confident are we that the 5th G is the right choice?

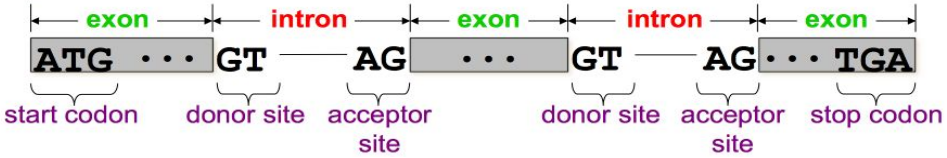
  – Posterior decoding using two dynamic programming algorithms – Forward and Backward – that sum over possible paths instead of choosing the best.

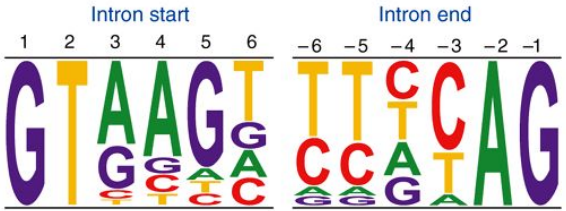
Eddy (2004)

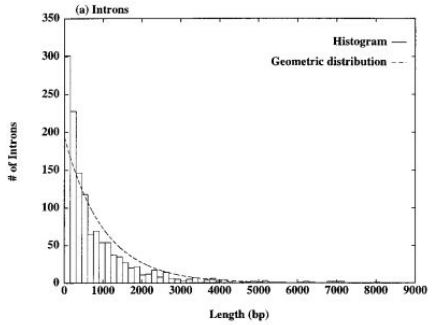
More realistic HMMs for modeling eukaryotic genes

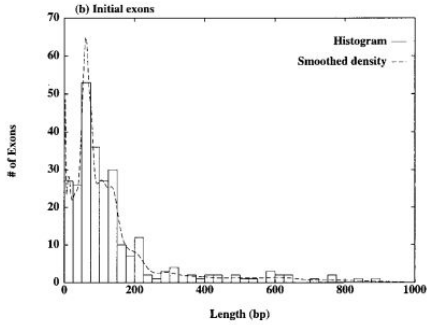
Adding more details:

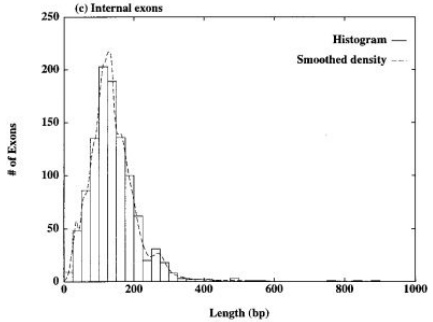
- Six-nucleotide consensus GTRAGT at the 5′ splice site.
- Similarly for the 3′ splice site.
- Add a 3′ exon state.
- Length constraints (Generalized hidden Markov models - GHMMs).

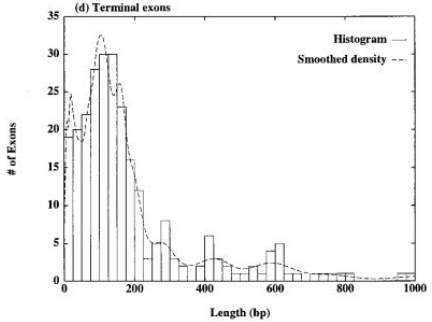










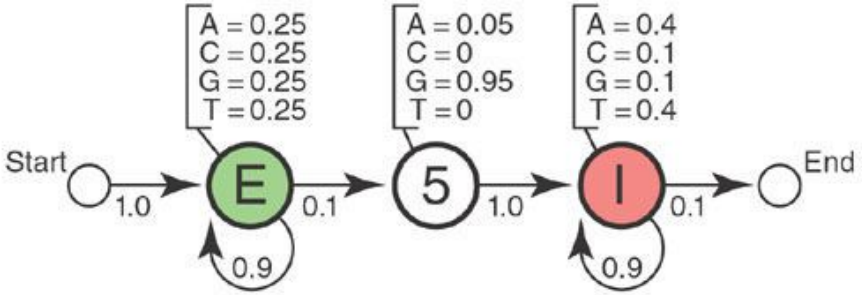


Eddy (2004); Burge & Karlin (1997)

More realistic HMMs for modeling eukaryotic genes

Adding more details:

- Six-nucleotide consensus GTRAGT at the 5′ splice site.
- Similarly for the 3′ splice site.
- Add a 3′ exon state.
- Length constraints (Generalized hidden Markov models - GHMMs).

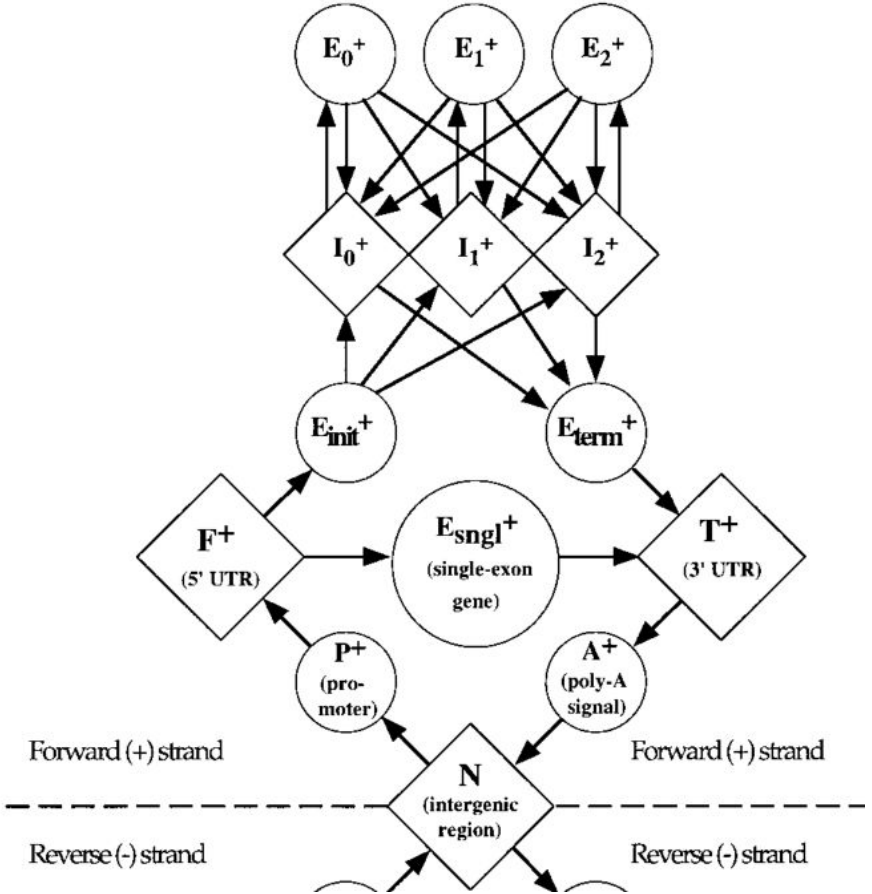


Burge & Karlin (1997)

More realistic HMMs for modeling eukaryotic genes

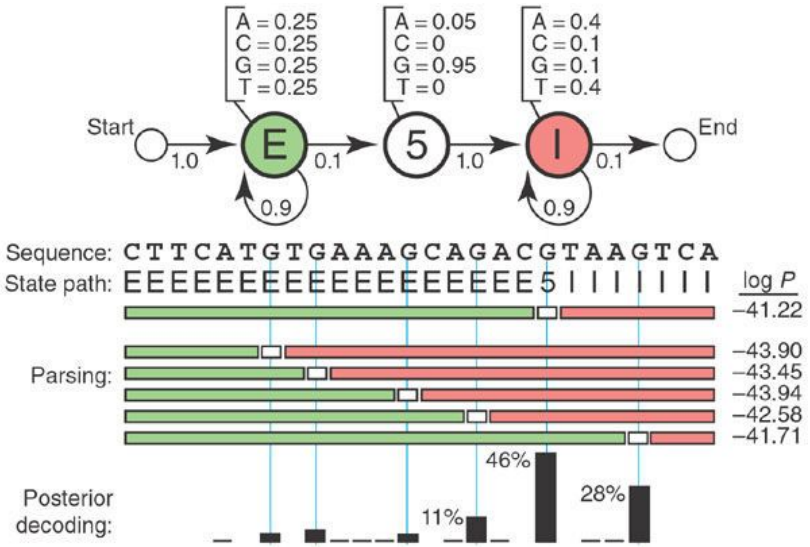
Adding more details:

- Six-nucleotide consensus GTRAGT at the 5′ splice site.
- Similarly for the 3′ splice site.
- Add a 3′ exon state.
- Length constraints (Generalized hidden Markov models - GHMMs).



Burge & Karlin (1997)

HMM: three fundamental problems



1. Problem 1 (Likelihood): Given an HMM **λ** and an observation sequence S, determine the likelihood P(S|**λ**).
2. Problem 2 (Decoding): Given an observation sequence S and an HMM **λ**, discover the best hidden state sequence π.
3. Problem 3 (Learning): Given an observation sequence S and the set of states in the HMM, learn the HMM parameters **θ**.

1. Problem 1 (Likelihood): Ability to **emit** a DNA sequence of a certain type.
2. Problem 2 (Decoding): **Recognize** DNA sequence of a certain type.
3. Problem 3 (Learning): **Learn** characteristics of sequences of different types.

Rabiner & Juang (1986)

Viterbi algorithm for decoding

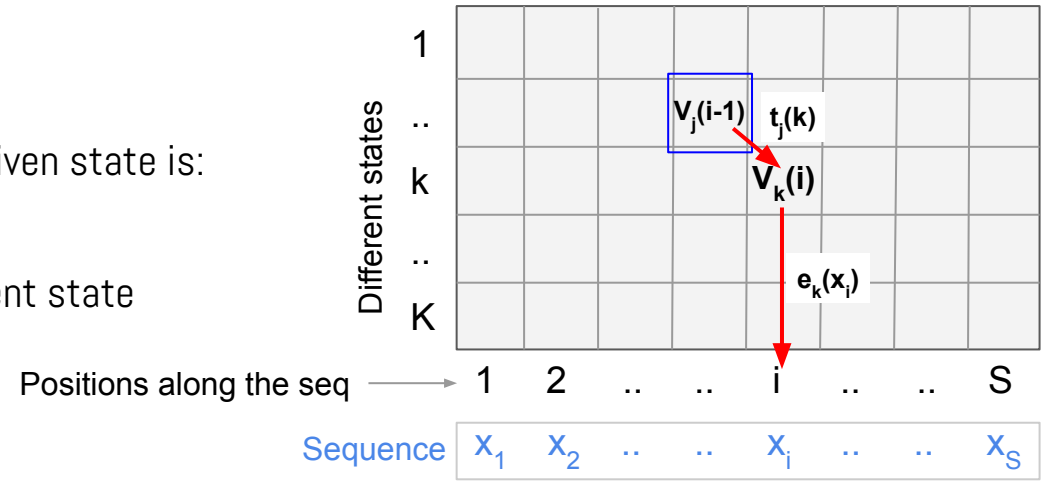
Dynamic programming: simplifying a complicated problem by breaking it down into simpler sub-problems in a recursive manner.

1. Store partial computation (max score to position **i** through state **k**):
   - Define $\mathbf{V_k(i)}$ = Prob. that the most likely path passes through state $\boldsymbol{\pi_i = k}$.

2. Assume we know the score in the previous position **(i-1)** for any state **j**: $\mathbf{V_j(i-1)}$. Now, calculate $\mathbf{V_k(i)}$ as function of: a) $\mathbf{V_j(i-1)}$, b) the transition prob. from **j**→**k**, & c) emission prob. of nucleotide $\mathbf{x_i}$ from state **k**:
   - $V_k(i) = e_k(x_i) * \max_j (t_j(k) V_j(i-1))$

DP: (optimal substructure) Best path through a given state is:

- Best path to previous state
- Best transition from previous state to current state
- Best path to the end state



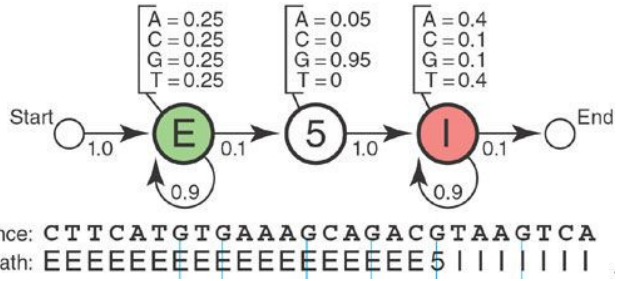
Viterbi algorithm for decoding

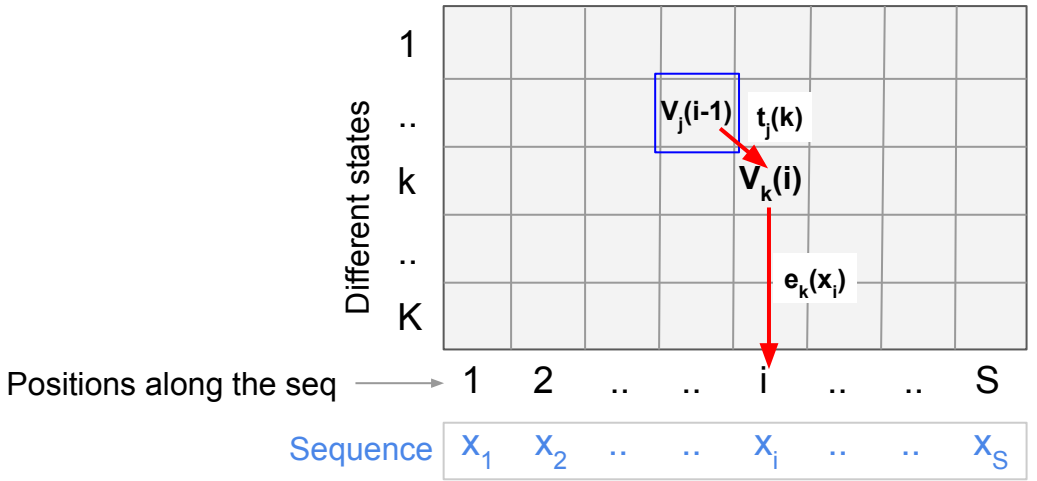
Input: $x_1, x_2, \ldots, x_S$

1. Initiatialization:
   - $V_0(0) = 1$, $V_k(0) = 0$, for all $k>0$
2. Iteration:
   - $V_k(i) = e_k(x_i) * \max_j (t_j(k)\, V_j(i-1))$
3. Termination:
   - $P(x, \pi^*) = \max_k V_k(S)$
4. Traceback:
   - Follow max pointers back

Running time: $O(K^2S)$

Space: $O(KS)$



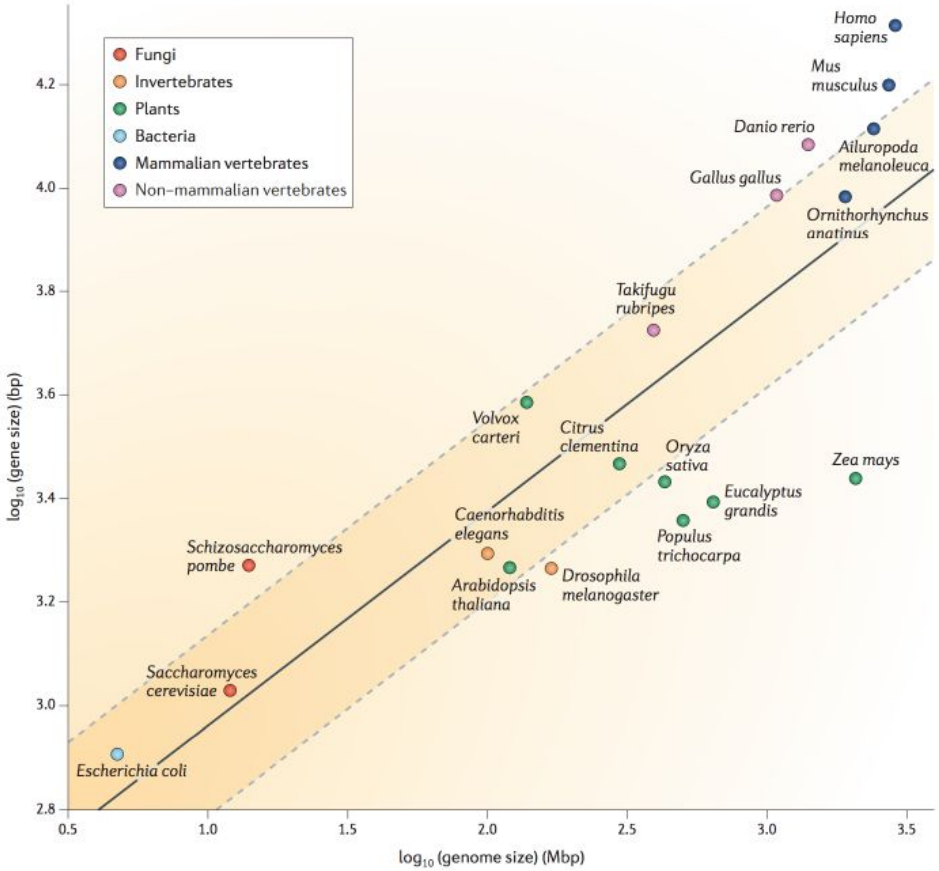


Some modern ab-initio gene prediction tools

Software	Description
Ab initio *and evidence-drivable gene predictors*	
Augustus	Accepts expressed sequence tag (EST)-based and protein-based evidence hints. Highly accurate
mGene	Support vector machine (SVM)-based discriminative gene predictor. Directly predicts 5′ and 3′ untranslated regions (UTRs) and poly(A) sites
SNAP	Accepts EST and protein-based evidence hints. Easily trained
FGENESH	Training files are constructed by SoftBerry and supplied to users
Geneid	First published in 1992 and revised in 2000. Accepts external hints from EST and protein-based evidence
Genemark	A self-training gene finder
Twinscan	Extension of the popular Genscan algorithm that can use homology between two genomes to guide gene prediction
GAZE	Highly configurable gene predictor
GenomeScan	Extension of the popular Genscan algorithm that can use BLASTX searches to guide gene prediction
Conrad	Discriminative gene predictor that uses conditional random fields (CRFs)
Contrast	Discriminative gene predictor that uses both SVMs and CRFs
CRAIG	Discriminative gene predictor that uses CRFs
Gnomon	Hidden Markov model (HMM) tool based on Genscan that uses EST and protein alignments to guide gene prediction
GeneSeqer	A tool for identifying potential exon–intron structure in precursor mRNAs (pre-mRNAs) by splice site prediction and spliced alignment

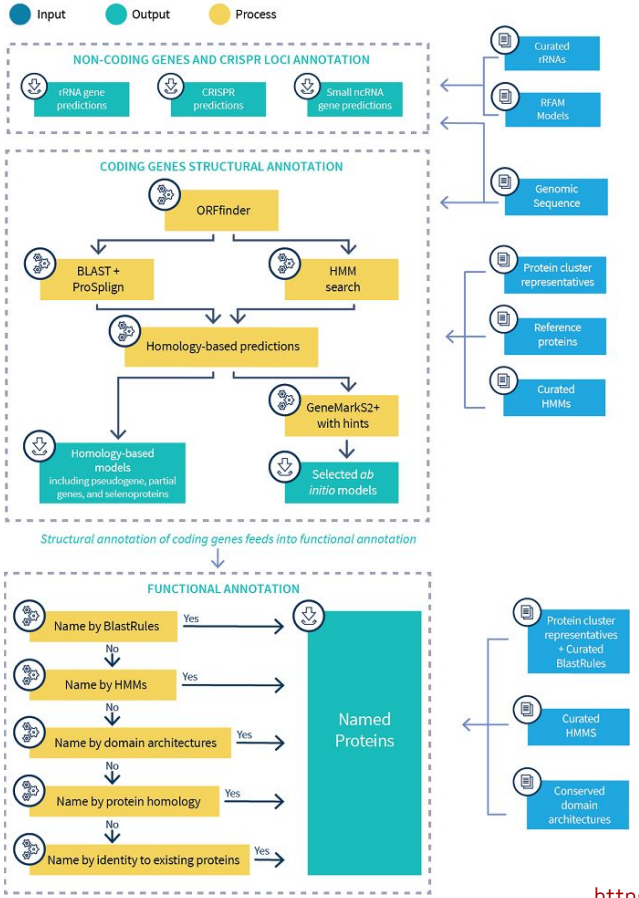
Yandell & Ence (2012)

Genomic complexity



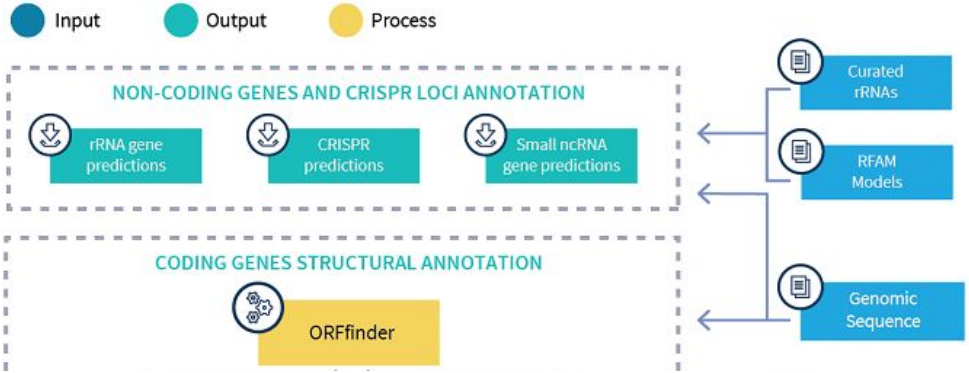
Yandell & Ence (2012)

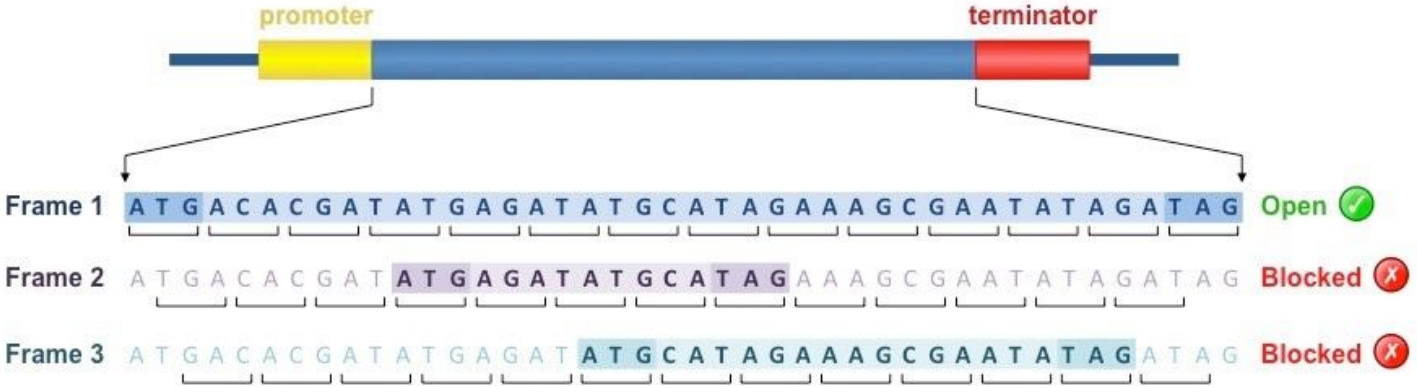
NCBI prokaryotic genome annotation pipeline



https://www.ncbi.nlm.nih.gov/genome/annotation_prok/process/

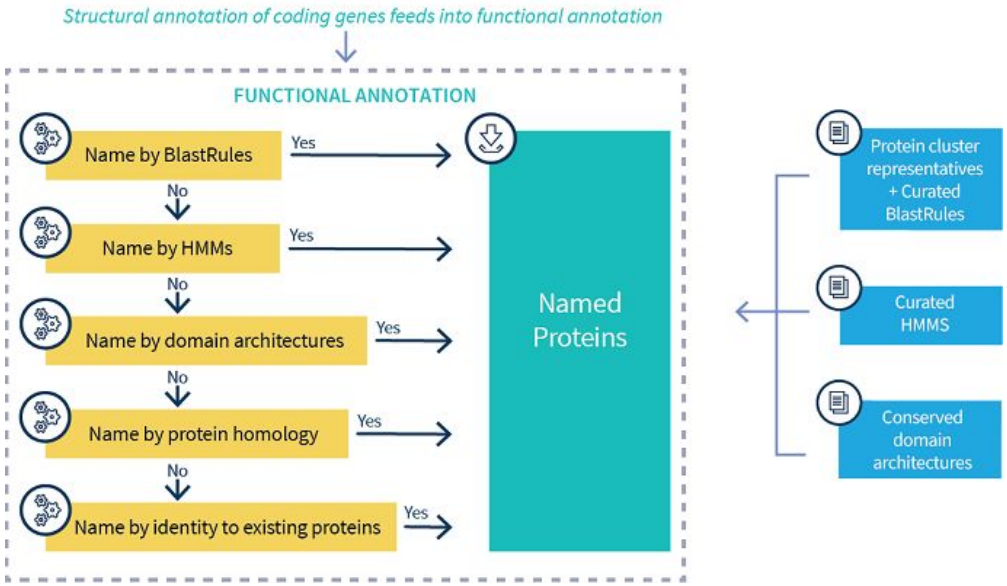
NCBI prokaryotic genome annotation pipeline





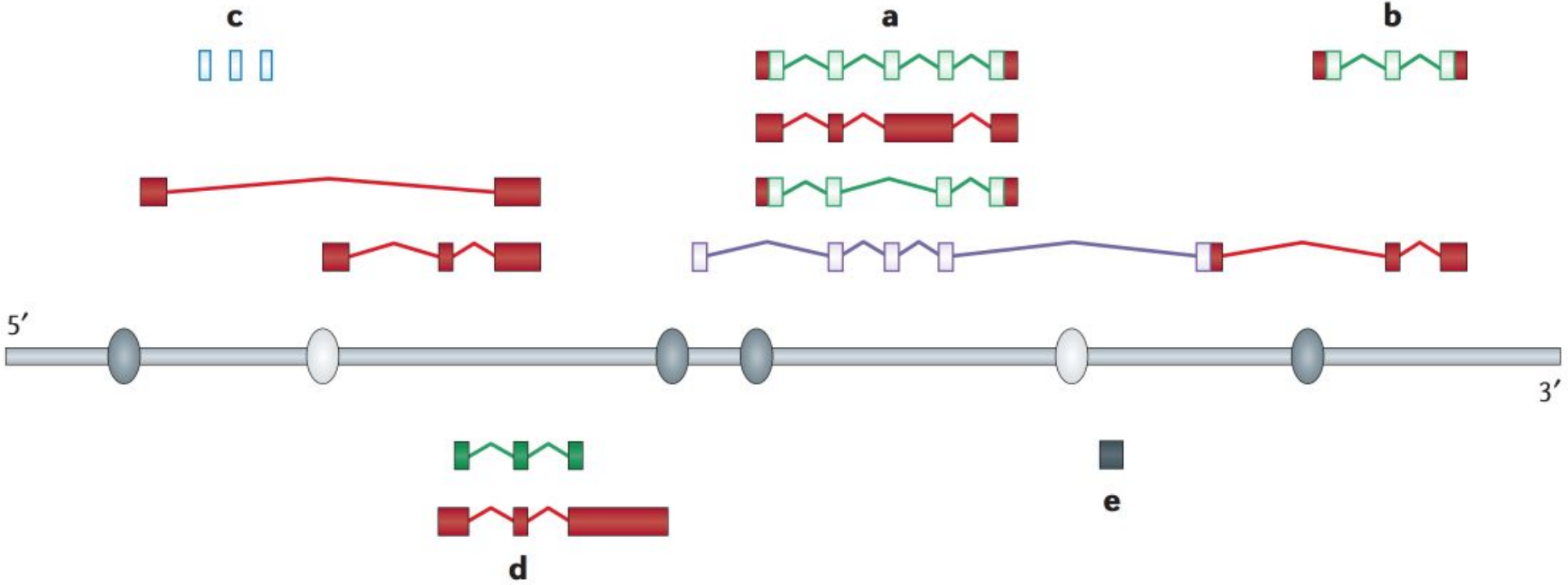
https://www.ncbi.nlm.nih.gov/genome/annotation_prok/process/

NCBI prokaryotic genome annotation pipeline



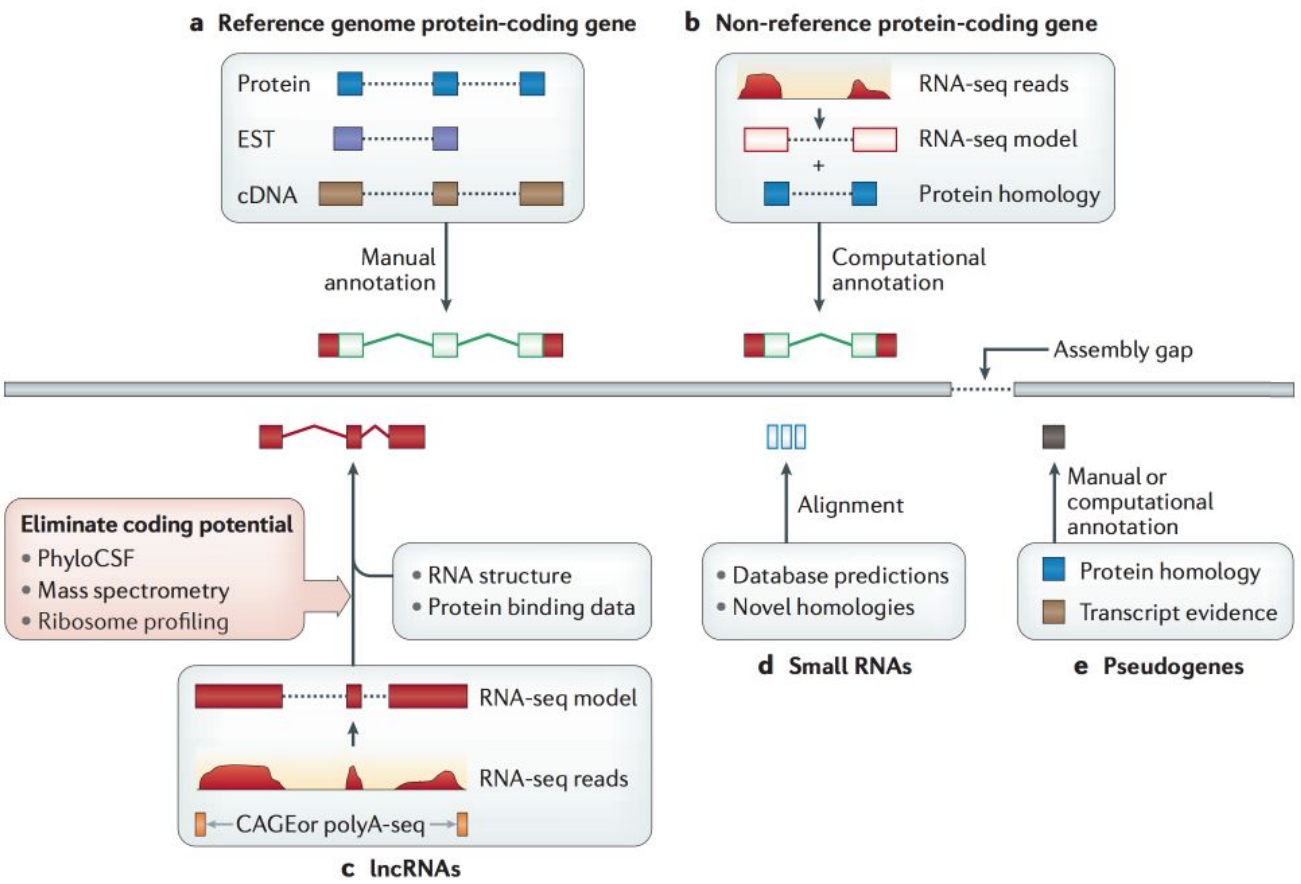
https://www.ncbi.nlm.nih.gov/genome/annotation_prok/process/

Genomic complexity – Types of eukaryotic genes & transcripts



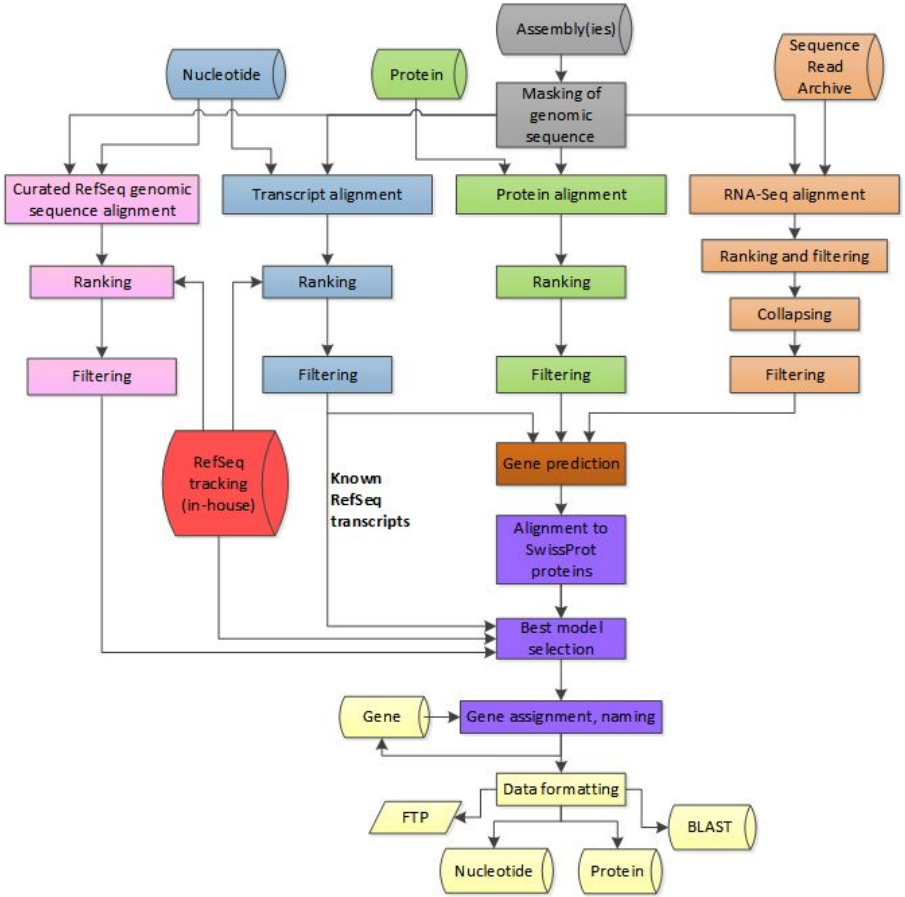
Mudge & Harrow (2016)

Annotation workflows for different gene types



Mudge & Harrow (2016)

NCBI eukaryotic genome annotation pipeline



Genomic sequence preparation (grey)

Alignments of transcripts (blue)

Alignment of proteins (green)

Alignment of short reads (orange)

Alignment of curated genomic sequences (pink)

Gene prediction based on all avail. Alignments (brown)

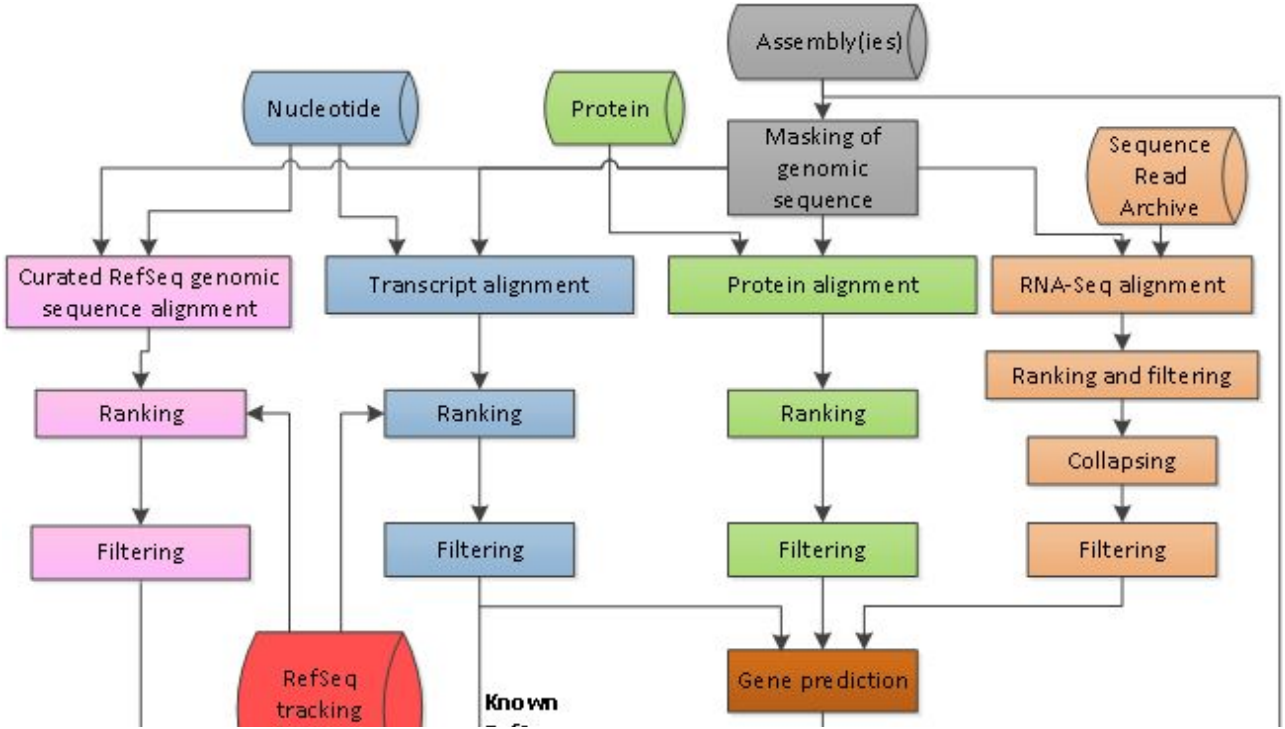
Internal tracking database of RefSeq seqs (red)

Selection of best models & protein naming (purple)

Formatting of ann sets for public resources (yellow)

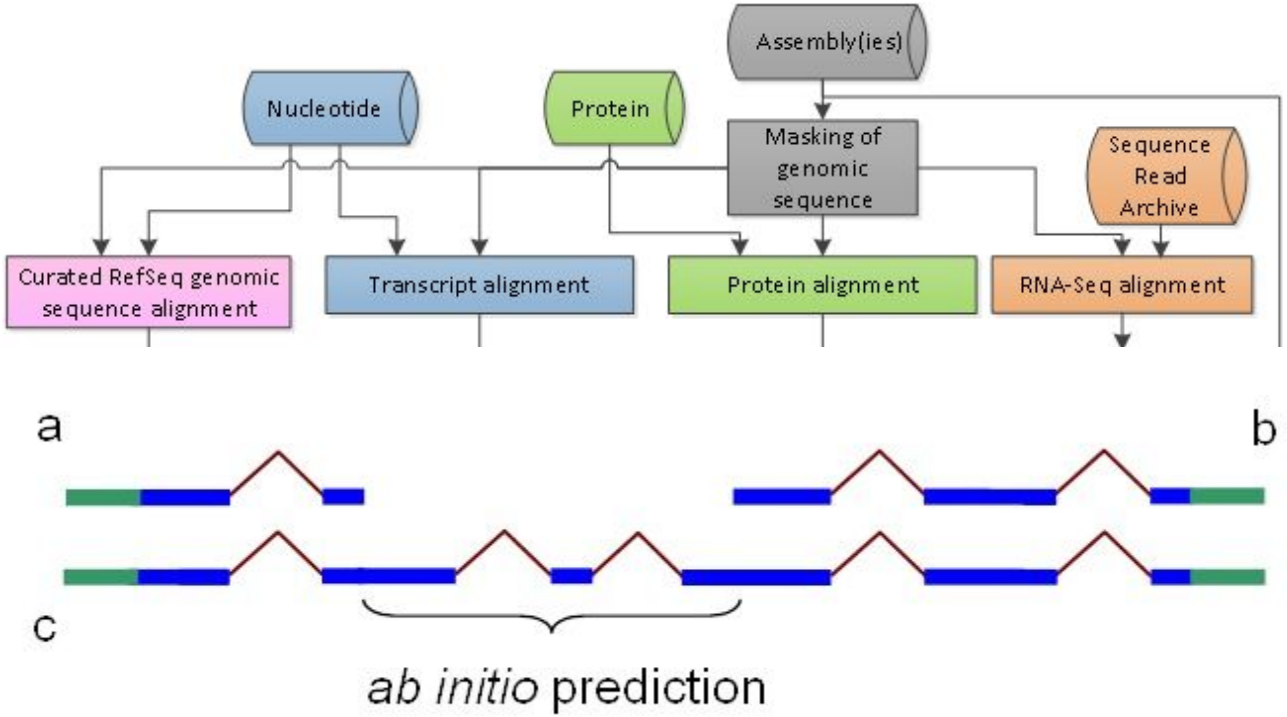
https://www.ncbi.nlm.nih.gov/genome/annotation_euk/process/

NCBI eukaryotic genome annotation pipeline



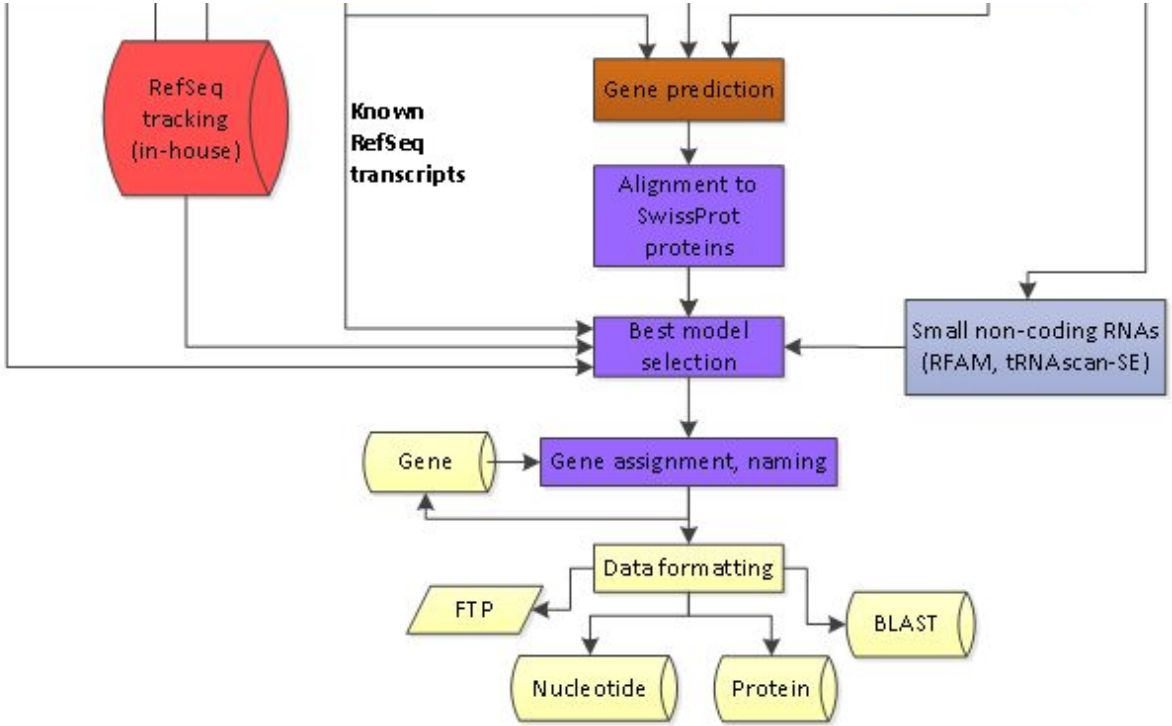
https://www.ncbi.nlm.nih.gov/genome/annotation_euk/process/

NCBI eukaryotic genome annotation pipeline



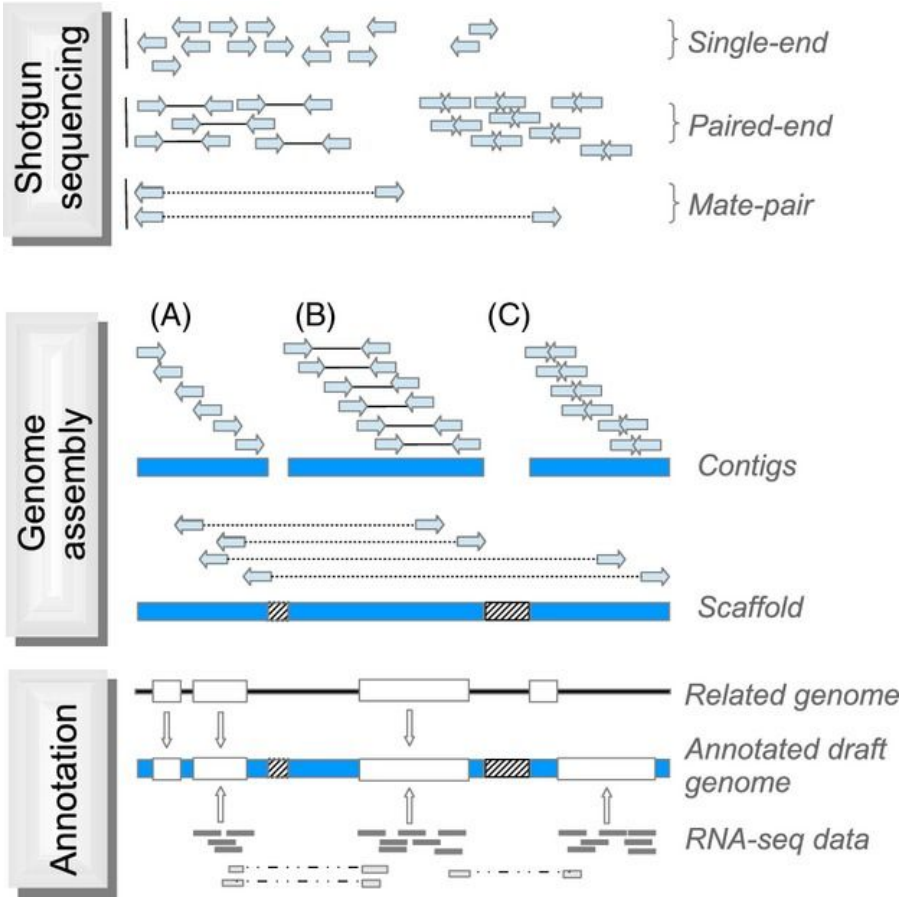
https://www.ncbi.nlm.nih.gov/genome/annotation_euk/process/

NCBI eukaryotic genome annotation pipeline



https://www.ncbi.nlm.nih.gov/genome/annotation_euk/process/

Genome assembly & annotation – Overview



Ekblom & Wolf (2014)