Bioinformatics Algorithms COS-BIOL-530/630 Lecture10

Days & Times	**Room**	**Meeting Dates**
Tu 2:00PM - 3:50PM	Thomas Gosnell Hall (GOS)-2178	01/13/2025 - 04/28/2025
Th 2:00PM - 3:50PM	Thomas Gosnell Hall (GOS)-2178	01/13/2025 - 04/28/2025

Instructor:
Fernando Rodriguez
email: frvsbi@rit.edu
Office: Orange Hall 1311

Gene Prediction
- Lecture10-

Announcements

Lecture10

Lab10

- Activity 10
- Discussion 10

Quiz 9: Open Friday April 4th 5 pm (**next week!**)

- Lecture/Lab 10 (Gene Prediction)
- Lecture/Lab 11 (High Throughput Sequencing)

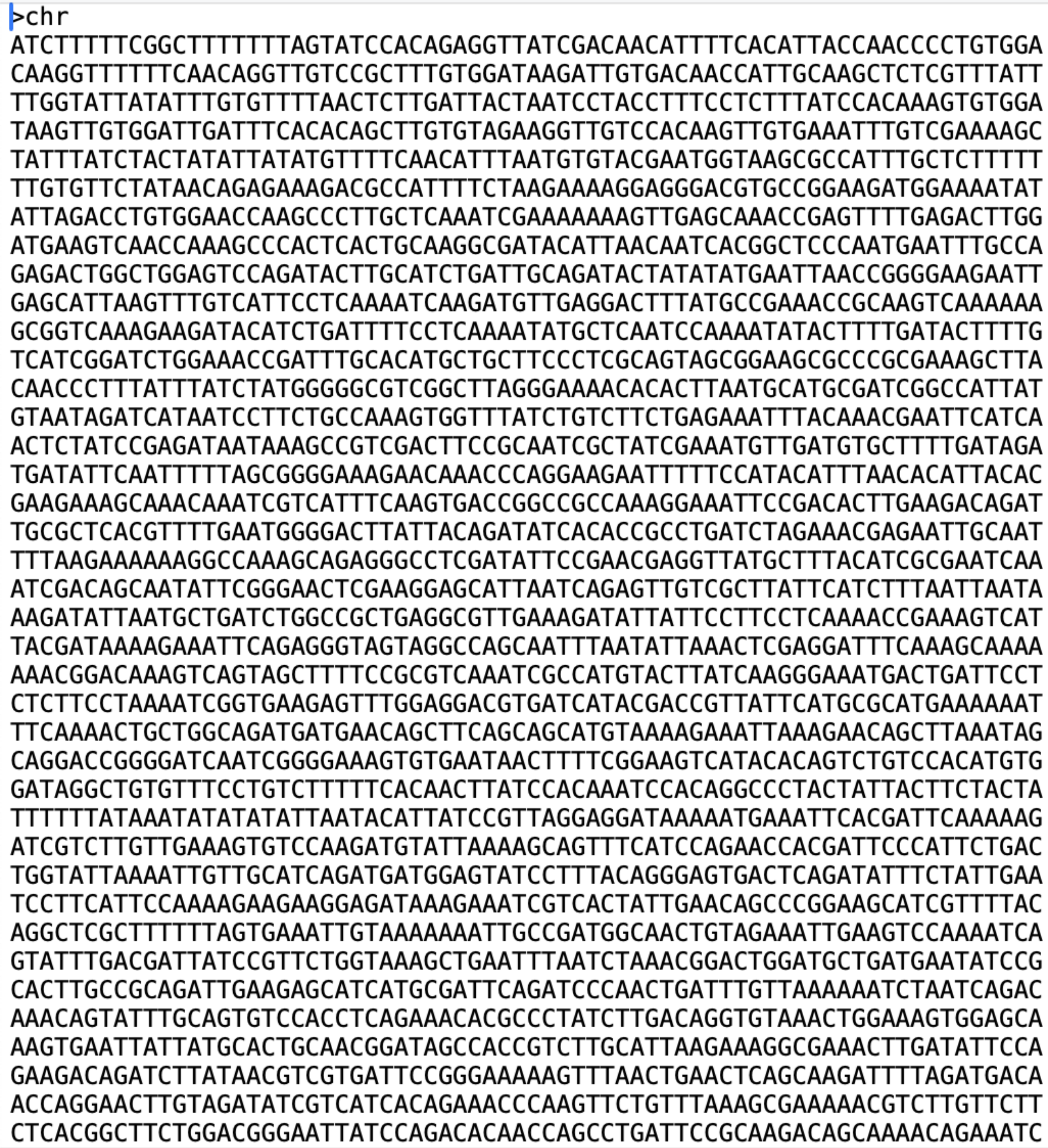
Gene Prediction
- Lecture10-

Topics:

- Gene recognition in sequences
- CpG islands
- Markov Model
- Hidden Markov Models

Gene Prediction
The problem:

We are given a sequence of **DNA**, and we wish to know which subsequence or concatenation of subsequences constitutes a **gene**.



```
>chr
ATCTTTTTCGGCTTTTTTTAGTATCCACAGAGGTTATCGACAACATTTTCACATTACCAACCCCTGTGGA
CAAGGTTTTTTCAACAGGTTGTCCGCTTTGTGGATAAGATTGTGACAACCATTGCAAGCTCTCGTTTATT
TTGGTATTATATTTGTGTTTTAACTCTTGATTACTAATCCTACCTTTCCTCTTTATCCACAAAGTGTGGA
TAAGTTGTGGATTGATTTCACACAGCTTGTGTAGAAGGTTGTCCACAAGTTGTGAAATTTGTCGAAAAGC
TATTTATCTACTATATTATATGTTTTCAACATTTAATGTGTACGAATGGTAAGCGCCATTTGCTCTTTTT
TTGTGTTCTATAACAGAGAAAGACGCCATTTTCTAAGAAAAGGAGGGACGTGCCGGAAGATGGAAAATAT
ATTAGACCTGTGGAACCAAGCCCTTGCTCAAATCGAAAAAAAGTTGAGCAAACCGAGTTTTGAGACTTGG
ATGAAGTCAACCAAAGCCCACTCACTGCAAGGCGATACATTAACAATCACGGCTCCCAATGAATTTGCCA
GAGACTGGCTGGAGTCCAGATACTTGCATCTGATTGCAGATACTATATATGAATTAACCGGGGAAGAATT
GAGCATTAAGTTTGTCATTCCTCAAAATCAAGATGTTGAGGACTTTATGCCGAAACCGCAAGTCAAAAAA
GCGGTCAAAGAAGATACATCTGATTTTCCTCAAAATATGCTCAATCCAAAATATACTTTTGATACTTTTG
TCATCGGATCTGGAAACCGATTTGCACATGCTGCTTCCCTCGCAGTAGCGGAAGCGCCCGCGAAAGCTTA
CAACCCTTTATTTATCTATGGGGGCGTCGGCTTAGGGAAAACACACTTAATGCATGCGATCGGCCATTAT
GTAATAGATCATAATCCTTCTGCCAAAGTGGTTTATCTGTCTTCTGAGAAATTTACAAACGAATTCATCA
ACTCTATCCGAGATAATAAAGCCGTCGACTTCCGCAATCGCTATCGAAATGTTGATGTGCTTTTGATAGA
TGATATTCAATTTTTAGCGGGGAAAGAACAAACCCAGGAAGAATTTTTCCATACATTTAACACATTACAC
GAAGAAAGCAAACAAATCGTCATTTCAAGTGACCGGCCGCCAAAGGAAATTCCGACACTTGAAGACAGAT
TGCGCTCACGTTTTGAATGGGGACTTATTACAGATATCACACCGCCTGATCTAGAAACGAGAATTGCAAT
TTTAAGAAAAAAGGCCAAAGCAGAGGGCCTCGATATTCCGAACGAGGTTATGCTTTACATCGCGAATCAA
ATCGACAGCAATATTCGGGAACTCGAAGGAGCATTAATCAGAGTTGTCGCTTATTCATCTTTAATTAATA
AAGATATTAATGCTGATCTGGCCGCTGAGGCGTTGAAAGATATTATTCCTTCCTCAAAACCGAAAGTCAT
TACGATAAAAGAAATTCAGAGGGTAGTAGGCCAGCAATTTAATATTAAACTCGAGGATTTCAAAGCAAAA
AAACGGACAAAGTCAGTAGCTTTTCCGCGTCAAATCGCCATGTACTTATCAAGGGAAATGACTGATTCCT
CTCTTCCTAAAATCGGTGAAGAGTTTGGAGGACGTGATCATACGACCGTTATTCATGCGCATGAAAAAAT
TTCAAAACTGCTGGCAGATGATGAACAGCTTCAGCAGCATGTAAAAGAAATTAAAGAACAGCTTAAATAG
CAGGACCGGGGATCAATCGGGGAAAGTGTGAATAACTTTTCGGAAGTCATACACAGTCTGTCCACATGTG
GATAGGCTGTGTTTCCTGTCTTTTTCACAACTTATCCACAAATCCACAGGCCCTACTATTACTTCTACTA
TTTTTTATAAATATATATATTAATACATTATCCGTTAGGAGGATAAAAATGAAATTCACGATTCAAAAAG
ATCGTCTTGTTGAAAGTGTCCAAGATGTATTAAAAGCAGTTTCATCCAGAACCACGATTCCCATTCTGAC
TGGTATTAAAATTGTTGCATCAGATGATGGAGTATCCTTTACAGGGAGTGACTCAGATATTTCTATTGAA
TCCTTCATTCCAAAAGAAGAAGGAGATAAAGAAATCGTCACTATTGAACAGCCCGGAAGCATCGTTTTAC
AGGCTCGCTTTTTTAGTGAAATTGTAAAAAAATTGCCGATGGCAACTGTAGAAATTGAAGTCCAAAATCA
GTATTTGACGATTATCCGTTCTGGTAAAGCTGAATTTAATCTAAACGGACTGGATGCTGATGAATATCCG
CACTTGCCGCAGATTGAAGAGCATCATGCGATTCAGATCCCAACTGATTTGTTAAAAAATCTAATCAGAC
AAACAGTATTTGCAGTGTCCACCTCAGAAACACGCCCTATCTTGACAGGTGTAAACTGGAAAGTGGAGCA
AAGTGAATTATTATGCACTGCAACGGATAGCCACCGTCTTGCATTAAGAAAGGCGAAACTTGATATTCCA
GAAGACAGATCTTATAACGTCGTGATTCCGGGAAAAAGTTTAACTGAACTCAGCAAGATTTTAGATGACA
ACCAGGAACTTGTAGATATCGTCATCACAGAAACCCAAGTTCTGTTTAAAGCGAAAAACGTCTTGTTCTT
CTCACGGCTTCTGGACGGGAATTATCCAGACACAACCAGCCTGATTCCGCAAGACAGCAAAACAGAAATC
```

Algorithms and Models in Bioinformatics

Model of a gene

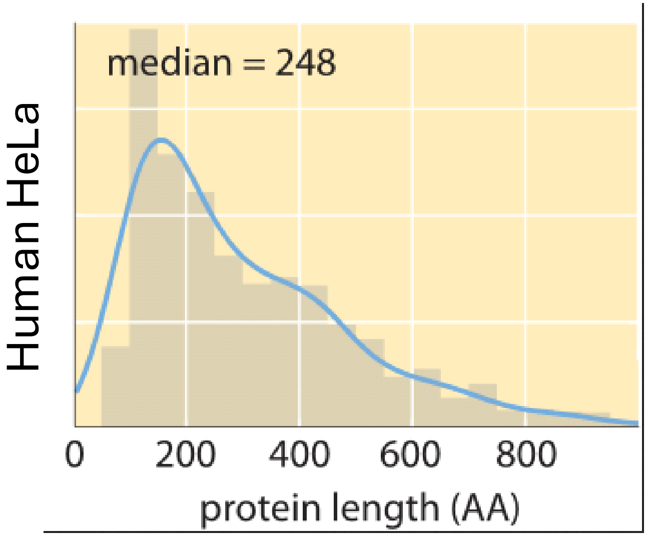
A gene is a sequence of nucleotides that encodes a protein sequence

- between 50 and 1000 residues in length
- A gene starts with Methionine
- And ends with a Stop Codon

Gene finding algorithm

Search for and identify all sequences that:

- start with ATG
- end with either TAG/TAA/TGA
- between 150 and 3,000 nucleotides in length.



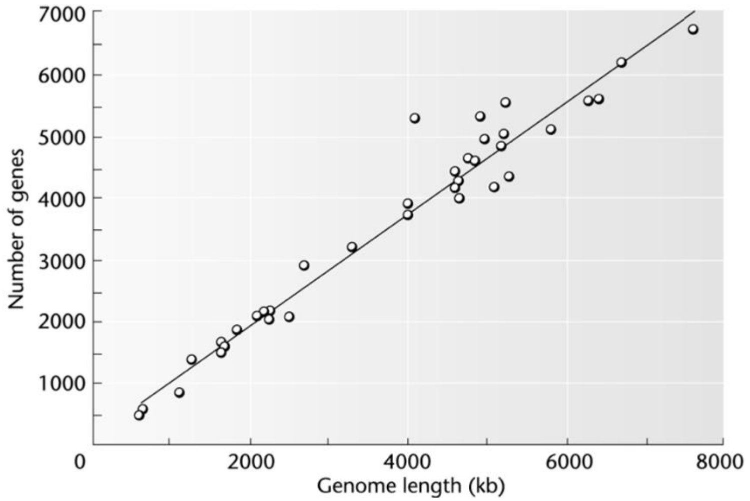
Source: https://book.bionumbers.org

```
Define the DNA sequence as a string

Create an empty list to store identified gene sequences

Set minimum_length = 150
Set maximum_length = 3000

For each position i in the DNA sequence from 0 to (length of sequence - 3):
    If the substring from i to i+2 is "ATG":  # Check for start codon
        For each position j from i+3 to (length of sequence - 2):
            If the substring from j to j+2 is "TAG" or "TAA" or "TGA":  #
                Calculate gene_length = j + 3 - i  # Length of the gene s
                If minimum_length <= gene_length <= maximum_length:
                    Extract the substring from i to j+2 as the gene s
                    Add the gene sequence to the list
                Break the inner loop  # Move to the next start codon

Output the list of identified gene sequences
```

Lecture 01!

Gene Prediction

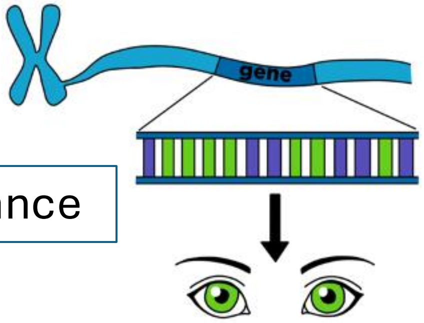
- There is a key difference between prokaryotes and eukaryotes.
  - > 85% of the prokaryotic genome is coding DNA
  - The portion of eukaryotic genome coding:
    - Fungi: 70% in *Saccharomyces cerevisiae* (yeast)
    - Plants: 20% in Arabidopsis
    - Human: 3%



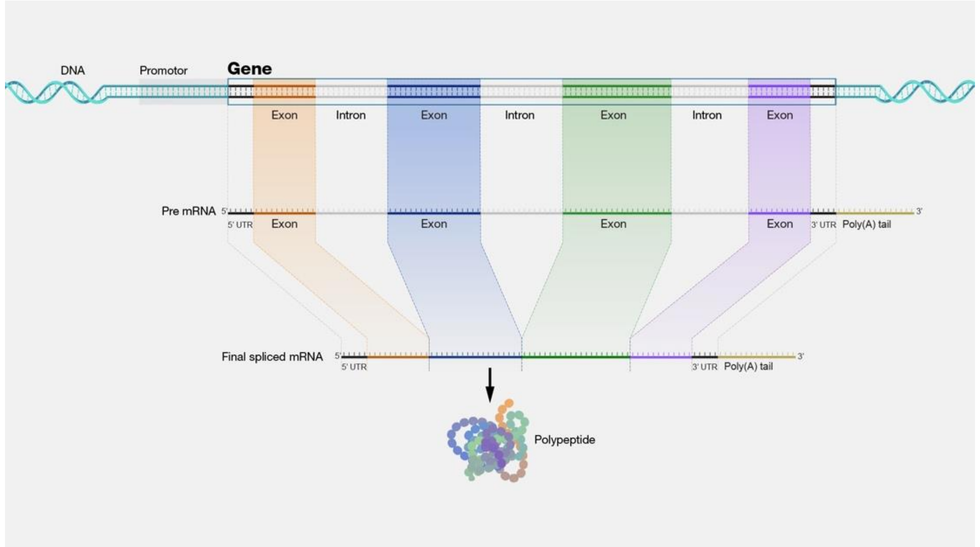
In prokaryotes most of the space is composed by genes.

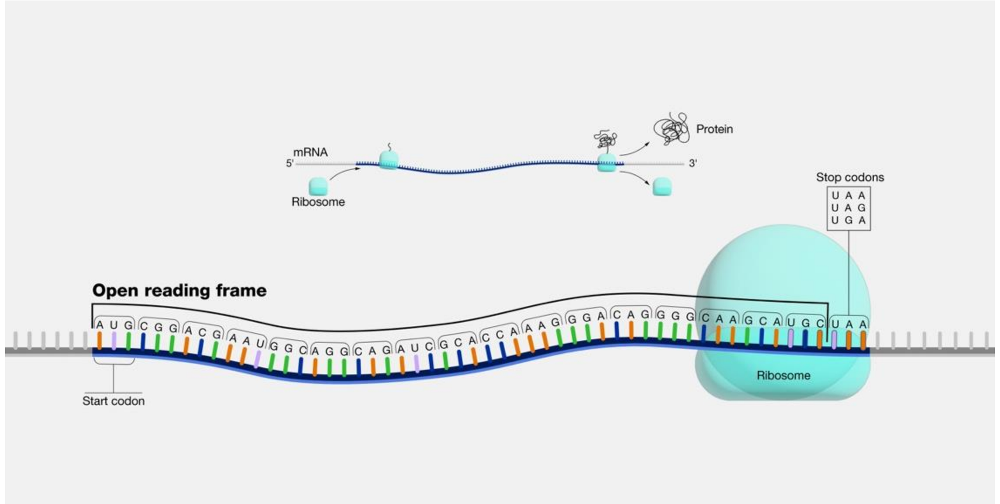
- We need to parse the genomic DNA to identify a complete gene with some knowledge (**a model**).

Gene *vs.* Open Reading Frame



The gene is considered the basic unit of inheritance



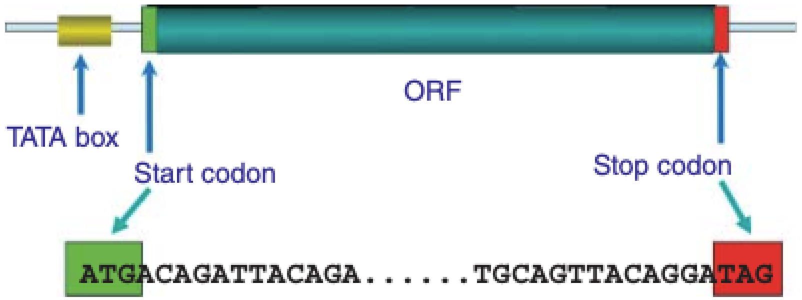

mRNA
5'
3'
Ribosome
Protein
Stop codons
U A A
U A G
U G A
Open reading frame
A U G C G G A C G A A U G G C A G G C A G A U C G C A C C A A A G G G A C A G G G G C A A G C A U G C U A A
Start codon
Ribosome

Source: genome.gov

An **open reading frame (ORF),** as related to genomics, **is a portion of a DNA sequence that does not include a stop codon** (which functions as a stop signal). A codon is a DNA or RNA sequence of three nucleotides (a trinucleotide) that forms a unit of genomic information encoding a particular amino acid or signaling the termination of protein synthesis (stop codon).

Prokaryotic genes

- A prokaryotic gene typically begins with a start codon (eg. ATG, GTG, TTG).
- Ends with one of the three stops codon (eg. TAG, TAA, or TGA).
- Most of the genes are organized in ***operons***: gene clusters of **more than one ORF** that are under the control of a shared set of regulatory sequences.
  - Promoters
  - Silencers
  - Terminators
  - Operators
- Regulatory sequences constitute 10—15% of the prokaryote genome.
- Promoters are located near the transcription start sites (TSSs) of genes, on the same strand and upstream of the gene or ORF.



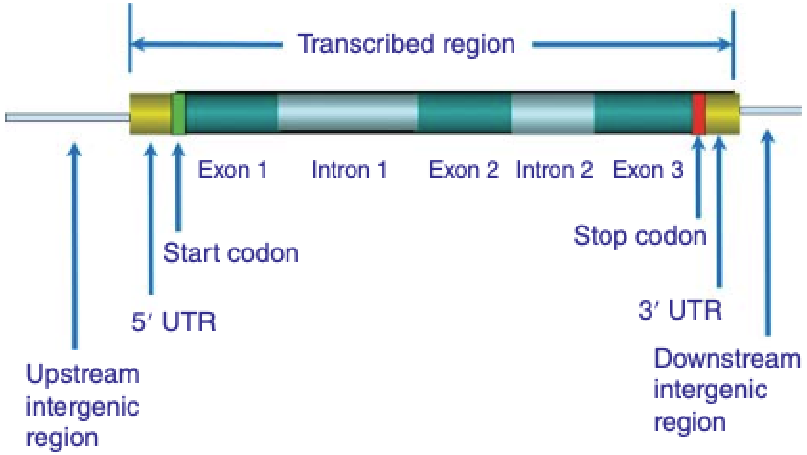
Source: Bioinformatics, Baxevanis

Eukaryotic genes

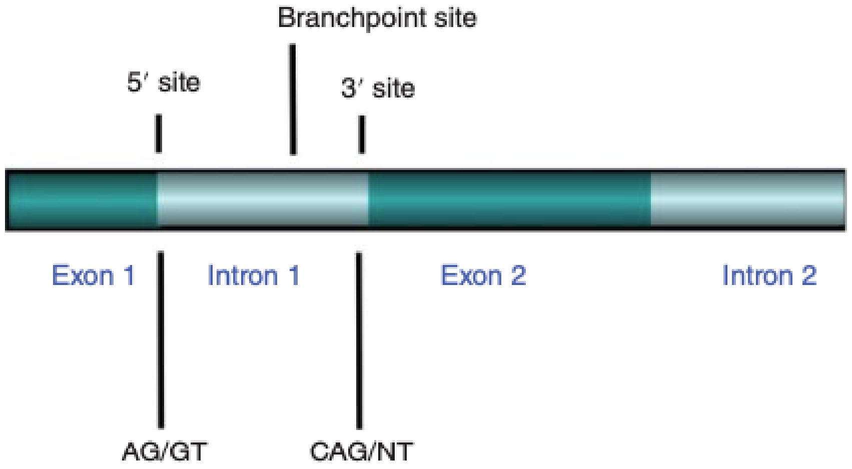
To find a eukaryotic gene, we must identify 4 signals:

- Start codon
- Stop codon
- Beginning of intron (donor site)
- End of intron (acceptor site)

Eukaryotic gene



Splice site region exon/intron



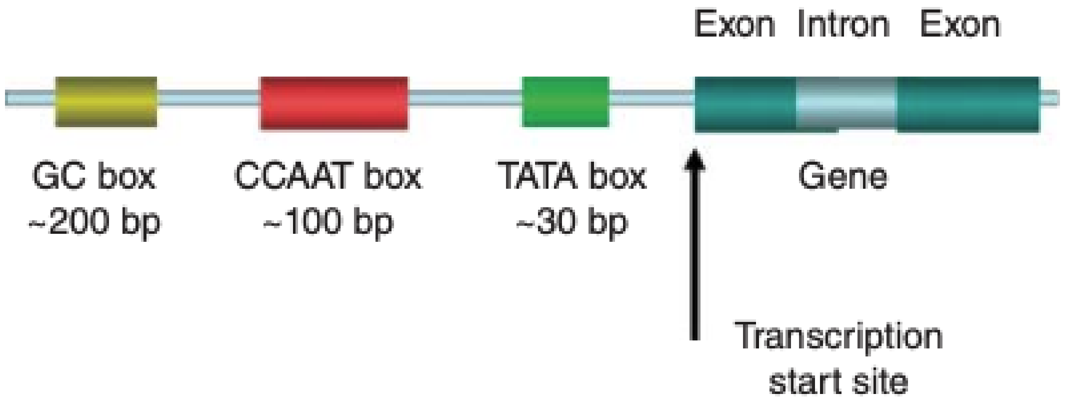
Source: Bioinformatics, Baxevanis

Eukaryotic genes

To find a eukaryotic gene, we must identify 4 signals:

- Start codon
- Stop codon
- Beginning of intron (donor site)
- End of intron (acceptor site)

*** It helps to find other signals outside the gene, such as promoters and ribosomal binding sites**



Source: Bioinformatics, Baxevanis

Start codon

What defines a start codon?

- Eukaryotes: ATG

- Prokaryotes: ATG, GTG, TTG

- In a random sequence, **the probability of an initiation codon is (1/4)*3 = 1/64 (for each).**

- There is usually a characteristic regulatory sequence upstream (promoter).

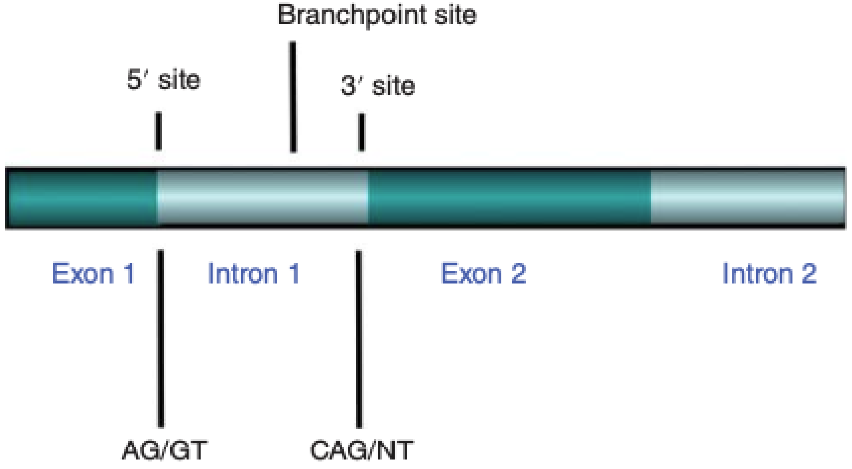
- However, regulatory element location and sequence are not consistent between species.

Stop codon

What defines a stop codon?

- TAA, TAG, TGA
- No sequence or regulatory element upstream.
- In a random sequence**, the probability of a stop codon is (1/4)*3 = 1/64 (for each).**

Splice sites



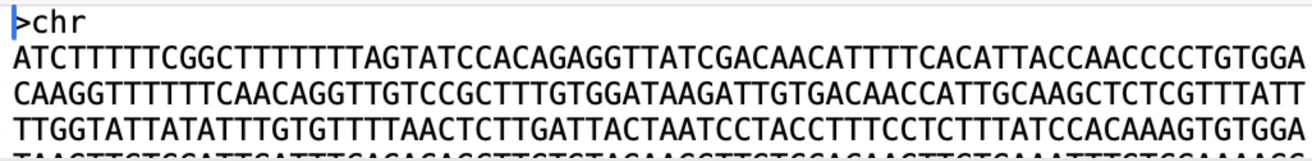
What defines a splice site (eukaryotes)?

- The donor is **almost** always GT.
- The acceptor is **almost** always AG.
- There are **certain consistencies** around the splice sites.
- However, the (average) size of introns varies between species.
- Exon base count is not always multiple of 3: introns can split codons!
- The split is not necessary in the same frameshift.

ORFs

What defines an open reading frame?

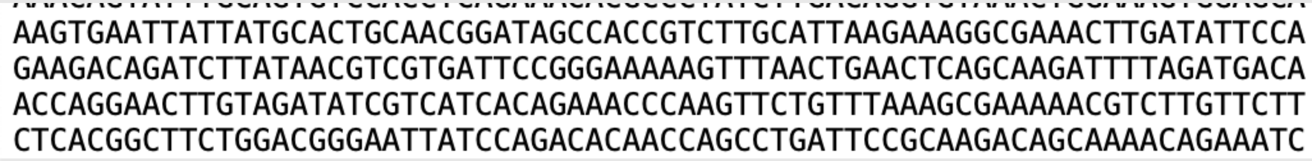
- Begins with **START codon**, ends with **STOP codon**.
- If we have a DNA sequence (not genome), an ORF must be the longest sequence without a stop codon.
- This is easy! In theory, an ORF is a gene. Indeed, in bacteria, it is a gene.
- The problem:
  - An ORF is ended by a stop codon. How do you know that the stop codon is the ***real* stop**?
  - Maybe the stop codon is just a random sequence embedded (randomly) in a non-coding area.
  - And the stop codon could come from any of the SIX reading frames (sense and anti-sense, 3 frames each).



```
>chr
ATCTTTTTCGGCTTTTTTTAGTATCCACAGAGGTTATCGACAACATTTTCACATTACCAACCCCTGTGGA
CAAGGTTTTTTCAACAGGTTGTCCGCTTTGTGGATAAGATTGTGACAACCATTGCAAGCTCTCGTTTATT
TTGGTATTATATTTGTGTTTTAACTCTTGATTACTAATCCTACCTTTCCTCTTTATCCACAAAGTGTGGA
```

Algorithm *Find_Stop_Codons*

```
Input: DNA_sequence (a string of nucleotides)
  Output: A list of positions of stop codons in each reading frame

  Define stop_codons as ["TAA", "TAG", "TGA"]
  Initialize stop_positions as an empty dictionary with keys "Frame 1", "Frame 2", and "Frame 3"

  For each frame in {0, 1, 2}:  // Three reading frames
    Initialize stop_positions["Frame " + (frame + 1)] as an empty list

    For i from frame to length(DNA_sequence) - 2 step 3:
      codon ← substring of DNA_sequence from i to i+2 (inclusive)

      If codon is in stop_codons:
        Append i (position of stop codon) to stop_positions["Frame " + (frame + 1)]

  Return stop_positions
```



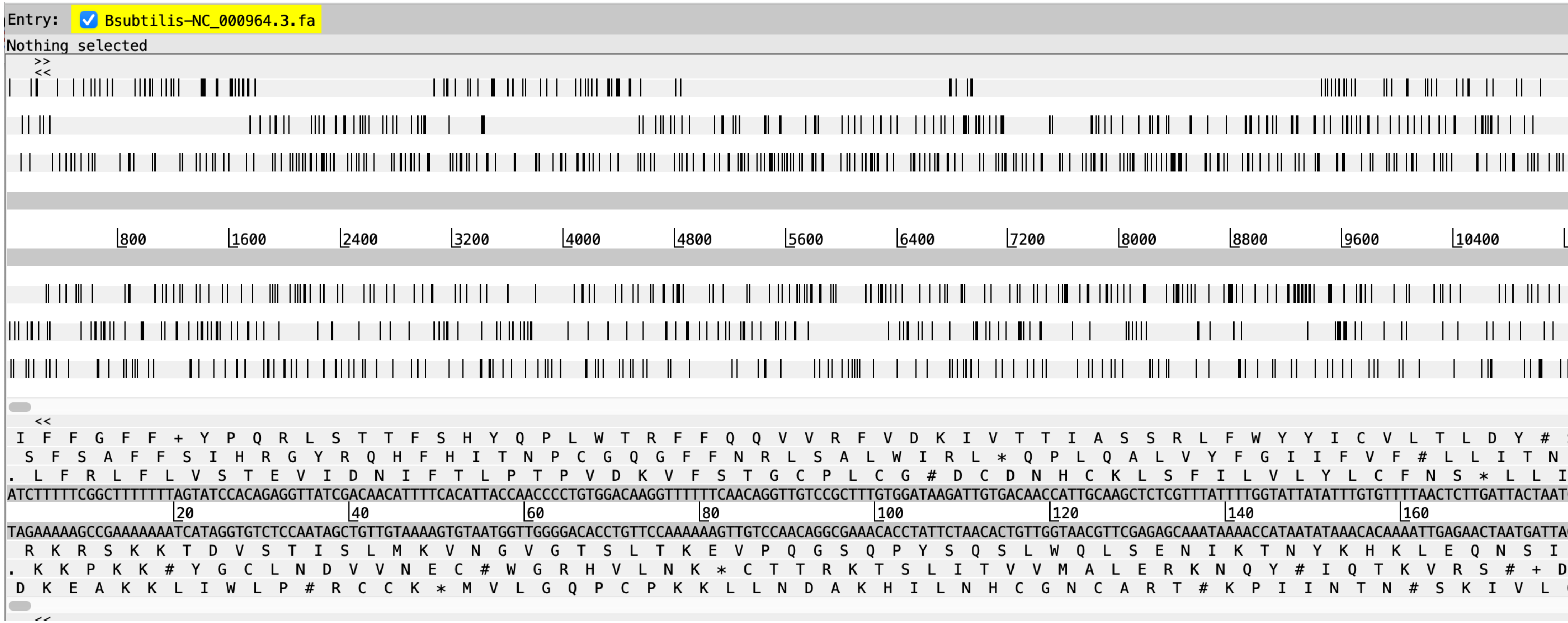
```
AAGTGAATTATTATGCACTGCAACGGATAGCCACCGTCTTGCATTAAGAAAGGCGAAACTTGATATTCCA
GAAGACAGATCTTATAACGTCGTGATTCCGGGAAAAAGTTTAACTGAACTCAGCAAGATTTTAGATGACA
ACCAGGAACTTGTAGATATCGTCATCACAGAAACCCAAGTTCTGTTTAAAGCGAAAAACGTCTTGTTCTT
CTCACGGCTTCTGGACGGGAATTATCCAGACACAACCAGCCTGATTCCGCAAGACAGCAAAACAGAAATC
```

Algorithm *Find_Stop_Codons*

For each frame

(+) 1 2 3

(+) 1 2 3



```
Entry: ☑ Bsubtilis-NC_000964.3.fa
Nothing selected
```

```
<<
 I  F  F  G  F  F  +  Y  P  Q  R  L  S  T  T  F  S  H  Y  Q  P  L  W  T  R  F  F  Q  Q  V  V  R  F  V  D  K  I  V  T  T  I  A  S  S  R  L  F  W  Y  Y  I  C  V  L  T  L  D  Y  #
  S  F  S  A  F  F  S  I  H  R  G  Y  R  Q  H  F  H  I  T  N  P  C  G  Q  G  F  F  N  R  L  S  A  L  W  I  R  L  *  Q  P  L  Q  A  L  V  Y  F  G  I  I  F  V  F  #  L  L  I  T  N
. L  F  R  L  F  L  V  S  T  E  V  I  D  N  I  F  T  L  P  T  P  V  D  K  V  F  S  T  G  C  P  L  C  G  #  D  C  D  N  H  C  K  L  S  F  I  L  V  L  Y  L  C  F  N  S  *  L  L  I
ATCTTTTTCGGCTTTTTTTAGTATCCACAGAGGTTATCGACAACATTTTCACATTACCAACCCCTGTGGACAAGGTTTTTTCAACAGGTTGTCCGCTTTGTGGATAAGATTGTGACAACCATTGCAAGCTCTCGTTTATTTTGGTATTATATTTGTGTTTTAACTCTTGATTACTAAT
                   20                  40                  60                  80                  100                 120                 140                 160
TAGAAAAAGCCGAAAAAAATCATAGGTGTCTCCAATAGCTGTTGTAAAAGTGTAATGGTTGGGGACACCTGTTCCAAAAAAGTTGTCCAACAGGCGAAACACCTATTCTAACACTGTTGGTAACGTTCGAGAGCAAATAAAACCATAATATAAACACAAAATTGAGAACTAATGATTA
  R  K  R  S  K  K  T  D  V  S  T  I  S  L  M  K  V  N  G  V  G  T  S  L  T  K  E  V  P  Q  G  S  Q  P  Y  S  Q  S  L  W  Q  L  S  E  N  I  K  T  N  Y  K  H  K  L  E  Q  N  S  I
. K  K  P  K  K  #  Y  G  C  L  N  D  V  V  N  E  C  #  W  G  R  H  V  L  N  K  *  C  T  T  T  R  K  T  S  L  I  T  V  V  M  A  L  E  R  K  N  Q  Y  #  I  Q  T  K  V  R  S  #  +  D
 D  K  E  A  K  K  L  I  W  L  P  #  R  C  C  K  *  M  V  L  G  Q  P  C  P  K  K  L  L  N  D  A  K  H  I  L  N  H  C  G  N  C  A  R  T  #  K  P  I  I  N  T  N  #  S  K  I  V  L
<<
```

Artemis software

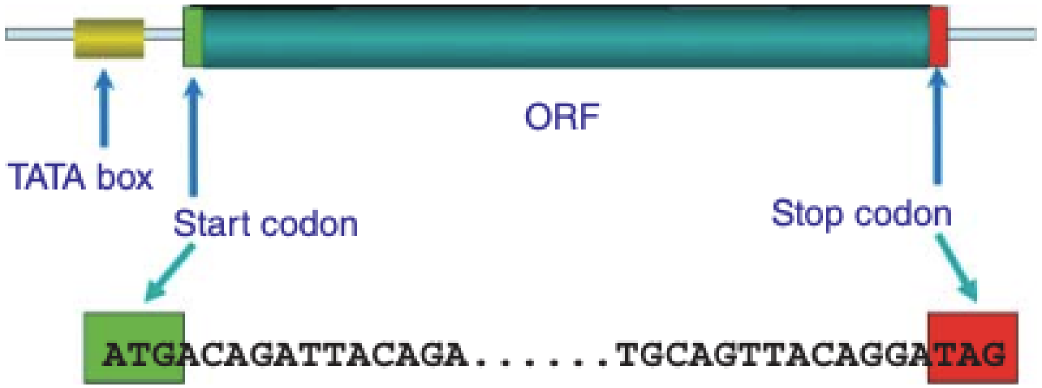
Gene Prediction
The solution:

- In prokaryotes:
    - Find the ORF

- In Eukaryotes:
    - Need to identify more structures (splice sites)

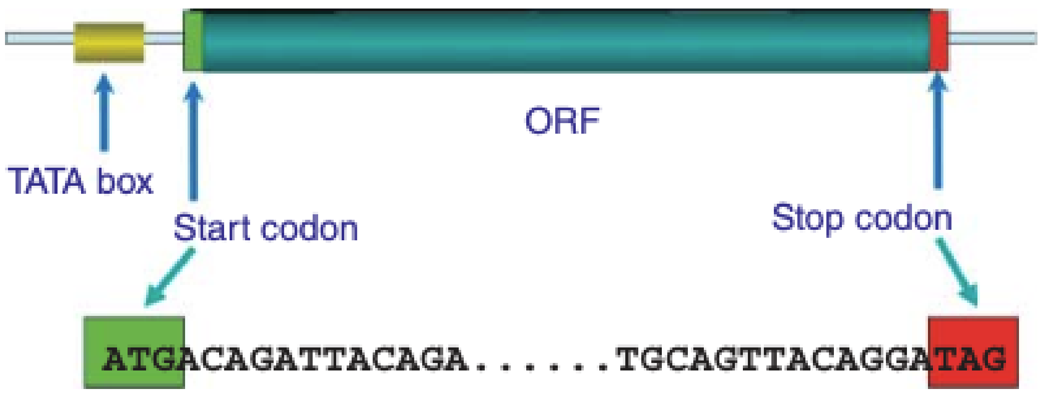
Gene Finding Strategy

in prokaryotes



Gene Finding Strategy

in prokaryotes



- Brute force:
  - Find ATG start codon
  - Longest ORF (>150 bases)
  - Move to the next ATG downstream
  - Repeat the process in the opposite strand
- Find motif signal (matrix profile)
  - TATA - Pribnow box

Gene Finding Models

- Markov Models (MMs)
- Hidden Markov Models (HMMs)

Models and Algorithms used in Computational Biology

A **model** is a parametric explanation of the observations of interest.

Probabilistic models/methods:

- Maximum Likelihood
- Bayesian
- Machine Learning
- Markov Chain Models
- Hidden Markov Models

An **algorithm** is a set of instructions for solving a problem, e.g. , inferring the optimal value of a model's parameter.

Algorithms/Methods:

- Sequence (string): sort/search algorithms
- Optimization algorithms:
    - Linear programing
    - Dynamic programing
    - Greedy algorithms
    - Heuristic methods

> **“All models are wrong, but some models are useful”**
>
> George Box

Correct *versus* incorrect Algorithms

Lecture 01!

Markov Models (MMs)

A Markov chain, model, or process refers to a series of observations in which the **probability of an observation depends on a number of previous observations**. The number of observations defines the “order” of the chain.

For example, in a first-order Markov model, the probability of an observation depends only on the previous observation. In a Markov chain of order 5, the probability of an observation depends on the five preceding observations.

A DNA sequence can be considered to be an example of a Markov model because the likelihood of **observing a particular base at a given position may depend on the bases preceding it**. In particular, in coding regions, it is well known that **the probability of a given base depends on the five preceding bases**, reflecting observed **codon biases** and dependencies between adjacent codons.

Such dependence is **not observed in non-coding regions**. When scanning an anonymous genomic region, one can compute how well the local nucleotide sequence conforms to the fifth-order dependencies observed in coding regions and assign appropriate coding likelihood scores.

CpG islands

- In the (human) genome, CpG dinucleotides should be rare (probabilities of C and G). **Why?**

- They are rich regions in CG dinucleotides.

- From 100-1000 bases long.

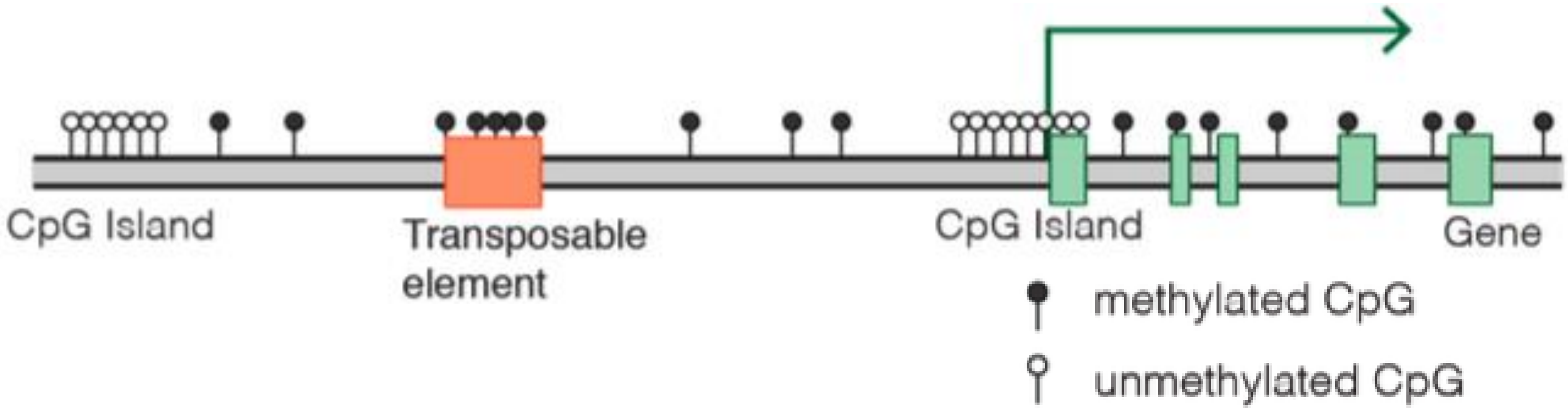
- The cytosine is usually modified by methylation (5mC).

- CpG regions in the genome play an essential role in regulation (suppressing nearby promoters/genes).

- CpG islands have other bases (A and T); they are just rich in CG dinucleotides.

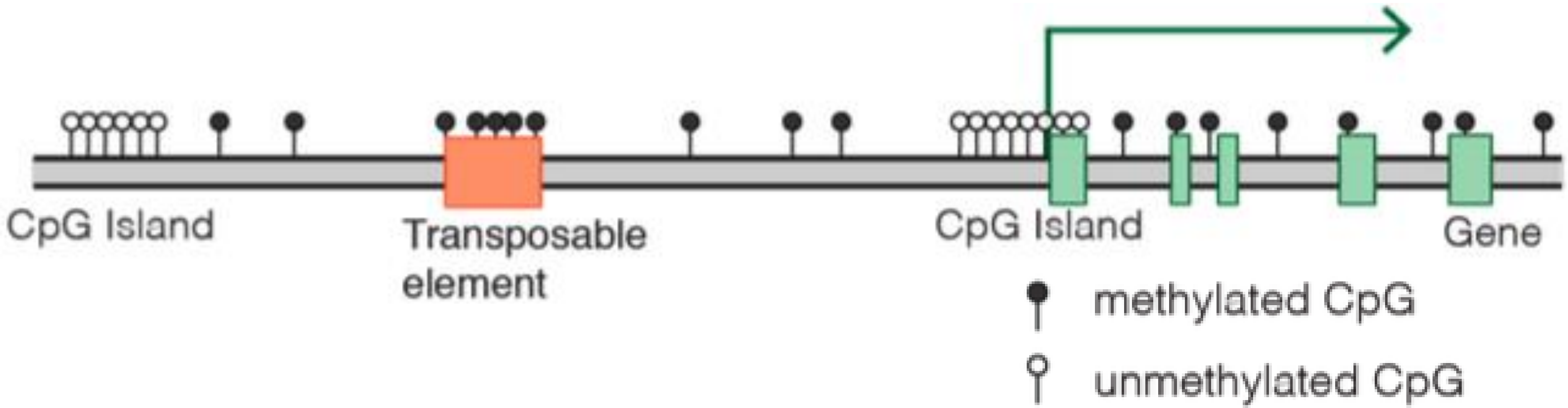
- Question: given a stretch of a genomic region, **how can we say if it comes from a CpG island or not**?

CpG islands



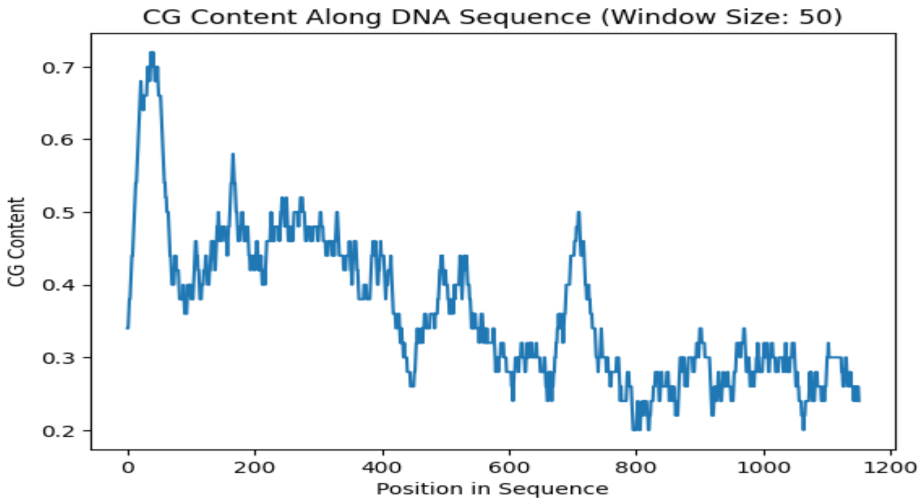
Check Seq_CG-content Notebook

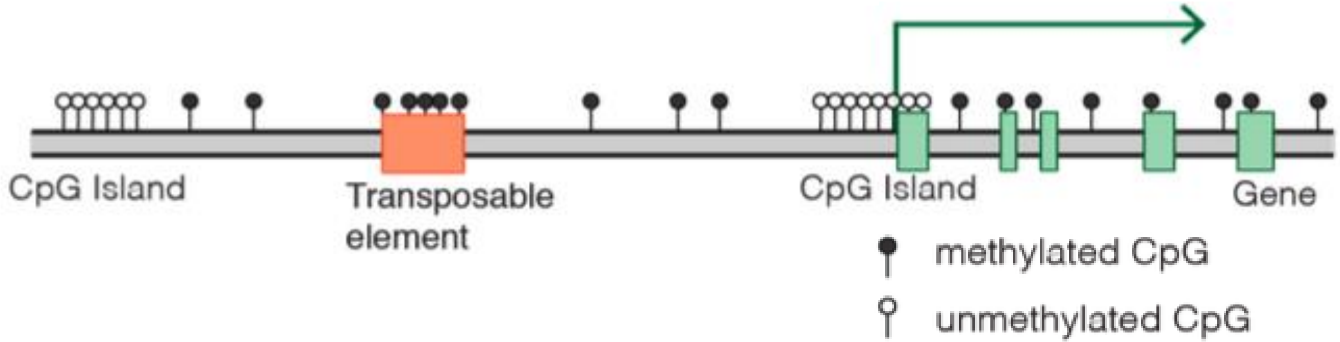
CpG islands



Is the sequence **ATCG** more likely to be from a CpG island?

Finding CpG islands





Problems with sliding window heuristics:

- What is the right window size?
  - Too small: break up real islands
  - Too large: we miss islands
- What cut-off (threshold) should we use?

*Check Seq_CG-content Notebook

Markov property example

Sequence: ATG ATG ATG ATG ATG ATG ATG ATG ATG ATG ATG ATG ATG ATG ATG ATG ATG

Table of transition probabilities

	A	G	C	T
A		↓		
G		↓		
C				
T		↑		

↓ : Low probability

↑ : High probability

- If you know you are looking at a sequence of (ATG)$_n$...
- The probability of the next character being a G?
  Depending on what character we are looking at:
  - If you are looking at a T: the odds are good that the next is a G
  - If you are looking at a A: the odds are weak that the next is a G
  - If you are looking at a G: the odds are weak that the next is a G

Markov property example

Table of transition probabilities

	A	G	C	T
A		↓		
G		↓		
C				
T		↑		

↓ : Low probability

↑ : High probability

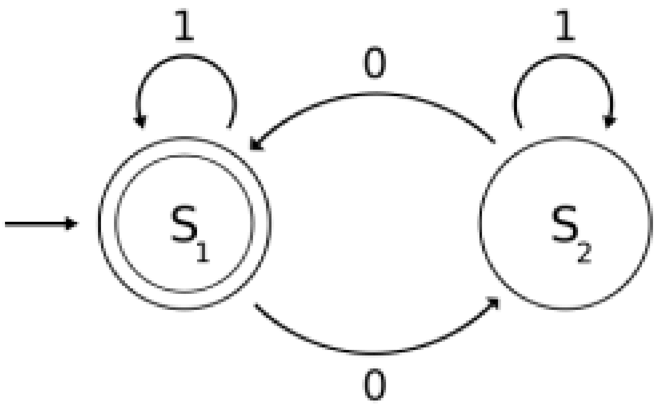
- As a result, we can model such a system with a series of transition probabilities.

- A table representing

  - If we are here…
  - What is the probability of getting there.
  - For all possible scenarios

Markov chains

- Markov Chains are implementations of models for systems with the Markov property.
- They are represented as **Deterministic Finite Automaton** (DFA).
- The **edges** (arrows) are represented as probabilities of a transition.
- The **vertices** (nodes) represent states.
- Often the DFA has “begin” and “end” states.



*A DFA requires **O(1)** memory* (constant)*,*
regardless of the length of the input.

Markov chains

1UP HIGH SCORE 2UP
00

CHARACTER / NICKNAME

-SHADOW "BLINKY"

-SPEEDY "PINKY"

-BASHFUL "INKY"

-POKEY "CLYDE"

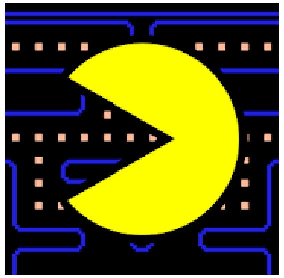
200

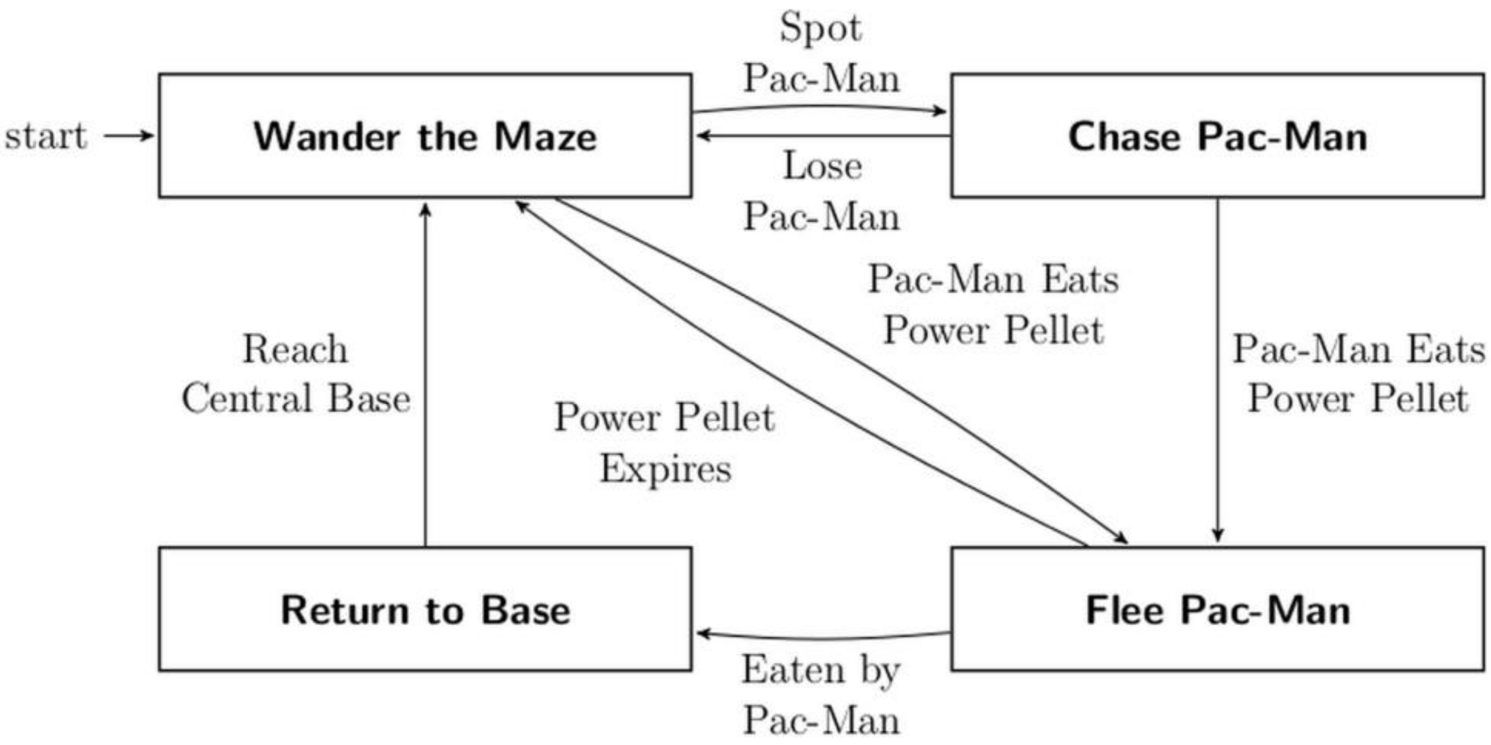
· 10 PTS

● 50 PTS

© 1980 MIDWAY MFG. CO.

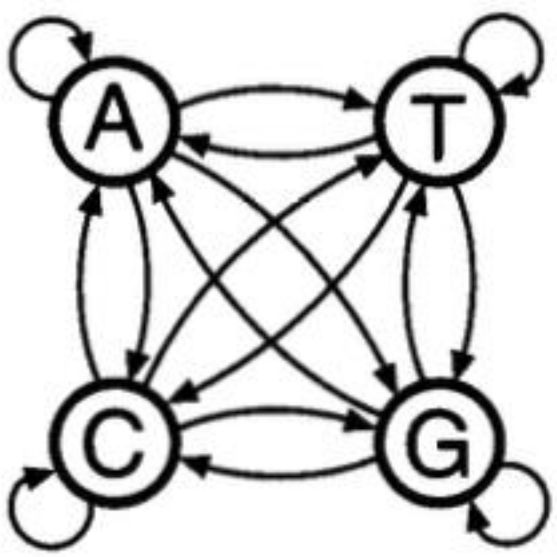
CREDIT 0





Pac-Man uses a four-state automaton

DNA Markov chains



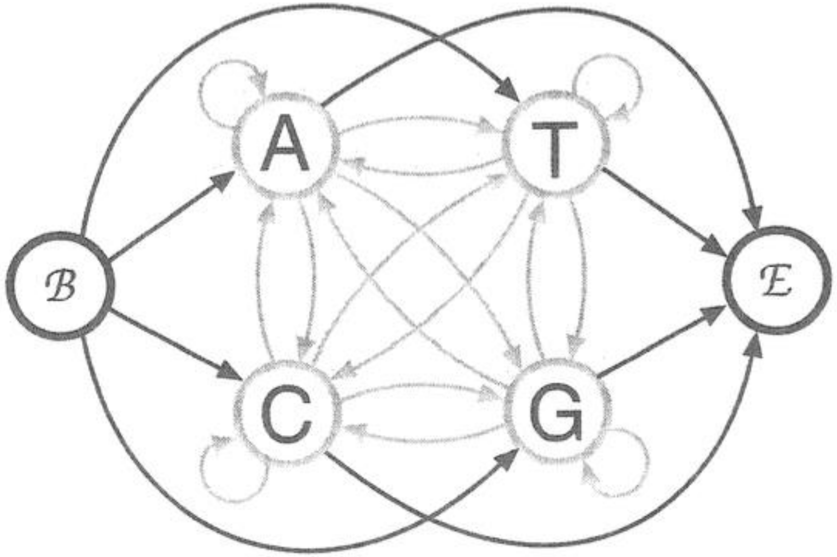
Source*: Biological sequence analysis*, Durbin

A Markov chain or DNA can be drawn with a state for each of the four letters A, C, G and T.

A probability parameter is associated with each arrow (edges): the probability of a certain residue following another residue.

The probability parameters are called **transition probabilities**.

DNA Markov chains



Source: *Biological sequence analysis*, Durbin

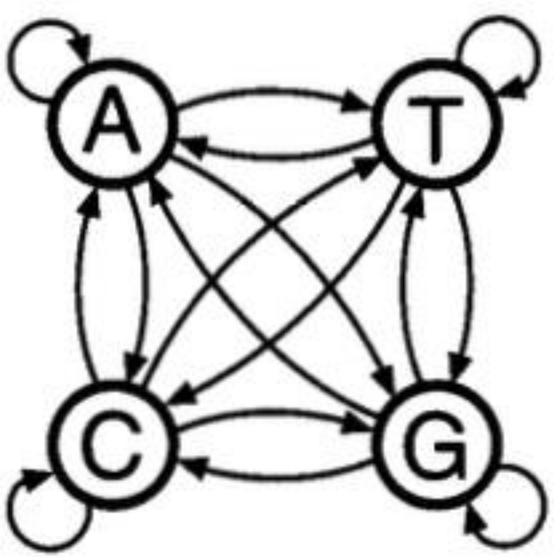
A Markov chain or DNA can be drawn with a state for each of the four letters A, C, G and T.

A probability parameter is associated with each arrow (edges): the probability of a certain residue following another residue.

The probability parameters are called **transition probabilities**.

Begin and end states can be added to the Markov Chain.

DNA Markov chains



When moving between states, it accumulates the product of probabilities.

$$\sum_{x} P(x)$$

The **probability for a given model** approaches to zero.

P = 0.5 * 0.2 * 0.1 * 0.3 ...

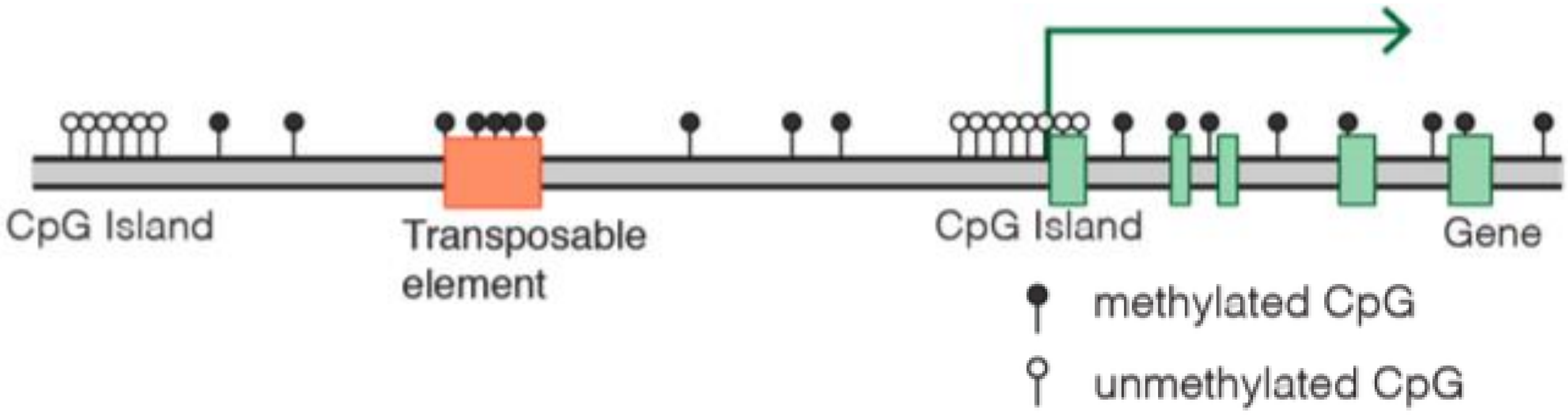
The longer the model runs, the smaller the result.

Each chain represents the probability of following the exact chain/string (eg. sequence nucleotide).

What is the sum of the probabilities of all possible sequences of length L?

L = 4
P (ACGT)
P (AGTC)
...
P (TGCA)

CpG islands



We have to train our model for:

- ➢ CpG island
- ➢ Non-CpG island (or CpG oceans)

Compute the probability of each possible transition.

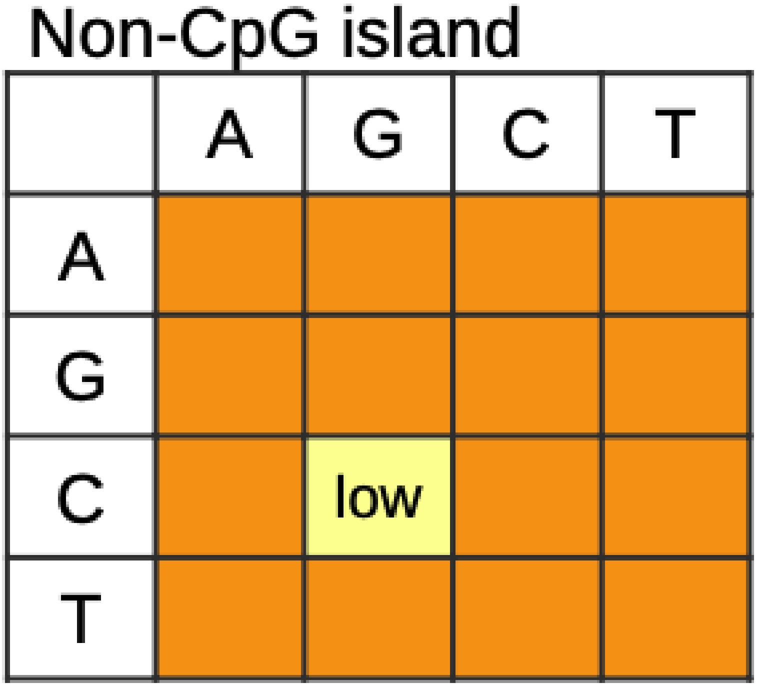
$$\frac{\sum_{\forall i,\forall j} transition_{island} i \rightarrow j}{number\ of\ all\ possible\ transitions_{island}}$$

CpG islands

Table of transition probabilities

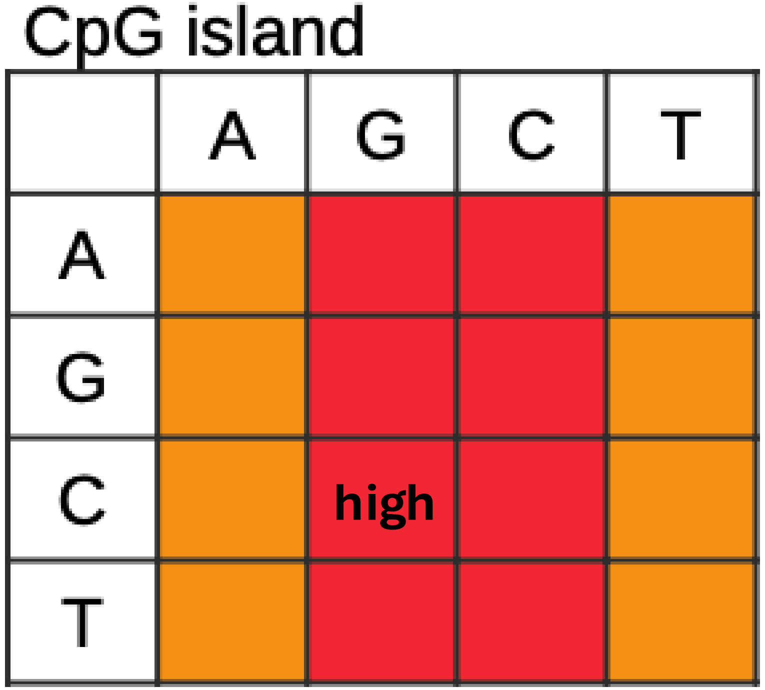
- Rows = from
- Columns = to

- Red = high probability
- Orange = medium
- Yellow = low



Non-CpG island

	A	G	C	T
A	🟧	🟧	🟧	🟧
G	🟧	🟧	🟧	🟧
C	🟧	🟨 low	🟧	🟧
T	🟧	🟧	🟧	🟧

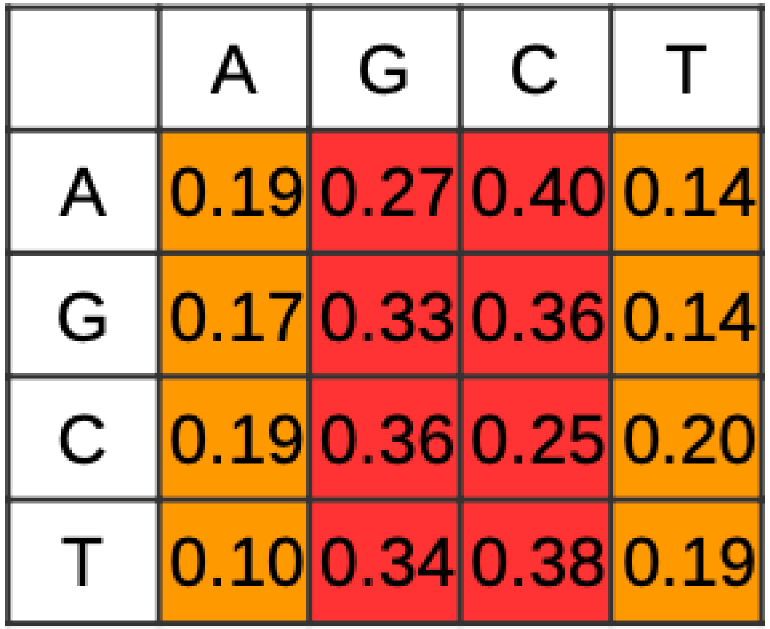


CpG island

	A	G	C	T
A	🟧	🟥	🟥	🟧
G	🟧	🟥	🟥	🟧
C	🟧	🟥 high	🟥	🟧
T	🟧	🟥	🟥	🟧

CpG islands

Probabilities



	A	G	C	T
A	0.19	0.27	0.40	0.14
G	0.17	0.33	0.36	0.14
C	0.19	0.36	0.25	0.20
T	0.10	0.34	0.38	0.19

<u>What's the probability of the sequence ATCG in a CpG island?</u>

x = ATCG

P(x) = P(x4|x3)P(x3|x2)P(x2|x1)P(x1)

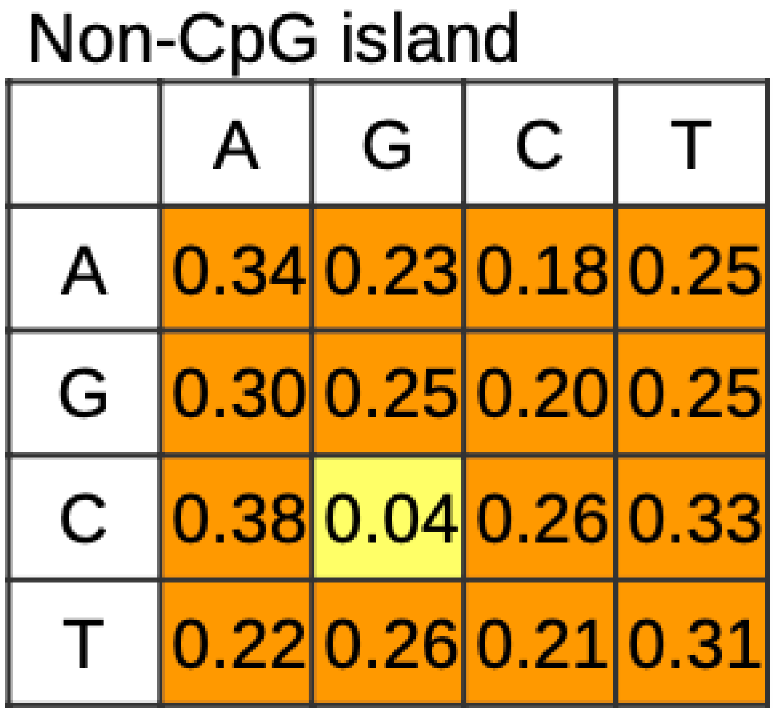
P(x) = P(G|C)P(C|T)P(T|A)P(A)

P(x)=0.36 *0.38 *0.14*0.16
*P(A) approx. = mean P(A|X)=0.16

Simulating
P(C | A) = 0.40
Building
P(C | A) = # times AC occurs / # times AX occurs

CpG islands

Probabilities

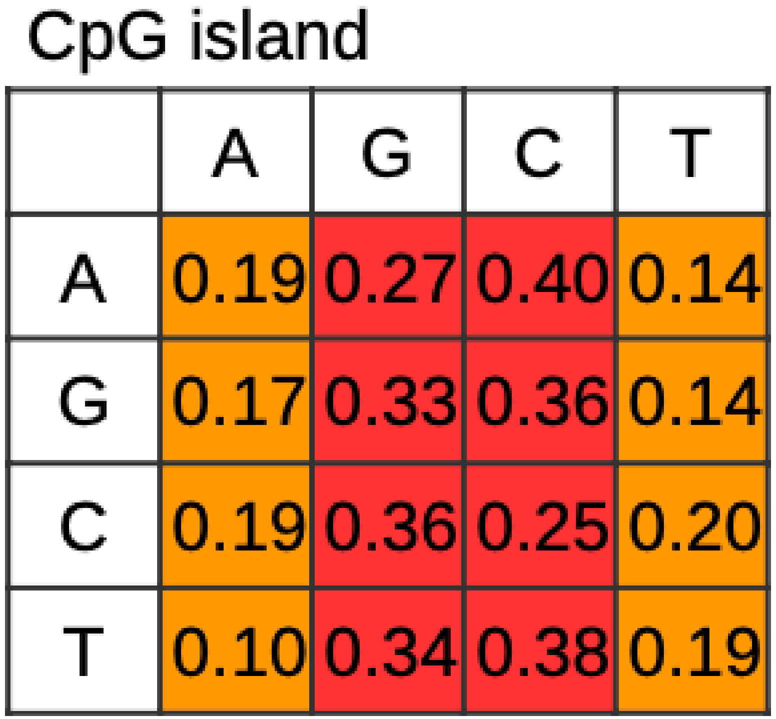


Non-CpG island

	A	G	C	T
A	0.34	0.23	0.18	0.25
G	0.30	0.25	0.20	0.25
C	0.38	0.04	0.26	0.33
T	0.22	0.26	0.21	0.31

Which one is more likely?

Which model/scenario (CpG island or non-CpG island) is more likely for the sequence ATCG to be?



CpG island

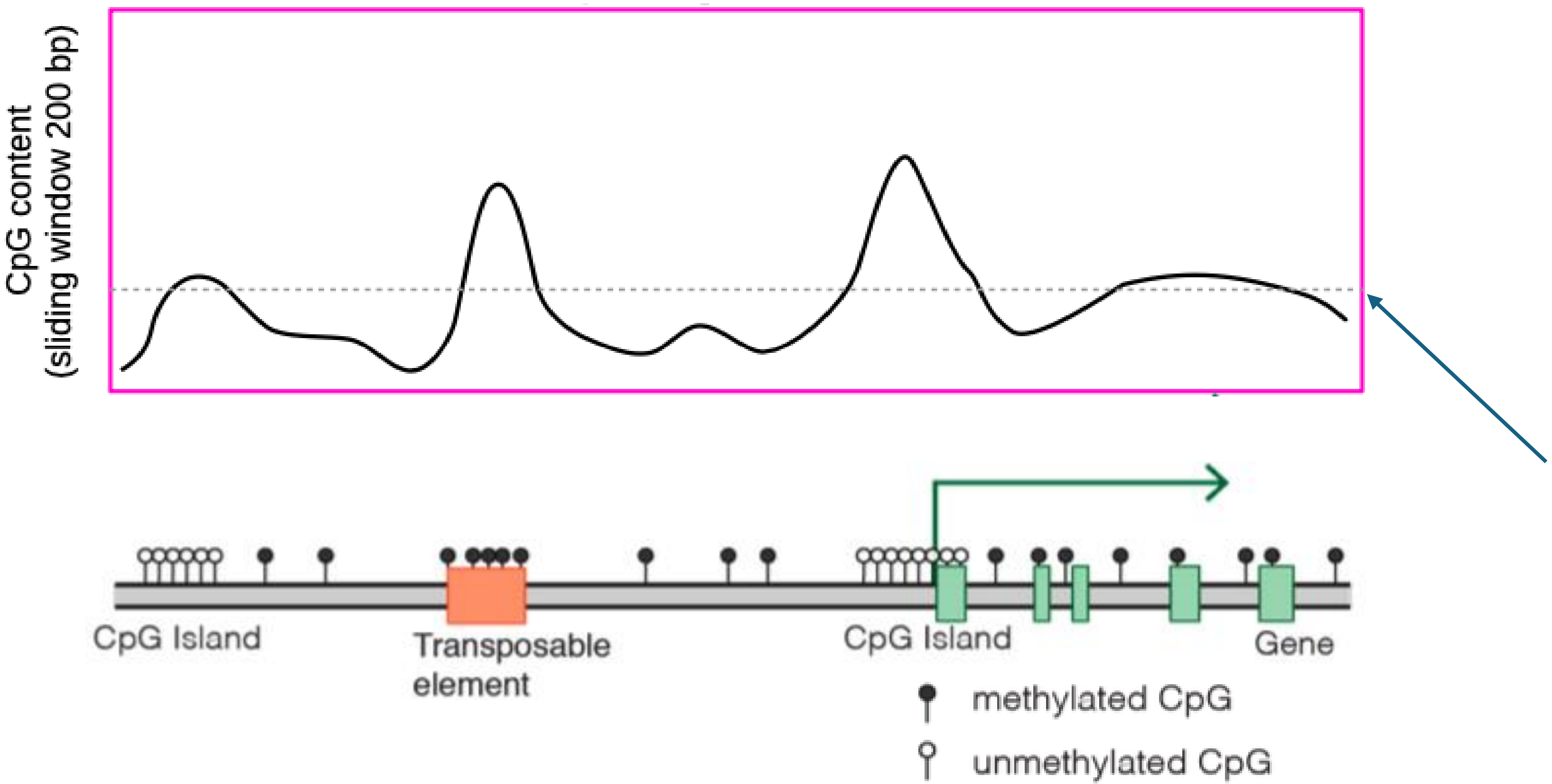
	A	G	C	T
A	0.19	0.27	0.40	0.14
G	0.17	0.33	0.36	0.14
C	0.19	0.36	0.25	0.20
T	0.10	0.34	0.38	0.19

x = ATCG
P(x) = 0.04*0.21*0.25*0.31
P(x) = 0.000651

4.7 time more likely

x = ATCG
P(x) = 0.36*0.38*0.14*0.16
P(x) = 0.00306

Finding CpG islands



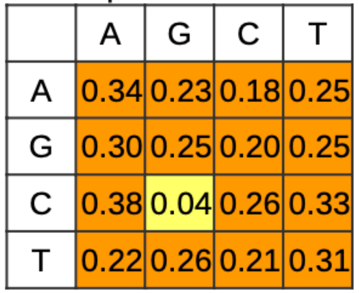
Problems with sliding window heuristics:

- What is the right window size?
  - Too small: break up real islands
  - Too large: we miss islands
- What cut-off (threshold) should we use?

*Check Seq_CG-content Notebook

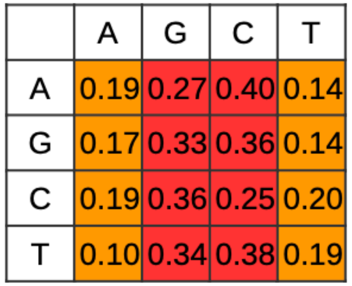
Finding CpG islands
Markov Models

Non-CpG island



	A	G	C	T
A	0.34	0.23	0.18	0.25
G	0.30	0.25	0.20	0.25
C	0.38	0.04	0.26	0.33
T	0.22	0.26	0.21	0.31

CpG island

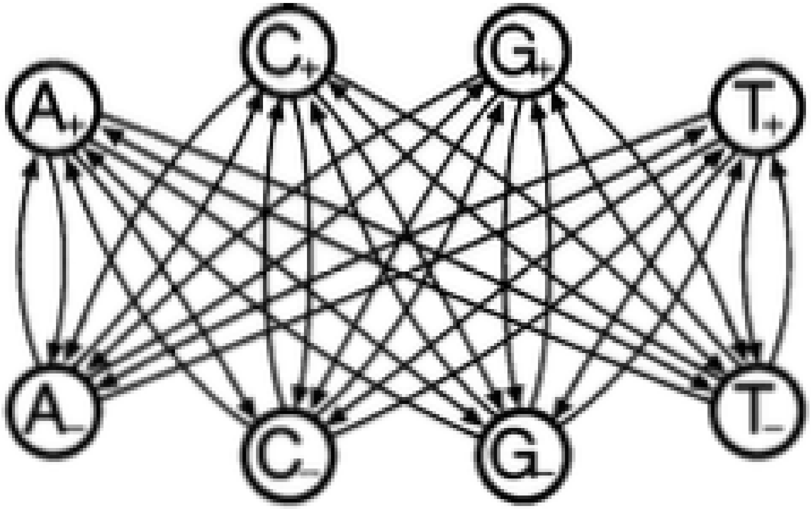


	A	G	C	T
A	0.19	0.27	0.40	0.14
G	0.17	0.33	0.36	0.14
C	0.19	0.36	0.25	0.20
T	0.10	0.34	0.38	0.19

- In order to train the model, you must know up front whether the training data came from an **island** or an **ocean**.
- The **cut-off problem is solved** with MMs: calculate which model is more likely.
- We can use Markov Chains for discrimination.
- But we still have to deal with size and boundary problems: “*Distinguishing the* ***shorelines***.”

Hidden Markov Models

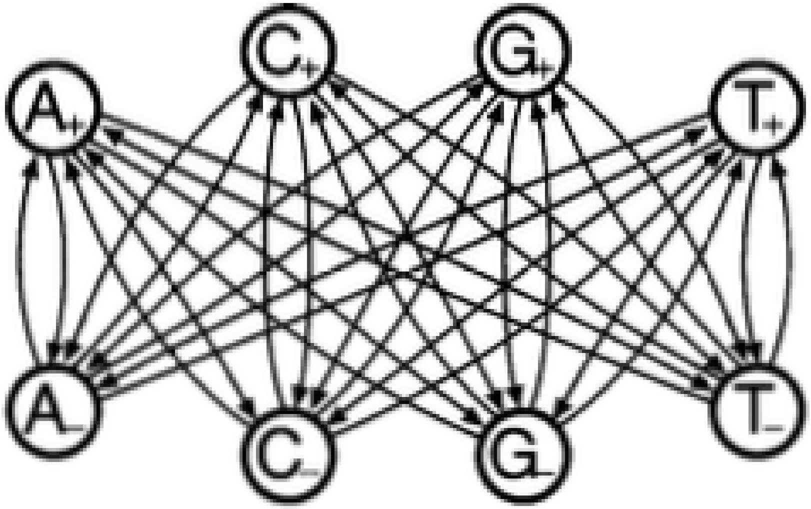
- We can use Hidden Markov Models (HMMs) to identify the boundaries of the CpG island.
- CpG island boundaries are ”sharp” but with variable length.
- We can implement a model with an additional set of transition probabilities and changes of states.
- To simulate in one model the “islands” in an “ocean” of non-island genomic sequence, we want to have both Markov chains present in the same model.
- We relabel the states with “+” and “-” symbols.
- Look for the most probable state path.



Source*: Biological sequence analysis*, Durbin

Hidden Markov Models

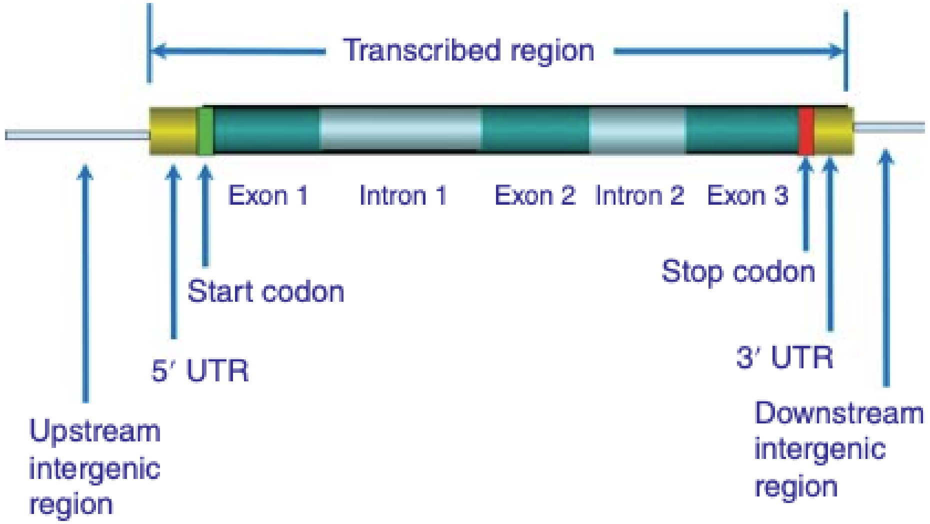
- A set of states (eg. CpG island, CpG ocean).
- A set of symbols to be determine (A, C, G, T)
- A set of **emission probabilities** for each symbol from each stated
- An index (eg. next nucleotide)
- A **transition probability** between successive states.
- **What other states can we look for in a genome?**



Source*: Biological sequence analysis*, Durbin

Hidden Markov Models

- A set of states (eg. CpG island, CpG ocean).
- A set of symbols to be determine (A, C, G, T)
- A set of emission probabilities for each symbol from each stated
- An index (eg. next nucleotide)
- A transition probability between successive states.
- **What other states can we look for in a genome?**
- Hidden Markov Models (HMM) forms the core component of most gene predictors.



Hidden Markov Models in gene prediction

- Hidden Markov models (HMMs) are used to provide a **statistical representation** of real biological processes.

- HMMs are used in speech recognition, facial recognition, and other applications.

- They have found widespread use in many areas of bioinformatics, including multiple sequence alignment, the characterization and classification of protein families, the comparison of protein structures, and the prediction of gene structure.

- In general, gene-finding methods use a raw nucleotide sequence as their input and, for each position in the sequence, they attempt to predict whether **a given base is most likely found in an intron, an exon, or within an intergenic region**.

- In making these predictions, the algorithm applied (variation of HMMs) must consider what is known about the structure of a gene for that specific genome or taxa.

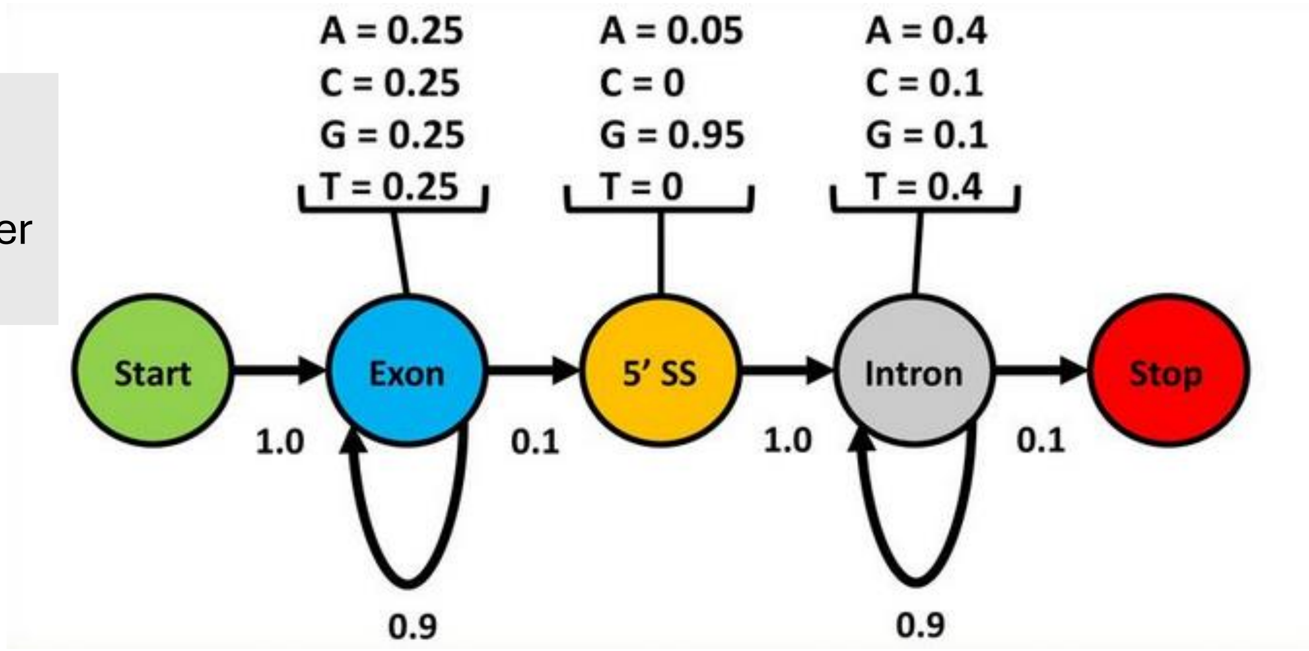
- Each of the elements – exons, introns, and so forth – are referred to as ***states***.

HMM Probabilities

- The probability of switching from one state to another (eg. exon -> intron) is called **transition probability**.

- The probability of observing a nucleotide (A, T, C, G) that is of a certain state (exon, intron, splice junction) is called an **emission probability**. (eg. the probability of observing an adenine in an exon).

The probability of switching from one state type to another (ex. Exon - Intron).

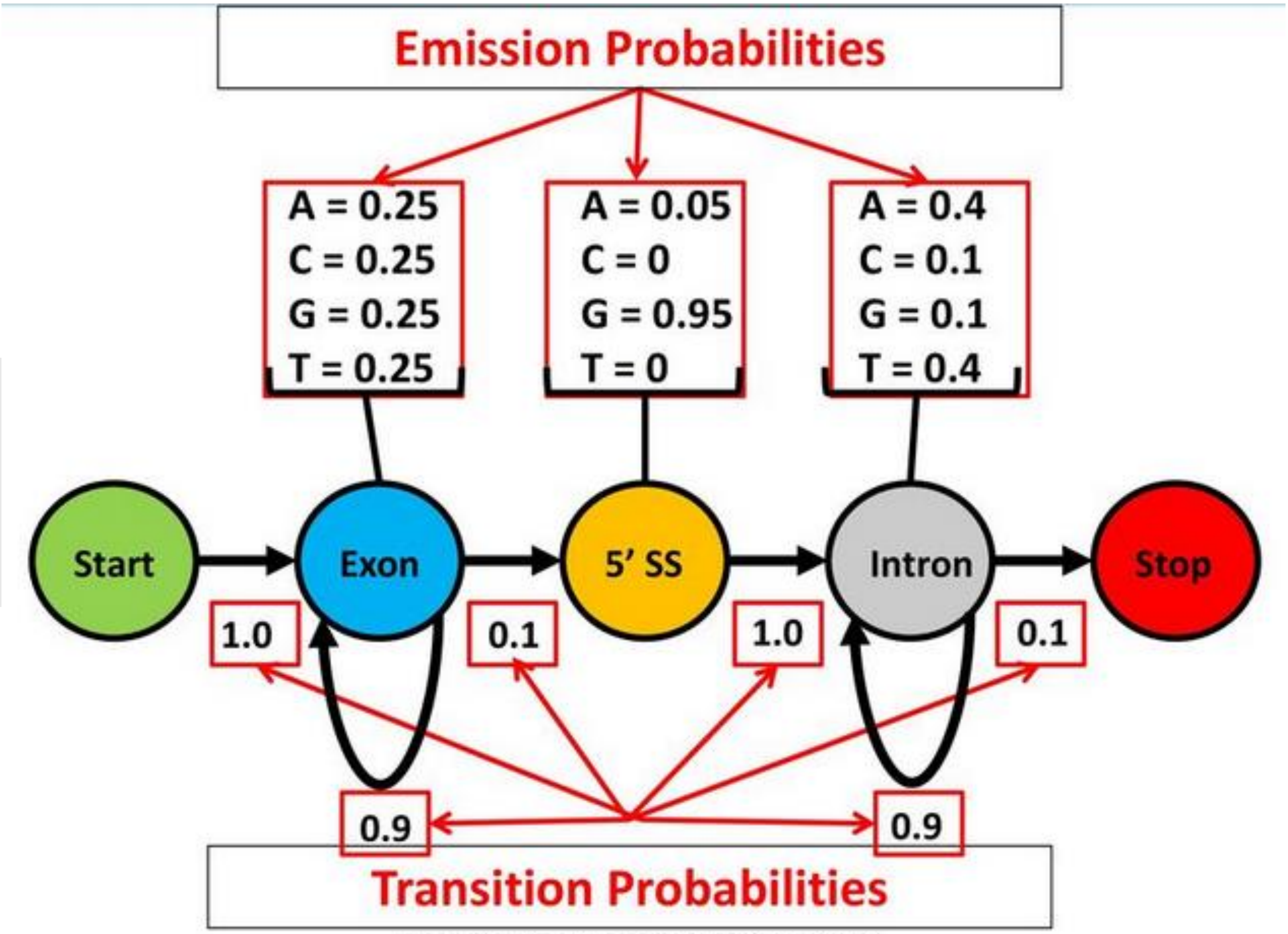


Source: *Introduction to HMMs*, Weisstein

SS Splice Site

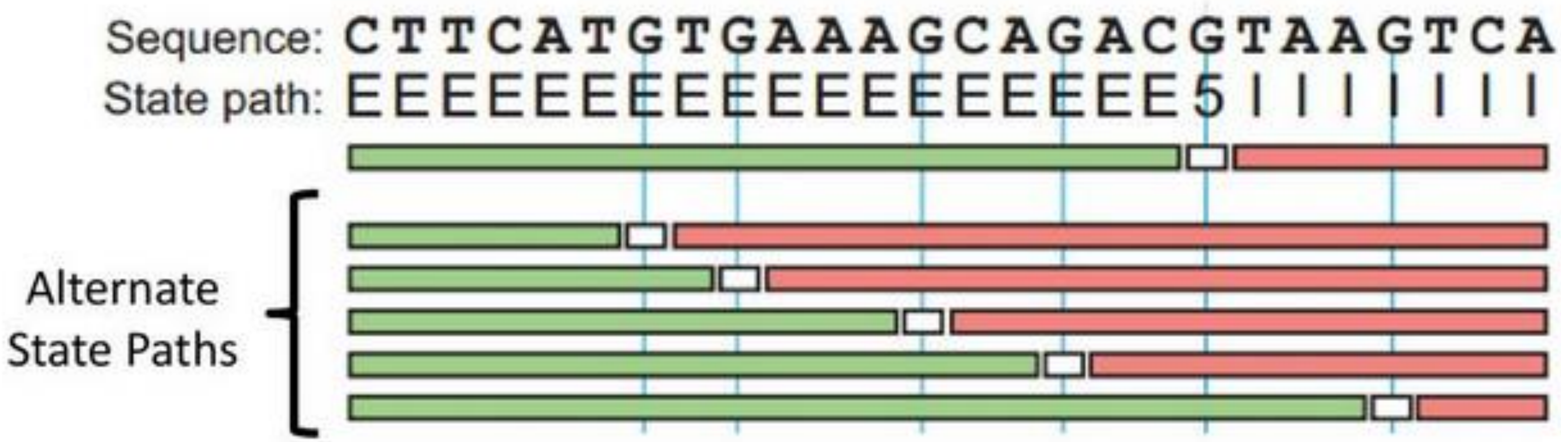
HMM Probabilities

The probability of switching from one state type to another (ex. Exon - Intron).



Source: *Introduction to HMMs*, Weisstein

HMM Probabilities


Sequence: CTTCATGTGAAAGCAGACGTAAGTCA
State path: EEEEEEEEEEEEEEEEEE5IIIIIII
Alternate State Paths

- A **state path** is the list of states (E: exon; I: intron; 5: 5’ splice junction).

- An HMM can produce many state paths for a single sequence.

- We use an algorithm to determine what is the most likely path given the emissions:
  - **Viterbi algorithm**:
    - Calculates a transition matrix.
    - Use dynamic programing to find the most probable path

Eukaryotic gene prediction

- We must train or know intron-exon and exon-intron junctions.

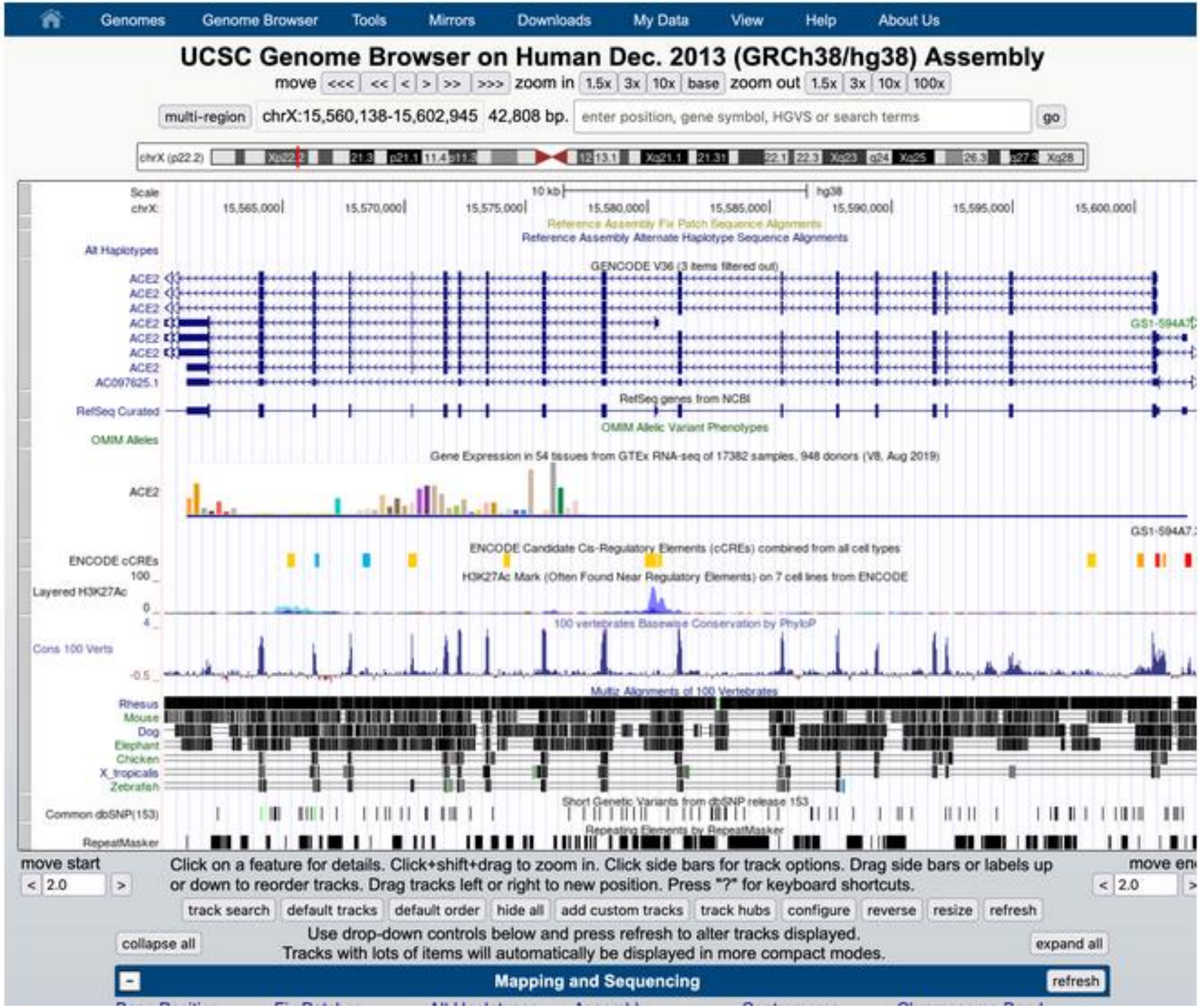
- The gene finder must find promoter motifs.

- Must have ORF awareness.

- HMMs are the core of several gene prediction algorithms:

    - GenScan
    - Augustus
    - GeneMark
    - GRAIL
    - Twinscan

- Gene prediction accuracy depends partly on transition probabilities calculated based on the training data.



Genomes
Genome Browser
Tools
Mirrors
Downloads
My Data
View
Help
About Us

UCSC Genome Browser on Human Dec. 2013 (GRCh38/hg38) Assembly

move <<< << < > >> >>> zoom in 1.5x 3x 10x base zoom out 1.5x 3x 10x 100x
multi-region

chrX:15,560,138-15,602,945 42,808 bp.

enter position, gene symbol, HGVS or search terms
go
chrX (p22.2)
Scale
10 kb
hg38
chrX:
15,565,000
15,570,000
15,575,000
15,580,000
15,585,000
15,590,000
15,595,000
15,600,000
Reference Assembly Fix Patch Sequence Alignments
Reference Assembly Alternate Haplotype Sequence Alignments
Alt Haplotypes
GENCODE V36 (3 items filtered out)
ACE2
ACE2
ACE2
ACE2
ACE2
ACE2
ACE2
AC097625.1
GS1-594A7.
RefSeq genes from NCBI
RefSeq Curated
OMIM Allelic Variant Phenotypes
OMIM Alleles
Gene Expression in 54 tissues from GTEx RNA-seq of 17382 samples, 948 donors (V8, Aug 2019)
ACE2
GS1-594A7.
ENCODE Candidate Cis-Regulatory Elements (cCREs) combined from all cell types
ENCODE cCREs
100 _
H3K27Ac Mark (Often Found Near Regulatory Elements) on 7 cell lines from ENCODE
Layered H3K27Ac
0
4 _
100 vertebrates Basewise Conservation by PhyloP
Cons 100 Verts
-0.5 _
Multiz Alignments of 100 Vertebrates
Rhesus
Mouse
Dog
Elephant
Chicken
X_tropicalis
Zebrafish
Short Genetic Variants from dbSNP release 153
Common dbSNP(153)
Repeating Elements by RepeatMasker
RepeatMasker
move start
< 2.0 >
move end
< 2.0 >

Click on a feature for details. Click+shift+drag to zoom in. Click side bars for track options. Drag side bars or labels up or down to reorder tracks. Drag tracks left or right to new position. Press "?" for keyboard shortcuts.

track search
default tracks
default order
hide all
add custom tracks
track hubs
configure
reverse
resize
refresh
collapse all
expand all

Use drop-down controls below and press refresh to alter tracks displayed.
Tracks with lots of items will automatically be displayed in more compact modes.

Thursday Lab10

Gene prediction in prokaryotes

GeneMarkS-2 (https://genemark.bme.gatech.edu/)