SCIE2100 | BINF6000

Bioinformatics

Genome Analysis II

Atefeh Taherian Fard, PhD

Australian Institute for Bioengineering and Nanotechnology

a.taherianfard@uq.edu.au

Outline

Lecture 1:

- Overview genome sequencing and sequencing technologies
- Genome re-sequencing
- De-novo genome assembly

Lecture 2:

- Gene features in prokaryotes
- Gene features in eukaryotes
- Computational approaches for gene prediction
- Functional genome annotation

Gene finding Approaches

- Physical, genetic or other ***experimental approaches***
  - *e.g.* Genetic knockouts

- ***Computational approaches***

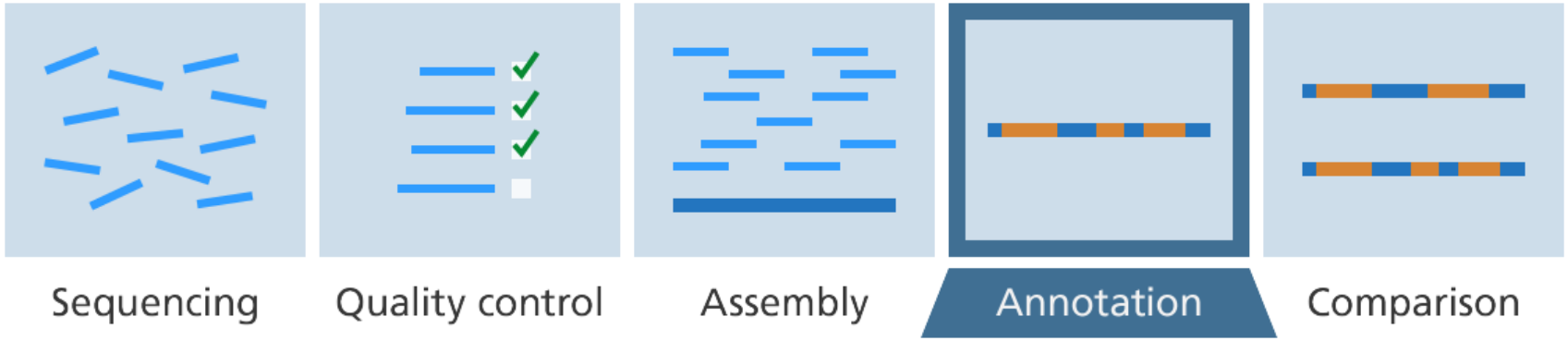
1) Identity search

2) Similarity search Homology based

3) *Ab initio* approaches

Slides adapted from Lutz Krause

www.yourgenome.org



Using computational methods, find all genes (or other elements) in a long, unannotated string of nucleotides.

```
ACCGGTCAATAGCCGCAGACTACGGCATTTCAGAGGGACAGGCACTATAGCAACTAGCAACCCCCGTATAATACAAGGAGGCT
CAAGCTCCACTCTGACTCTCAACTTATTACGCTGTCACTCGATACGGCAGGGGCATTTAGACTTACGGCATATACCCGCCCGA
TCCAGCTTACGATACTACTGCTACTGGATACCCTGATAGCCAATCATTACGACTACTACTACGGCATTTCAGACCCGACAGGC
ACTAGAGCAACTAGCAACACCCGTATAATACAAGGAGGCTCAAGCTCCAGCTCTCACTGCAGCTATGTGGTGCACACATGTGC
ATCGTATGACTCAGTCGATGCTATCACGTACATCGTGTGGGTGCACACCACCCATGCCCTGATAGCCCCTGATTTTAGCCCCA
GCATTATTTTTCCGACGAGATCACGTACCCCCTACGGCATTTCAGAGGGACAGGGGACGCGCCCAATTACGACTACTACTACG
GCATTTCAGACCCGACAGGCACTAGAGCAACTAGCAACACCCGTATAATACAAGGAGGCTCAAGCTCCAGCCTTTTCAACAGA
CCGGGCGTTACGGTAAAAAAAAATCCGGCCGTACGGACTACTGGATACCGCAGACTACGGCATTTCAGAGGGACAGGCACTAT
AGCAACTAGCAACCCCCGTATAATACAAGGAGGCTCAAGCTCCACTCTGACTCTCAACTTATGACAGGGGACGATGACTCAGT
CGATTTCGCTATCACGTAAAACATCGTGTGGGGTGCACACCACCGCATGCCCTTTCAGGATAGCCCCTGATTTTAAGCCCCAG
CATTATTTGGTTCCGACGAGATCACGTACCCCACTACGGCATTTCAGAGGGACACTCAGTCGATGCTATCACGTACATCGTGT
GGGTGCTTACACCACGCCATGCCCTGATAGCCCCTGGGGATTTTAGCCCCAGCATTAATTTCTTCCGACGAGCCCTCAGACCC
GACAGGGGCACTAGAGCAACTATATAAGCAACACCCGTATACCCATACCAAGGAGGCTCAAGCTCCAGCCTTCTTCAACAGGA
CCGGGCGGGATTCCACATCATTCATGGGCAGCATCCCAGCAAACCCACGGCATAAGGACCACCCCTCGGCTAAGCAATCGCAT
AATACGGCGCTGCGCTCTACGTCTAGAGCTCACCATCTTACGAGGCTCTACCTCTATG...[3 BILLION or so MORE]
```

Aim: To identify transcriptional unit

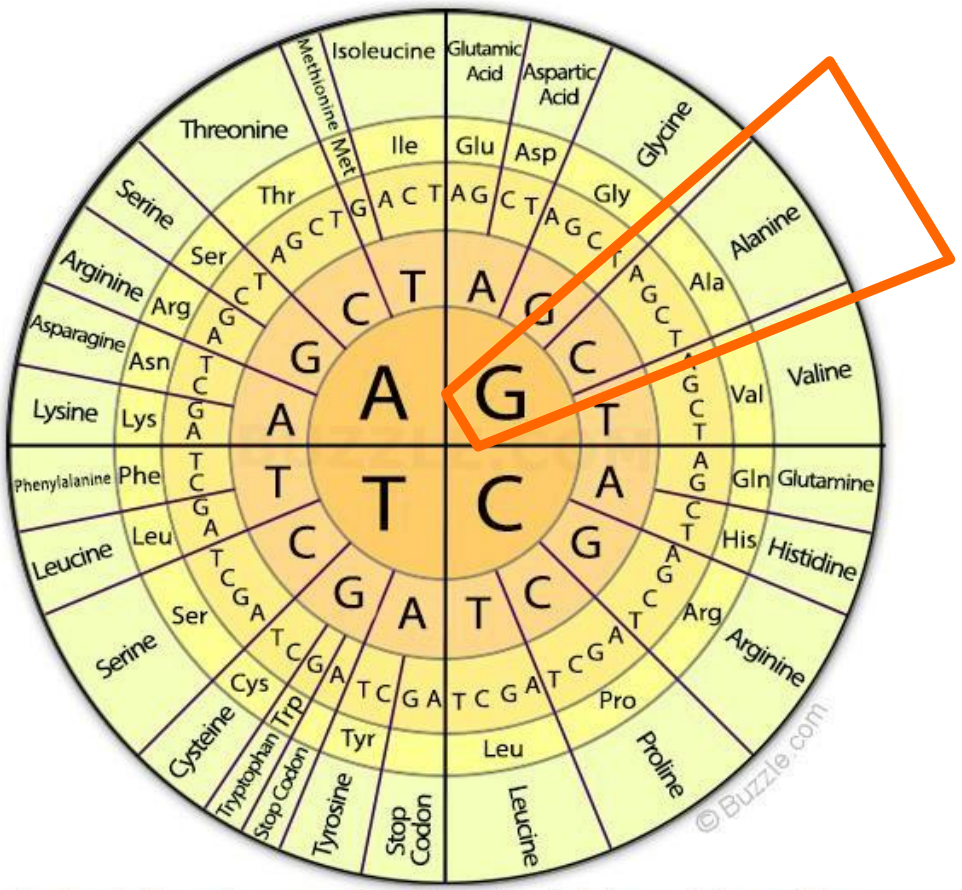
What do we know? Know only approximately what they look like

How? Find their locations and boundaries as accurately as possible, overlook as few as possible, and report as few non-genes as possible.

Adapted from Mark Ragan

Codon bias

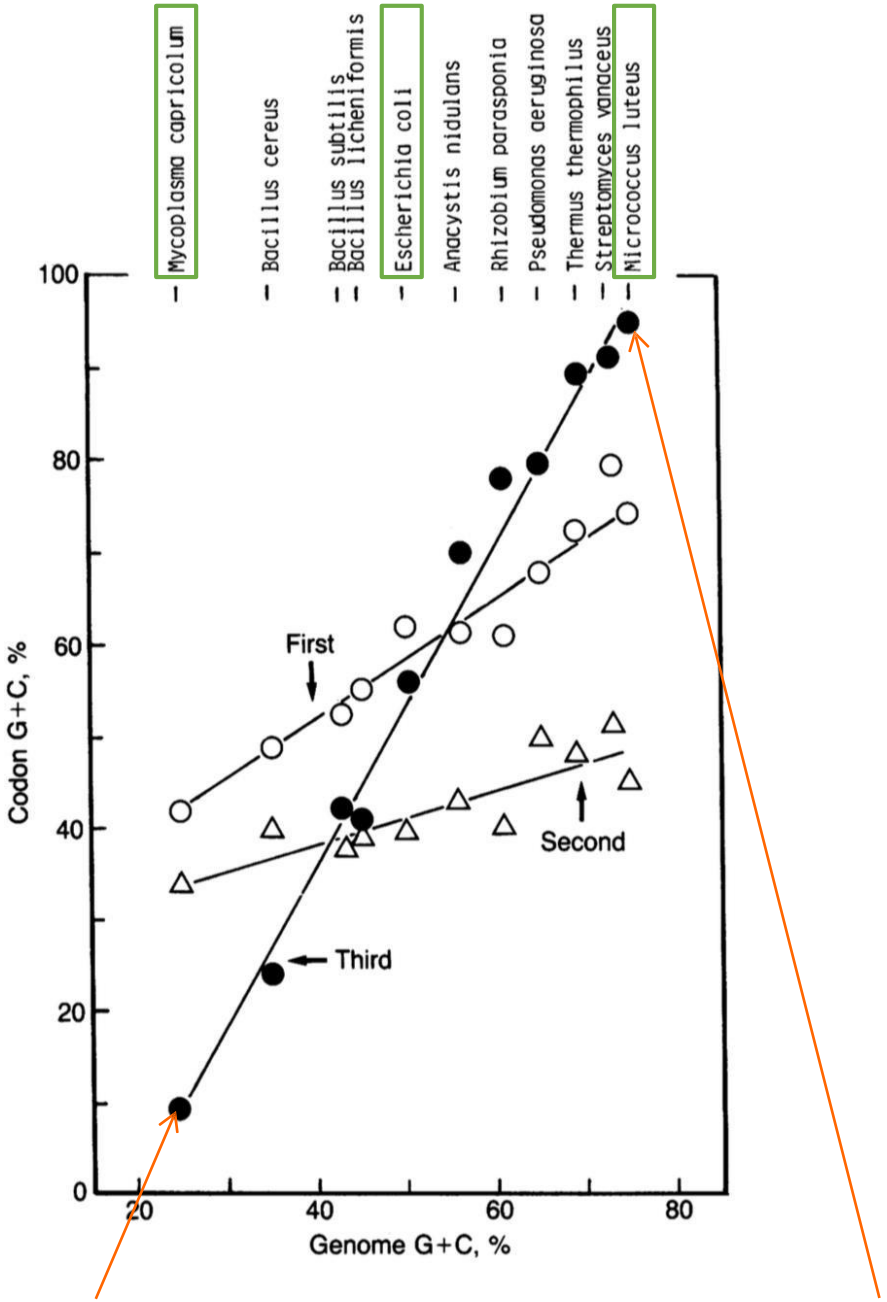
- 61 codons encoding 20 amino acids, genetic code is redundant
- Codon bias: Each organism seems to prefer a different set of codons over others

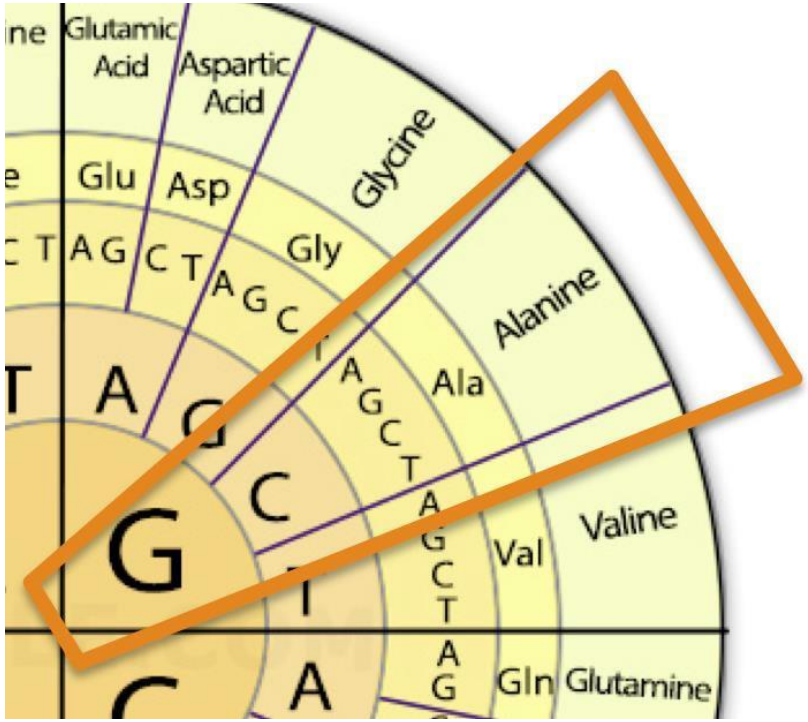


To decode the codon, move from the center circle towards the periphery.

- This redundancy is mainly at the 3rd codon position
- Example 1: The codons for alanine can have ANY base in the 3rd position
- Examples 2: Leu is encoded by 6 codons. But human in nuclear genes it's most frequently coded by **CTG**, and only rarely by **TTA** or **CTA**

http://www.buzzle.com/articles/codon-chart.html





Very high correlation between **genomic** GC content and **3rd codon position GC** usage

=> Codons in GC-rich genomes will have higher frequency of GC at 3rd position

GCA, GCT usage more common

GCC, GCG usage more common

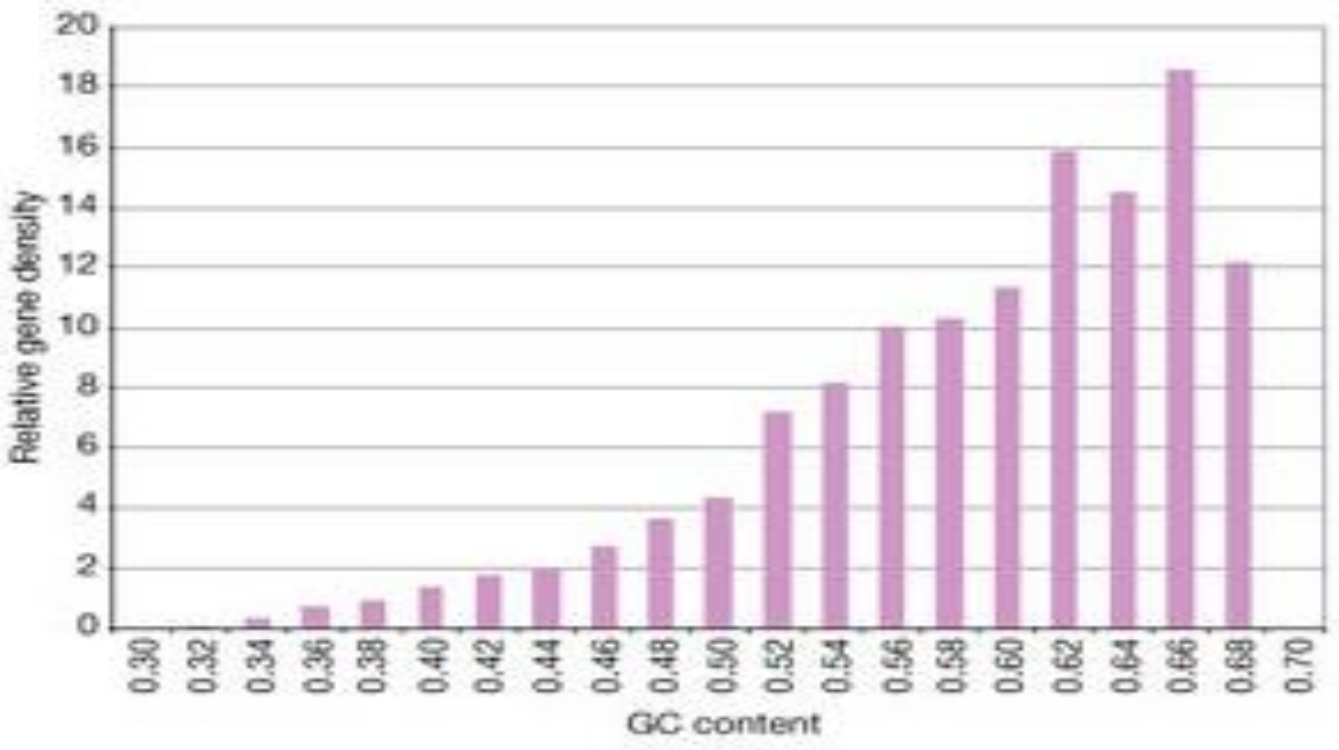
Muto and Osawa, 1987

CpG islands, (G+C)-rich regions and genes

- For slightly more than half of human genes, transcriptional start site occurs in a CpG-rich region
- *This is particularly true for "housekeeping" genes:*

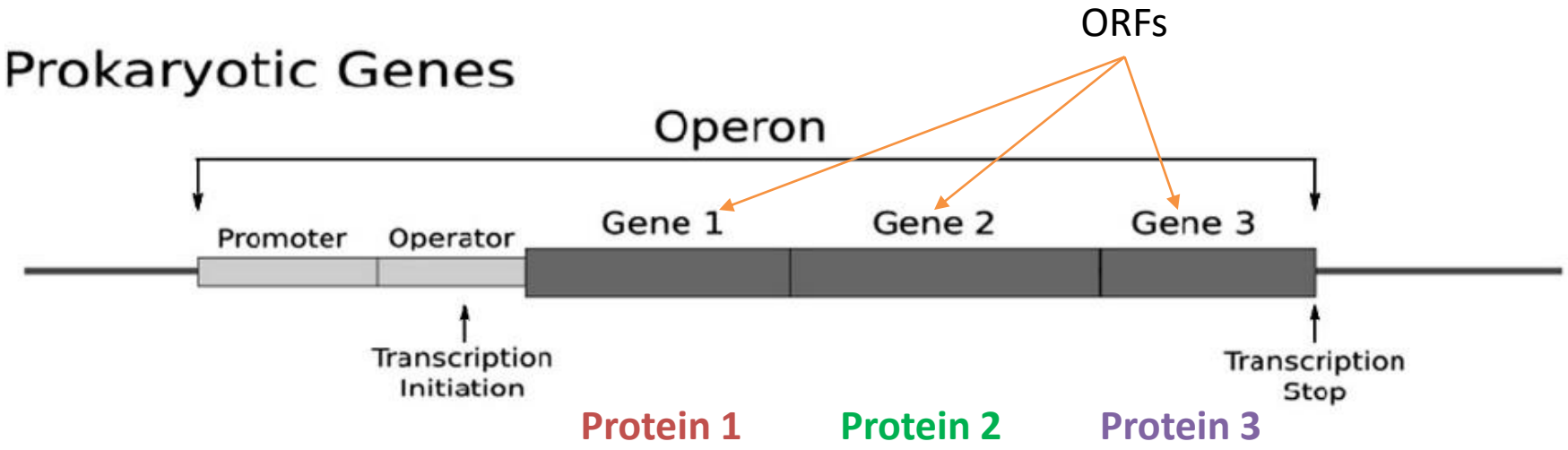
the first coding exon usually occurs in a CpG island

- Mammalian genomes show elevated gene density in (G+C)-rich regions



Brown, Genomes 2

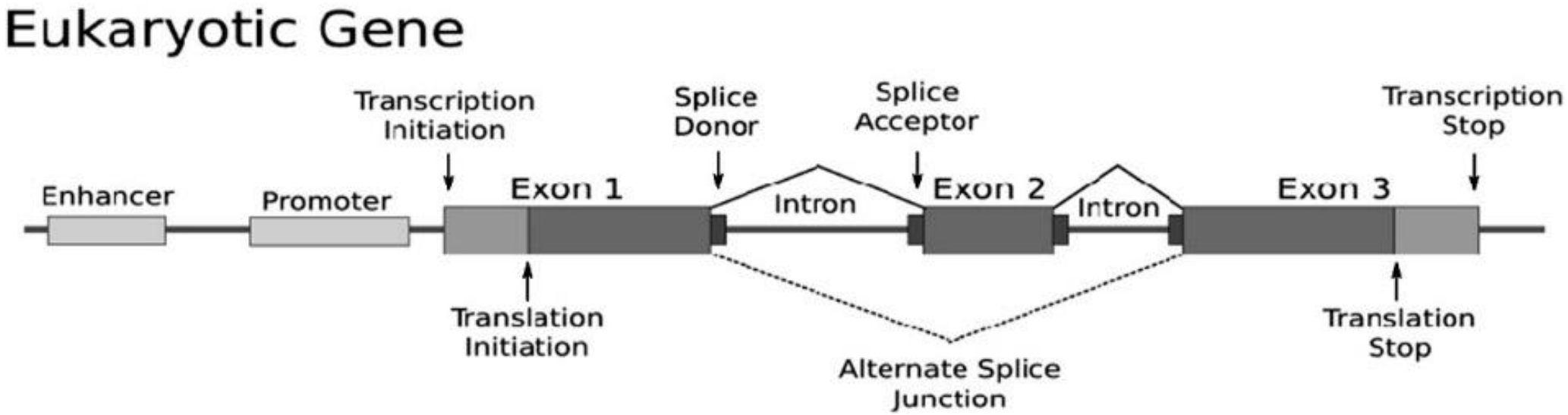
Gene features in Prokaryotes



Protein coding genes in prokaryotes:

- Are open reading frames (ORFs)
- Begin with start codon (usually ATG for Methionine)
- End with stop codon
- No in-frame stop-codon

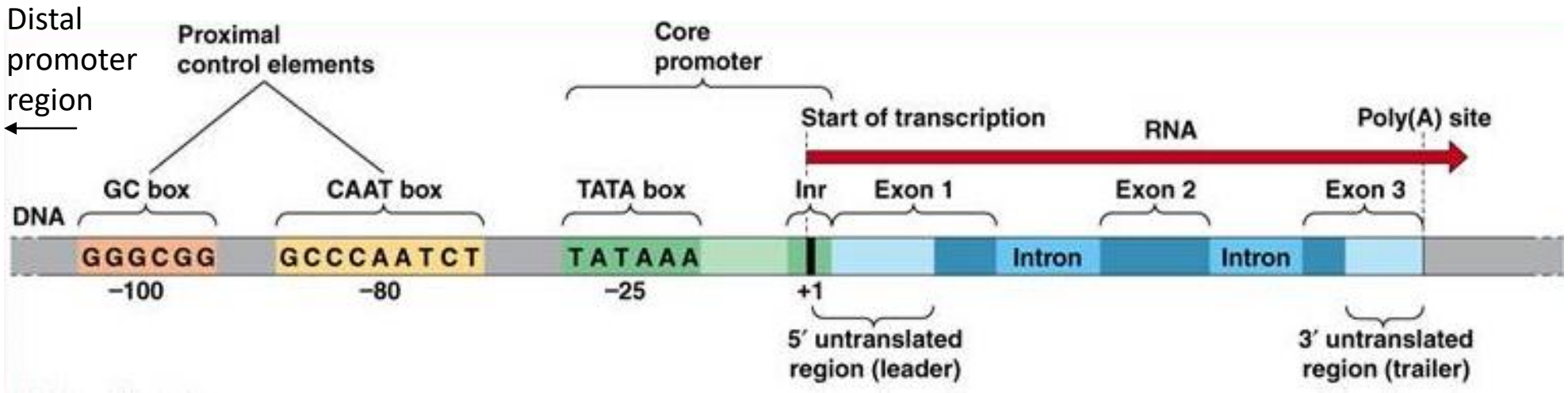
Gene features in Eukaryotes



Protein coding genes in Eukaryotes:

- Exons and introns
- Regulatory sequences (promoter and enhancer)
- Splice donor and accepter sides
- 5’ Cap and 3’ PolyA tail

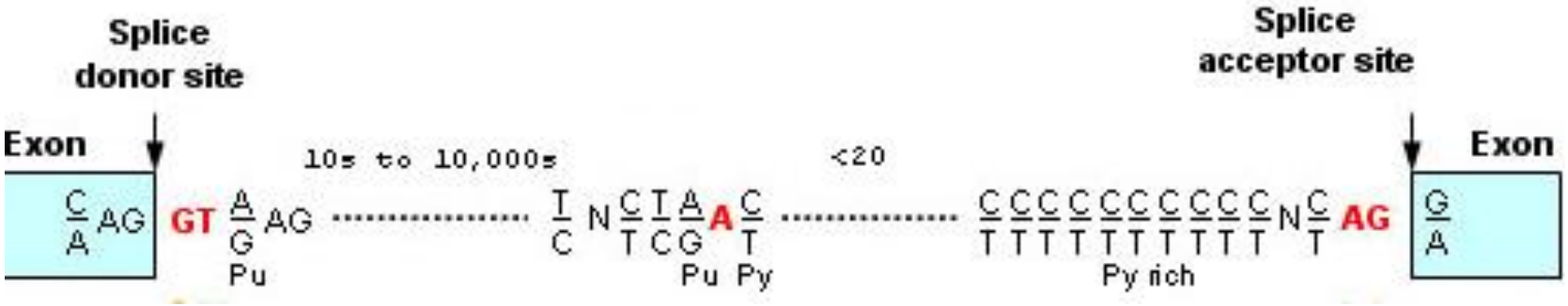
Regulatory promoter regions in Eukaryotic genes



- Regulatory sequences have distinct features, recognised by DNA-binding proteins to regulate gene expression
- Sequence motifs, may be conserved
- In each species, different sets of genes have different motifs

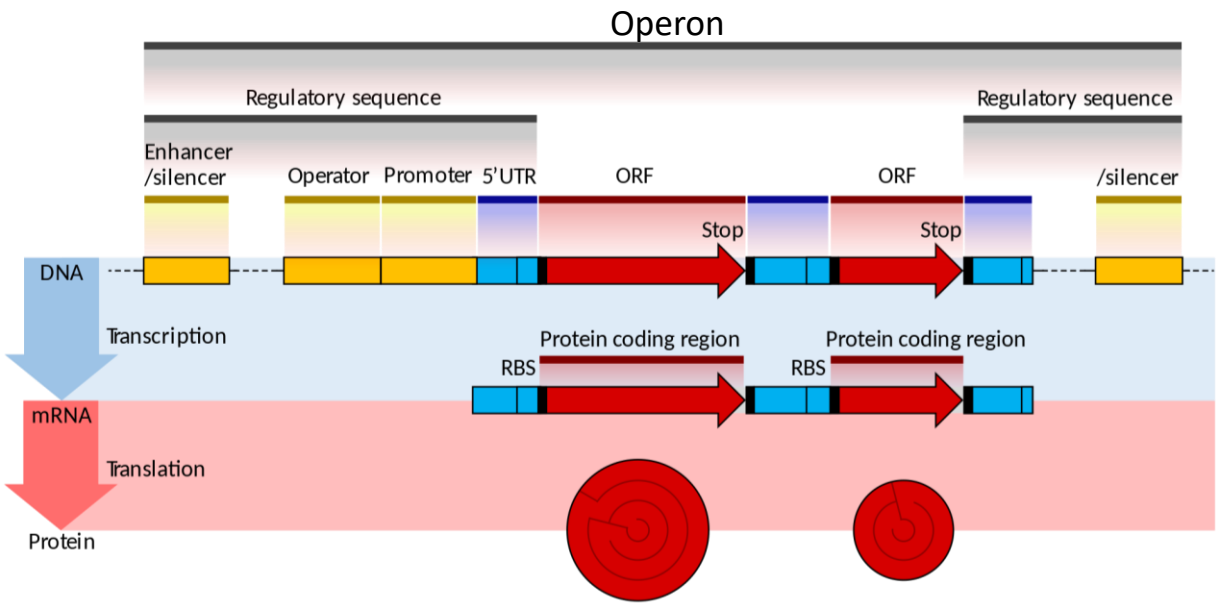
http://www.mun.ca/biology/

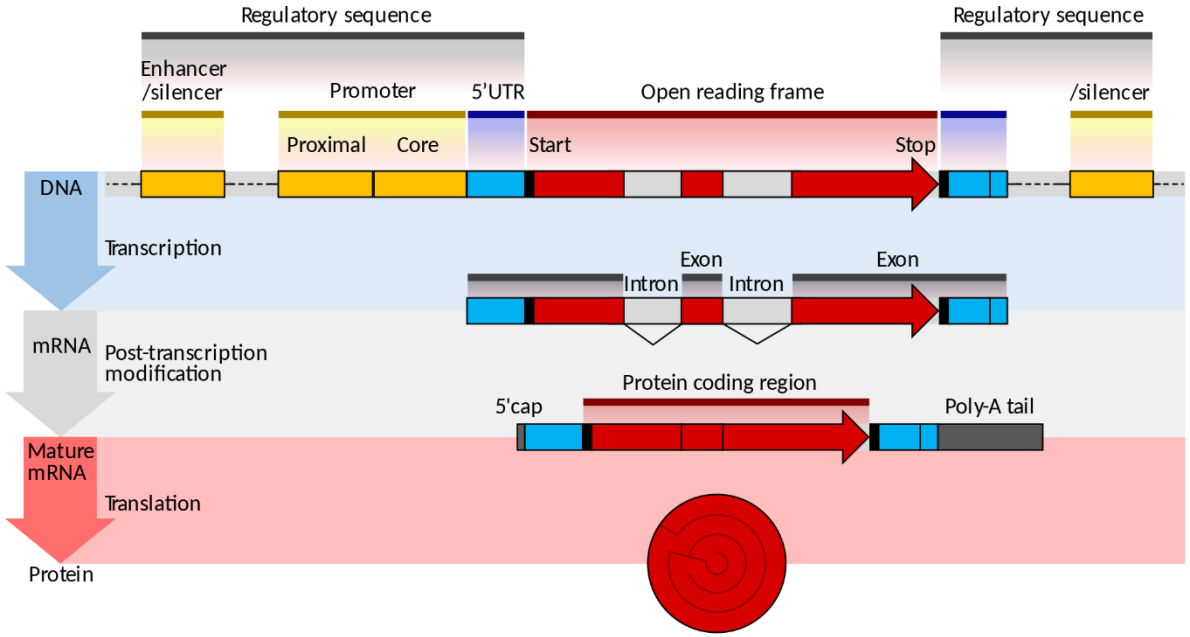
Intron-exon splice sites



- The GT-AG rule
- Numbers indicate the number of nucleotide bases between each conserved region
- Pu: Purines (A and G)
- Py: Pyrimidines (C and T)
- N: Any nucleotide

Summary: Gene features in Prokaryotics vs Eukaryotes





- Codon bias and GC rich regions
- Transcriptional start and stop sites
- ORFs: Start and stop codons
- 5’ UTR: Ribosomal Binding Sequence site
- 3’UTR

- Codon bias and GC rich regions
- Promoter regions
- Intro and Exon splice site
- ORFs: Start and Stop codons
- 5’ UTR: 5’ Cap (G cap site)
- 3’ UTR: PolyAs

Shafee & Lowe (2017)

Gene finding Approaches

- ***Computational approaches***

1) Identity search

- Looks for exact match the query and target sequence!

2) Similarity search- Homology based

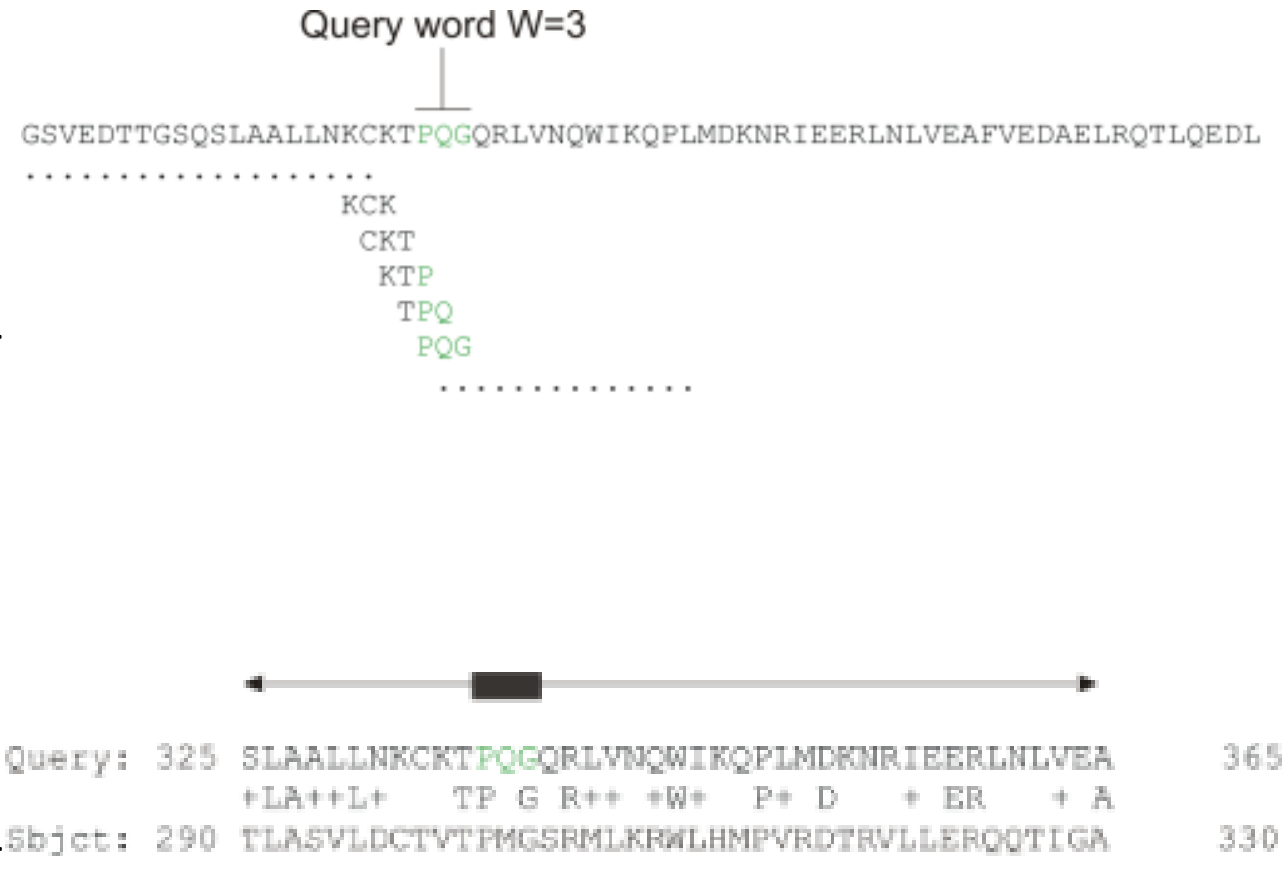
- Translate genome in all 6 reading frames, map translated sequence to known proteins (*e.g.* using BLAST)
- Only identifies known genes
- Species specific
- Unknown genes and evolved genes with low sequence identity are missed
- Compare genome to close relative organism and identify conserved regions

3) *Ab initio* approaches

- Evaluate ***specific sequence properties of genes*** to distinguish between non-coding and coding regions
- Identifies also new genes, but may produce many false predictions

Similarity search (e.g. BLAST)

- BLAST identifies homologous sequences using a heuristic method which initially finds short matches between two sequences.
- When finding a match between a query sequence and a hit sequence, the starting point is the *words* that the two sequences have in common.
- After initial finding of words (seeding), the BLAST algorithm will extend the (only 3 residues long) alignment in both directions.
- Each time the alignment is extended, an alignment score is increases/decreased.
- When the alignment score drops below a predefined threshold, the extension of the alignment stops.
- If the obtained alignment receives a score above a certain threshold, it will be included in the final BLAST result.

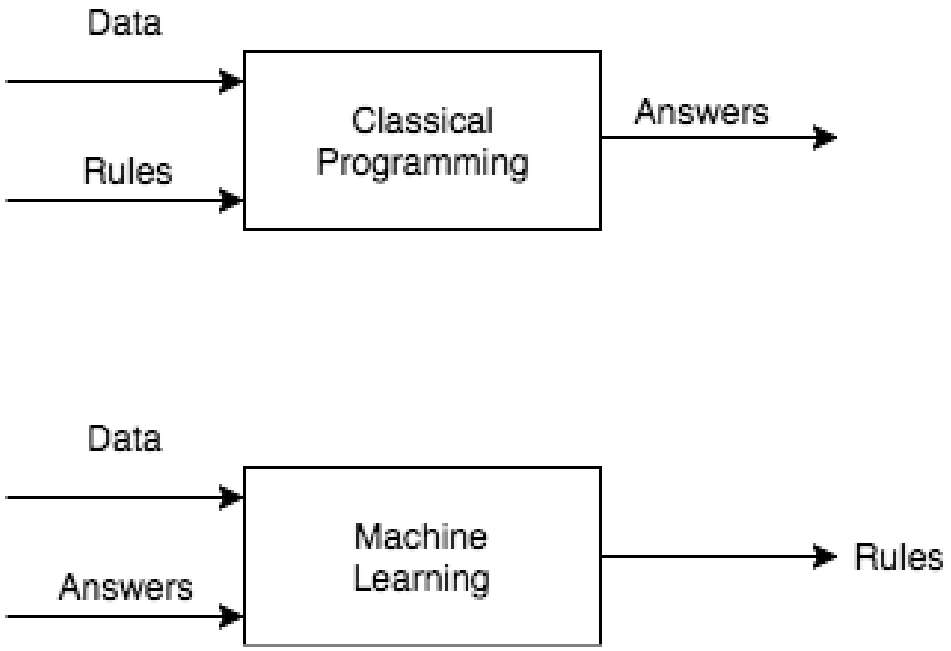


http://resources.qiagenbioinformatics.com/

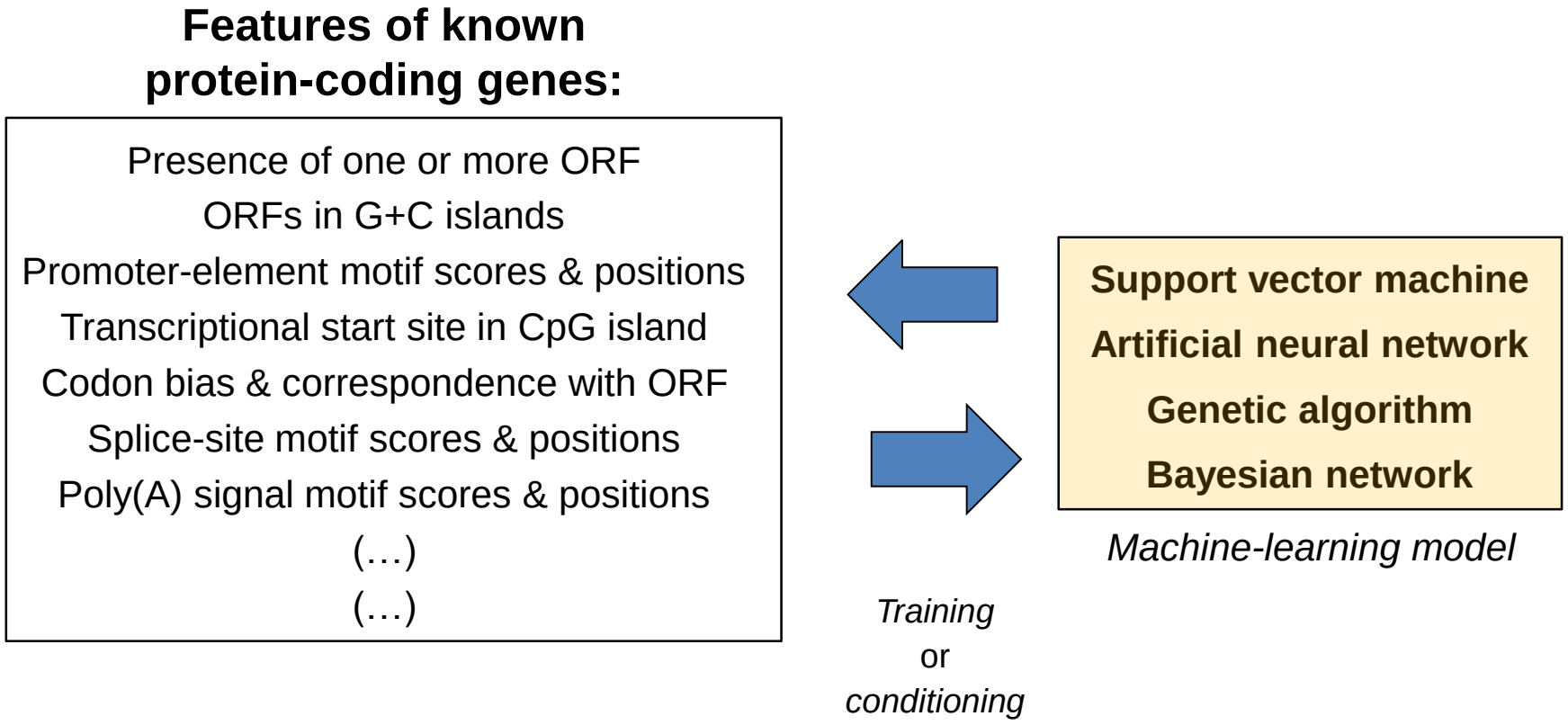
Machine learning approach to *ab initio* gene finding

- A branch of artificial intelligence
- Enabling computers make successful predictions using past experiences
- Based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention

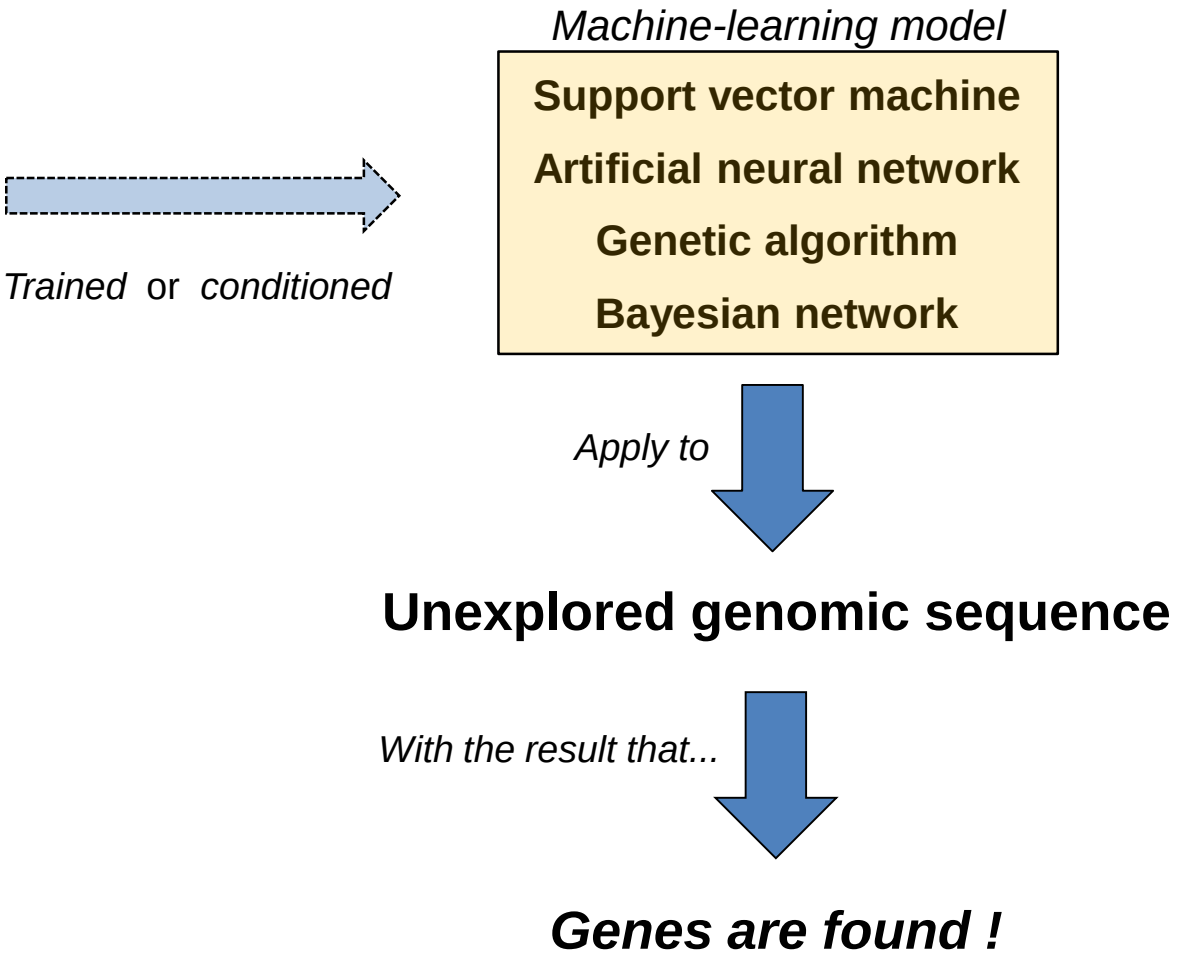
Machine Learning vs Traditional programming



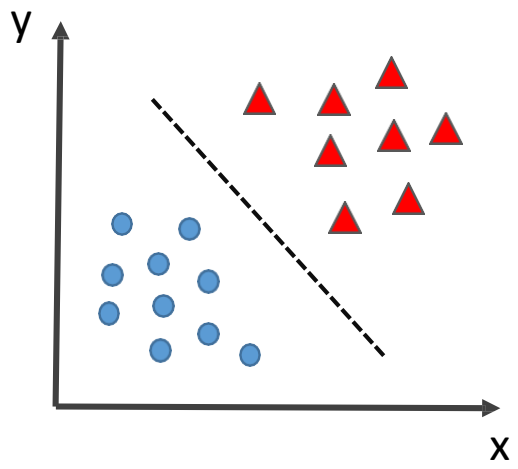
Machine learning approach to ab initio gene finding



Machine learning approach to ab initio gene finding



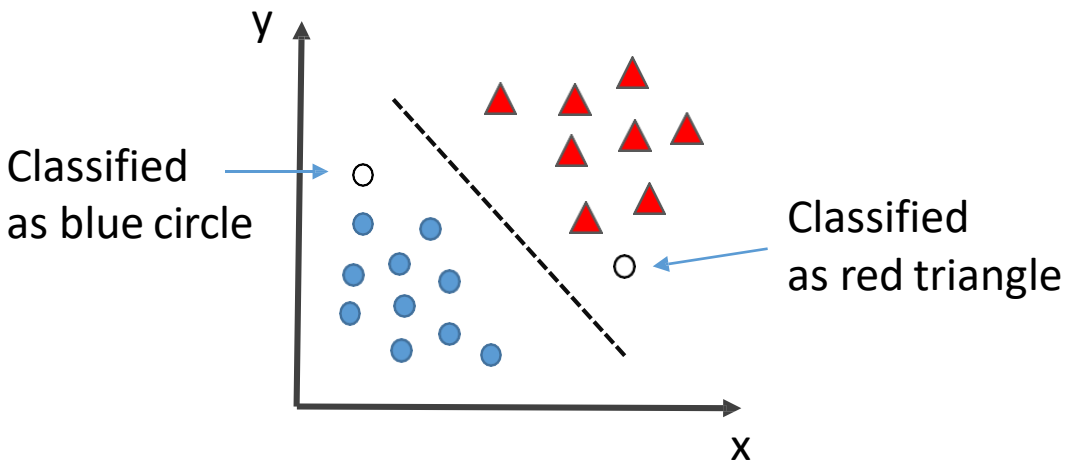
Support Vector Machines (SVMs)



Training:

- Given a training set of items from two classes a SVM learns a hyperplane that separates the items from both classes

Support Vector Machines (SVMs)



Training:

- Given a training set of items from two classes a SVM learns a hyperplane that separates the items from both classes

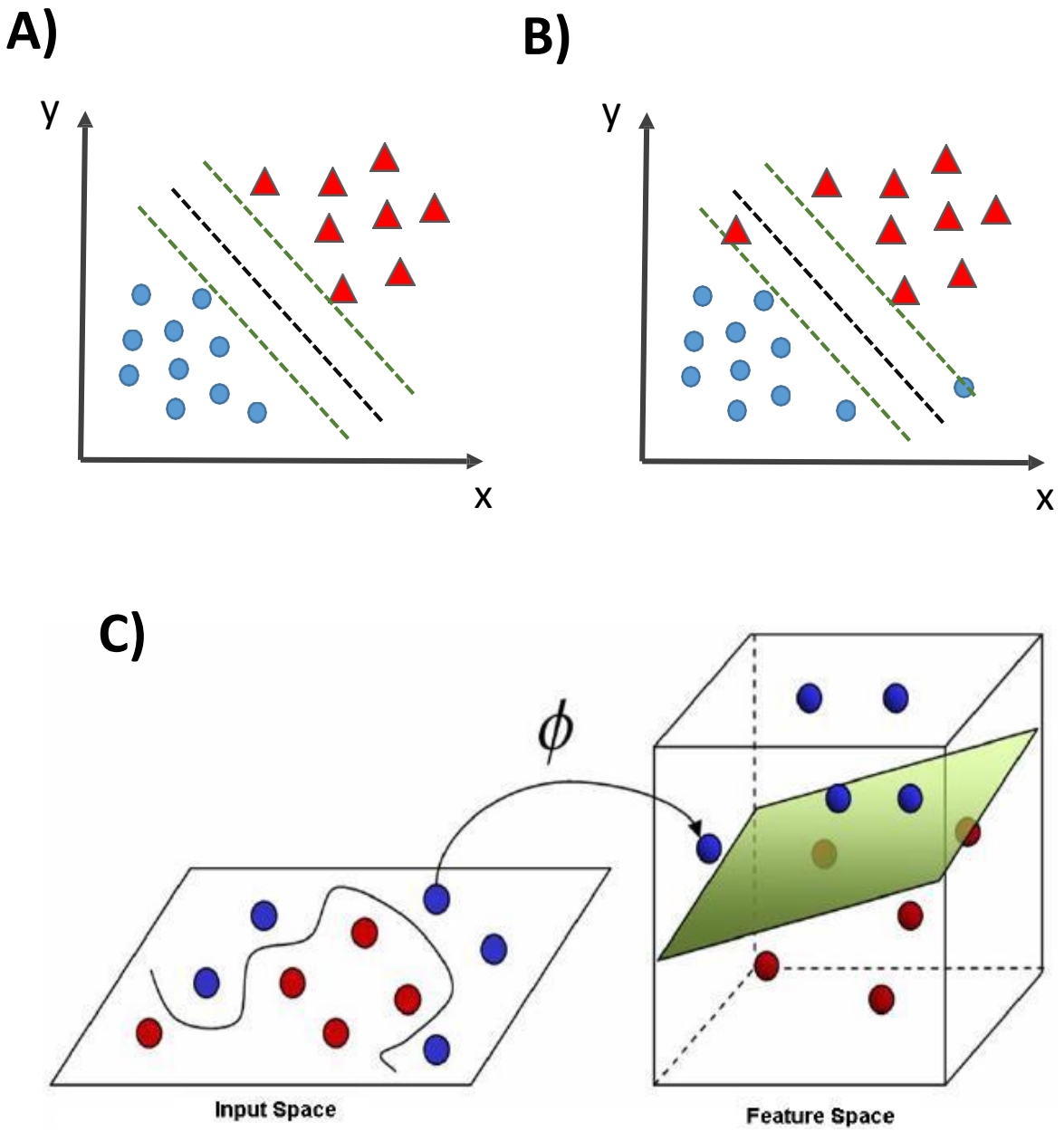
Classification:

- New items with unknown class affiliation classified depending on the side of the hyperplane

a) SVMs learn hyperplanes with maximal margins to improve generalization ability

b) Over fitting avoided by allowing misclassification of outliers of training set (called softmargin hyperplane)

c) Non-linear classifier achieved by mapping data to higher dimensional feature space via non- linear mapping function (kernel). By learning hyperplane in feature space, non-linear classifier achieved in input space



SVMs for Gene Identification

- SVMs can be used to classify ORFs into coding and non-coding (*e.g. Krause et al. NAR 2008*)

Training:

- Generate training set of ORFs known to be coding (positive training set) and ORFs known to be non-coding (negative training set)
- For each ORF, count codon frequencies
- Represent each ORF as 61 dimensional vector of codon frequencies
- Train SVM to distinguish between to sets of ORFs

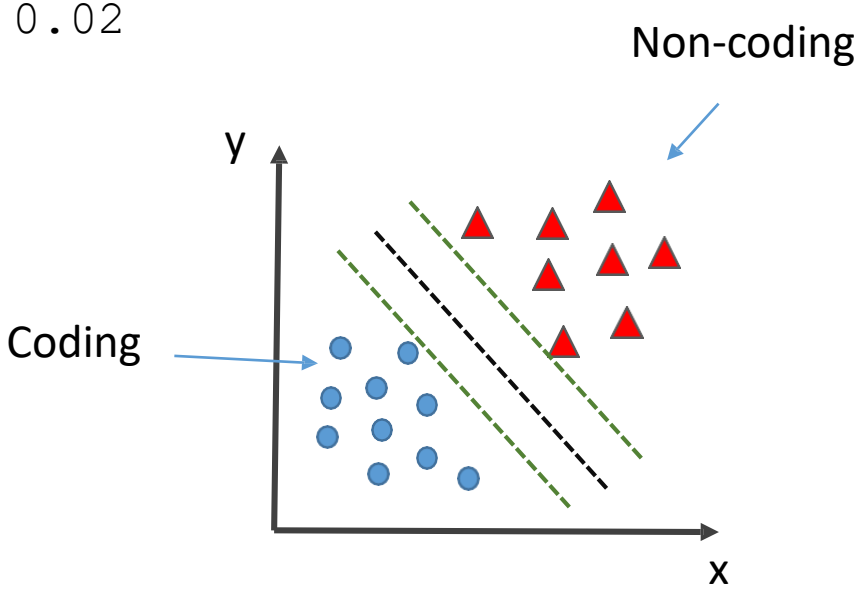
ORF: ATGGCTATCGACGAAAACAAACAGAA…

Codons:

ATG
GCT
ATC
GAC
etc.

Our **ORF** has the frequencies:

```
ATG = 0.1
GCT = 0.02
ATC = 0.001
GAC = 0.02
```



SVMs for Gene Identification

Classification (gene prediction):

- Extract all ORFs of genome
- Represent each ORF as 61 dimensional vector of codon frequencies
- Apply trained SVM to classify all ORFs into coding and non-coding depending on side of hyperplane
- scikit-learn library in Python
  - sklearn.svm

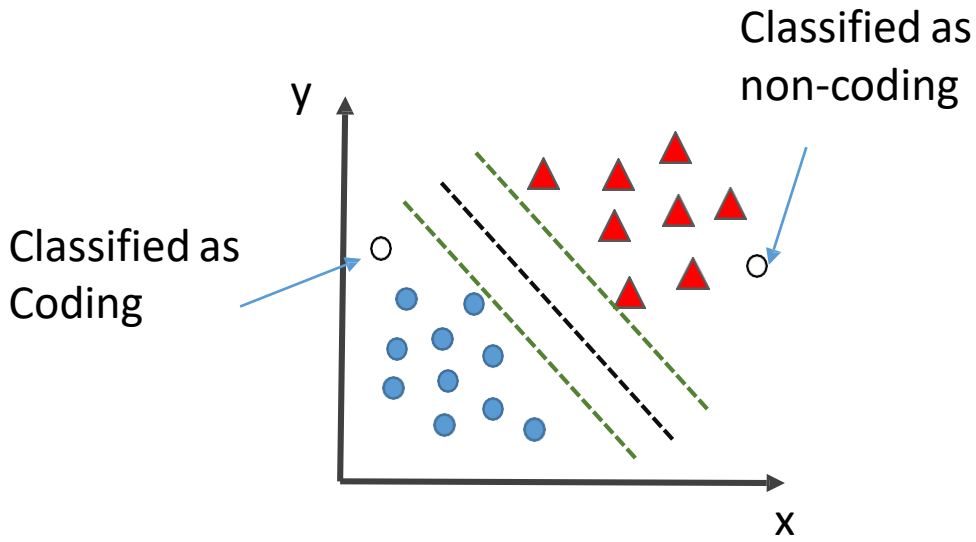
ORF: ATGGCTATCGACGAAAACAAACAGAA…

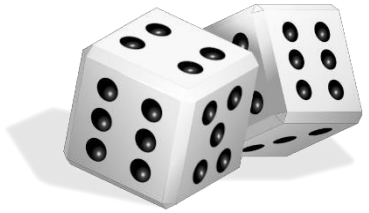
Codons:

ATG
GCT
ATC
GAC
etc.

Our **ORF** has the frequencies:

```
ATG = 0.1
GCT = 0.02
ATC = 0.001
GAC = 0.02
```





Hidden Markov Models

The dishonest casino:

Known information:

- Casino has 2 die, fair dice, loaded dice
- Casino player switches back & forth between dies
- Once either of the dice is used, it will continue to be used for a while

Number	Probability	
	Fair	Loaded
1	1/6	1/10
2	1/6	1/10
3	1/6	1/10
4	1/6	1/10
5	1/6	1/10
6	1/6	**1/2**

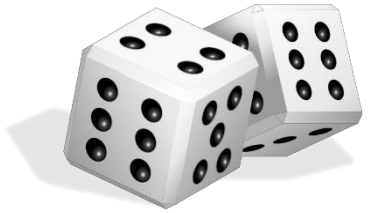
Observations:

- Sequence of roles:

  3 5 3 1 3 6 3 6 4 4 1 6 2 …

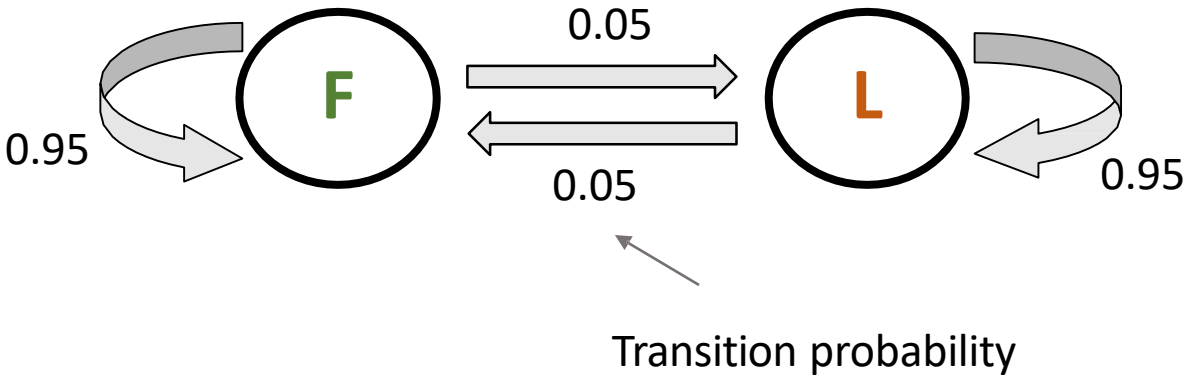
Question:

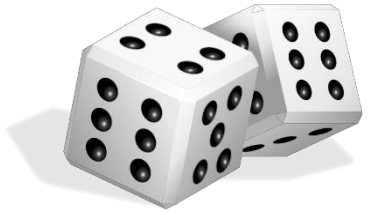
- Which dice used for each role?



Dishonest Casino Example

Number	Probability	
	Fair	Loaded
1	1/6	1/10
2	1/6	1/10
3	1/6	1/10
4	1/6	1/10
5	1/6	1/10
6	1/6	1/2



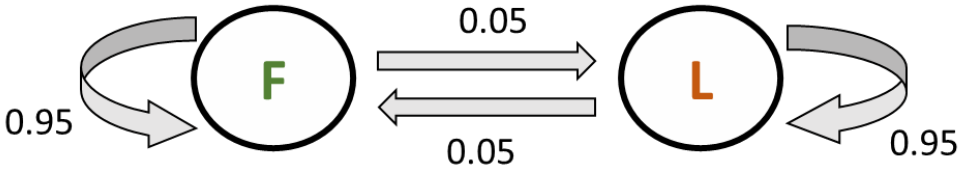


Dishonest Casino Example

Observation:

Sequence of roles:

Obs: 3 1 **6** 2 5 2 3 1 3 **6** 3 **6 6** 4 **6 6** 2 **6** …



Hidden information:

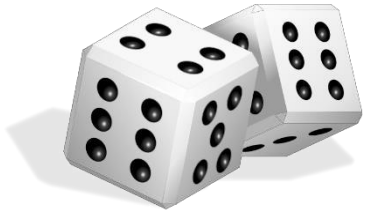
Sequence of states, e.g.

S1: F F F F F F F F F L L L L L L L L L….

S2: F F F F F F F F F F F F F F F F F F….

S3: L L L F F F F F F L L L L L L L L L…..

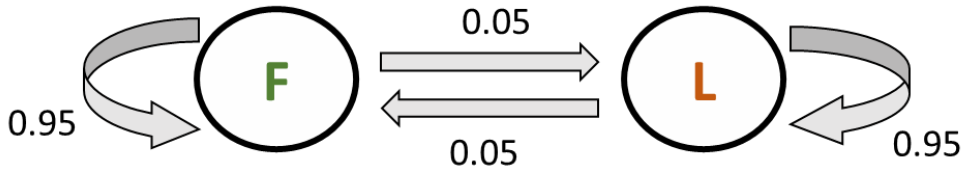
Number	Probability	
	Fair	Loaded
1	1/6	1/10
2	1/6	1/10
3	1/6	1/10
4	1/6	1/10
5	1/6	1/10
6	1/6	**1/2**



Dishonest Casino Example

Obs: 3 1 **6** 2 5 2 3 1 3 **6** 3 **6** **6** 4 **6** **6** 2 **6**

S1: F F F F F F F F F L L L L L L L L L



Transition to L state

P(Obs|S1) = 1/6 * 0.95 * 1/6 * 0.95 … * 0.05 * 1/2 * 0.95 * 1/10 * 0.95 * 1/2 … = 3.4e-14

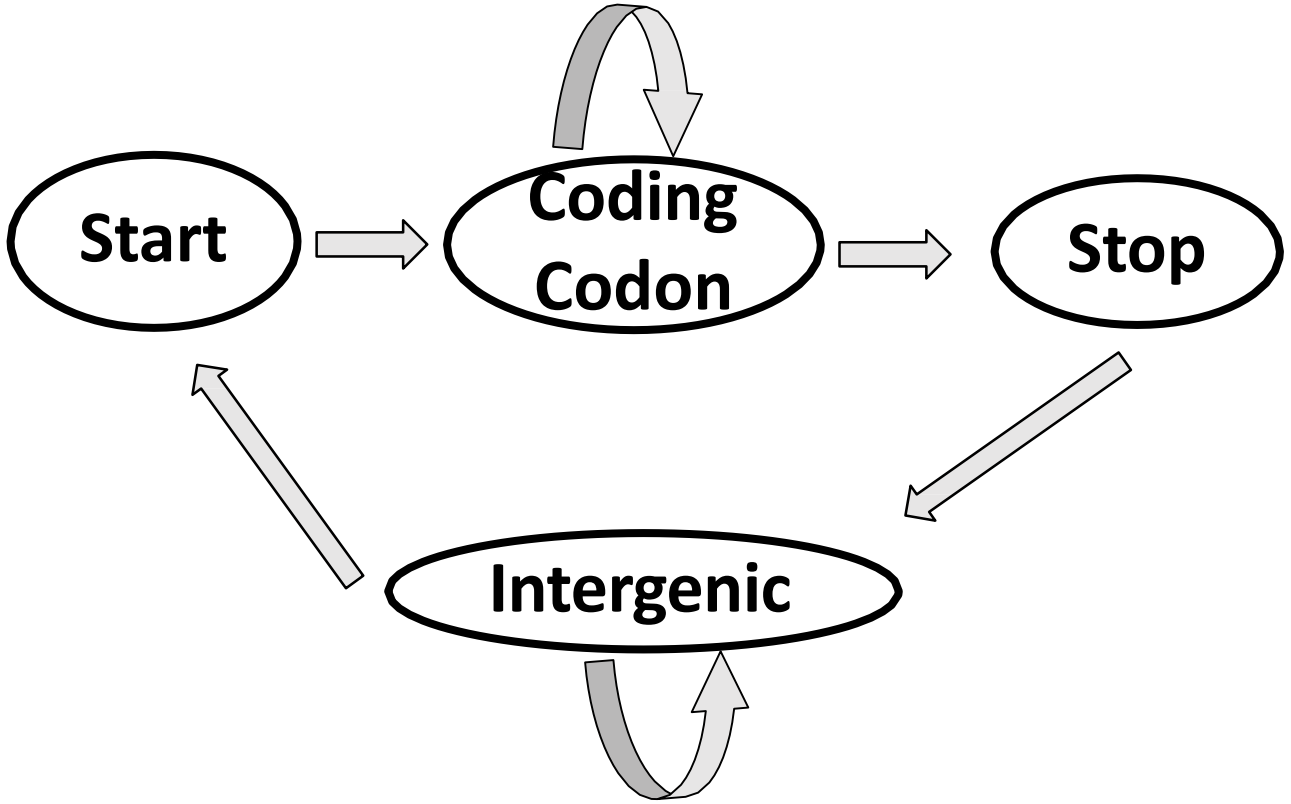
P(Obs|S2) = 4.1e-15

…..

Aim: Identify most likely path through model, which is S1 in this case, 9 roles fair dice, 9 roles loaded dice

Number	Probability Fair	Loaded
1	1/6	1/10
2	1/6	1/10
3	1/6	1/10
4	1/6	1/10
5	1/6	1/10
6	1/6	**1/2**

Simple HMM for Gene Identification in Prokaryotes



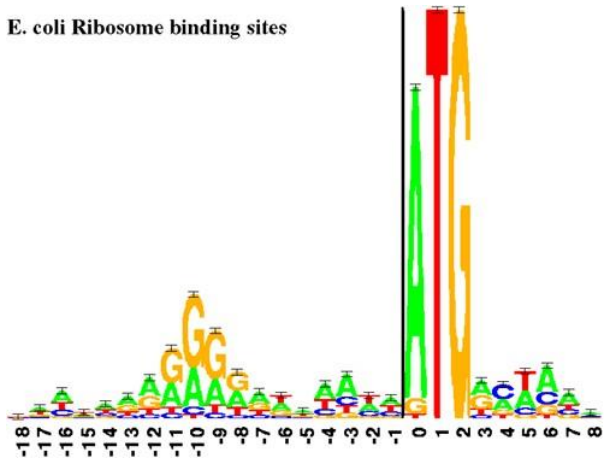
Training: Train model to learn codon frequencies of coding and non-coding sequences

Classification: Given observed DNA sequence, find most likely path through model to divide sequence into coding and non-coding regions

…CCTATC **ATG** GCT ATC GAC GAA AAC AAA … **TAA** CCTTATACTAG…

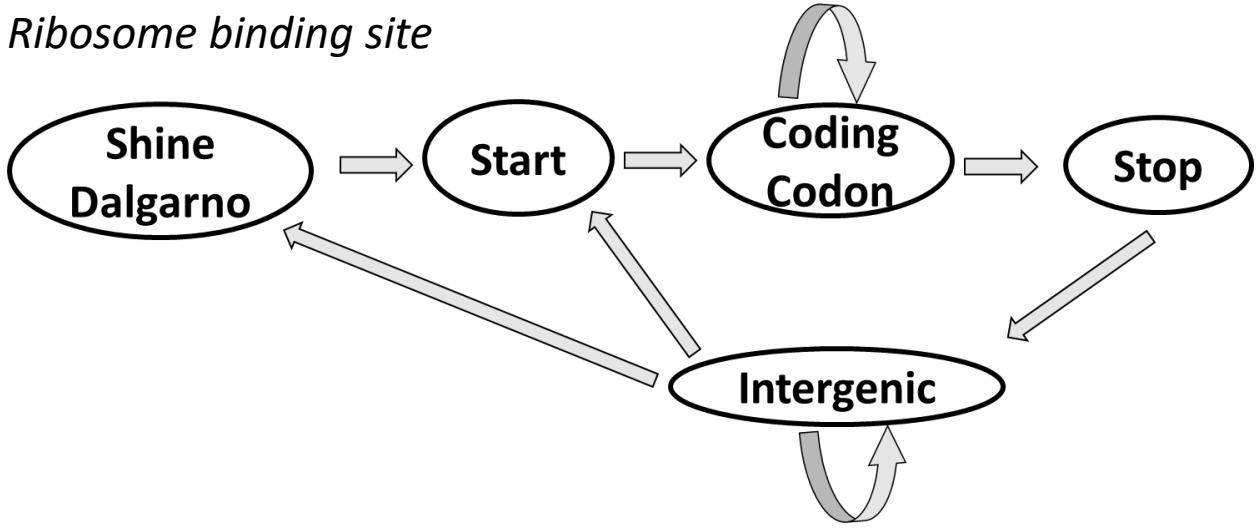
More Complex HMM for Gene Identification in Prokaryotes

Include signal for ribosomal binding site



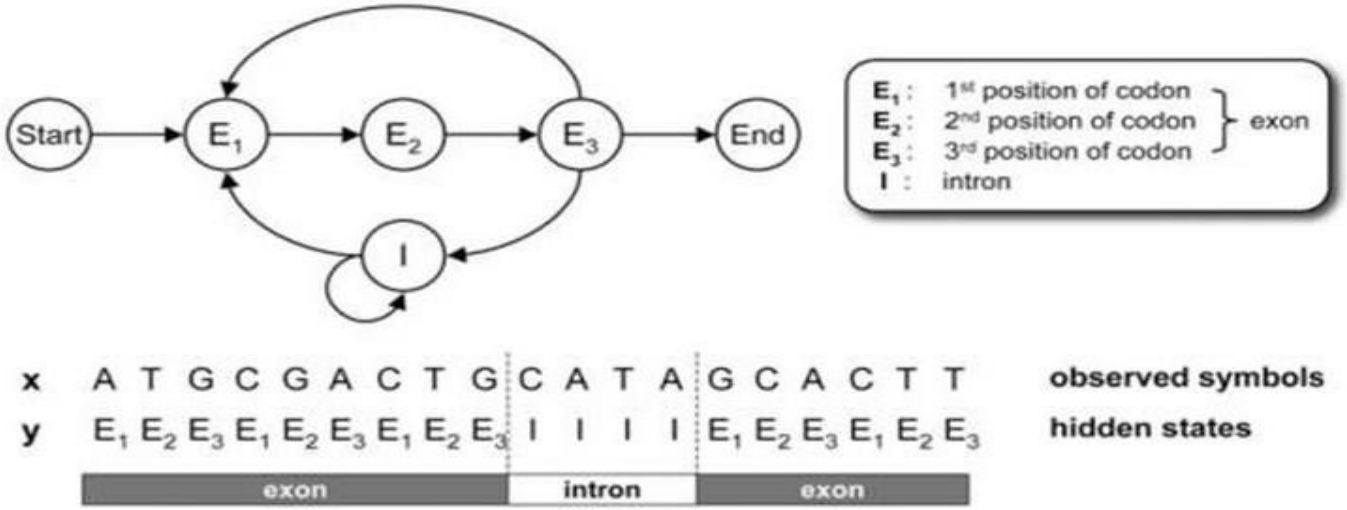
A/G-rich region about 10 bases upstream of the start codon
Helps recruit ribosome to mRNA

Ribosome binding site

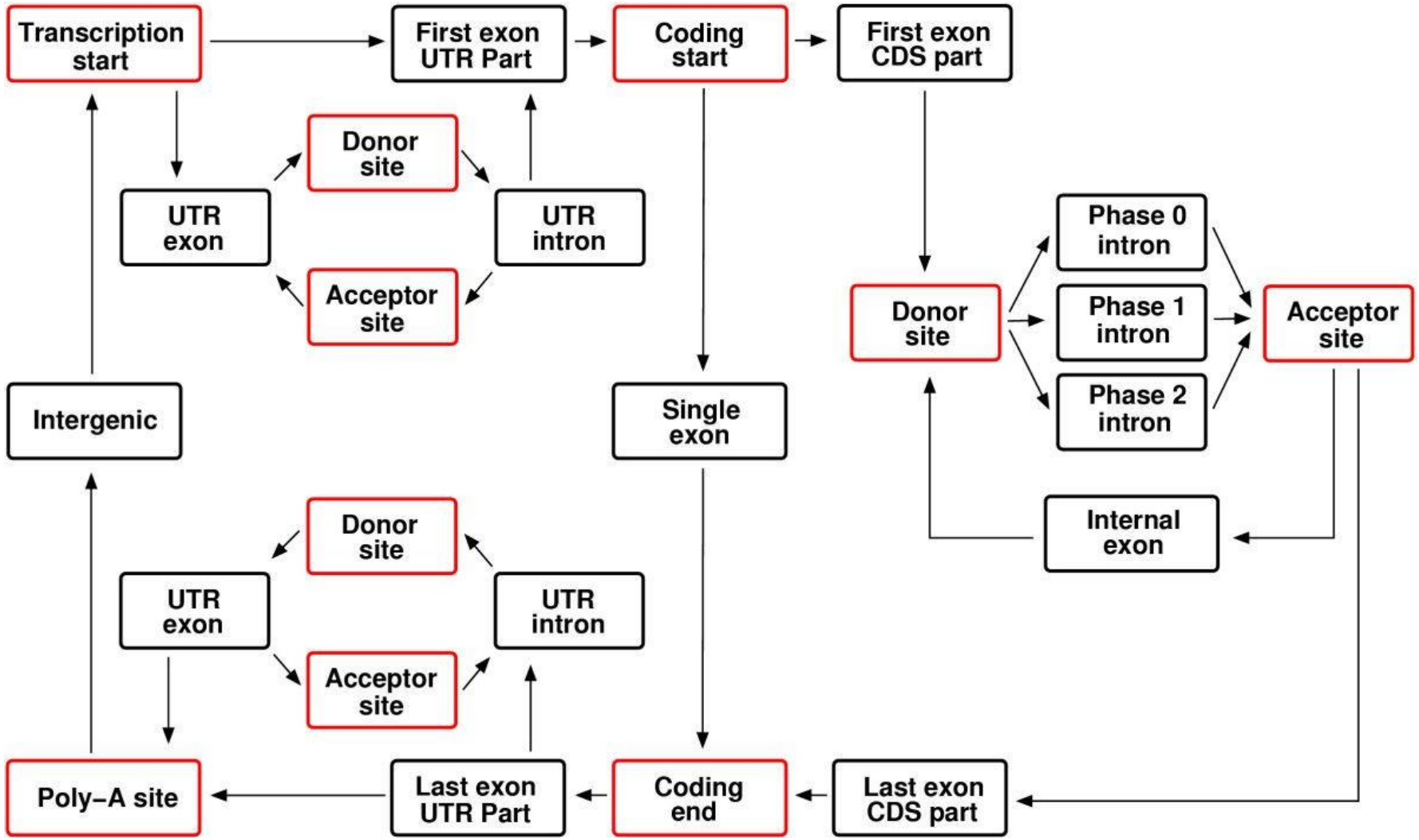


Gene Prediction in Eukaryotic Genomes

- **A Simple HMM for Modeling Eukaryotic Genes**



- hmmlearn() Python package. https://github.com/hmmlearn/hmmlearn

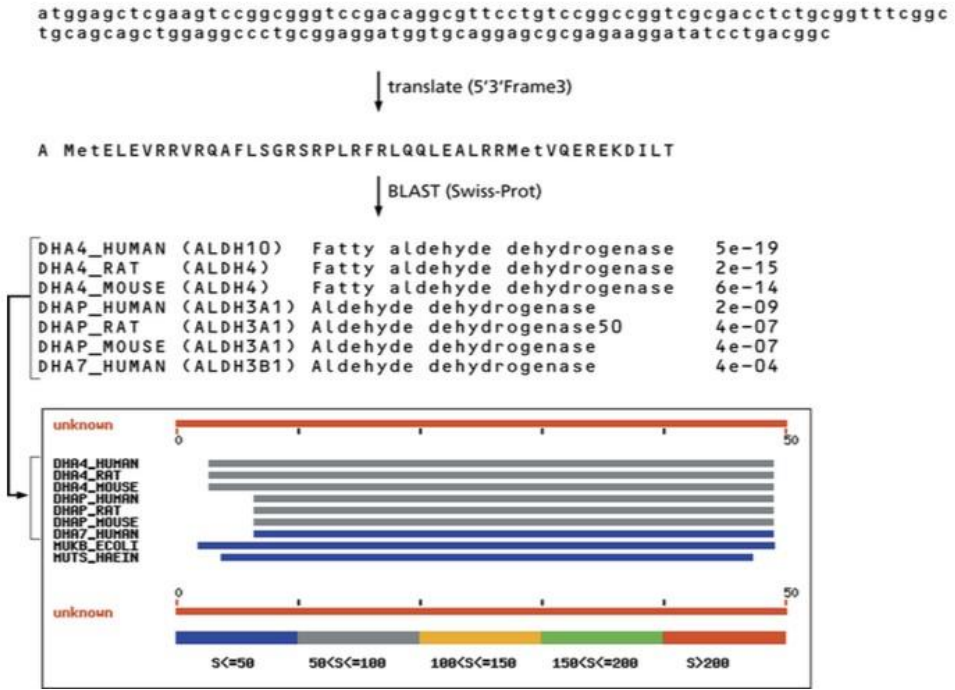

Transcription start
First exon UTR Part
Coding start
First exon CDS part
Donor site
UTR exon
UTR intron
Acceptor site
Phase 0 intron
Donor site
Phase 1 intron
Acceptor site
Phase 2 intron
Intergenic
Single exon
Internal exon
Donor site
UTR exon
UTR intron
Acceptor site
Poly–A site
Last exon UTR Part
Coding end
Last exon CDS part

Overview Characteristics that Can be Incorporated into HMM

- Coding regions begin with start codon and end with stop codon
- Coding regions do not contain stop codons
- Motifs of splice sites
- Sequence characteristics of UTRs
- Sequence characteristics of promoter regions
- Length distribution of coding regions, exons and introns
- polyA site

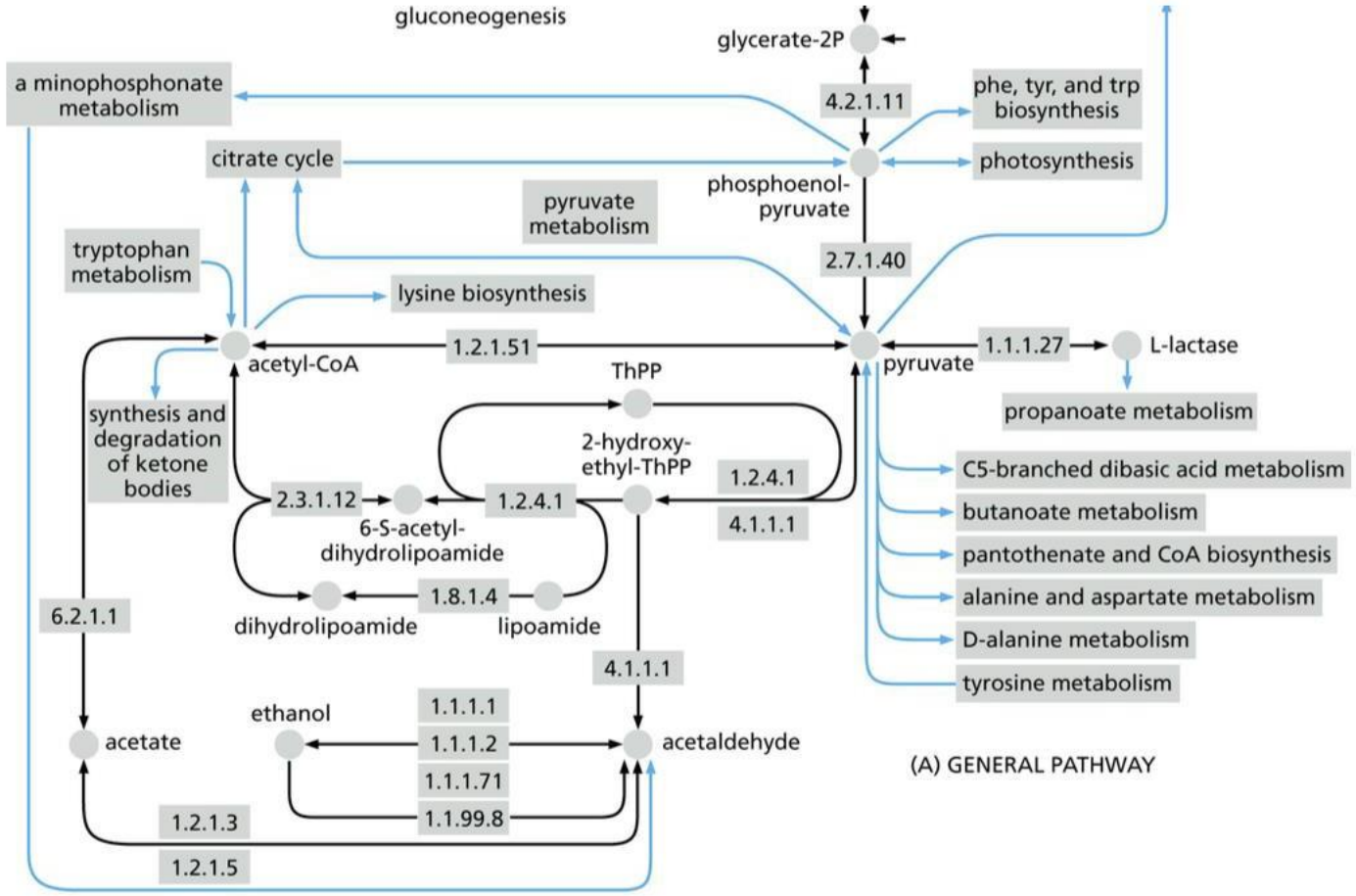
Genome annotation

- Once you have identified bona fide genes, what do you do with them?
- Use homology searchers (e.g. BLAST) to identify putative functions for genes, as well as identify other genetic elements such as functional RNAs and repeats
- Gene ontology provides a standard vocabulary to describe gene functions
- Annotations can be done for a whole genome/set of genomes using online services (e.g. IMG-M, KAAS)



Pathway analysis

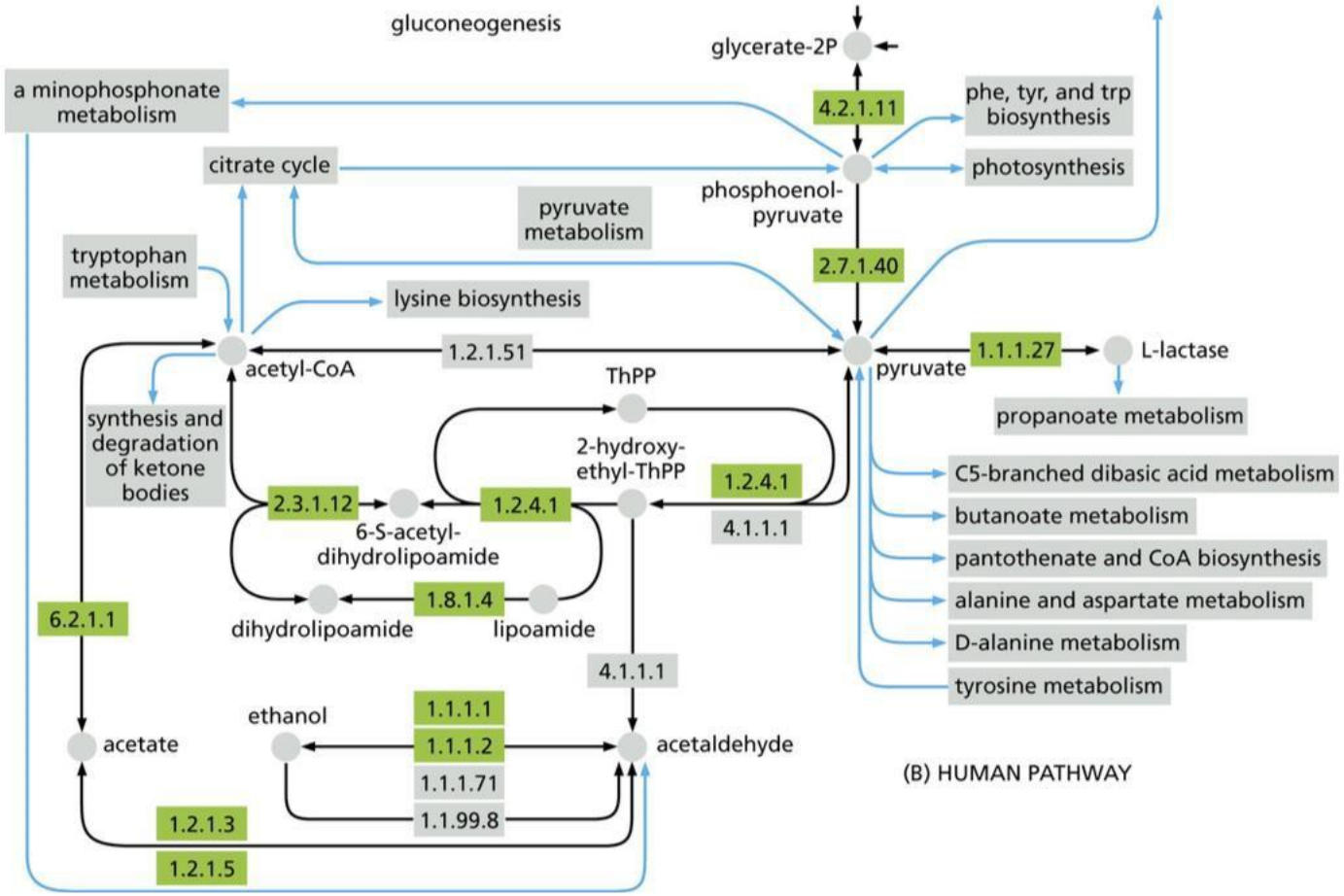
Once genes have been annotated, they are then fit into metabolic pathways to try and decipher the metabolism of the organism



Each gray box with a number represents an enzyme/protein in the pathway

Pathway analysis

We can color in the boxes based on which proteins we find in the organism



Also, if you have most of the proteins in the pathway, but one or two are missing, it suggests functions for some of your “unknown”/hypothetical proteins

Pathway Databases

- KEGG: Kyoto Encyclopedia of Genes and Genomes (https://www.genome.jp/kegg/)
- Reactome (https://reactome.org/)
- GO: Gene Ontology (http://geneontology.org/)

Take-Home points

- Eukaryotic gene-finding is more difficult because of the presence of introns and exons, and also because exons are short and some exons may have no coding sequence
- Exon finding can be aided by using transcriptional signals and splice sites
- Determining correct starting positions and length of exons is essential as all predicted exons must eventually be linked together to form a total coding sequence
- Once you have found genes, they can be annotated and used to infer metabolism
- Practical things you might want to know:
  - How to identify ORFs
  - Differences between prokaryotic and eukaryotic genes, and
  - Bioinformatics strategies for eukaryotic gene finding
  - How to read pathway maps