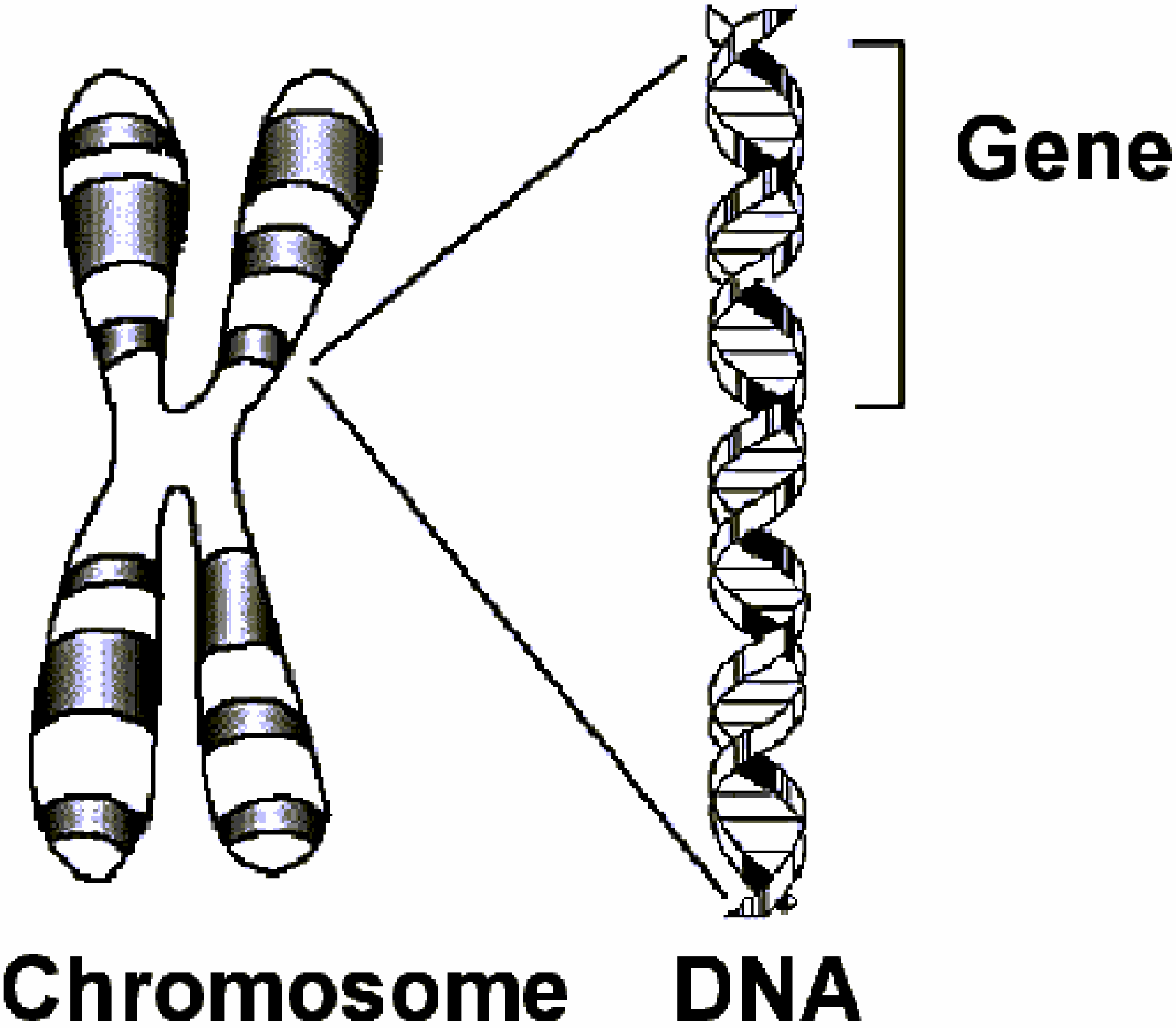
Gene Structure Prediction

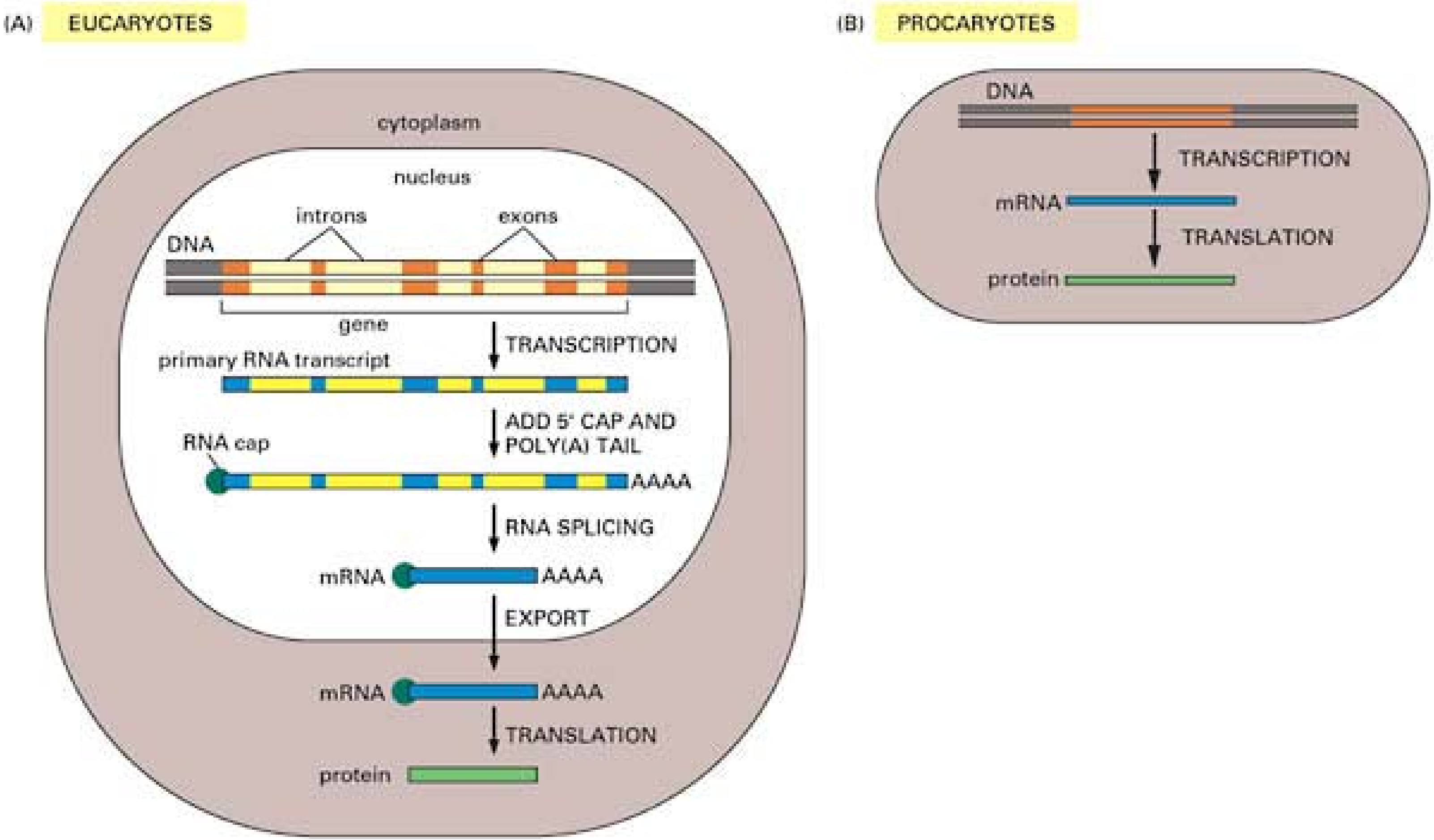
Jianlin Cheng, PhD

School of Electrical Engineering and Computer Science
University of Central Florida

2006


Gene
Chromosome
DNA

Gene Structure

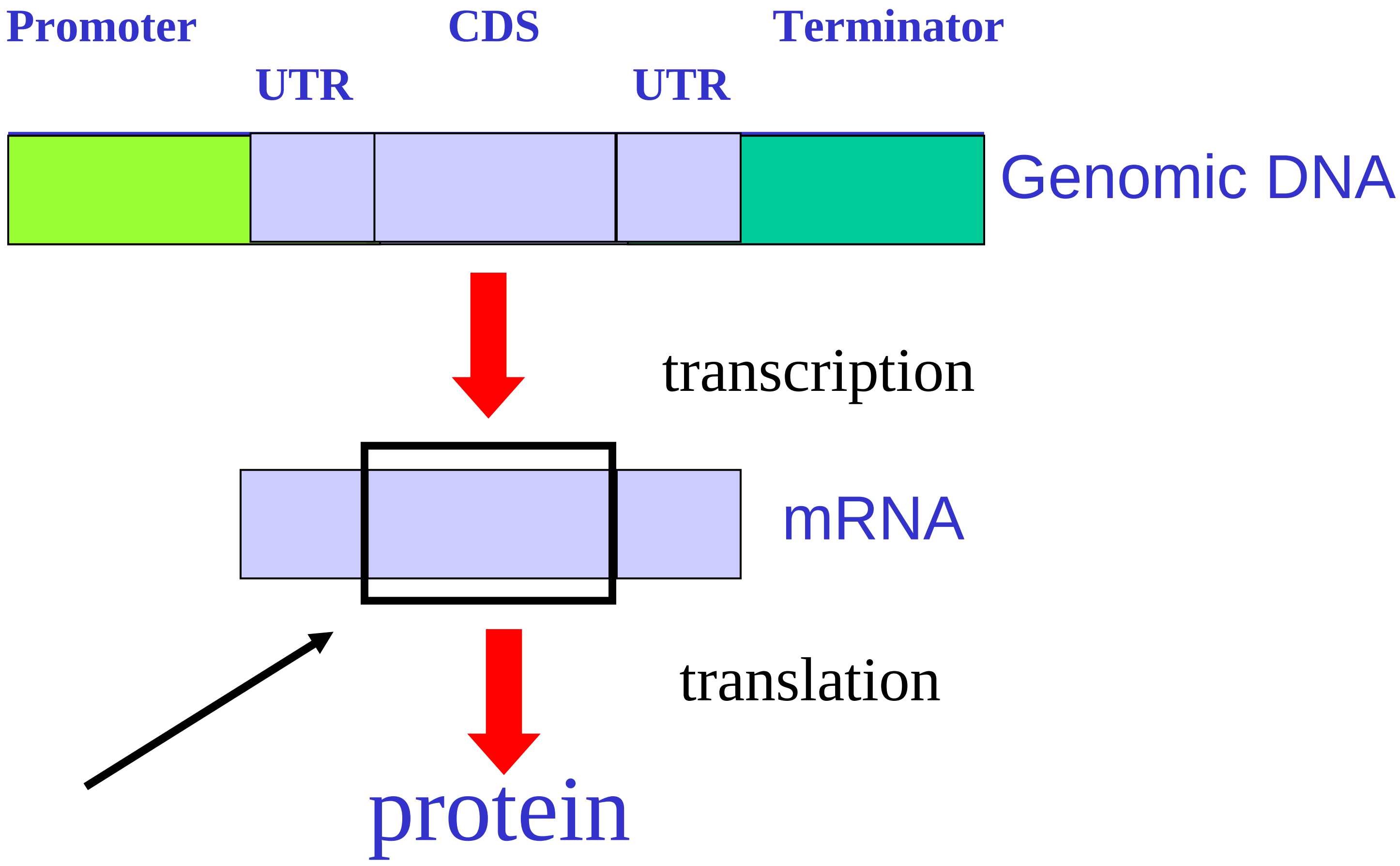


Ambuj Singh, 2005

Prokaryote gene structure

- **Promoter** : RNA polymerase binding consisting of a number of subunits
    - minus 10 site:
        - Pribnow box (TATAAT)
        - Sigma-specific
    - minus 35 site:
        - Sigma-specific
- **Transcription start site**
- **Coding region** (ORF): aa sequence in protein
    - Translational start site (AUG)
    - Translational stop site (UAA, UAG,UGA)
- **Transcription stop site**

Prokaryote Gene Structure

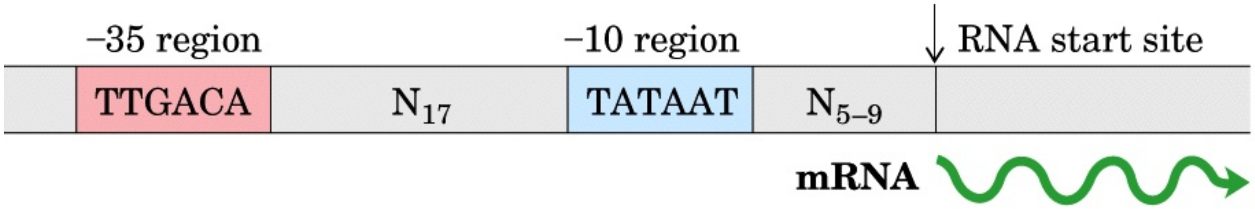


Ambuj Singh, 2005

UTR: a transcribed but non-coding region.

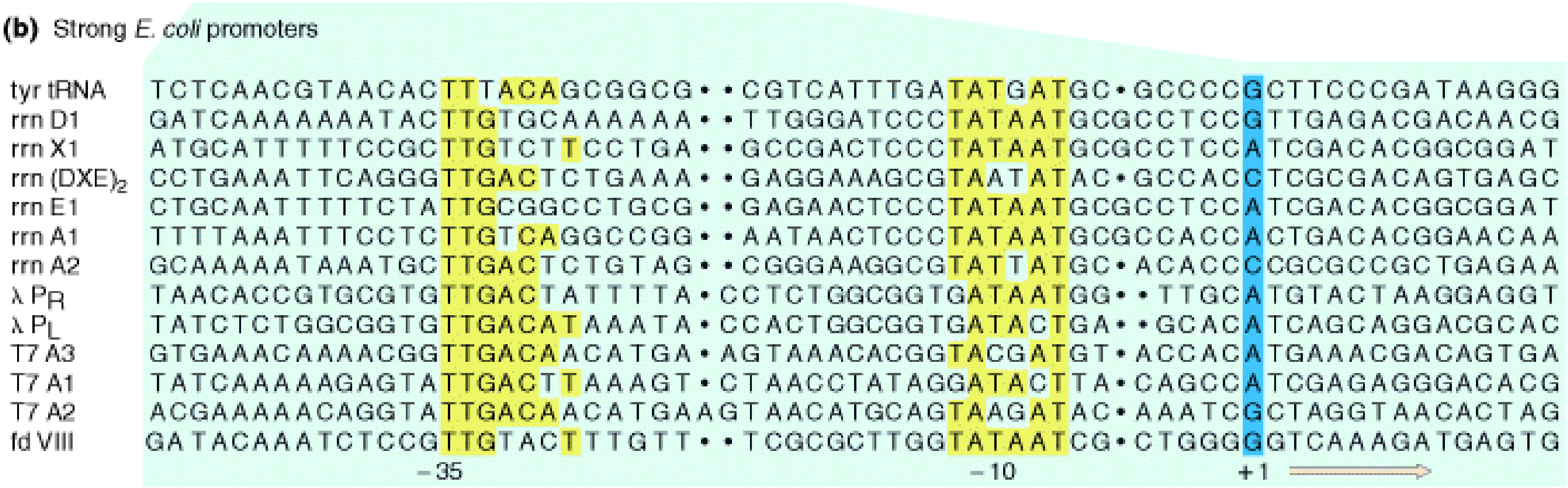
Prokaryote promoter example

- Pribnow box located at –10 (6-7bp)
- Promoter sequence located at -35 (6bp)



Consensus sequences

- Promoters sequences can vary tremendously.
- RNA polymerase recognizes hundreds of different promoters

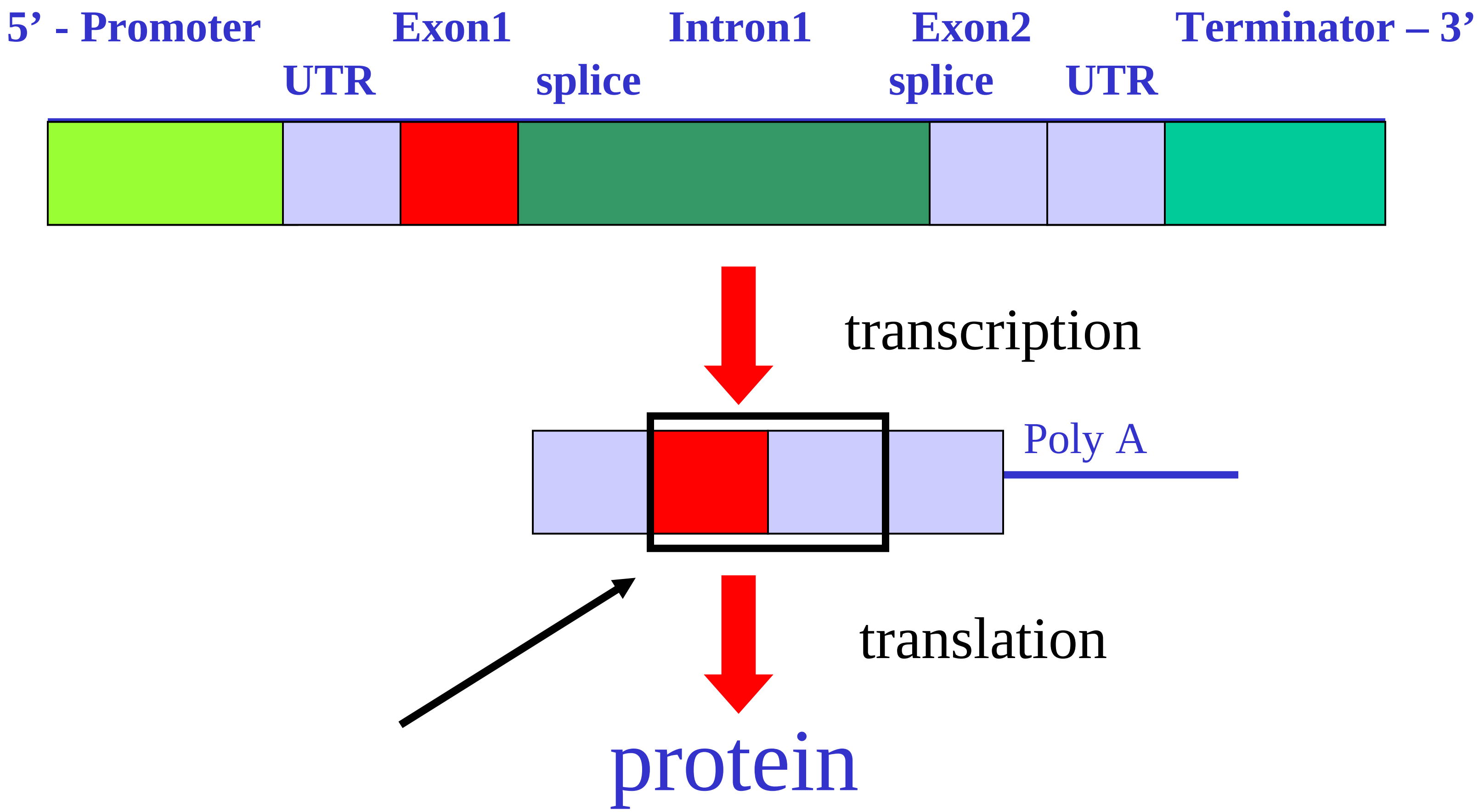


(b) Strong *E. coli* promoters

tyr tRNA	TCTCAACGTAACACTTTACAGCGGCG••CGTCATTTGATATGATGC•GCCCCGCTTCCCGATAAGGG
rrn D1	GATCAAAAAAATACTTGTGCAAAAAA••TTGGGATCCCTATAATGCGCCTCCGTTGAGACGACAACG
rrn X1	ATGCATTTTTCCGCTTGTCTTCCTGA••GCCGACTCCCTATAATGCGCCTCCATCGACACGGCGGAT
rrn $(DXE)_2$	CCTGAAATTCAGGGTTGACTCTGAAA••GAGGAAAGCGTAATATAC•GCCACCTCGCGACAGTGAGC
rrn E1	CTGCAATTTTTCTATTGCGGCCTGCG••GAGAACTCCCTATAATGCGCCTCCATCGACACGGCGGAT
rrn A1	TTTTAAATTTCCTCTTGTCAGGCCGG••AATAACTCCCTATAATGCGCCACCACTGACACGGAACAA
rrn A2	GCAAAAATAAATGCTTGACTCTGTAG••CGGGAAGGCGTATTATGC•ACACCCCGCGCCGCTGAGAA
λ P_R	TAACACCGTGCGTGTTGACTATTTTA•CCTCTGGCGGTGATAATGG••TTGCATGTACTAAGGAGGT
λ P_L	TATCTCTGGCGGTGTTGACATAAATA•CCACTGGCGGTGATACTGA••GCACATCAGCAGGACGCAC
T7 A3	GTGAAACAAAACGGTTGACAACATGA•AGTAAACACGGTACGATGT•ACCACATGAAACGACAGTGA
T7 A1	TATCAAAAAGAGTATTGACTTAAAGT•CTAACCTATAGGATACTTA•CAGCCATCGAGAGGGACACG
T7 A2	ACGAAAAACAGGTATTGACAACATGAAGTAACATGCAGTAAGATAC•AAATCGCTAGGTAACACTAG
fd VIII	GATACAAATCTCCGTTGTACTTTGTT••TCGCGCTTGGTATAATCG•CTGGGGGTCAAAGATGAGTG

Ambuj Singh, 2005

Eukaryote gene structure



5’ - Promoter

UTR

Exon1

splice

Intron1

splice

Exon2

UTR

Terminator – 3’

transcription

Poly A

translation

protein

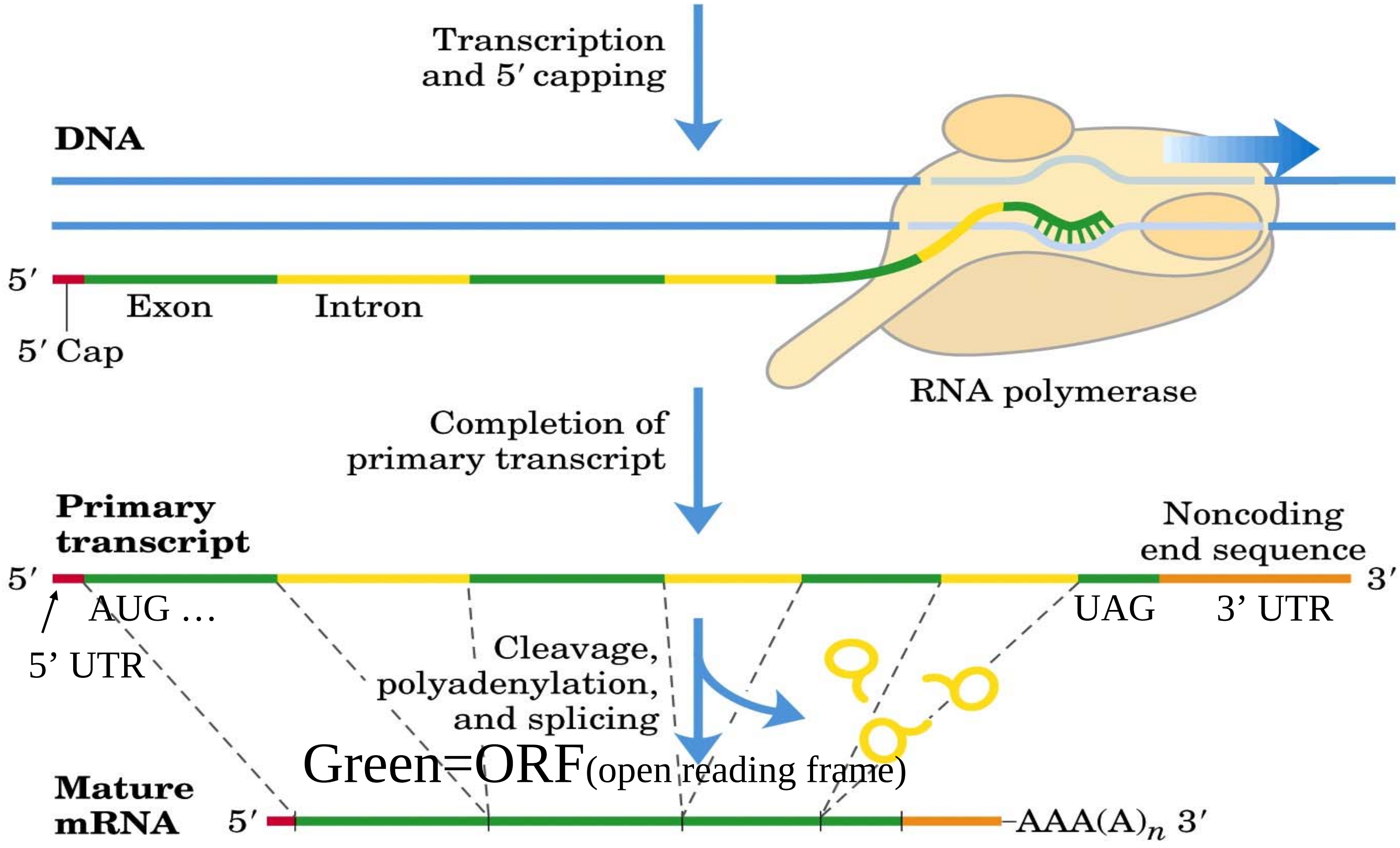
Ambuj Singh, 2005

Eukaryote gene structure

- TATA box located at –25
  - TATA(A/T)A(A/T)
  - Recognized by TATA-binding protein
- Initiator sequence at +1
  - YYCARR; Y is C/T, R is G/A
  - +1 is usually the A
- Transcription factors bind to promoters
  - Position specific scoring matrix (PSSM)
- Possible distant regions acting as enhancers or silencers (even more than 50 kb).
  - More complex mechanism than prokaryotes

Eukaryote gene structure vs. prokaryote gene structure

- No operons
- Capping at 5’ end and polyadenylation at 3’ end
  - Transport of mRNA out of nucleus
  - Effects stability and efficiency of translation
- Introns
- Alternative splicing
- CpG islands around promoter regions
  - CpG tends to methylate and mutate
  - Conservation implies function

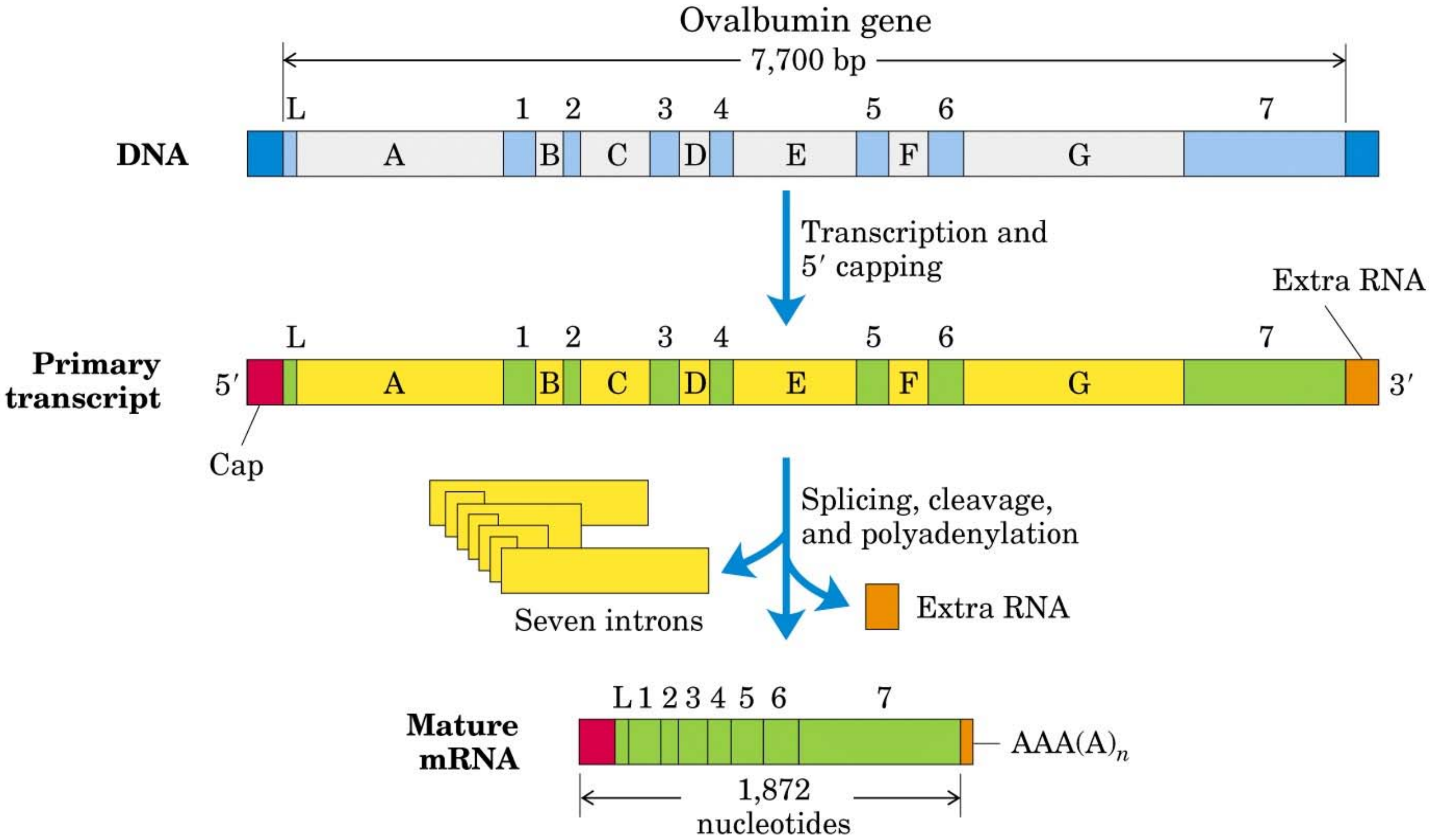


Green=ORF(open reading frame)

The linear order is never violated; it is simply interrupted

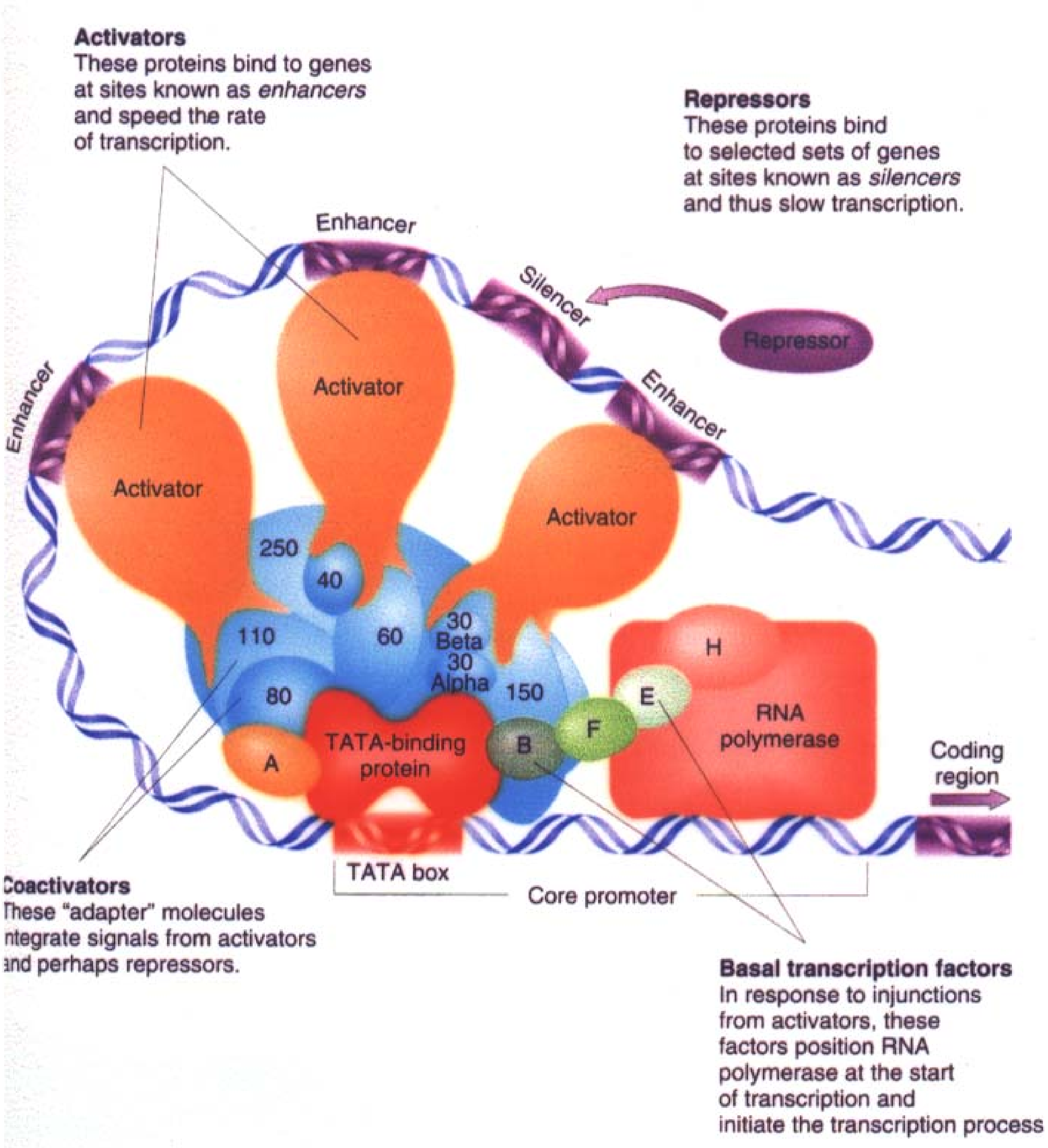
Ambuj Singh, 2005

Summary of the three steps in pre-mRNA processing



- The final mRNA may represent less than 5% of the transcribed DNA sequence

Ambuj Singh, 2005



Activators
These proteins bind to genes at sites known as *enhancers* and speed the rate of transcription.

Repressors
These proteins bind to selected sets of genes at sites known as *silencers* and thus slow transcription.

Coactivators
These "adapter" molecules integrate signals from activators and perhaps repressors.

Basal transcription factors
In response to injunctions from activators, these factors position RNA polymerase at the start of transcription and initiate the transcription process

Ambuj Singh, 2005

Gene Prediction Problems

- Prokaryotes: easy. Predict promoter region or start of coding region is able to determine a gene.
- Eukaryotes: hard. Need to predict promoter, transcription/translation start region, splice sites, coding regions. All these prediction can be considered in isolation or altogether.

Gene Structure Prediction Methods

- Homology Based Method
- Ab-Initio Methods

  Markov Model

  Hidden Markov Model

  Neural Network

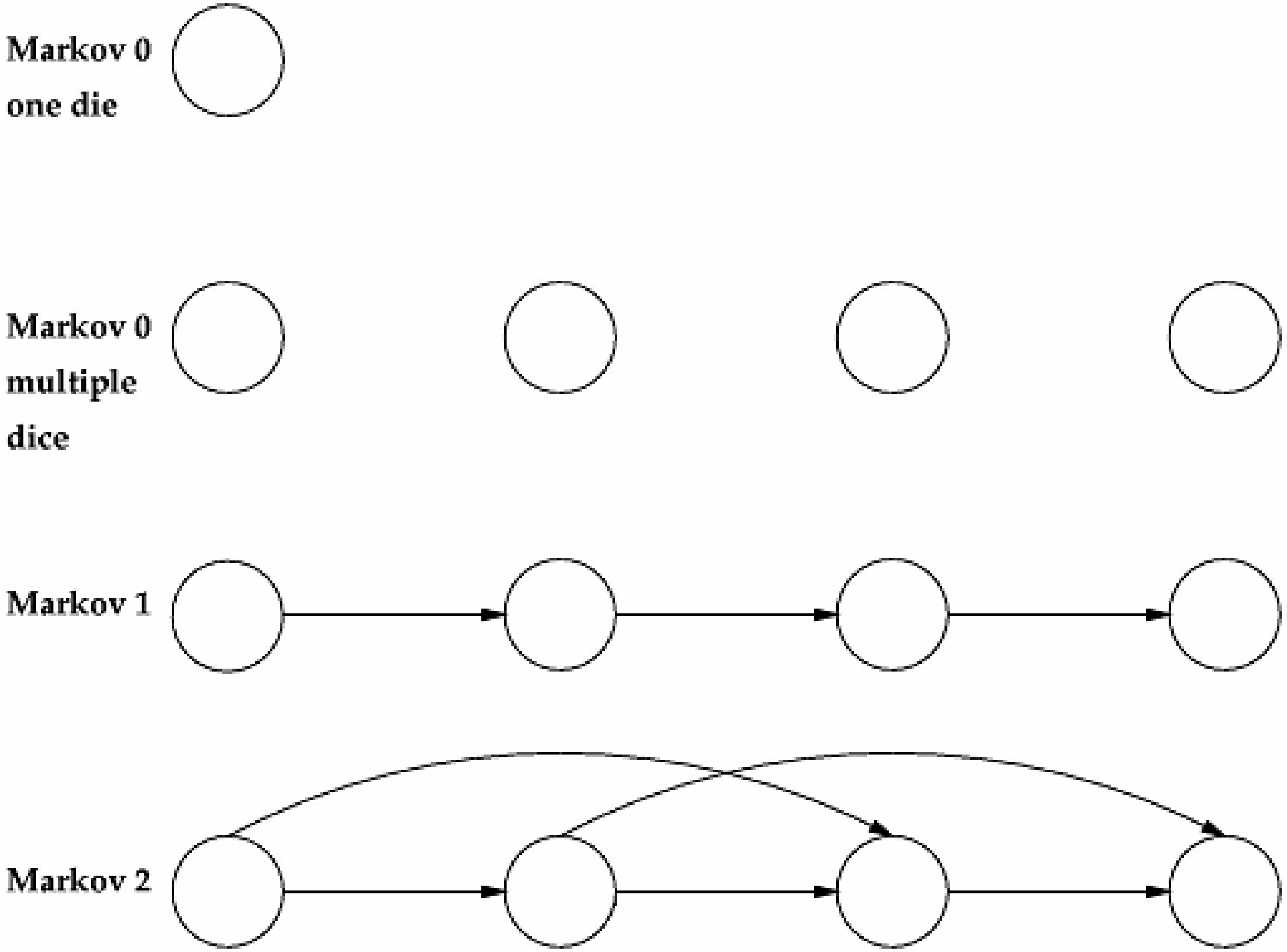
Homology Based Methods

- Given a genomic sequence, search against cDNA or EST libraries
- GenomeScan (genes.mit.edu/genomescan.html)
- EST2Genome (bioweb.pasteur.fr/seqanal/interfaces/est2genome.html)
- Consensus-based programs: GeneComber (www.bioinformatics.ubc.ca/genecomber/index.php)

Markov Model

- A Markov chain is a sequence of random variables X_1, X_2, X_3, … with Markov property, namely that, given the present state, the future and past states are independent.
- $P(X_{n+1}=x|X_n=x_n,\ldots,X_1=x_1,X_0=x_0)=P(X_{n+1}=x|X_n=x_n)$. (first order Markov Model)
- The possible values of X_i form a countable set S called the state space of the chain.
- A finite state machine is an example of a Markov chain.
- The probability of transition from one state to another state is called transition probability.

Markov Models

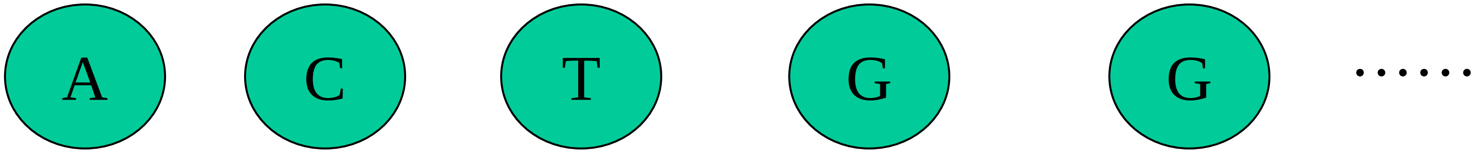


Markov Model for Gene Prediction

- DNA sequences can be considered to be generated by two Markov Chains
- One chain generates coding regions (gene). another chain generates non-coding regions.
- Each state in the chain can has four values: A, C, G, T

0-Order Markov Model

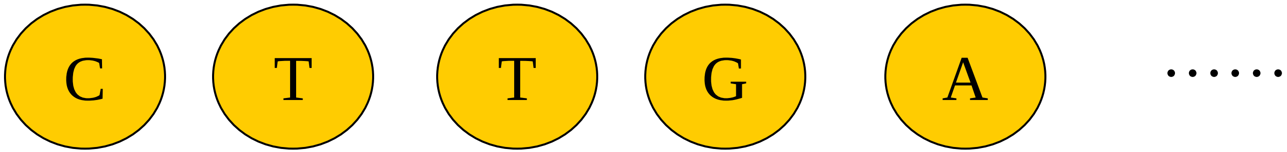
Coding Region:



For all coding sequences:

$P_c(A)$ = total num of A / total num of nucleotides
$P_c(C)$ = total num of C / total num of nucleotides
$P_c(G)$ = total num of G / total num of nucleotides
$P_c(T)$ = total num of T / total num of nucleotides

Non-Coding Region:



For all non-coding sequences:

$P_n(A)$ = total num of A / total num of nucleotides
$P_n(C)$ = total num of C / total num of nucleotides
$P_n(G)$ = total num of G / total num of nucleotides
$P_n(T)$ = total num of T / total num of nucleotides

Gene Prediction Using 0-order Markov Model

ACTGAGACAATGCCTA….

Under coding model:

$$P(A|\text{coding}) = P_c(G) * P_c(A) * P_c(G) * \ldots$$

Under non-coding model:

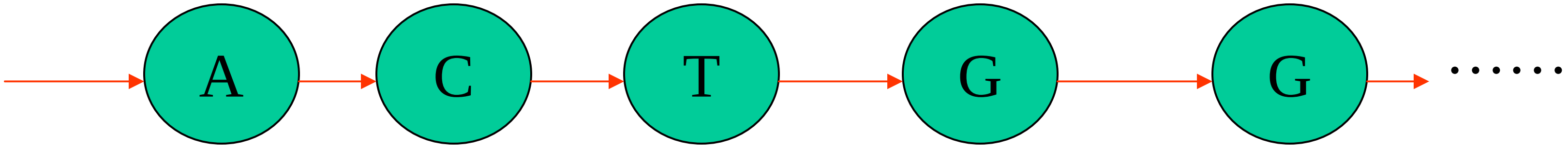
$$P(A|\text{non-coding}) = P_n(G) * P_n(A) * P_n(G) * \ldots$$

If P(A|coding) > P(A|non-coding), it is in a gene. Otherwise, it is not in a gene.

Window size is usually pretty large, e.g., 101.

1st-Order Markov Model

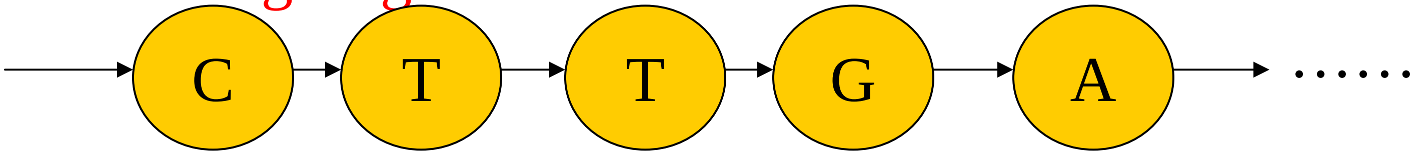
Coding Region:



For all coding sequences:

To compute P(x|y). x, y
$P_c(C|A)$ = total num of AC / total num of A
$P_c(T|C)$ = total num of CT / total number C
$P_c(G|T)$ = total num of TG / total number of T
$P_c(G|G)$ = total num of GG / total number of G
……. (16 different conditional probabilities)

Non-Coding Region:



For all non-coding sequences:

To compute P(x|y). X, y.
$P_n(T|C)$ = total num of CT / total num of C
$P_n(T|T)$ = total num of TT / total num of T
$P_n(G|T)$ = total num of GT / total num of T
$P_n(A|G)$ = total num of AG / total num of G
……

Gene Prediction Using 1st-order Markov Model

ACTGGGACAATGCCTA….

Under coding model:

$$P(seq|coding) = P_c(A) * P_c(C|A) * P_c(T|C) * \ldots$$

Under coding model:

$$P(seq|non\text{-}coding) = P_n(A) * P_n(C|A) * P_n(T|C) * \ldots$$

If P(seq|coding) > P(seq|non-coding), it is gene. Otherwise, it is not a gene.

Higher Order Markov Model for Gene Prediction

ACTGGGACAATGCCTA….

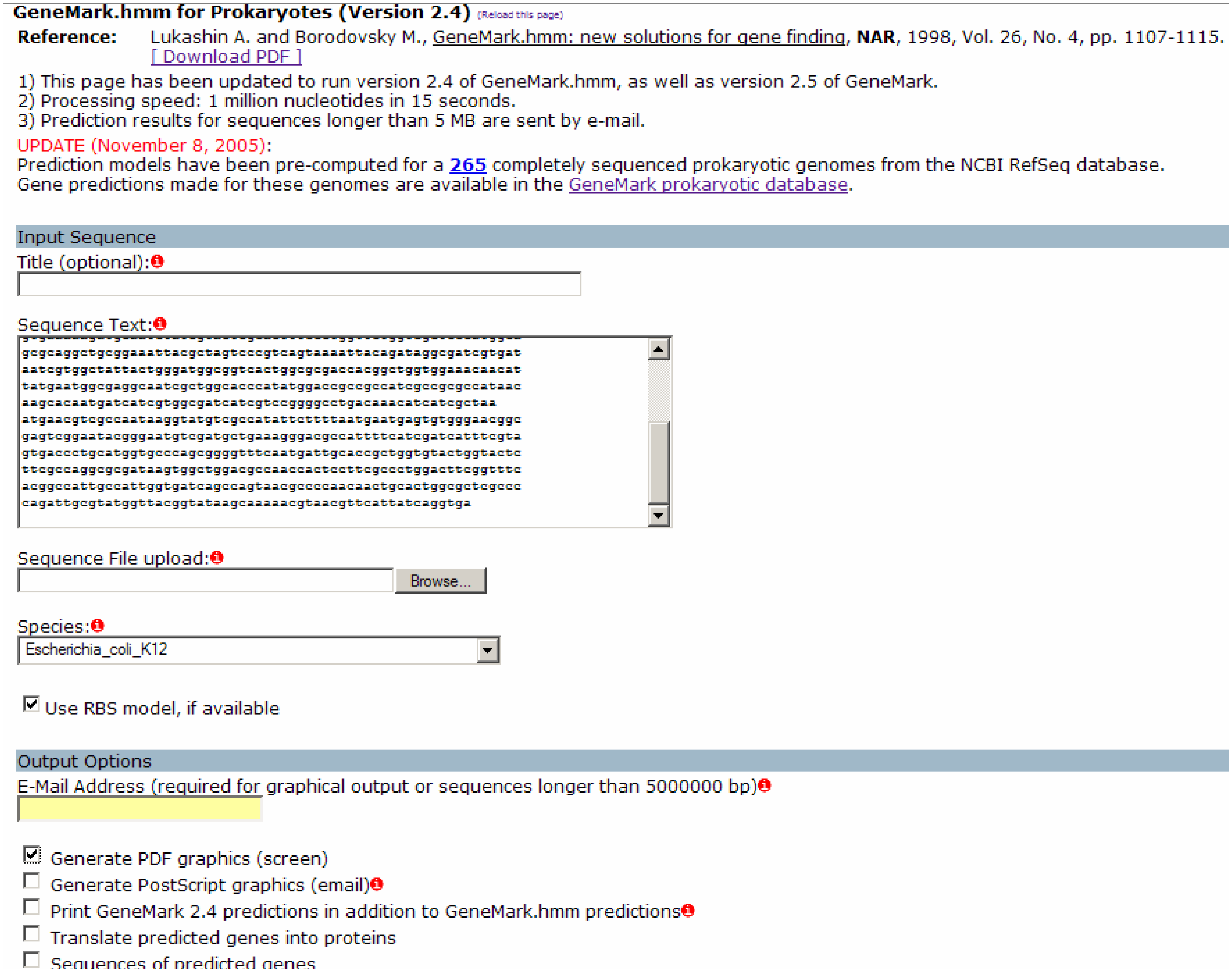
Second order:
P(T|AC), P(G|CT),….   (64 conditional probabilities)
P(z|xy) =   #xyz / #xy

Third order:
P(T|ACG), P(G|AAA), (256 conditional probabilities)
….

The best Markov Model for gene prediction uses 5th order.
(biological meaning?)

GeneMark

http://exon.gatech.edu/GeneMark/



GeneMark.hmm for Prokaryotes (Version 2.4) (Reload this page)

Reference: Lukashin A. and Borodovsky M., GeneMark.hmm: new solutions for gene finding, **NAR**, 1998, Vol. 26, No. 4, pp. 1107-1115. [Download PDF]

1) This page has been updated to run version 2.4 of GeneMark.hmm, as well as version 2.5 of GeneMark.
2) Processing speed: 1 million nucleotides in 15 seconds.
3) Prediction results for sequences longer than 5 MB are sent by e-mail.

UPDATE (November 8, 2005):
Prediction models have been pre-computed for a **265** completely sequenced prokaryotic genomes from the NCBI RefSeq database.
Gene predictions made for these genomes are available in the GeneMark prokaryotic database.

Input Sequence

Title (optional):

Sequence Text:

```
gcgcaggctgcggaaattacgctagtcccgtcagtaaaattacagataggcgatcgtgat
aatcgtggctattactgggatggcggtcactggcgcgaccacggctggtggaaacaacat
tatgaatggcgaggcaatcgctggcacccatatggaccgccgccatcgccgcgccataac
aagcacaatgatcatcgtggcgatcatcgtccggggcctgacaaacatcatcgctaa
atgaacgtcgccaataaggtatgtcgccatattcttttaatgaatgagtgtgggaacggc
gagtcggaatacgggaatgtcgatgctgaaagggacgccattttcatcgatcatttcgta
gtgaccctgcatggtgcccagcggggtttcaatgattgcaccgctggtgtactggtactc
ttcgccaggcgcgataagtggctggacgccaaccactccttcgccctggacttcggtttc
acggccattgccattggtgatcagccagtaacgccccaacaactgcactggcgctcgccc
cagattgcgtatggttacggtataagcaaaaacgtaacgttcattatcaggtga
```

Sequence File upload:

Species: Escherichia_coli_K12

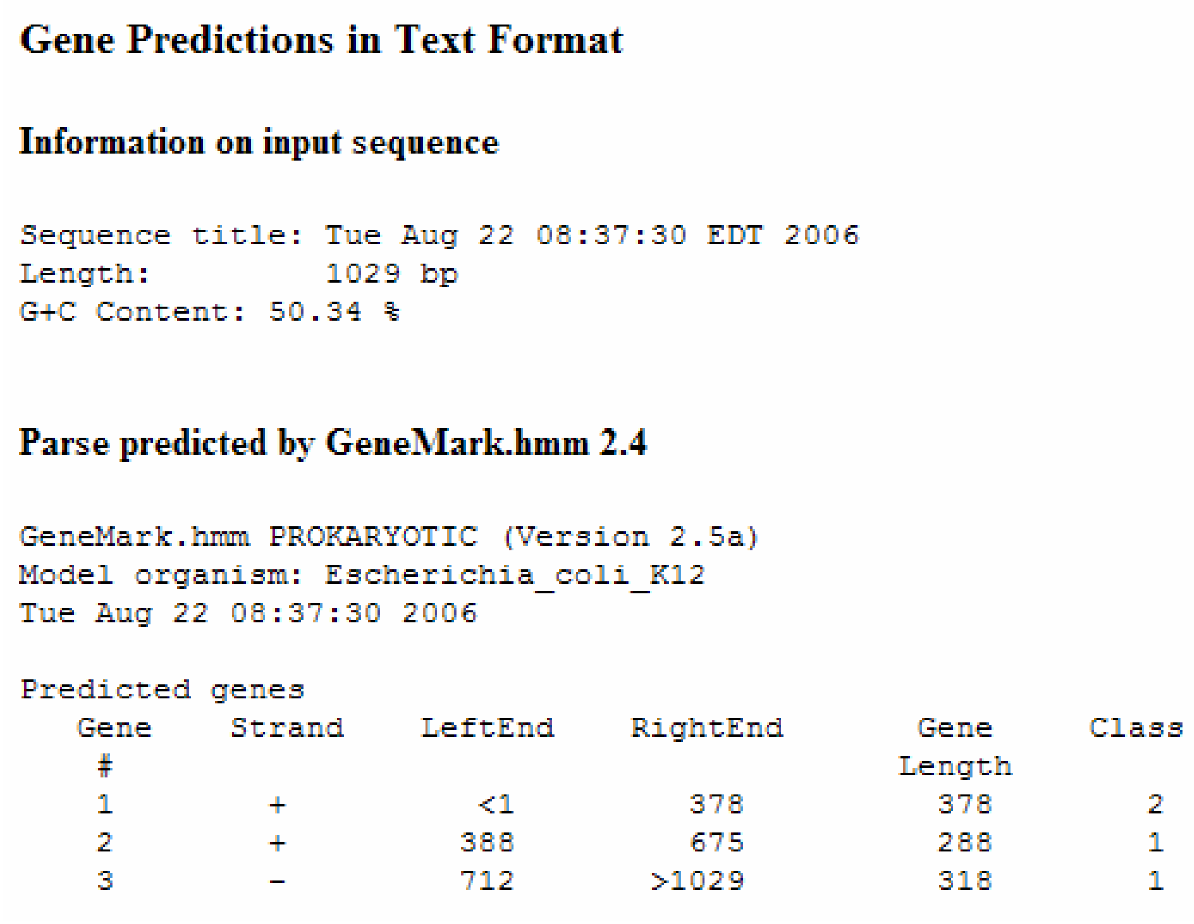
☑ Use RBS model, if available

Output Options

E-Mail Address (required for graphical output or sequences longer than 5000000 bp)

☑ Generate PDF graphics (screen)
☐ Generate PostScript graphics (email)
☐ Print GeneMark 2.4 predictions in addition to GeneMark.hmm predictions
☐ Translate predicted genes into proteins
☐ Sequences of predicted genes

Output of GeneMark



Gene Predictions in Text Format

Information on input sequence

```
Sequence title: Tue Aug 22 08:37:30 EDT 2006
Length:         1029 bp
G+C Content: 50.34 %
```

Parse predicted by GeneMark.hmm 2.4

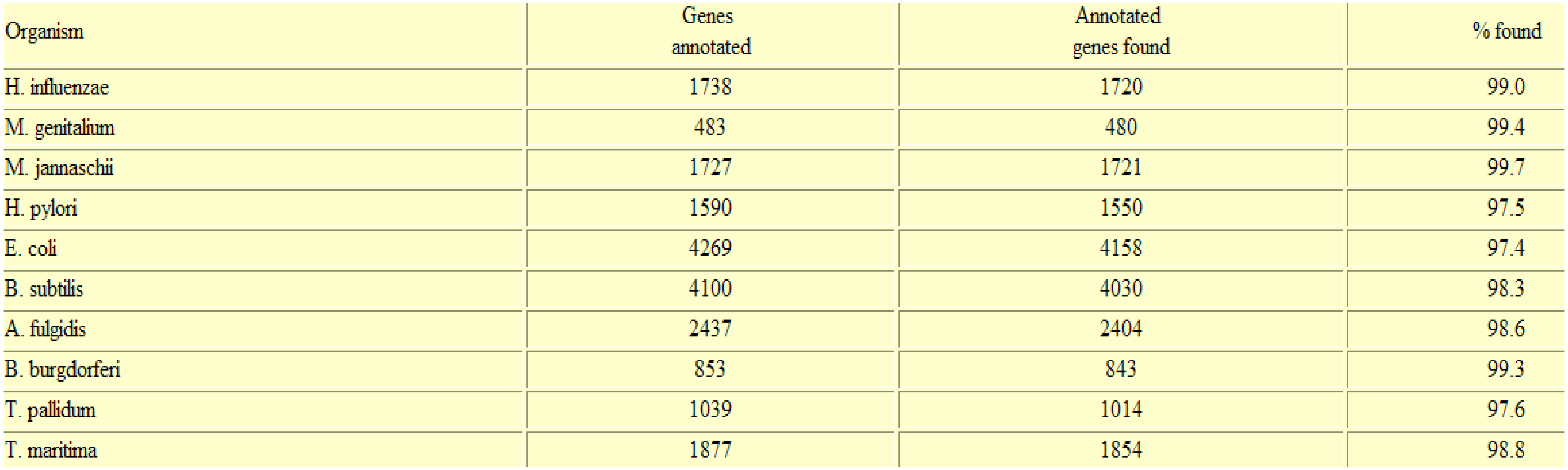
```
GeneMark.hmm PROKARYOTIC (Version 2.5a)
Model organism: Escherichia_coli_K12
Tue Aug 22 08:37:30 2006

Predicted genes
   Gene     Strand     LeftEnd     RightEnd        Gene      Class
    #                                             Length
    1         +          <1          378           378         2
    2         +         388          675           288         1
    3         -         712        >1029           318         1
```

Glimmer

- Download page: http://www.tigr.org/~salzberg/glimmer.html#perf
- Mainly for bacteria and archaea
- Use interpolated Markov Model: train model from 1st ord to 8th ord and weight them according to prediction accuracy.

Accuracy of Glimmer



Organism	Genes annotated	Annotated genes found	% found
H. influenzae	1738	1720	99.0
M. genitalium	483	480	99.4
M. jannaschii	1727	1721	99.7
H. pylori	1590	1550	97.5
E. coli	4269	4158	97.4
B. subtilis	4100	4030	98.3
A. fulgidis	2437	2404	98.6
B. burgdorferi	853	843	99.3
T. pallidum	1039	1014	97.6
T. maritima	1877	1854	98.8

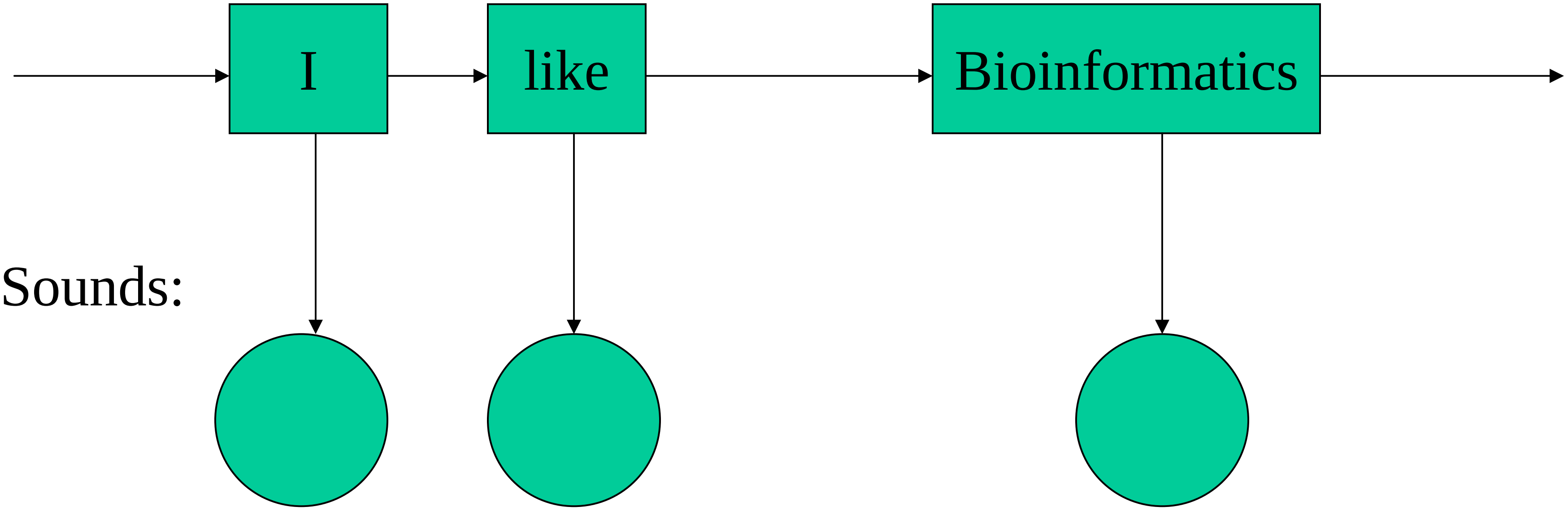
It is pretty accurate for prokaryotes.

Hidden Markov Model

- HMM is Markov process where states are hidden (unseen), but the variables emitted from states can be observed.
- Challenge is to determine the hidden parameters from the observable parameters.

Speech Recognition Example

Words:



I

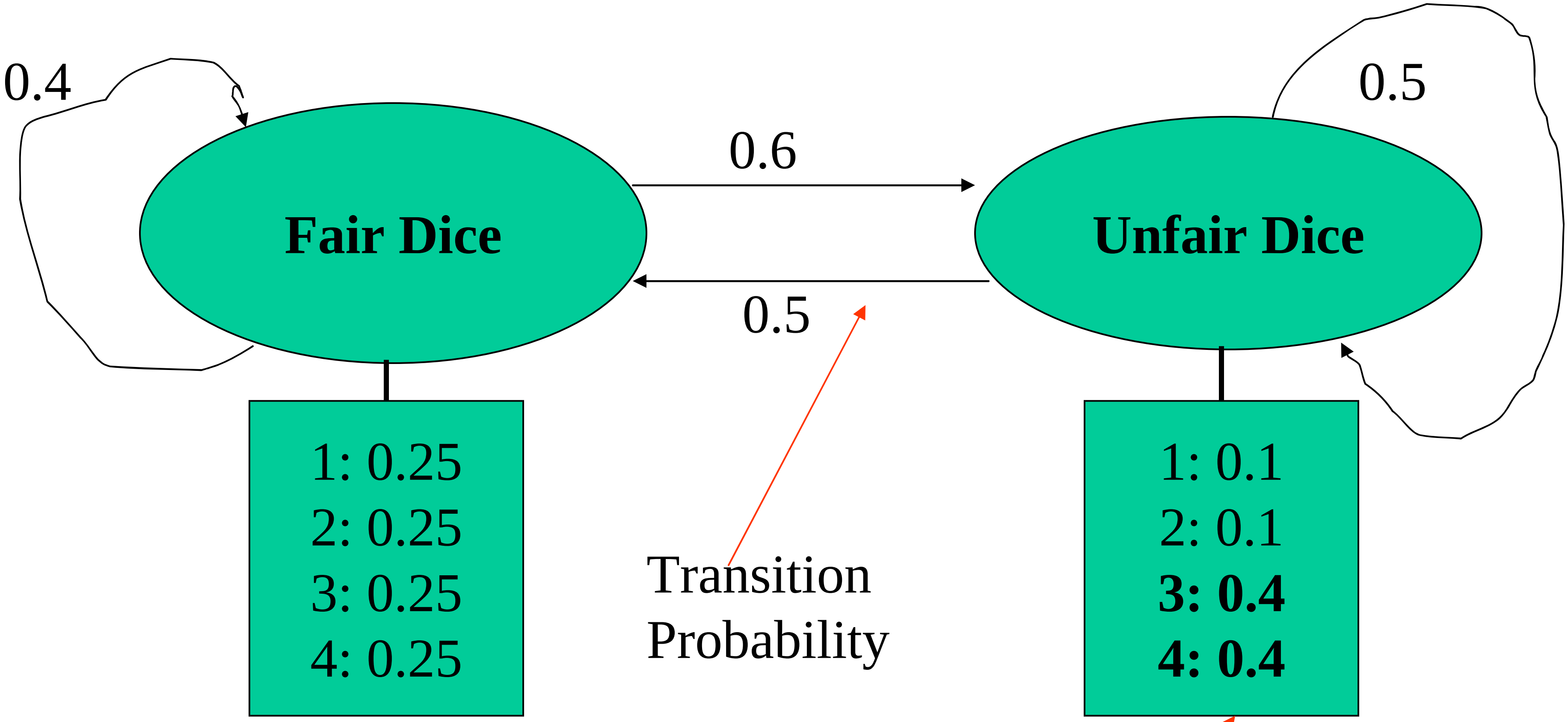
like

Bioinformatics

Sounds:

Goal: infer words from sounds

Gambling Example



0.4

0.5

0.6

0.5

Fair Dice

Unfair Dice

1: 0.25
2: 0.25
3: 0.25
4: 0.25

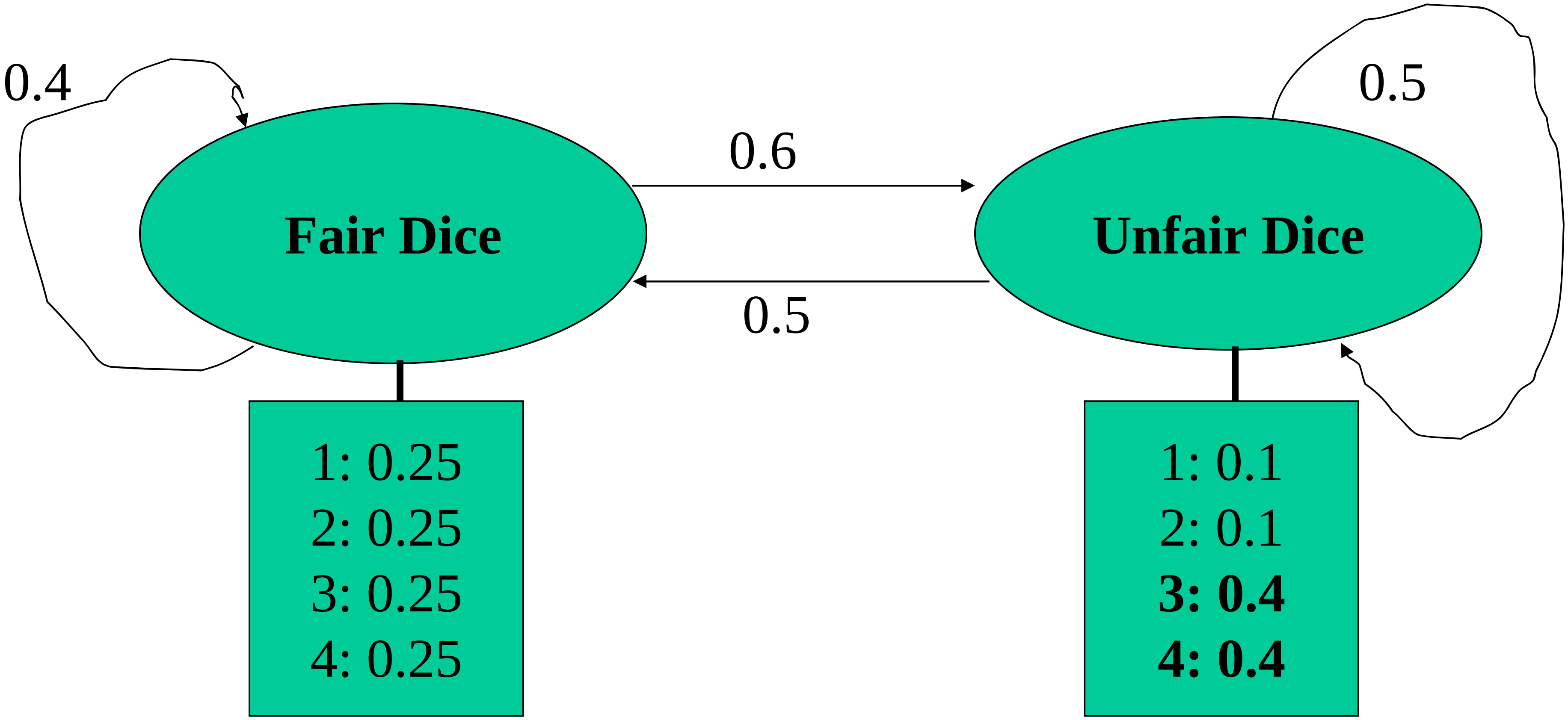
Transition
Probability

1: 0.1
2: 0.1
3: 0.4
4: 0.4

Observations: 12312421341213443432434443

Emission Probability

Gambling Example



0.4

0.6

0.5

0.5

Fair Dice

Unfair Dice

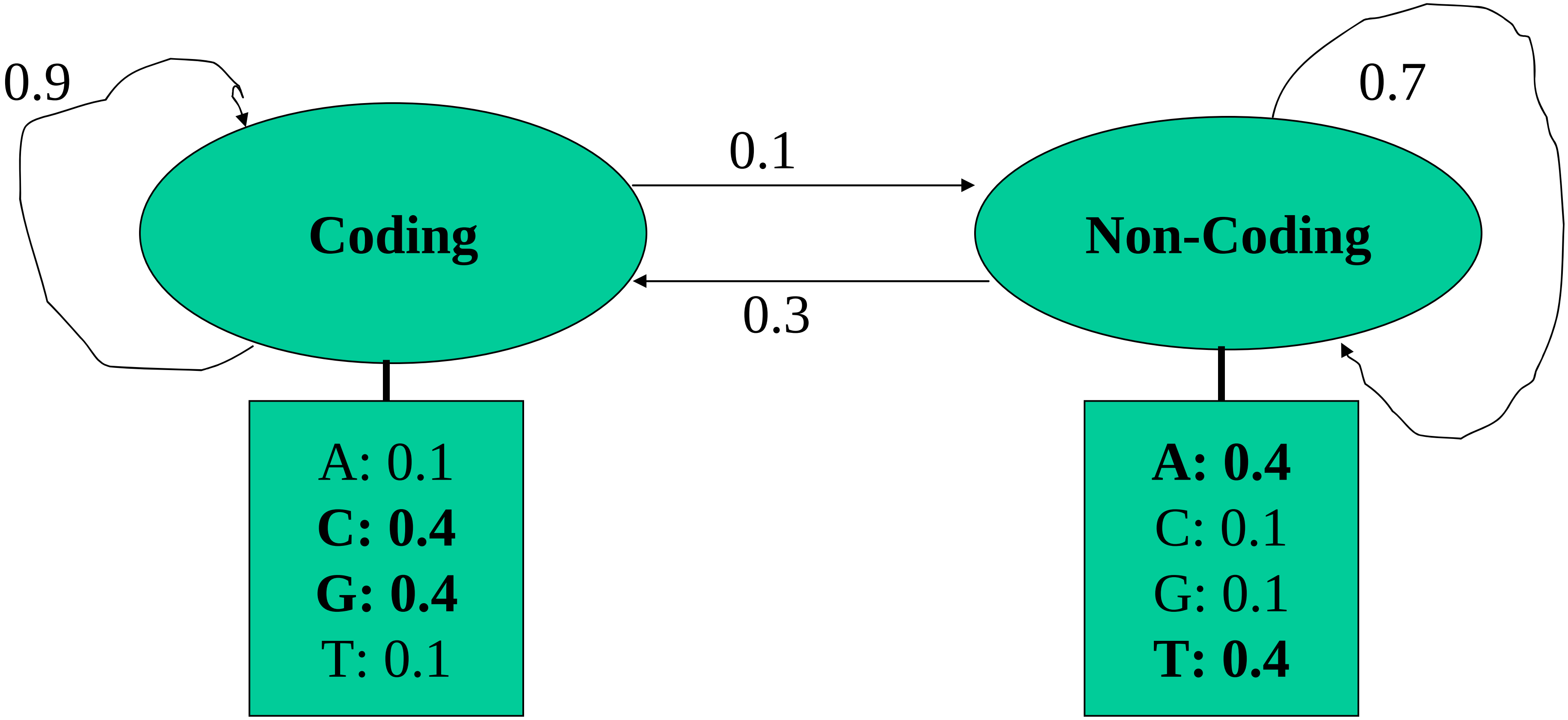
1: 0.25
2: 0.25
3: 0.25
4: 0.25

1: 0.1
2: 0.1
3: 0.4
4: 0.4

Observations: 12312421341213443432434443

State: FFFFFFFFFFFFFUUUUUUUUU

A simple gene prediction example



Observations: ATATCGGCCCGACCCGGGGTACTA

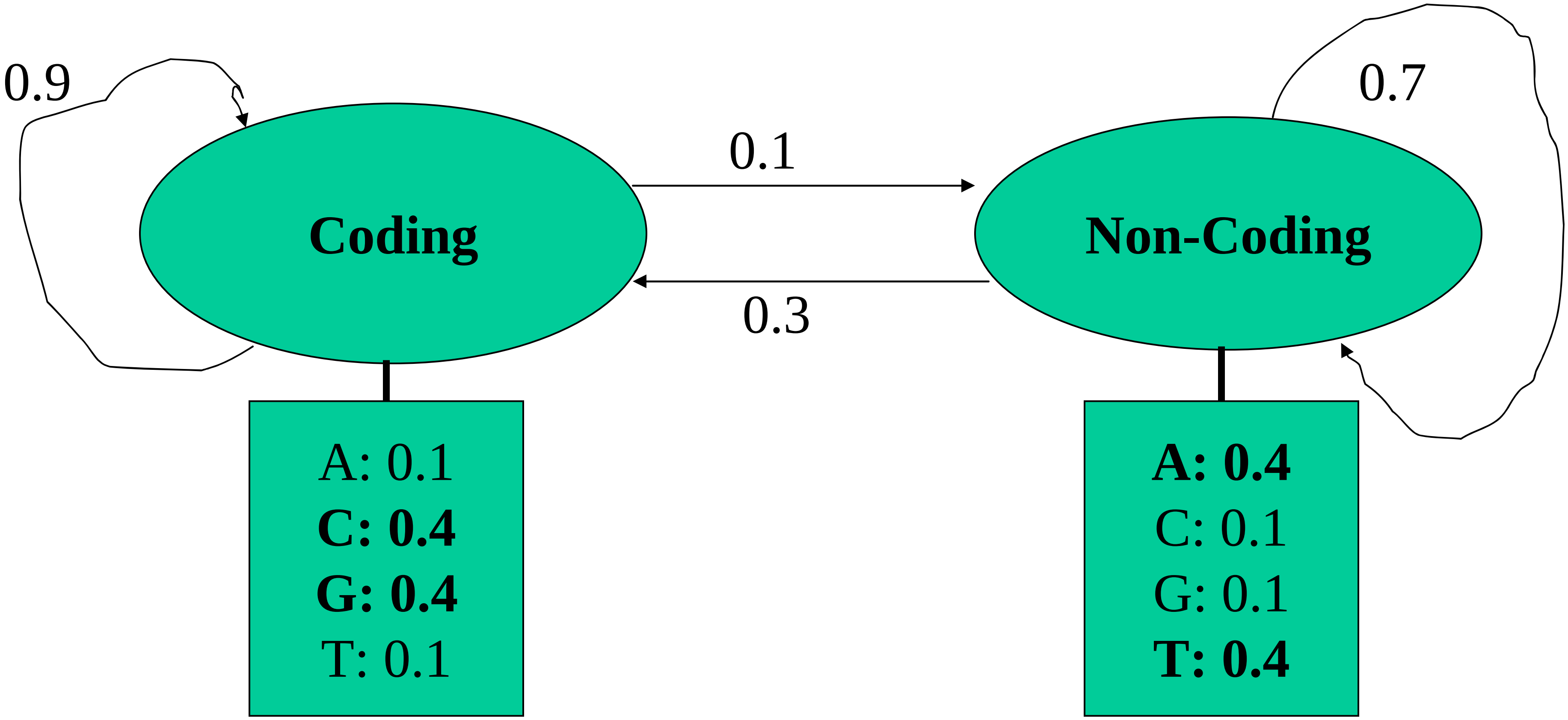
State:

Three Problems in HMM

1. **Prediction / Evaluation**: Given parameters of the model, compute the probability of an output sequence(Forward / backward Algorithm)
2. **Decoding**: Given parameters of the model, find most likely sequence of hidden states. (Viterbi Algorithm)
3. **Learning**: Given a set of sequences generated by the model, learn the most likely model parameters (transition/emission probabilities) (Baum-Welch Algorithm)

In gene prediction, we first use coding and non-coding sequences to train a HMM and then use known HMM to make prediction for a new sequence.

HMM Prediction



0.9

0.7

0.1

0.3

Coding

Non-Coding

A: 0.1
C: 0.4
G: 0.4
T: 0.1

A: 0.4
C: 0.1
G: 0.1
T: 0.4

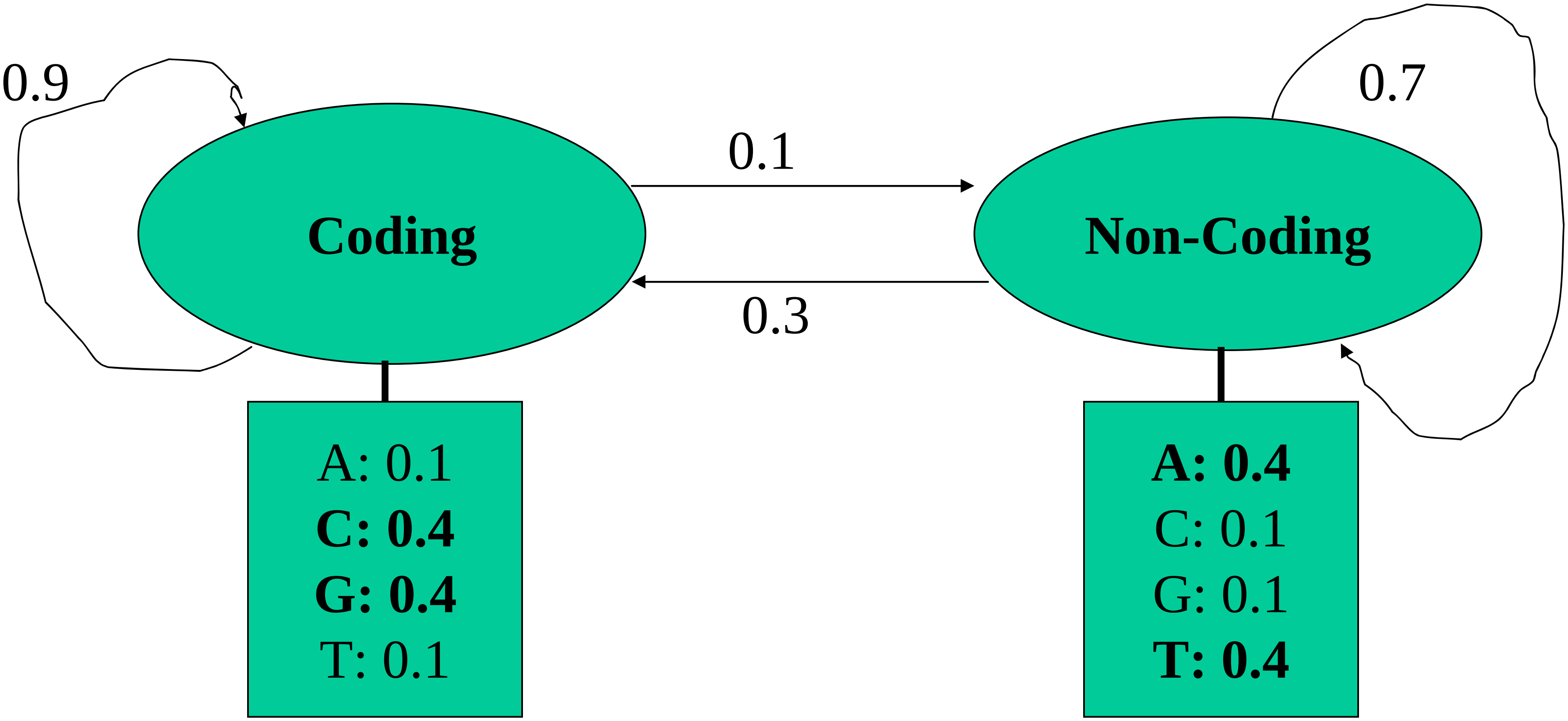
Observations: ATCT

Path 1: ***N->N->C->N*****, what is probability?**

Path 2: ***C->C->N->C*****, what is probability?**

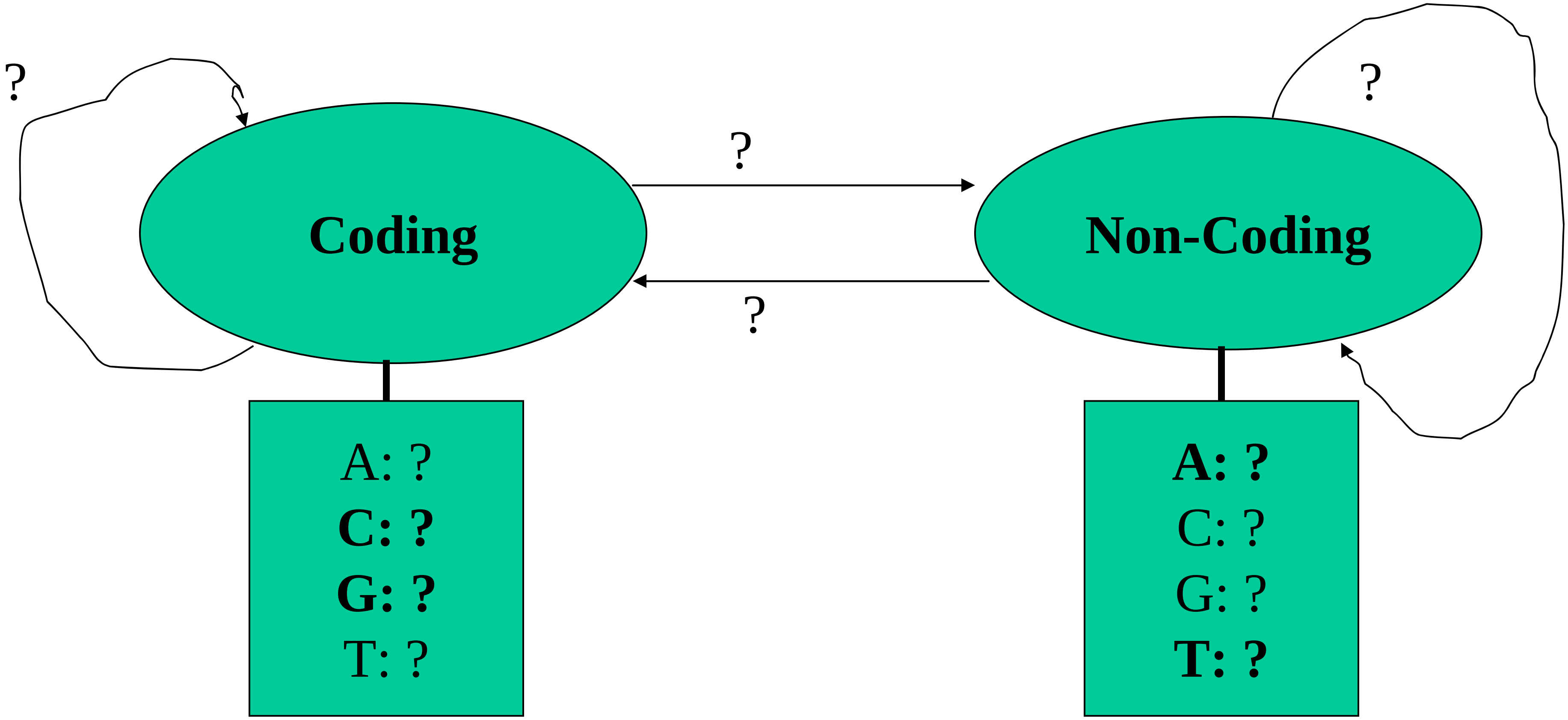
Goal: find the max probability that sequence is generated from HMM

HMM Decoding



Observations: ATCT
Best path?
Goal: find the path with maximum probability.

HMM Learning



?

Coding

?

?

Non-Coding

?

A: ?
C: ?
G: ?
T: ?

A: ?
C: ?
G: ?
T: ?

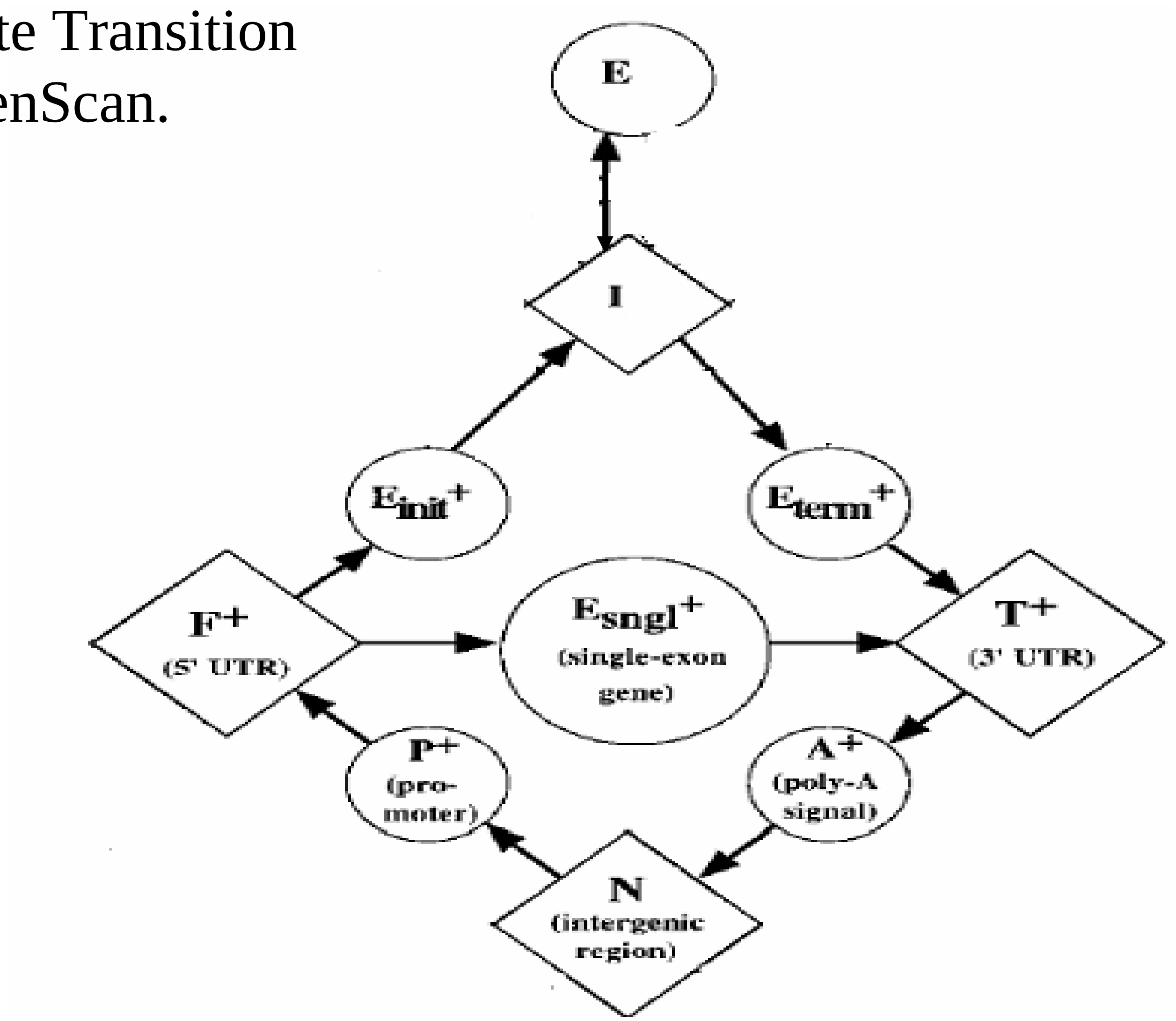
Observations: ATCT, CCCT, GTAC, TTAC,…

Goal: find the parameter values to fit data well.

GENSCAN

(genes.mit.edu/GENSCAN.html)

Simplified State Transition Diagram of GenScan.



454 LIFE SCIENCES
Genome Sequencer 20
Roche

Fast DNA Sequencing Machine: 25 million in four hours

Nature: http://www.nature.com/news/2006/060918/full/443258a.html

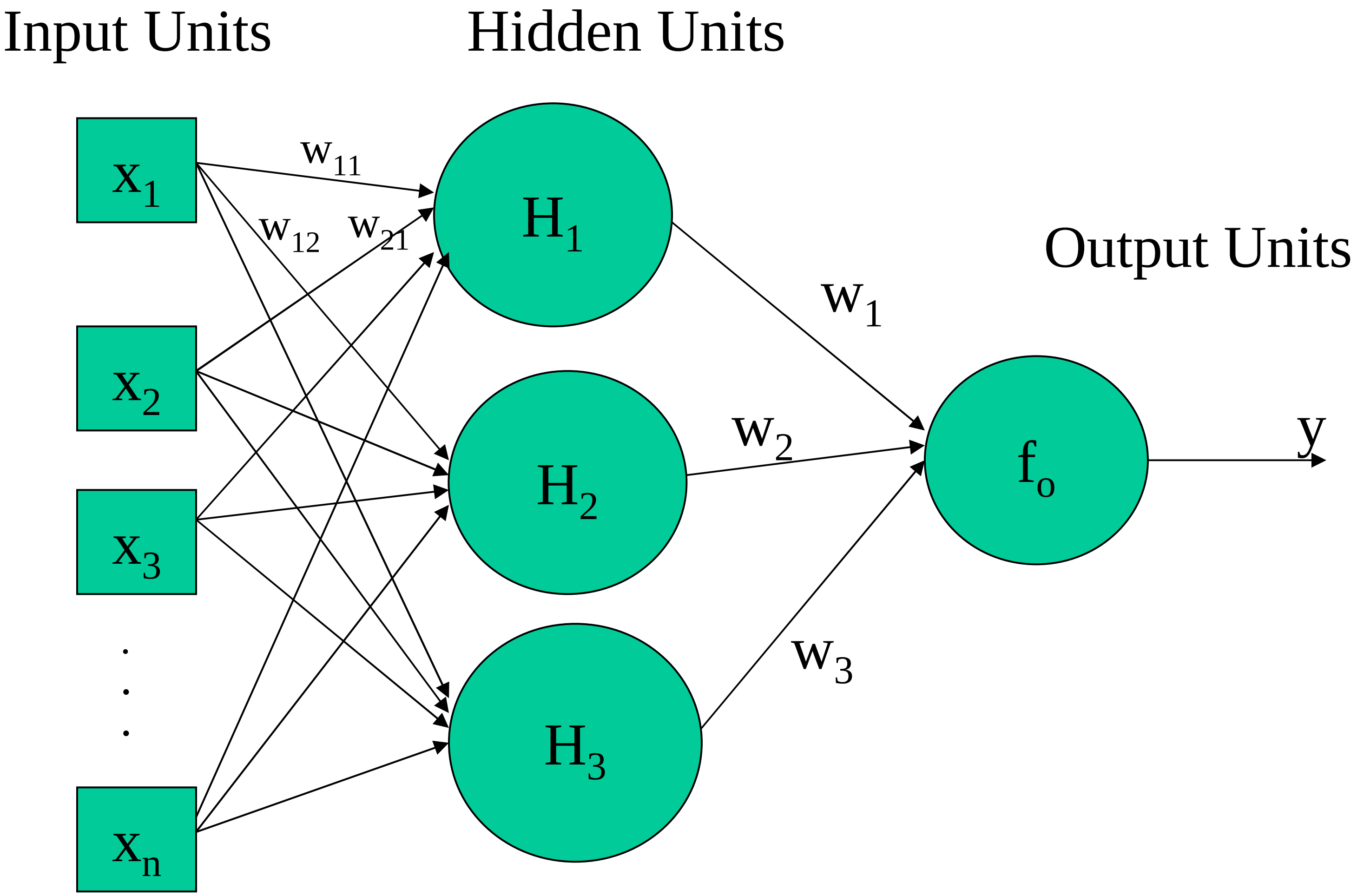
Neural Network

- Generative versus discriminative
- Neural network is a general, powerful classification / pattern recognition tool.
- Inputs to NN are features that describe the subject.
- Output of NN is a class label (or category) assigned to the subject.

Example of NN applications

- Given a set of words of a news article, predict its category (sports, politics, science, technology)
- Given a set of features describing a sequence of DNA, predict if it is coding region (exon) or not (intron)
- Goal is to learn a function to map input features to the target (category, real value)

A General Neural Network



Each weighted connection means the product of the output of one unit and the weight is sent to another unit as input. Each hidden unit and output unit have a transfer function to convert the sum of inputs into an output. Let transfer function of hidden unit be f_h (e.g., identity function) output unit to be f_o (e.g., sigmoid function, $1/(1+e^{-x})$).

Neural Network is a Universal Function Approximator

We can represent neural network as an function:

$$y = f_o(\sum_i w_i f_h(\sum_j x_j w_{ij}))$$

This function is universal, which means that any function y=*f*(x) can be approximated by this function accurately, given a set of appropriate weights W.

So, the key is to adjust weights W to make neural network to approximate the function of our interest. e.g., given input of sequence features, tell if it is a gene or not (1: yes, 0: no)?

Adjust Weights by Training

- How to adjust weights?
- Adjust weights using known examples (training data) $(x_1,x_2,x_3,\ldots x_n,y)$. This process is called training or learning
- Try to adjust weights so that the difference between the output of the neural network and y (called target) becomes smaller and smaller.
- Goal is to minimize Error (difference)

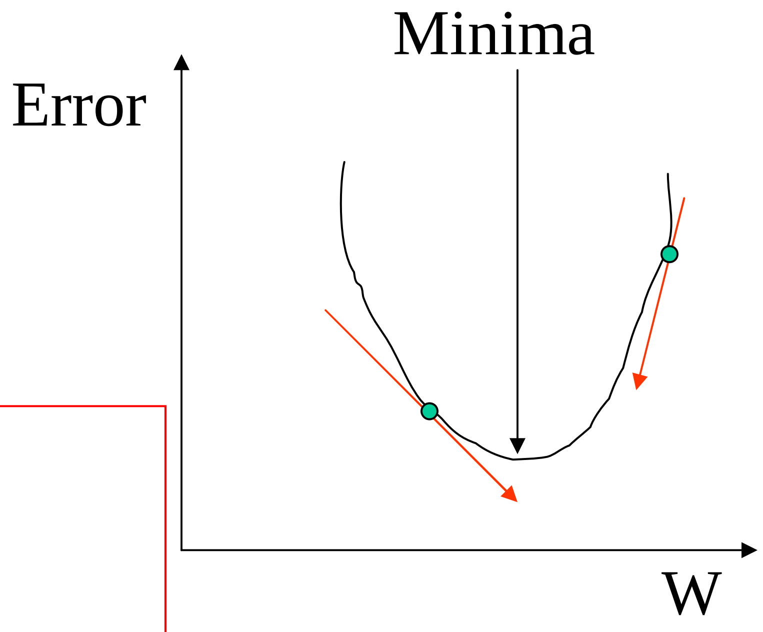
Adjust Weights using Gradient Descent (back-propagation)

Known:

Data: $(x_1,x_2,x_3,\ldots,x_n)$ (y)

Unknown weights w:

$w_{11}, w_{12},\ldots..$



```
Randomly initialize weights
Repeat
    for each example, compute output $o$
        calculate error $E = (o-y)^2$
        compute the derivative of $E$ over $w$: $dw=\frac{\partial E}{\partial w}$
        $w_{\text{new}} = w_{\text{prev}} - \eta * dw$
Until error doesn't decrease or max num of iterations
```

Note: η is learning rate or step size.

Prediction and Test Phase

- Weights are known.
- Given an input vector *X*, neural network will generate an output *O*.
- For binary classification/prediction, there is only one output. If *O* > 0.5, it is positive (gene), else, it is negative (not gene).
- Evaluate neural network on test data

Neural Network Tools

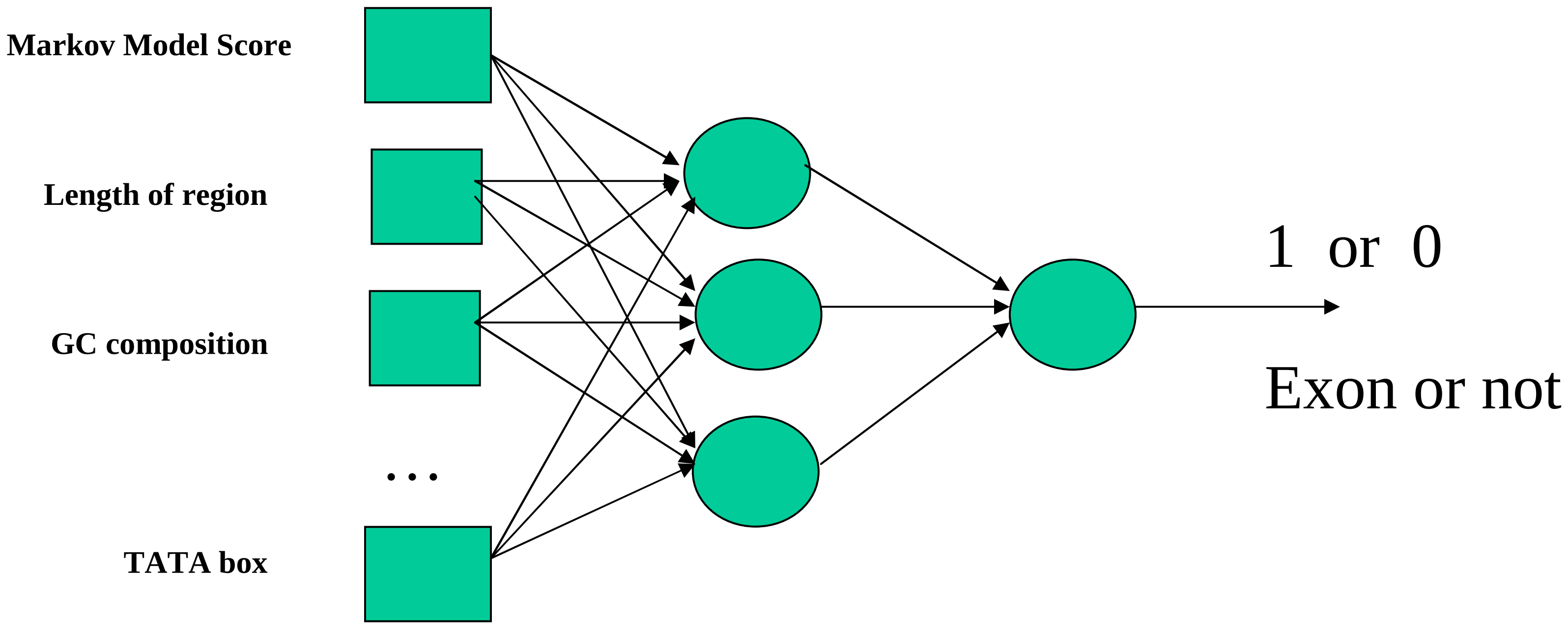
- Neural network has become a standard classification tool.
- The key thing left for user is to extract features (or inputs X), assign outputs, and control training.
- Pick a standard tool to train a neural network model (weights) and use it in prediction.
- Some tools: Weka (Java), NNClass (C++), and Neural Networks in MatLab

NNClass: http://www.eecs.ucf.edu/~jcheng/cheng_software.html
Weka: http://www.cs.waikato.ac.nz/ml/weka/

Neural Network for Gene Prediction

Given a sequence ACGGGGAATTCGTAGCT…, predict if it is an exon (coding region) or not.

Extract features from the sequence and feed them into neural Network.



Markov Model Score

Length of region

GC composition

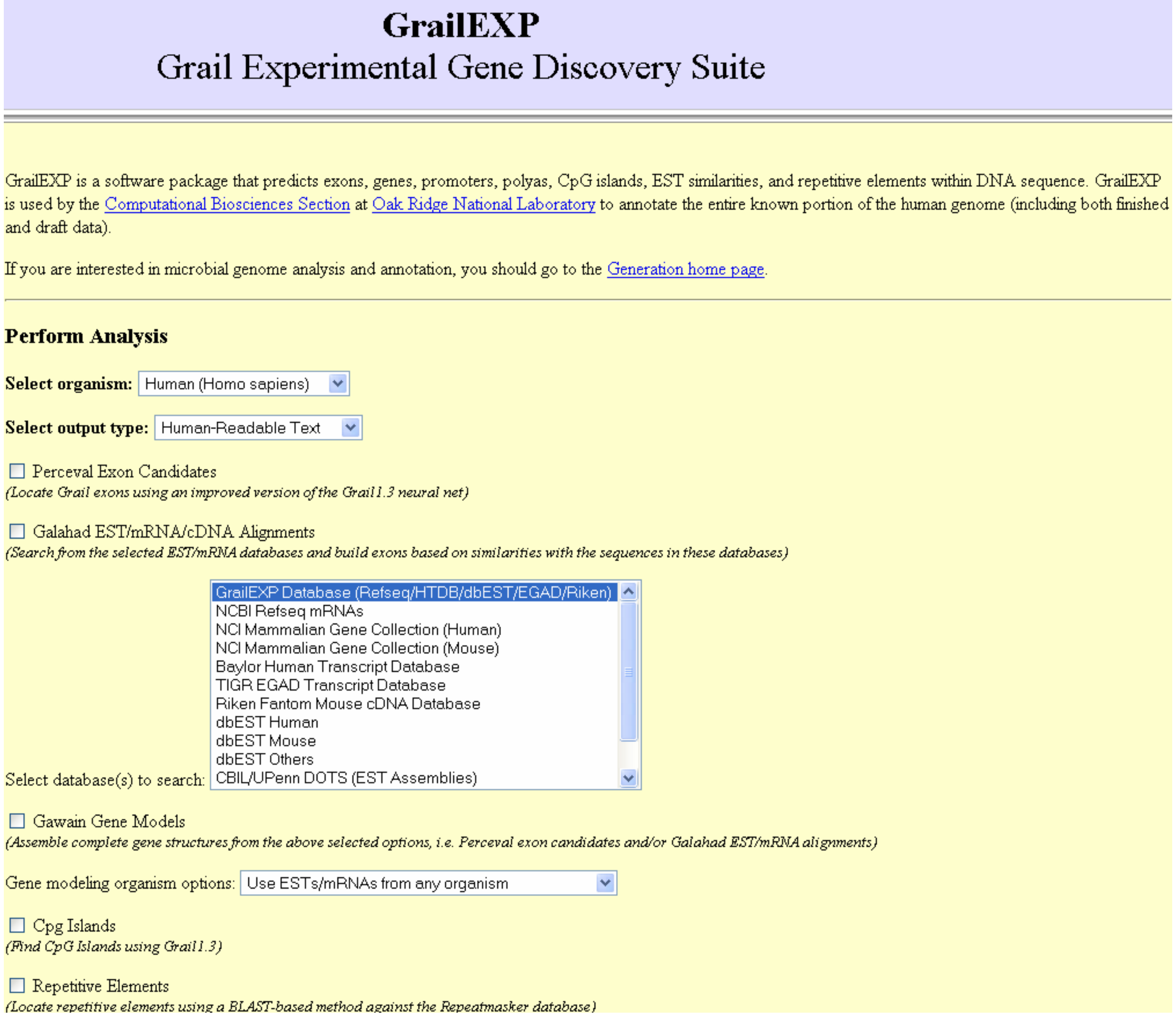
…

TATA box

1 or 0

Exon or not

Grail



GrailEXP

Grail Experimental Gene Discovery Suite

GrailEXP is a software package that predicts exons, genes, promoters, polyas, CpG islands, EST similarities, and repetitive elements within DNA sequence. GrailEXP is used by the Computational Biosciences Section at Oak Ridge National Laboratory to annotate the entire known portion of the human genome (including both finished and draft data).

If you are interested in microbial genome analysis and annotation, you should go to the Generation home page.

Perform Analysis

Select organism: Human (Homo sapiens)

Select output type: Human-Readable Text

□ Perceval Exon Candidates
(Locate Grail exons using an improved version of the Grail1.3 neural net)

□ Galahad EST/mRNA/cDNA Alignments
(Search from the selected EST/mRNA databases and build exons based on similarities with the sequences in these databases)

Select database(s) to search:

- GrailEXP Database (Refseq/HTDB/dbEST/EGAD/Riken)
- NCBI Refseq mRNAs
- NCI Mammalian Gene Collection (Human)
- NCI Mammalian Gene Collection (Mouse)
- Baylor Human Transcript Database
- TIGR EGAD Transcript Database
- Riken Fantom Mouse cDNA Database
- dbEST Human
- dbEST Mouse
- dbEST Others
- CBIL/UPenn DOTS (EST Assemblies)

□ Gawain Gene Models
(Assemble complete gene structures from the above selected options, i.e. Perceval exon candidates and/or Galahad EST/mRNA alignments)

Gene modeling organism options: Use ESTs/mRNAs from any organism

□ Cpg Islands
(Find CpG Islands using Grail1.3)

□ Repetitive Elements
(Locate repetitive elements using a BLAST-based method against the Repeatmasker database)

Web:http://compbio.ornl.gov/grailexp/

Gail combine both neural network and homology search

Other Tools

- Grail: http://compbio.ornl.gov/grailexp (Neural Network and EST database search)
- HMMgene: www.cbs.dtu.dk/services/HMMgene

  (use HHM)
- GeneParser: http://beagle.colorado.edu/~eesnyder/GeneParser.html (dynamic programming and neural network)

Ten Topics

- 1. Introduction to Molecular Biology and Bioinformatics
- 2. Pairwise Sequence Alignment Using Dynamic Programming
- 3. Practical Sequence/Profile Alignment Using Fast Heuristic Methods (BLAST and PSI-BLAST)
- 4. Multiple Sequence Alignment
- 5. **Gene Identification**
- 6. Phylogenetic Analysis
- 7. Protein Structure Analysis and Prediction
- 8. RNA Secondary Structure Prediction
- 9. Clustering and Classification of Gene Expression Data
- 10. Search and Mining of Biological Databases, Databanks, and Literature