Markov Models for Genome Annotation

The bigger picture

- In bioinformatics, we want to gather, analyze, integrate, and visualize data to form a biological opinion
  - Sometimes that data comes from what we know already
    - Online databases (GenBank, Pubmed, etc)
  - Sometimes that data comes from hints and guesses about other data
    - Simulation based on physical laws

The bigger picture

- In bioinformatics, we want to gather, analyze, integrate, and visualize data to form a biological opinion
  - Sometimes that data comes from what we know already
    - Online databases (GenBank, Pubmed, etc)

    Knowledge Based
  - Sometimes that data comes from hints and guesses about other data
    - Simulation based on physical laws

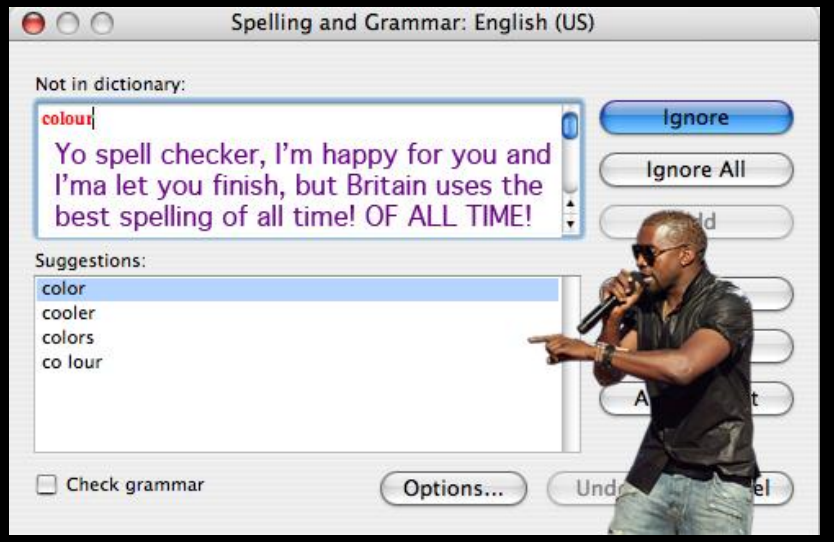
    Ab Initio

Knowledge based and Ab Initio Predictions

- Both approaches can be very successful or fail spectacularly in their predictive power

Knowledge based and Ab Initio Predictions

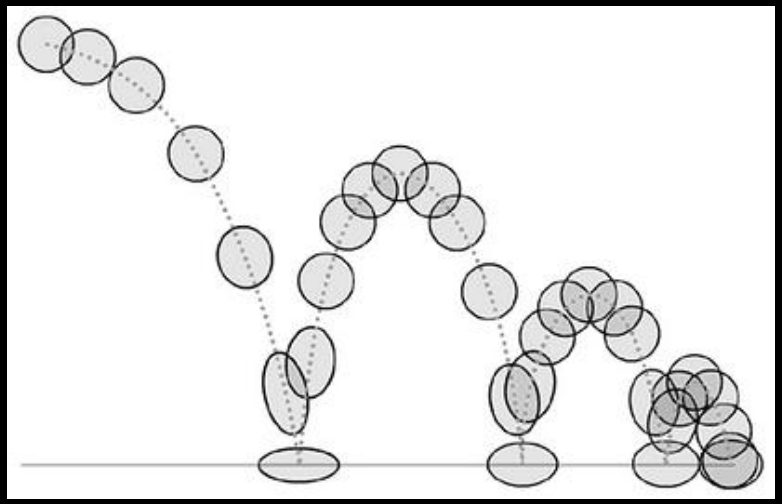
- Both approaches can be very successful or fail spectacularly in their predictive power
- A case for Knowledge Based Predictions: Spellchecking
- The computer uses dictionaries to predict the word the user is trying to spell
- Using English “spelling rules” will fail, because of all the exceptions in the language.



Knowledge Based spellcheck would be more successful than ab initio methods.. if anyone even tried to make them.

Knowledge based and Ab Initio Predictions

- Both approaches can be very successful or fail spectacularly in their predictive power
- A case for Ab initio predictions: mechanical simulation
- Based on Newtonian Physics, the computer predicts instantaneous position, velocity, accel.
- Knowledge based methods would look up a huge range of data, be less accurate



Ab initio simulation techniques enable the generality of physical systems in a way that knowledge based methods can't

Most technologies are in between

- Most effective methods use information from ab initio computations and from knowledge
  - Navigation planning software
    - Speed-limit information for theoretical trip time (ab initio)
    - Maps and historical traffic information (knowledge base)
  - Protein Folding software
    - Predicts the 3D structure of proteins based on their amino acid sequence
    - Physical simulation of molecules (ab initio)
    - Existing structures of related proteins (knowledge base)

Glimmer, gene prediction software

- The most accurate gene prediction technique is BLAST itself
  - If your gene exists, or has a close homolog in genBank, BLAST will find it
  - But not every gene has homologs in genBank
- We are sequencing new organisms every day, and there are many new genes we don't know about
- Glimmer is a combination of ab initio and knowledge based gene prediction
- GeneMark uses methods with similar mathematical foundations, structures

The original Glimmer paper:

Salzberg SL, Delcher AL, Kasif S, White O. "Microbial gene identification using interpolated Markov Models." Nucleic Acids Research (1998) 26(2): 544-8

Outline

- What is the Markov Property?
- What is a Markov Model?
- How do we apply Markov Models to gene finding?
- What is an Interpolated Markov Model (IMM)?
- How do we apply IMMs to gene finding?
- How well does Glimmer work, anyway?

Stochastic and Deterministic

- A Deterministic process is a rule by which the state of a system changes.
  - A mechanical clock is a good example of a deterministic process
    - 1:00:00, then
    - 1:00:01, then
    - 1:00:02..
- A stochastic process (also random process) is a nondeterministic rule by which a system changes

Andrei Markov

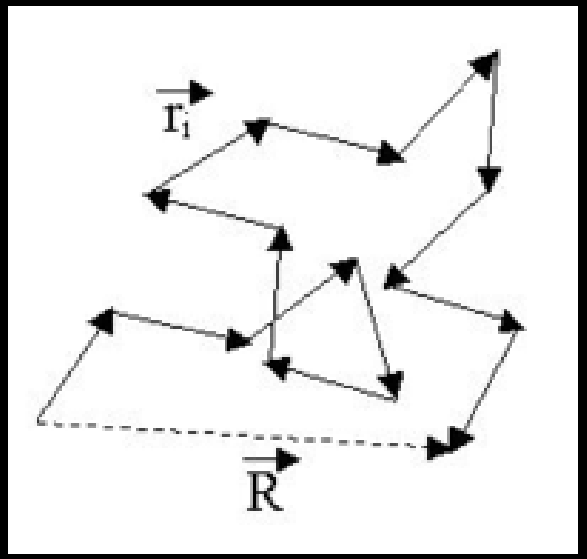
1856-1922

Studied Stochastic Processes

Examples of Stochastic Processes

- The order of ball numbers that come out of a lottery ball roller
  - The probability of any one of the balls is intended to be as even as possible
- A random walk between a discrete or continuous set of destinations
  - Destination selection may not be uniformly random: a person walking might have preferences, etc

Lotto Ball Roller



Random Walk

These processes are all models

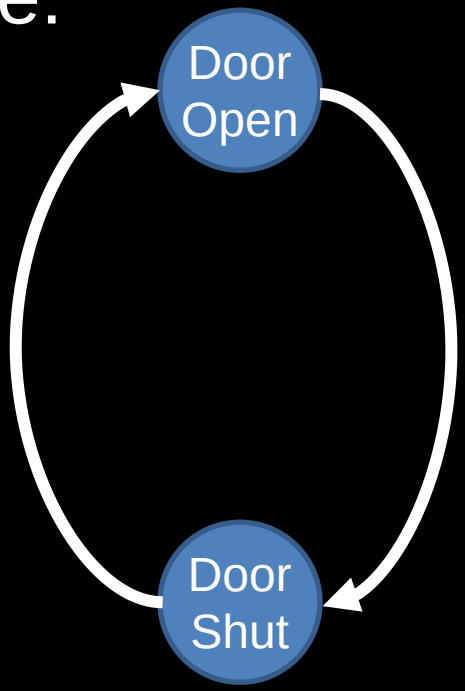
- One way to understand the behavior of systems is to build models based on rules that (we think) approximate the system
  - Hopefully our approximation captures the way the system works “in theory”
  - while eliminating a lot of things that make it hard to understand
- Think of this as reverse engineering
- If you had to figure out how a complex gear system worked, you would start writing down A turns B turns C…
- This is a model of gear connectivity

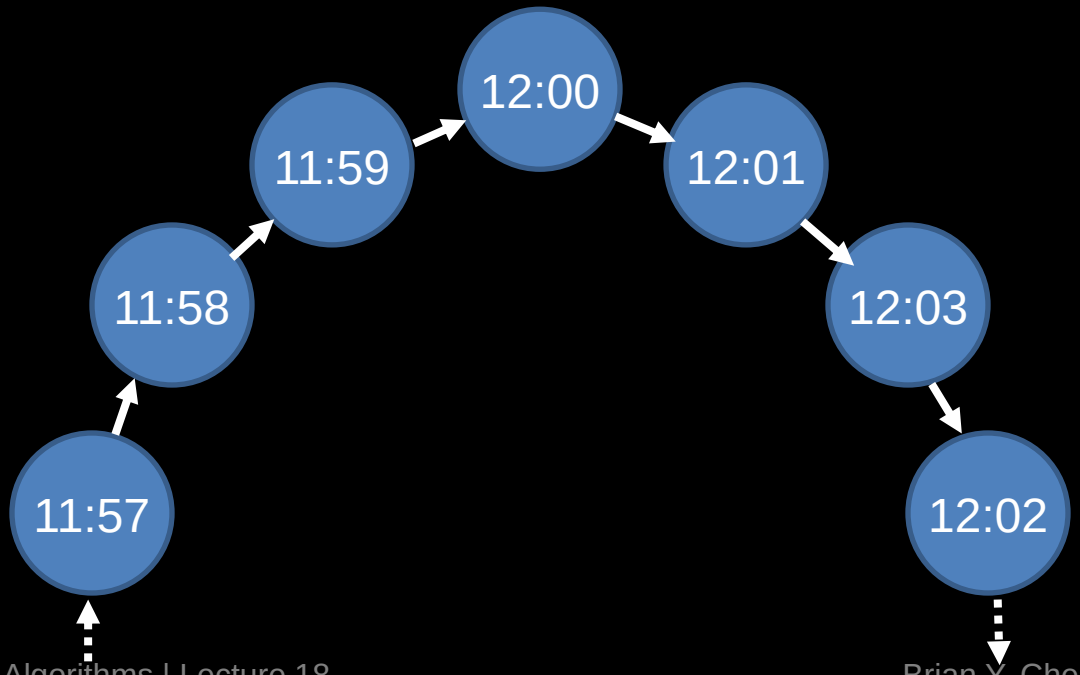
Why mathematicians were interested in models

- Many natural, physical, and mechanical systems can be modeled
- People are always designing more and more complex machines, and they don't always know if they will (or even can) work
- Mathematicians want a theoretical way to prove certain outcomes are guaranteed based the properties of model-able systems:
  - Very large CPU designs: do they have errors?
  - Unusual aircraft wing designs: do they fly?
  - Theoretical drug designs: do they bind?

A simple way to think of a model: as a graph

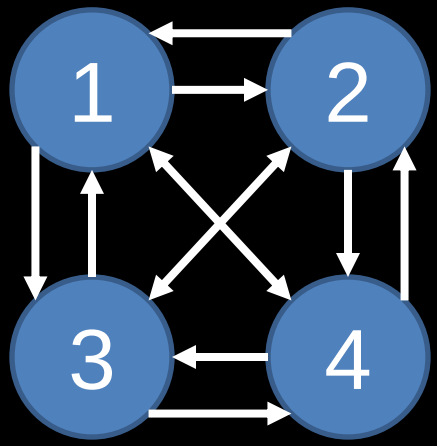
- Processes have states and transition properties
  - The Door in Star Trek has two states: open and closed
  - A digital 12 hour clock that does not count seconds has 12x60 states. One for each possible minute.
- Deterministic models always move to the next state:

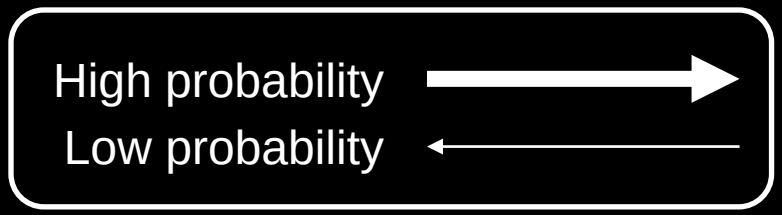




- Stochastic models move to different states based on a certain probability
  - 4 lottery balls w/ replacement (order of ball selects)

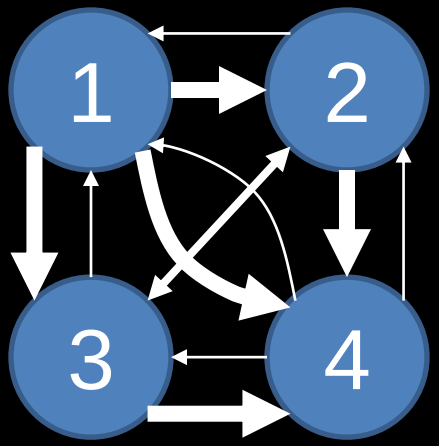
All Balls the Same





Some Balls Different

- The “4” is extra small and falls out easily
- the “1” ball is too large and tends to not fall out



What is a Markov Process?

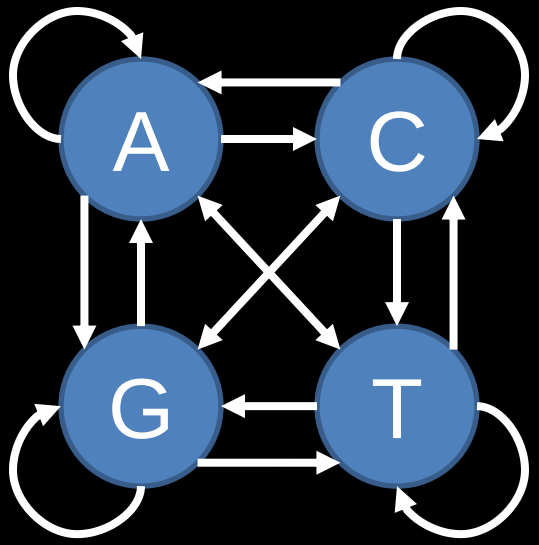
- A Markov Process is a Stochastic Process that fulfills the Markov Property:
  - The next state of the system is dependent only on randomness and the current state
    - not previous states
- Markov Processes have no memory

- This is not true for all systems, of course, but this is a good simplification to work with

Modeling DNA with Markov Models

- We can model the sequence of nucleotides as a record of the states of a Markov Process

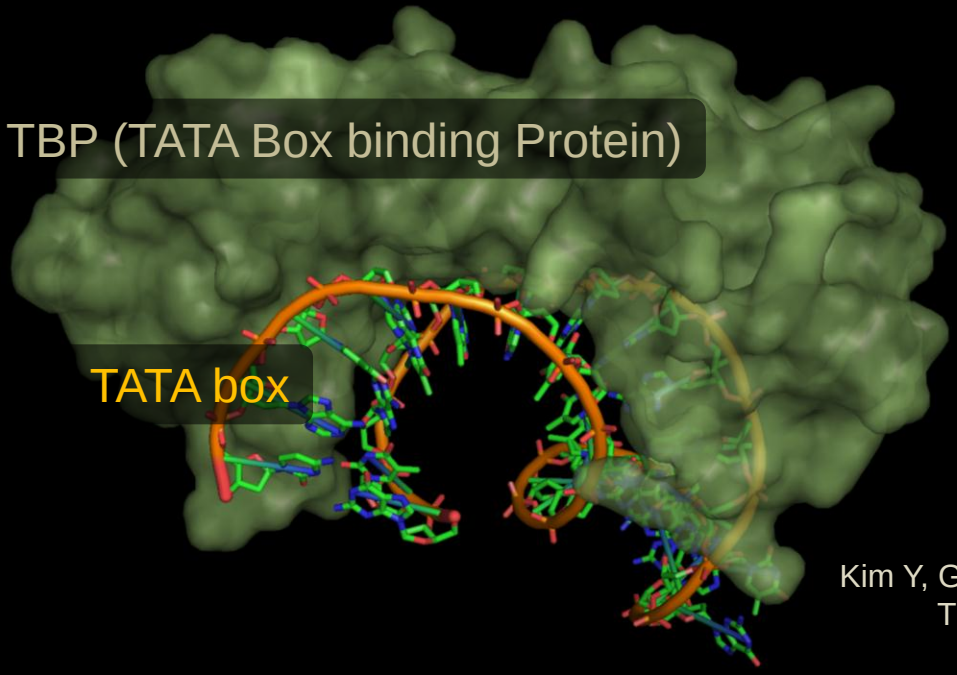
A C G G T A G C C G A

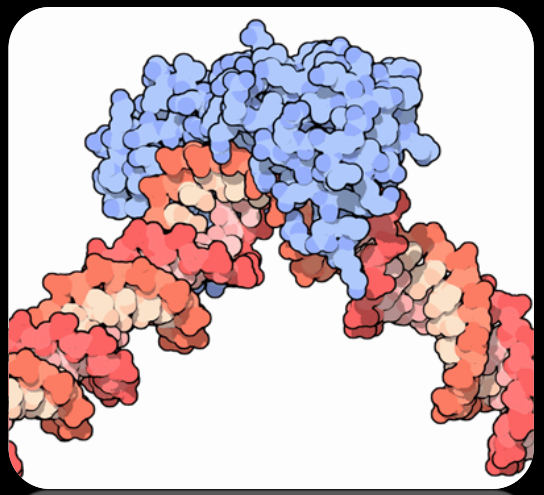


A first order Markov Model

Problems with first order Markov Models

- DNA is a complicated sequence; the next nucleotide doesn't simply depend on the last nucleotide, even if randomized
- If it did, the TATA box couldn't exist
  - The 'TATA' sequence adds additional flexibility to the DNA molecule that allows TBP to bind





Kim Y, Geiger JH, Hahn S, Sigler PB. Crystal structure of a yeast TBP/TATA-box complex. Nature, 1993. 365(6446):512-20.

Higher order Markov Models

- Model the next nucleotide based on several previous nucleotides
- The Markov Property must be maintained,
- Accomplished by increasing the # of states
- Each node connects to only four other nodes, eg:

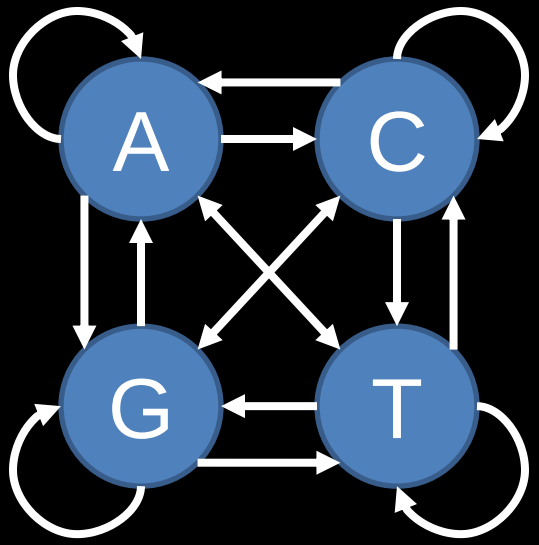
CA → AA
CA → AC
CA → AG
CA → AT

A second-order Markov Model for DNA (without edges)

Training Markov Models to find genes

- Transition probabilities (edges weights) are set to reflect the frequency of transitions observed in training data: gene sequences.

A C G G T A G C C G A



Transitions Observed

AA:0	GA:0
AC:1	GC:1
AG:0	GG:1
AT:0	GT:1
CA:0	TA:1
CC:1	TC:0
CG:2	TG:0
CT:0	TT:0

- Observed transitions are translated into edge frequencies

But training Markov Models can be tricky

- Training data: sequences that code for protein
- You want to use higher order Markov Models because they store more sophisticated information
- But higher order Markov Models have exponentially more states: 4^0, 4^1, 4^2, 4^3, 4^4, 4^5..
  - As a result, it becomes hard to find enough training data to be representative of all transitions.
  - There are useful 8^{th} order transitions, but most 8^{th} order transitions don't occur often enough to be well categorized.

Some intuition on higher order Models

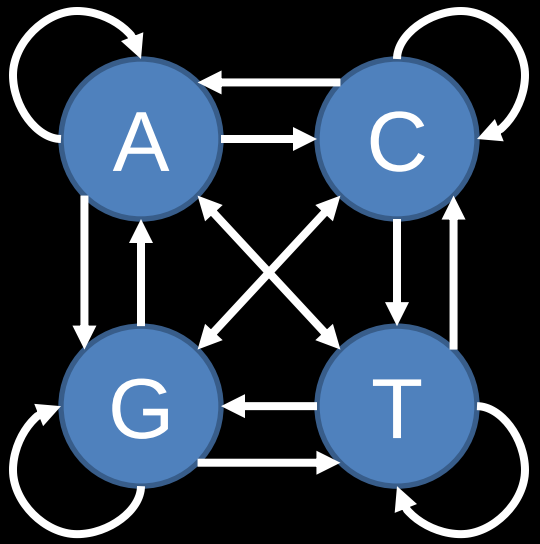
- Recall that codons of three nucleotides code for amino acids in proteins
- If you look at this critically, you see that amino acids depend most on the first and second bases
- The lower significance of the third nucleotide creates a periodicity in the Model:
  - "care", "care", "don't care"
  - "hi/low", "hi/low", "50/50"

Codons Found in Messenger RNA

First Base	Second Base: U	Second Base: C	Second Base: A	Second Base: G	Third Base
U	Phe	Ser	Tyr	Cys	U
	Phe	Ser	Tyr	Cys	C
	Leu	Ser	Stop	Stop	A
	Leu	Ser	Stop	Trp	G
C	Leu	Pro	His	Arg	U
	Leu	Pro	His	Arg	C
	Leu	Pro	Gln	Arg	A
	Leu	Pro	Gln	Arg	G
A	Ile	Thr	Asn	Ser	U
	Ile	Thr	Asn	Ser	C
	Ile	Thr	Lys	Arg	A
	Met	Thr	Lys	Arg	G
G	Val	Ala	Asp	Gly	U
	Val	Ala	Asp	Gly	C
	Val	Ala	Glu	Gly	A
	Val	Ala	Glu	Gly	G

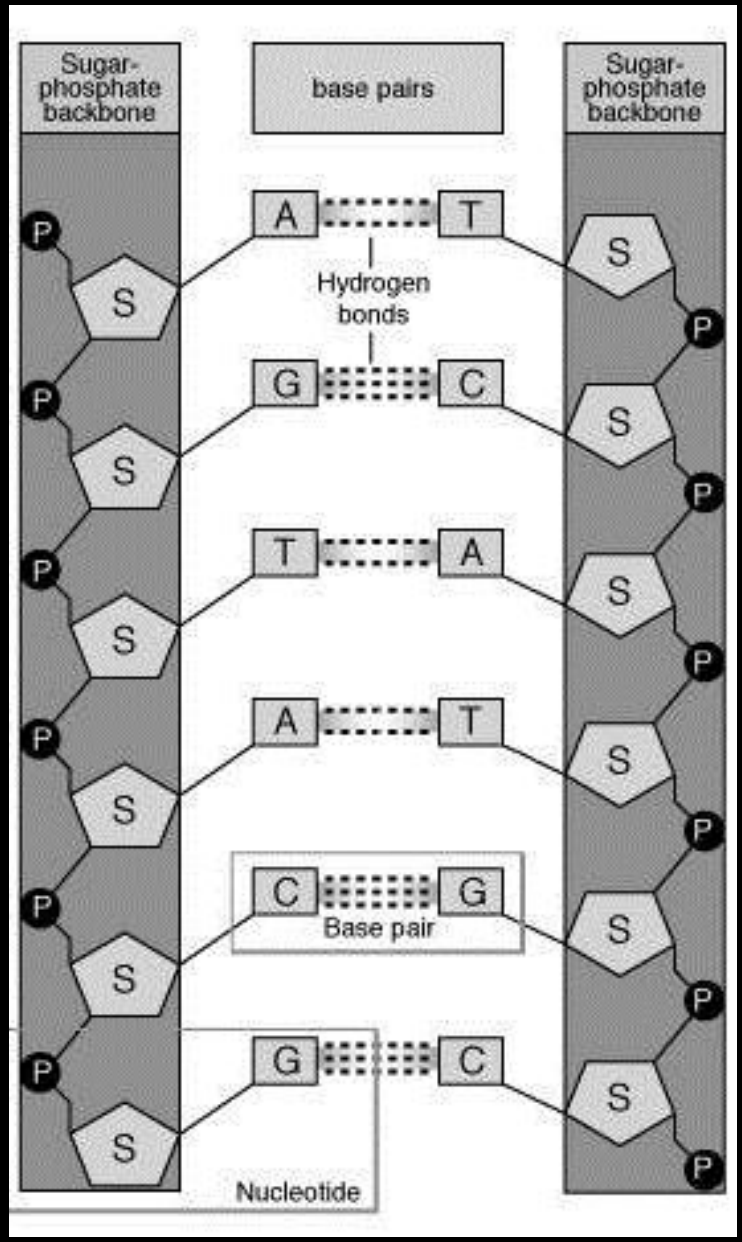
Using Markov Models to Predict Genes

- The transitions in the (trained) model M now reflects the transitions observed in real genes
- Given a sequence S = $s_1 s_2 s_3 s_4 s_5 s_6 s_7 \ldots$
- We want to know which parts of S code for a gene
- We do this by measuring the probability of each transition in the states of M, for each s_i
- We can plot these probabilities
  - High values in gene regions
  - Low values between genes



Using Markov Models to Predict Genes

- Glimmer runs six Markov Models simultaneously
  - Three run on the forwards strand
  - Three run on the backwards (complementary) strand
- Since you do not know the frame, you run with offsets of 0, 1, 2, so that one model must be in-frame
- Result: Models Named 1, 2, 3, -1, -2, -3



Scoring data from six different Models

- Based on the six different models, a single test sequence is scored in all models, and the highest scoring model is used as the output
- Alternatively, very high scoring regions in all frames can be returned
  - This is more appropriate for Virii, which re-read their genomes in different frames sometimes.

```
GLIMMER (ver. 3.02; iterated) predictions:
   orfID start end frame score
-------- ---- ---- -- ----
>Cauliflower Mosaic Virus
Orf00001   13  303 +1 6.63
Orf00002  370 1347 +1 3.15
orf00003 1349 1828 +2 7.24
orf00004 1821 2219 +3 2.92
orf00007 2201 4024 +2 3.32
orf00009 4083 5672 +3 3.11
orf00012 5629 7410 +1 2.87
```

Here Glimmer reports high scoring predictions for all frames (from NCBI Glimmer)

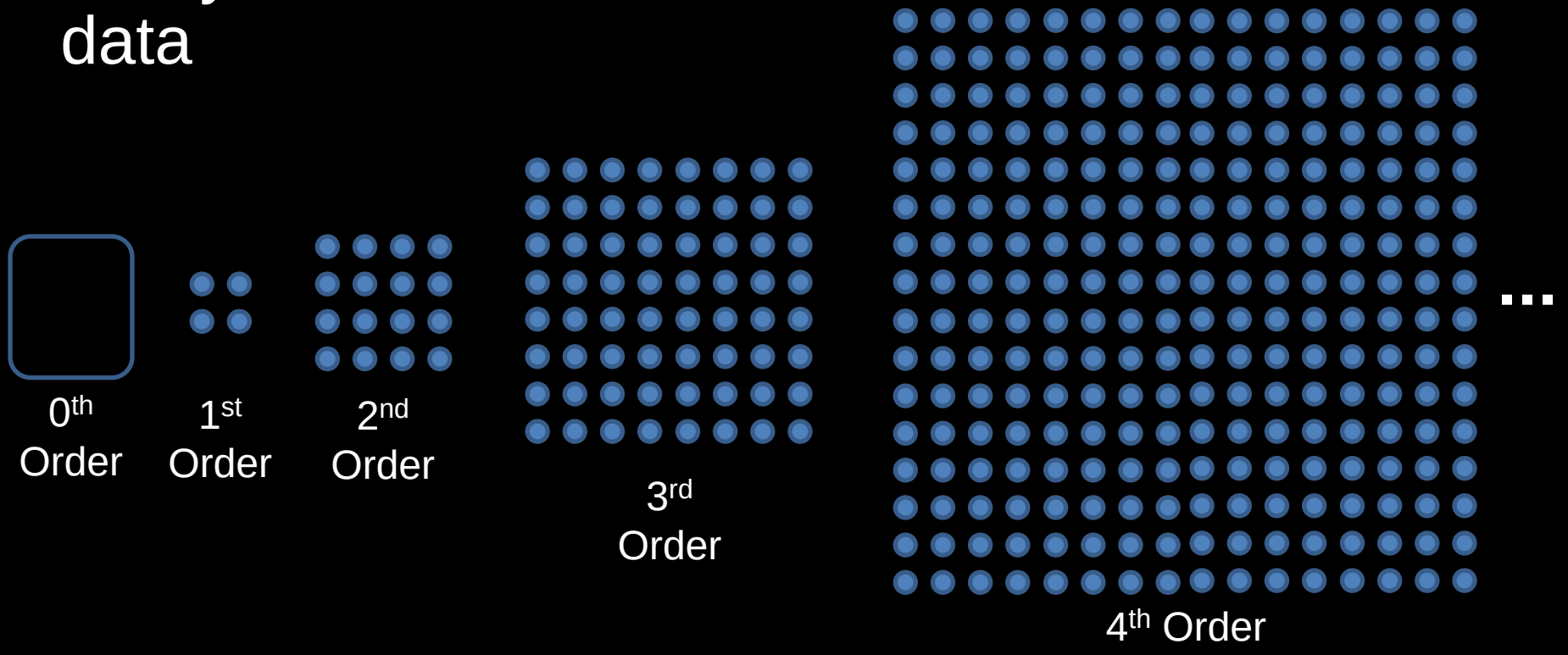
Interpolated Markov Models in Glimmer

- Glimmer maintains an Interpolated Markov Model for each frame, instead of fixed-order Markov Models
- Interpolated Markov models are intended to get the advantages of a higher-order model while avoiding the training difficulties

- How does an Interpolated Markov Model work?

Interpolated Markov Model

- Not really a Markov Model, but a predictor based on an assembly of Markov Models
- An 8th order interpolated Markov Model has 9 fixed-order models: 0th, 1st, 2nd, .. 8th.
- Every fixed-order model is trained on the same data



Training an Interpolated Markov Model

- The idea is to use the highest order model for which there is sufficient state transition information from the training data.
- A minimum threshold of 400 samples is required for every transition
  - Some transitions thus do not exist in the higher fixed order Markov Models

Making Predictions with Interpolated MMs

- For each transition, the edge probability is determined in each internal Fixed-order Markov Model
- A linear combination of scores is assembled from each Markov Model (including zero scores from higher order models that had insufficient information)

$$Si = w_1m_1 + w_2m_2 + w_3m_3 + \ldots w_8m_8$$

- W_i increases as i increases, to rely more on higher order Markov Models when possible.

Testing Glimmer on the Influenza Genome

- Test Data: A highly selective list of genes
  - Only genes with > 500 base reading frame
    - Reading frames this long almost never occur in DNA that does not code for a protein
  - The gene does not overlap any other gene longer than 500 bases
    - Overlapping genes have unusual statistical properties
  - Total: 1717 annotated genes for test data
- Comparison:
  - Glimmer vs a Fixed-length 5th order Markov Model
    - Run both on the Training Genome
    - Captures the periodicity in two codons
    - Enough training data to fill state transitions for this dataset

Experimental Results

Model	# Genes found	Genes Missed	Additional genes
Glimmer (IMM)	1680/1717 (97.8%)	37	209
5th order markov	1574/1717 (91.7%)	143	104

- Of the 37 genes missed by Glimmer, the 5th order Markov model found one
- Overall result:
  - Higher sensitivity (e.g. more genes found)

$$\text{Sensitivity} = \frac{TP}{FN + TP}$$

  - Lower specificity (e.g. more false positives)

$$\text{Specificity} = \frac{TN}{TN + FP}$$

Testing Glimmer on *Helicobacter Pylori*

- The previous experiment tested Glimmer on it's own training data.
  - That test does not demonstrate that Glimmer can predict new genes
  - It does demonstrate that it can recall things based on it's training
  - It also demonstrates that it can recall better than a 5^{th} Order Markov Model

- New Test: Using an ad hoc training set, predict genes in Helicobacter Pylori

Training Set

- Open reading frames > 500 bases were identified from H. Pylori (e.g. start-stop codon pairs)
- Glimmer is trained on this crude set
- An annotated set of genes is selected from the training set based on these conditions:
  - Protein-level sequence similarity to public sequence archives
  - A gene predicted by GeneMark
  - Intergenic ORFs (e.g. not coding) found by not being in databases nor identified by GeneMark
  - The annotated set is the “correct set of genes”

Results on the H. Pylori dataset

- Glimmer finds 1548 out of 1590 correct genes

Comparison to GeneMark

- Using only the 974 genes found in sequence databases “highest accuracy”
  - Glimmer finds 21 genes that GeneMark misses
  - GeneMark finds 1 gene that Glimmer misses
  - Agreement: 945/975 genes
- Comparison to Genemark-HMM “new version”
  - GeneMarkHMM misses 23
  - Glimmer finds 15 that GeneMarkHMM misses, GeneMarkHMM finds none that Glimmer misses
  - Agreement on 951/974 genes

Conclusions on Glimmer

- This is a reasonably reliable way to find genes within bacterial genomes
  - It also happens to work okay for virii
- Interpolated Markov Models enable information from higher order Markov models to be used without depending on incomplete training data elsewhere in the model
- Glimmer is very sensitive, but it still identifies a lot of false positives, which is problematic
  - Needs ways to eliminate genes that are not real

Markov Models overall

- A technology that has been applied extensively in prediction applications, and has worked very well.
- All trainable algorithms are inevitably limited or enabled by their training data; the best models cannot succeed without good training data – if such data does not exist, they don't work.

- Understanding when Markov Models will definitely make accurate predictions, and when they cannot, is a very important aspect of their use
  - Many people just blindly try these things, and don't understand why they work / don't work.

Questions

- Note: Project 1 grades will be out today, via email