# HMMs

In the context of gene finding and CpG islands

# Overview

- CpG islands
  - Markov Chains and Hidden Markov Models to tackle CpG islands in DNA.
- Gene finding

# CpG Islands

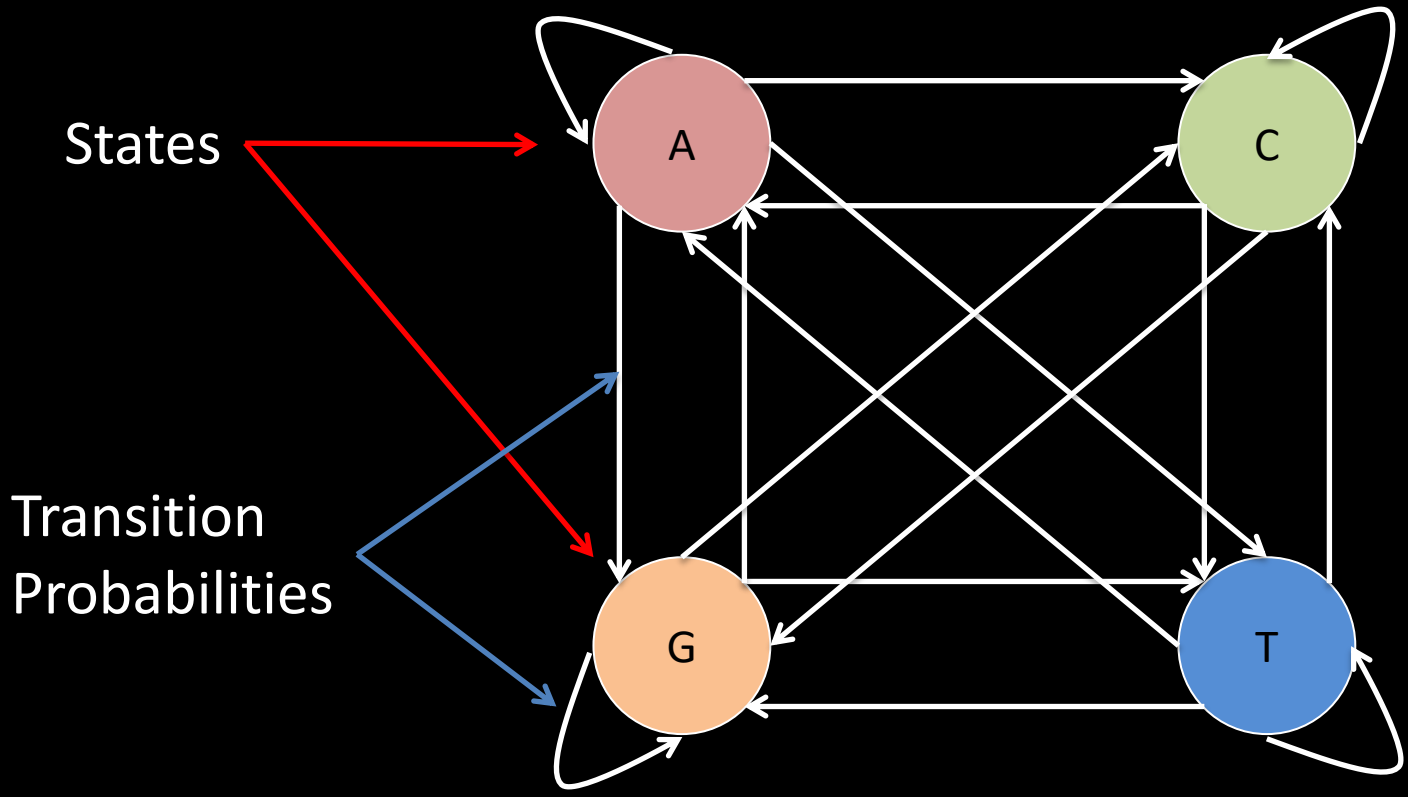
- CpG islands are regions of the genome with an abnormally high occurrence of CG di-nucleotides.
  - Regions of l > 200 base pairs.
  - Occurrence of CGs higher than (#C + #G) / l
- CpG islands are often found at the 5’ end of genes (promotor region) -> gene finding.
- Methylation of CpGs in a promotor region can lead to silencing of genes.

Gardiner-Garden M, Frommer M (1987). CpG islands in vertebrate genomes. Journal of Molecular Biology 196 (2): 261–282. doi:10.1016/0022-2836(87)90689-9

# Model of a CpG Island

- If we formally define a CpG island like this:
  - A DNA sequence of length >=200 with a C+G content of >=50% and a ratio of observed-to-expected number of CpG's that is above 0.6,
- Then we can model such a feature with a Markov Chain.

# A Markov Chain



# CpG Islands: models

CpG Island (model +)

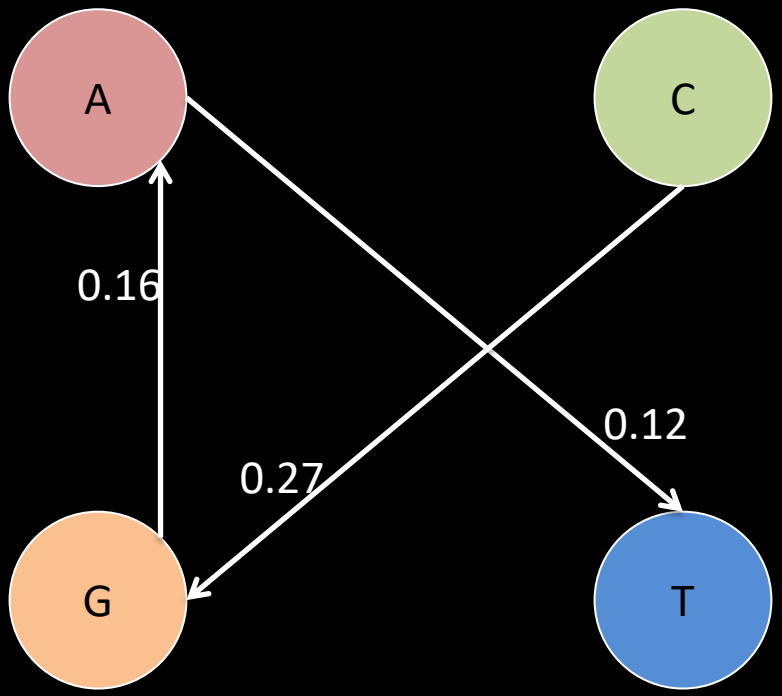
| | A | C | G | T |
|---|---|---|---|---|
| A | 0.180 | 0.274 | 0.426 | 0.120 |
| C | 0.171 | 0.368 | **0.274** | 0.188 |
| G | 0.161 | 0.339 | 0.375 | 0.125 |
| T | 0.079 | 0.355 | 0.384 | 0.182 |

Not CpG Island (model -)

| | A | C | G | T |
|---|---|---|---|---|
| A | 0.399 | 0.205 | 0.285 | 0.210 |
| C | 0.322 | 0.298 | **0.078** | 0.302 |
| G | 0.248 | 0.246 | 0.298 | 0.208 |
| T | 0.177 | 0.239 | 0.292 | 0.292 |

Data from Durbin et al, Biological Sequence Analysis, Cambridge University Press, 1998, p. 50.

# CpG Markov Chain Example




| | A | C | G | T |
|---|---|---|---|---|
| A | 0.180 | 0.274 | 0.426 | 0.120 |
| C | 0.171 | 0.368 | **0.274** | 0.188 |
| G | 0.161 | 0.339 | 0.375 | 0.125 |
| T | 0.079 | 0.355 | 0.384 | 0.182 |

- What is the probability of the string CGAT in the markov chain model for a CpG island?
- 0.27*0.16*0.12=0.0053
- For the non CpG model?
- 0.08*0.25*0.21=0.0041

# Comparing Sequence and Models

- Let there be a sequence $X = (x_1, x_2, \ldots, x_n)$
- And a Markov Chain MC with states ACGT and transition probabilities $a_{xy} = P(state\ x \mid state\ y)$
- Then the probability of X under the Markov model is

$$P(X \mid MC) = \prod_{i=2}^{n} a_{x_{i-1}x_i}$$

- CpG application: Compare model+ with model- to decide whether a sequence is likely to be a CpG island or not with the following score (log odds):

$$S(X) = \log \frac{P(X \mid MC+)}{P(X \mid MC-)} = \sum_{i=2}^{n} \log \frac{a^{+}_{x_{i-1}x_i}}{a^{-}_{x_{i-1}x_i}}$$

# More Examples

S(CGCGCGCGCG) = 1.1 (scores normalised by sequence length)

S(ATATATATAT) = -0.87

S(AGTCGAGCG)?

= 0.5

Known sequences show a cutoff at around -0.05 and 0.1, with an uncertain region in between.

R-code for CpG island MC available at: http://www.r-bloggers.com/discrimination-between-cpg-islands-and-random-sequences-using-markov-chains/

Beware: There is an error in the code to calculate S: It does not use the correct transitions, but only every other!

# Training a Markov Chain for CpG islands

- To train a model on data one simply counts all observed di-nucleotides in known CpG islands, and derives the transition probabilities from the frequencies accordingly.
- The same is done for a set of known non CpG islands.
- This prior knowledge allows us to explore new sequence contents in other
  - Parts of the genome
  - Organisms (needs similar properties)

# Markov Chain Limitations

- Given good training data, a Markov Model can answer the question for us: is this sequence likely to be a CpG island?
- But what if our question is: where in this large sequence (genome) are the CpG islands?
- Discuss.
- Enter: Hidden Markov Models (HMMs)

# HMM Motivation

- Since Markov chains only compares the likelihood that an entire sequence of observations is generated by the pre-trained model, it is unsuited to the localization task.
- But we can create a larger model that can generate both CpG islands and non CpG islands.
- Then, we can calculate the probability of a path through this model.
- A path can change in and out of CpG island states, therefore identifying the location of CpG islands.

# HMMs

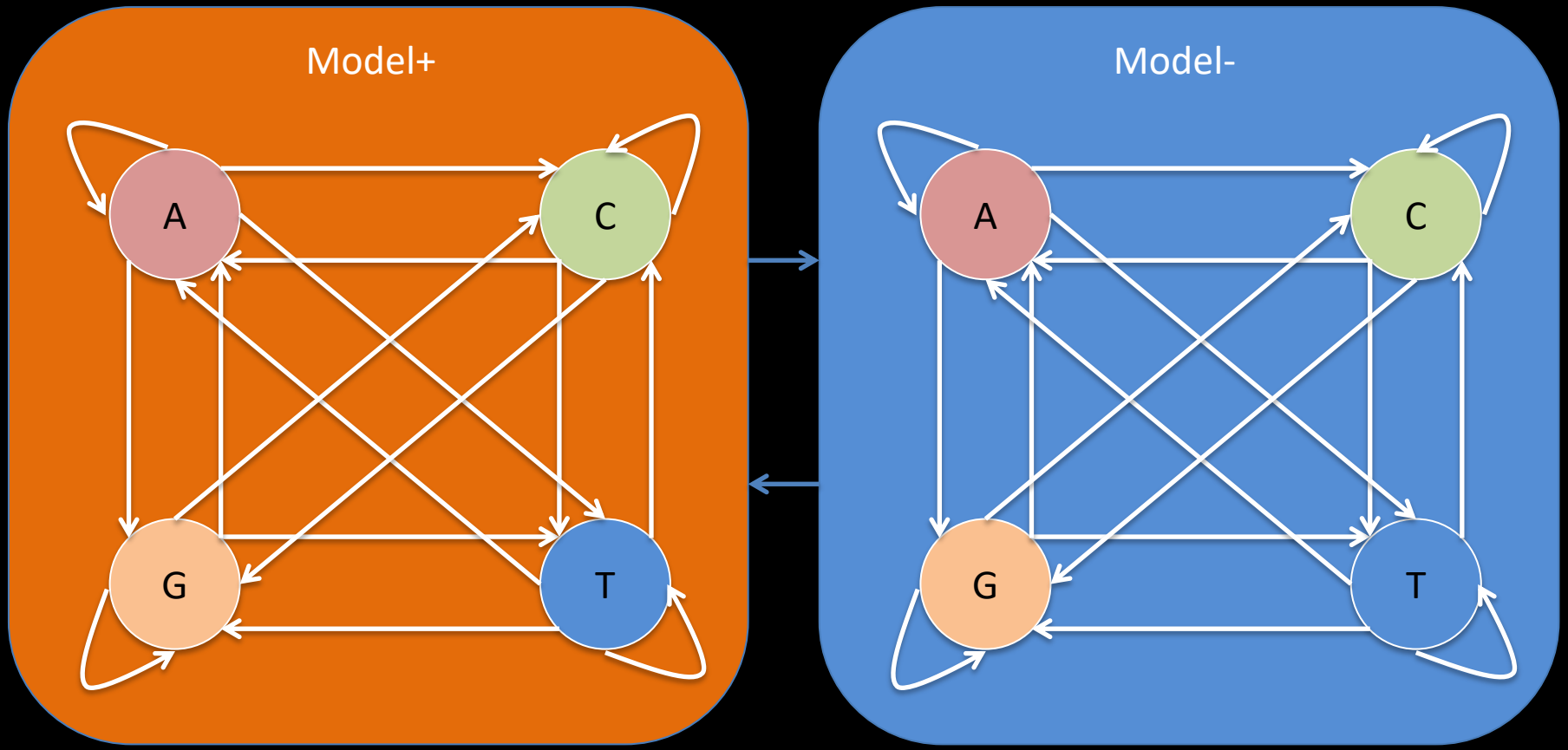
- A Hidden Markov Model (HMM) has states and transition probabilities like a Markov Chain.
- Additionally, each state has emission probabilities.
- The clue: We observe emissions, but do not know the states that they come from.
  - HMMs can compute the underlying states.

Baum, L. E.; Petrie, T. (1966). "Statistical Inference for Probabilistic Functions of Finite State Markov Chains". The Annals of Mathematical Statistics

# HMMs and CpGs

- The emissions are nucleotides.
- The states encode sequence context (nucleotides) *as well as* CpG island state (in island/outside island).
- The transition probabilities within the two new states are the same as in our Markov Chains before (describing the di-nucleotide probabilities).
- What are the transition probabilities between the two states?

# HMMs and CpGs




Note: the transitions go between every state in *CpG Island (Model+)* to every state in *Not CpG Island (Model-).* Only two arrows drawn for simplicity. The emission probabilities in each of the 8 states is 100% for whatever nucleotide is the label (this is not generally the case in a HMM)

# HMM Transition Probabilities

- To accurately specify a CpG island HMM, we need more training data:
  - How does a CpG Island look like? Model+ vs Model-
  - How many CpG islands are expected in our sequence? Probability to switch from Model+ to Model-

| # | A+ | C+ | G+ | T+ | A- | C- | G- | T- |
|---|---|---|---|---|---|---|---|---|
| A+ | 0.1762237 | 0.2682517 | 0.4170629 | 0.1174825 | 0.0035964 | 0.0054745 | 0.0085104 | 0.0023976 |
| C+ | 0.1672435 | 0.3599201 | **0.267984** | 0.1838722 | 0.0034131 | 0.0073453 | 0.005469 | 0.0037524 |
| G+ | 0.1576223 | 0.3318881 | 0.3671328 | 0.1223776 | 0.0032167 | 0.0067732 | 0.0074915 | 0.0024975 |
| T+ | 0.0773426 | 0.3475514 | 0.375944 | 0.1781818 | 0.0015784 | 0.0070929 | 0.0076723 | 0.0036363 |
| A- | 0.0002997 | 0.0002047 | 0.0002837 | 0.0002097 | 0.2994005 | 0.2045904 | 0.2844305 | 0.2095804 |
| C- | 0.0003216 | 0.0002977 | 0.0000769 | 0.0003016 | 0.3213566 | 0.2974045 | **0.0778441** | 0.3013966 |
| G- | 0.0001768 | 0.0002387 | 0.0002917 | 0.0002917 | 0.1766463 | 0.2385224 | 0.2914165 | 0.2914155 |
| T- | 0.0002477 | 0.0002457 | 0.0002977 | 0.0002077 | 0.2475044 | 0.2455084 | 0.2974035 | 0.2075844 |

# CpG Island HMM: Island Prediction

- Given such a HMM, we can now answer the question "where are the CpG islands?":
  - The sequence CGCG has several possible paths through the HMM:
  - C+G+C+G+, C-G-C-G-, C+G-C+G-, C+G+,C-,G-, etc.
  - Each has a probability as before. (Note that in a general HMM the transition probabilities are multiplied with the emission probabilities, but here they are 1).
  - Which path is the most likely?
  - The *hidden* states (and more specifically the +/- component) on *that* path tell us the location of CpG islands.

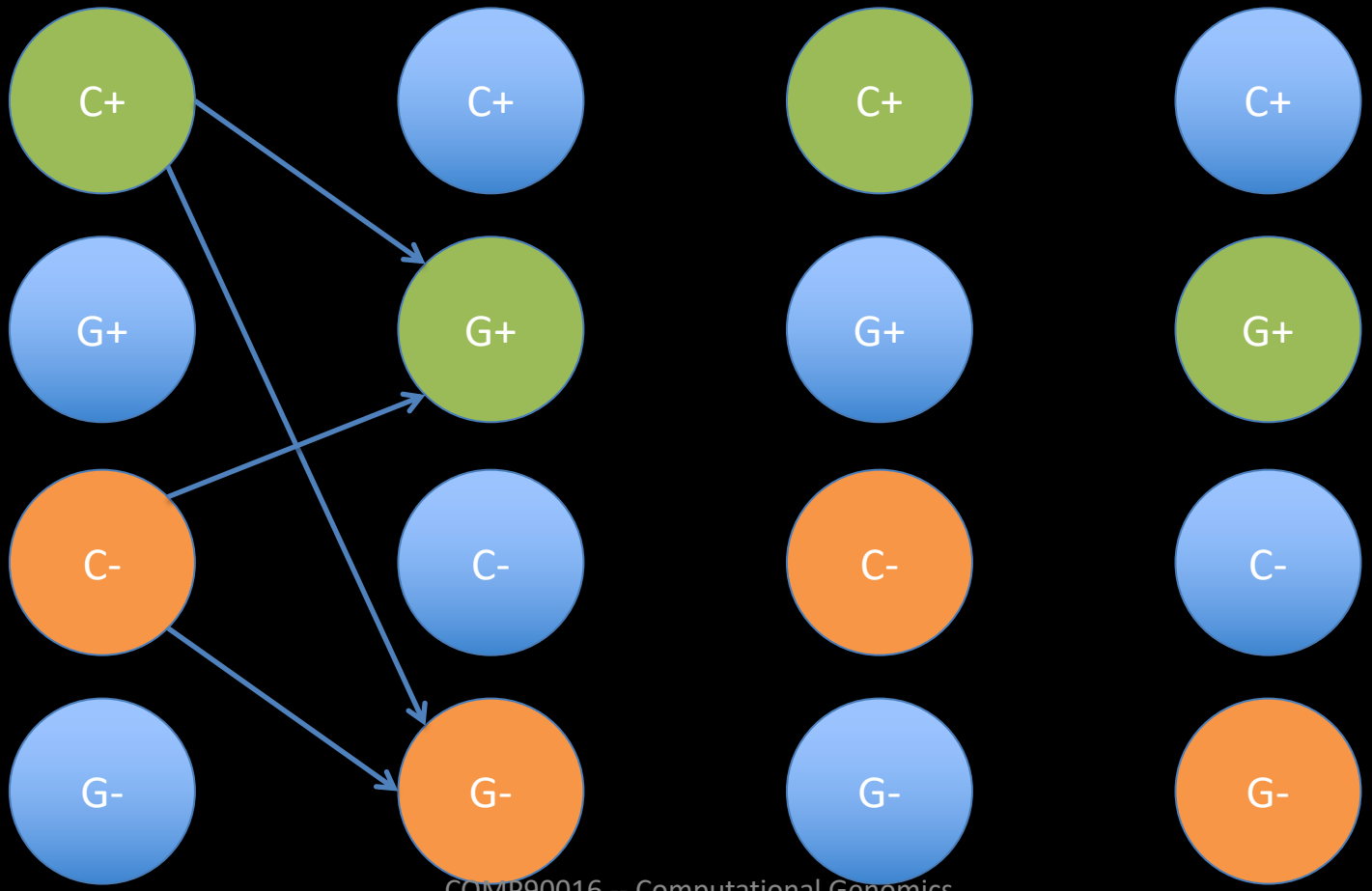
# Computational Complexity of Path Probability

- To compute a single path:
  - O(n) – n the length of the path (sequence).
- How many different paths are there to generate the sequence?
  - In this particular case: $2^n$. Each symbol comes from either a + state or a - state.
  - > overall complexity of $O(n\ 2^n)$ ☹
  - In general even worse! EVERY state could emit EVERY symbol!
- A seemingly intractable exponential problem…
- Sounds familiar?

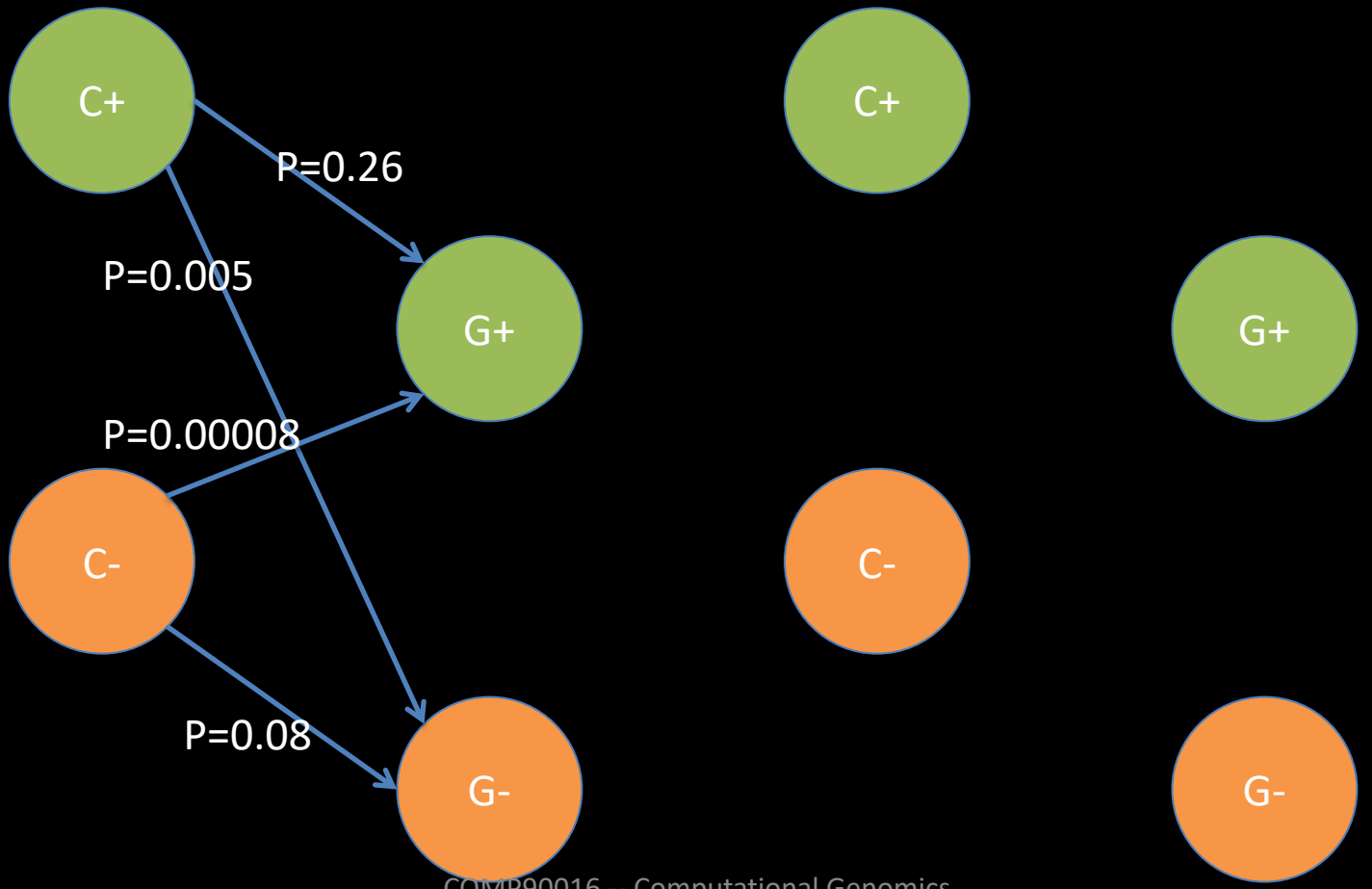
# Viterbi Algorithm

- Dynamic programming to the rescue once again!
- We exploit the fact that, since we are only interested in the best path, not all paths, we can ignore many sub-optimal solutions.
- As with alignment we are going to exploit the notion of
  - subsequences (in this case prefixes like in Needleman-Wunsch)
  - And optimal sub-solutions ("you can't find a cheaper sub-path than this, so ignore all the alternatives.")

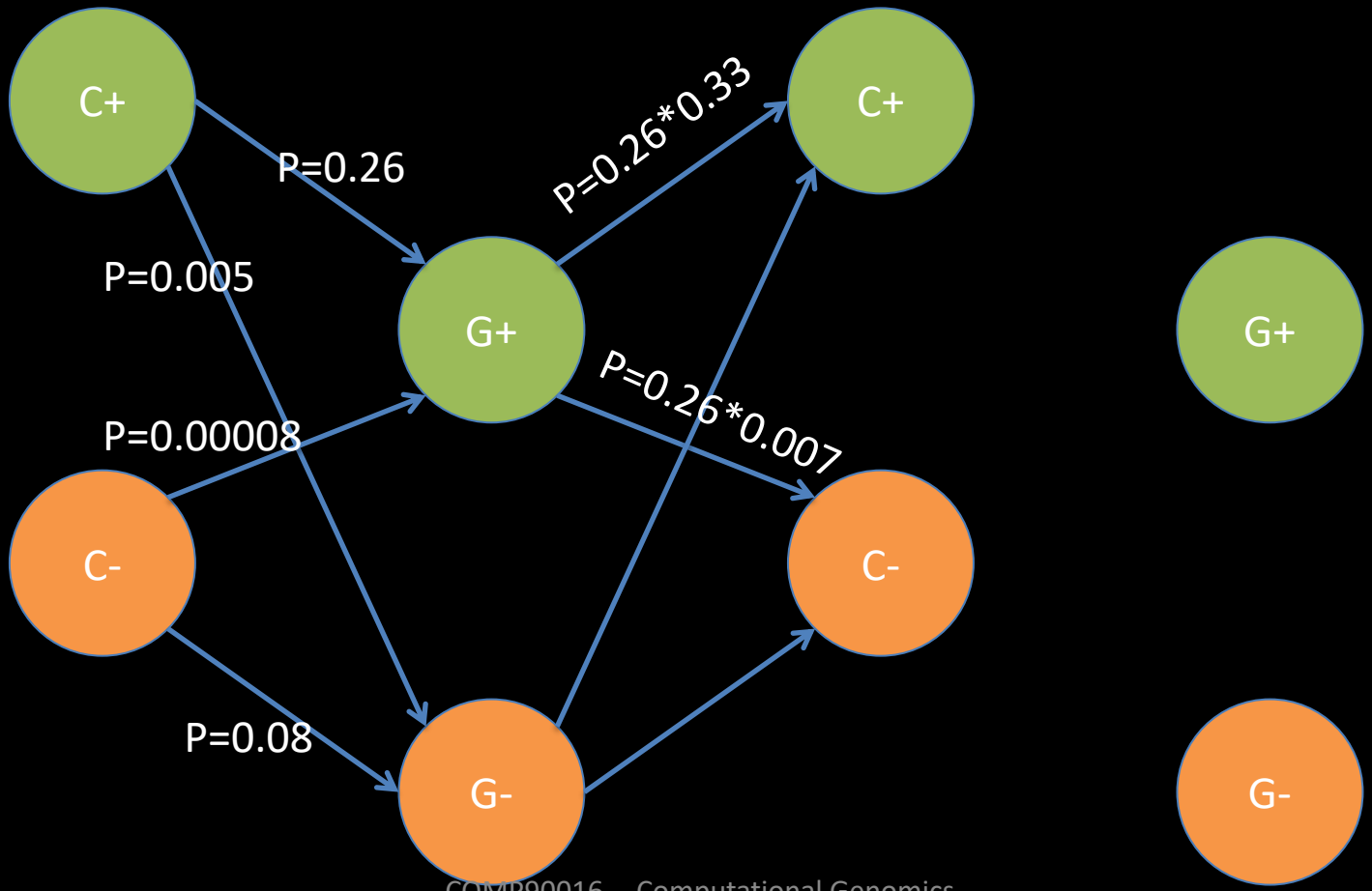
# Consider CGCG again



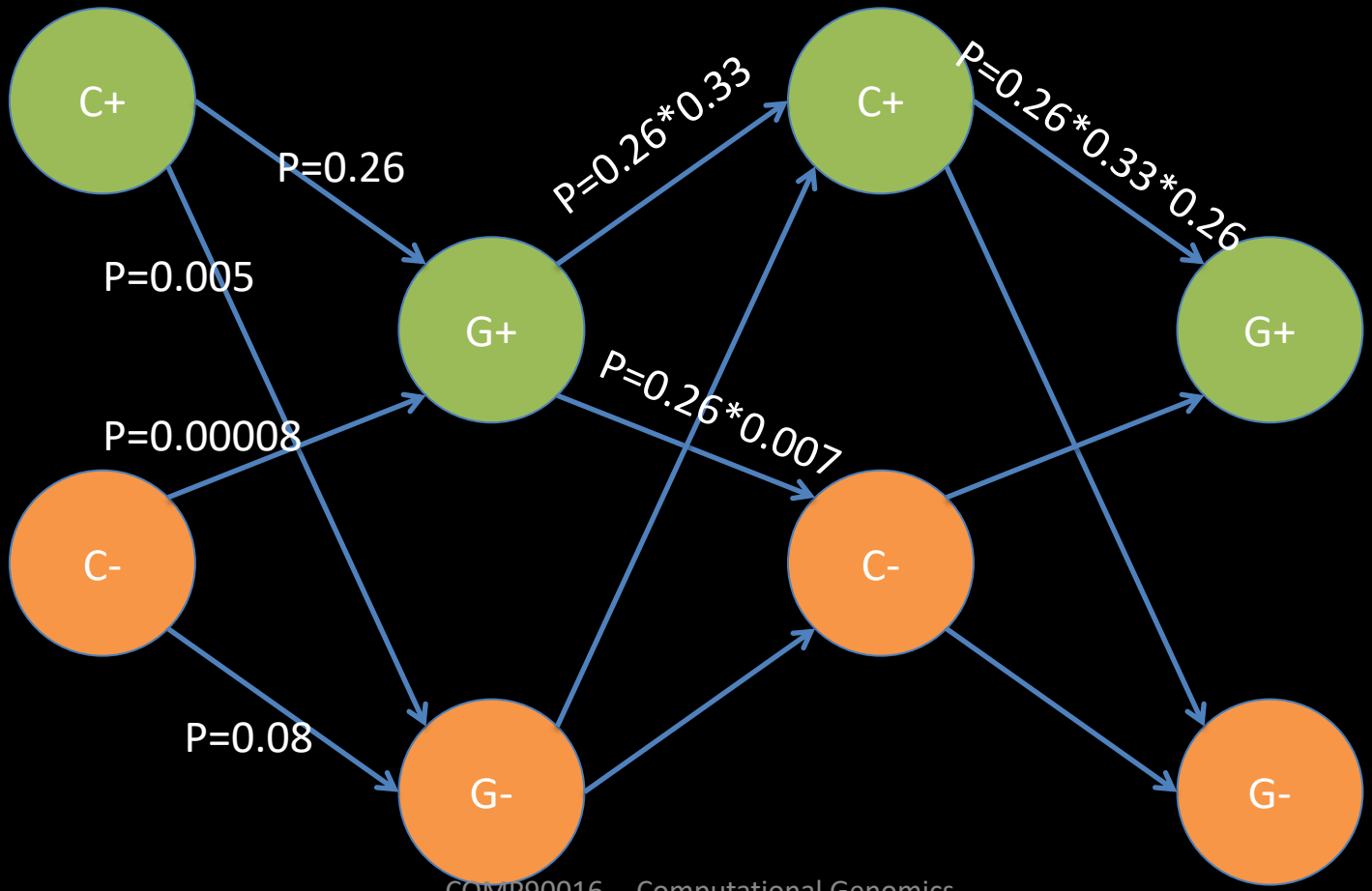
# Consider CGCG again



- We now know that the best path to get to G+ in the second position has probability 0.26. Therefore, we do not have to explore any path starting in C- AND going through G+ in the second position , because it will ALWAYS be worse than the C+G+ option!
- However, paths from C- to G- have to be explored still, since they could end up better in the long run.



- In the end, we only have calculate 12 probabilities to establish that C+G+C+G+ is the most likely path (and that the entire sequence is likely to be a CG island according to the hidden states).



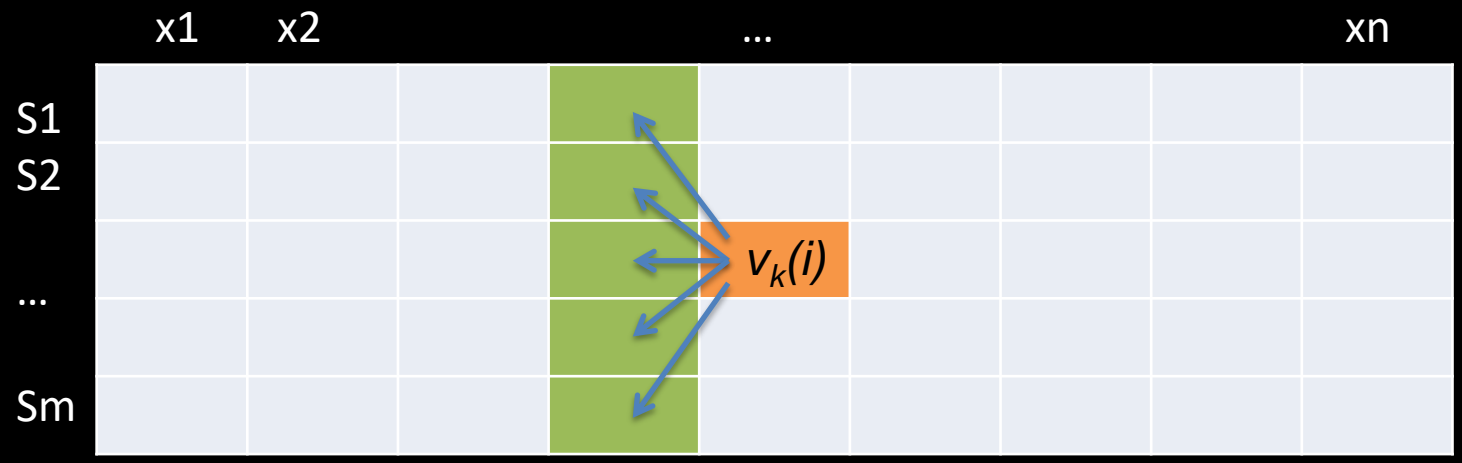
# Viterbi DPA

- Given:
  - $v_k(i)$ = most probable path ending in state $k$ with observation (emission) $x_i$.
- Then:
  - Probability of being in state $l$ after observing symbol $x_{i+1}$ is
  - $v_l(i+1) = e_l(x_{i+1}) * \max_k(v_k(i) a_{kl})$

# Viterbi DPA: Recurrences

- Initialization (sequences start in begin state *0*):
  - $v_0(0)=1$; $v_k(0)=0$, k>0
  - May assume flat base distribution: $a_{0k}$ = 0.25 for all bases (k=1-4).
- Recursion *(i=1…L)*:
  - $v_l(i) = e_l(x_i)*max_k(v_k(i-1)a_{kl})$
  - $ptr_i(l) = argmax_k(v_k(i-1)a_{kl})$
- Termination (models explicit termination state *t*):
  - $P(x,\pi^*) = max_k(v_k(L)a_{kt})$
- Traceback *(i=L…1):*
  - $\pi_{i-1}^* = ptr_i(\pi_i^*)$

# Viterbi Algorithm Overview




To compute one cell Viterbi looks at all cells from the last iteration
Viterbi is analogous to aligning a sequence to a set of states – sort of.

# Viterbi DPA: Technical detail

- Probabilities are small.
- Calculate $log(v_l(i))$ to prevent underflow.
- Overall complexity:
  - For each element of the sequence, we have to calculate the $v_k$ for each state k.
  - To calculate one $v_k$ we need to compute and compare |K| probabilities and choose the max.
  - $O(n\ m^2)$ if n the length of the sequence and m the number of states.

# Using R to Implement HMM

- The following code runs our HMM on any input (Excel spreadsheet on the LMS):

```
library(HMM)
transitions <- read.xlsx('Talk - Lecture - 6 HMMs.xlsx',1)
emissions <- read.xlsx('Talk - Lecture - 6 HMMs.xlsx',2)
hmm=initHMM(States = transitions[,1], Symbols=c("A","C","G","T"),
   transProbs = as.matrix(transitions[,2:9]), emissionProbs = t(as.matrix(emissions)))
obs=("ATATCTTATCGCAGCGCGCTGCGAGCCGATTAGTAT")
viterbi(hmm,strsplit(obs, "")[[1]])
 [1] A+ T+ A+ T+ C+ T+ T+ A+ T+ C+ G+ C+ A+ G+ C+
     G+ C+ G+ C+ T+ G+ C+ G+ A+ G+ C+ C+ G- A- T-
     T- A- G- T- A- T-
                 ATATCTTATCGCAGCGCGCTGCGAGCCGATTAGTAT
```

I suggest trying this on a range of data present on http://www.r-bloggers.com/discrimination-between-cpg-islands-and-random-sequences-using-markov-chains/

# A Larger Example

obs=CCGTCCTGTGGACCACTCGAAGCCAGGTGCCTCGCGATCCGTAACCATCTCGAGTTCCAGGGGCCCCGGCTGGCCGCCTATCCTCCTCCACTCAGCCGGTGCCCCGCGTCCAGCCGAGTTGCTGTCAACTCCAGTCGGGCGGCCAGCAGGGCTCC
CCGGACGAGCCGCCCGCCACGCCGCCCACCTCTCGCGGCCGGCACCCCGCGCCGCGGCCTCCTCCGGGCTCACCTAGGGCCACCTCGCACGCTTTGCGCAGCTGGGAGTGATGCGCCTTCTTCACTTCCTTGTCGGCCAGAATCTTCTCCAGGGCCCG
GGTCAGGAACATGTTCTTCGTCTTCTTGCCCTCATACATGGACGCGGAGGAGGCGGCGGCGGCGGTCCCGGCGGCTGGAGGGAAGGCGGAGGAGGGGGACGGGGACCGGGATGAGGACGGGGAGGGAGAAAGGCGAGGAGGTGGCGGT
GGGGGCGAGGAGGCGTGGAGGGCAGCGGCGGCACCAGGAAGCGGCCGGCGACGCGGAGCGGCCGCGGGGCCGCCAAGGGCGTCGAAAGCCTCGCCACGCTCGAGGGGCCGCCGCCCGTCCGGCCTCGGCGCCGTCGGCC
CGGGGGTCGCTTCCCGGCCGCCCGCCTCACTCGTCCCTACGGCCCGGCGGCGGCGGCACTCGGCCCGGGCGGCCGTCTTCCCGGGGCCGAGGGCCTAGGTGGCGGCGCCTCACACTGCCAGCAGGCGCGAGGCTGCCCTAGCTACCGTGGGGACCA
GCATCCGCCGCAGCCGCCGACTGGCCTCCATCACCGCCGACCTGCCCCGGACGCCATGTTGAGAGGAAACGACGCGTCACGCAGCGGCCGAACGGCTGCGGAGAGGAAGCGGCGGCGGCGGCGGTGCCAGCGCCAGCCCAGAGCTGCCGCGGCG
CCGAGAGGCAAGGGGCGGAGGGCGGCGCGGGAGGAGCGGCCGGAGCTGCAACTCTACGGCGGCCCGGCGGGGACTCGCCGCCTCCCGCGTTCGCCTGCAGGGCAGCGTGGGGCGTGGCGGGCGTCCTGCTGCGCGGCACGGCGTGGCGCGGG
GCTTCTGCTCTTGCGGGCTACTGGGATGCGTGATCGGAGCACAGGGGCGCTCCCAAGGCCGGGACGACAGGACCCCTGCCTACCGTGCACGGGCTTCCCCAACTTTCTGGACAGCCGGTCGCCGGGGAGCAAAGTTCCTCGCGGTCCTAGGGCCACA
GGAACACCCGCGTCGCTCCCTCAGGCGCACCTGGTTTCCGCCTGCCACACG

viterbi(hmm,strsplit(obs, )[[1]])

```
G+ G+ A+ G+ G+ G+ G+ A+ G+ A+ A+ A+ G+ G+ A+ G+ G+ G+ A+ A+ A+ A+ G+ G+ A+ A+ A+ G+ A+ G+ G+ G+ G+ A+ G+ A+ A+ G+ G+ G+ A+ G+ A+
A+ G+ G+ A+ G+ G+ G+ G+ A+ A+ A+ G+ G+ G+ G+ A+ A+ A+ A+ A+ G+ G+ G+ G+ A+ A+ G+ G+ A+ A+ G+ G+ A+ G+ G+ G+ A+ G+ G+ G+ G+ G+ G+
G+ G+ G+ A+ G+ G+ G+ A+ A+ G+ G+ A+ A+ G+ A+ G+ G+ G+ G+ A+ G+ A+ G+ G+ G+ A+ A+ G+ G+ A+ A+ A+ G+ G+ A+ G+ G+ A+ G+ G+ A+ A+ G+
G+ G+ G+ A+ A+ G+ G+ A+ A+ A+ G+ A+ A+ G+ G+ A+ A+ G+ A+ A+ A+ A+ G+ G+ G+ G+ G+ G+ A+ A+ A+ G+ A+ A+ A+ G+ G+ A+ G+ A+ G+ G+
G+ A+ G+ A+ G+ G+ A+ G+ G+ G+ A+ G+ G+ G+ G+ G+ G+ A+ G+ A+ A+ G+ G+ A+ A+ G+ A+ G+ G+ G+ G+ A+ G+ A+ G+ G+ A+ G+ A+ A+ G+ G+ G+
G+ G+ G+ G+ G+ A+ A+ A+ G+ G+ G+ A+ G+ G+ G+ A+ A+ A+ A+ G+ G+ A+ G+ G+ G+ G+ A+ G+ A+ G+ A+ G+ G+ G+ G+ A+ G+ A+ G+ A+ A+ G+ G+
A+ A+ A+ A+ A+ G+ A+ A+ G+ A+ G+ A+ G+ G+ G+ G+ G+ G+ G+ G+ A+ G+ G+ G+ G+ G+ G+ G+ A+ G+ G+ A+ A+ G+ G+ A+ A+ A+ A+ G+ G+ G+ G+
G+ G+ G+ G+ A+ A+ A+ A+ G+ G+ G+ A+ A+ A+ G+ G+ A+ A+ A+ A+ A+ G+ A+ G+ A+ G+ G+ G+ G+ G+ G+ A+ G+ G+ G+ G+ G+ G+ G+ A+ G+ G+ G+
G+ G+ C- T- C- T- C- T- C- C- C- T- C- T- C- C- C- C- C- C- C- T- C- C- T- C- C- T- C- C- T- C- C- T- C- C- T- T- T- T- C- C-
T- C- C- T- T- C- C- C- C- C- C- C- C- C- T- C- C- C- C- C- C- C- C- C- C- C- C- T- C- C- C- C- C- T- T- C- C- C- C- T- C- C-
C- C- C- T- C- C- C- C- C- C- C- C- C- C- C- C- C- C- C- T- C- C- C- C- C- C- C- T- C- C- T- C- C- T- C- C- C- C- C- T- C- C- C-
C- C- C- C- T- C- T- C- C- C- C- C- C- T- C- C- T- C- C- T- C- C- T- C- T- T- C- C- C- C- C- C- T- C- C- T- T- C- C- T- C- C- T-
C- T- C- C- C- C- T- C- C- T- T- C- T- C- C- C- C- T- T- C- T- T- C- C- C- C- C- T- C- T- T- C- C- C- C- A+ G+ G+ G+ G+ A+ G+ G+
A+ G+ A+ G+ G+ G+ A+ A+ A+ A+ A+ A+ G+ G+ A+ G+ G+ A+ G+ G+ G+ A+ A+ A+ A+ A+ G+ G+ A+ G+ A+ G+ G+ G+ A+ G+ G+ G+ A+ A+ G+ G+ G+
G+ A+ A+ G+ A+ G+ G+ A+ G+ G+ A+ A+ G+ G+ G+ A+ A+ A+ A+ A+ G+ G+ A+ G+ G+ G+ G+ G+ G+ A+ A+ G+ G+ A+ G+ G+ G+ G+ A+ G+ G+ G+ G+ A+
G+ G+ G+ A+ G+ G+ G+ G+ G+ A+ G+ A+ A+ G+ G+ G+ A+ A+ G+ A+ A+ G+ A+ A+ G+ A+ A+ G+ A+ A+ G+ A+ G+ G+ G+ A+ A+ G+ G+ G+ A+ A+ A+ G+ A+ A+
G+ G+ A+ G+ G+ G+ G+ G+ G+ G+ A+ A+ A+ A+ G+ G+ A+ A+ A+ A+ A+ G+ G+ G+ A+ A+ A+ G+ A+ A+ G+ A+ A+ G+ A+ G+ G+ G+ G+ A+ G+ A+ G+
G+ A+ G+ G+ A+ A+ G+ A+ A+ A+ G+ A+ G+ A+ A+ A+ A+ G+ G+ A+ G+ G+ G+ G+ A+ A+ G+ G+ A+ G+ G+ A+ G+ A+ A+ A+ A+ A+ G+ G+ A+ A+ G+
A+ G+ G+ G+ A+ G+ G+ A+ G+ A+ A+ G+ G+ A+ G+ G+ A+ A+ G+ G+ A+ A+ G+ G+ G+ G+ G+ A+ G+ G+ A+ G+ G+ A+ G+ G+ A+ A+ G+ G+ G+ A+ G+
G+ G+ G+ C- C- C- T- C- T- T- C- T- C- T- T- C- C- C- C- C- C- C- C- C- T- C- C- T- C- T- C- T- T- C- T- C- T- C- C- T- C- C- T-
T- C- C- C- T- C- C- T- T- C- T- C- C- C- C- C- C- C- C- C- C- C- T- C- C- T- C- C- T- C- C- T- C- C- T- C- C- T- C- T- T- C- C- T-
C- T- T- C- C- T- T- T- C- C- C- C- T- T- C- T- T- C- T- C- C- T- C- T- T- C- C- C- C- C- C- T- C- C- C- C- C- C- T- C- C- C- C-
C- C- T- C- C- T- C- T- C- C- C- C- C- C- C- C- C- T- C- C- T- T- C- C- C- C- T- T- C- T- C- C- T- T- T- T- C- T- C- C- T- C- C- T-
T- T- C- C- T- C- C- C- C- C- T- T- T- C- T- T- C- T- T- T- T- T- T- C- T- C- T- T- T- C- T- T- T- C- T- C- C- C- C- T- C- C- T-
C- T- C- C- C- C- T- C- T- C- C- T- C- C- C- T- C- T- T- T- T- C- T- T- C- T- C- T- C- C- T- C- T- C- C- T- C- T- C- C- T- C- T-
C- C- C- C- T- T- T- T- T- C- T- T- T- T- T- C- T- C- C- C- T- T- C- T- T- C- C- C- C- T- C- T- C- T- C- C- T- T- C- C- C- C- T-
C- T- C- C- C- C- C- T- C- T- T- T- T- T- C- C- C- C- T- T- C- C- C- C- T- C- C- T- C- C- C- A+ G+ G+ G+ G+ G+ A+ G+ G+ G+ A+ G+
G+ A+ G+ A+ G+ A+ G+ A+ A+ A+ G+ G+ G+ G+ G+ G+ G+ A+ A+ G+ G+ G+ G+ G+ A+ A+ A+ G+ A+ A+ G+ G+ A+ A+ G+ G+ A+ G+ G+ A+ A+ A+
A+ A+ A+ G+ A+ A+ A+ A+ G+ G+ G+ G+ G+ G+ A+ G+ A+ A+ G+ G+ G+ G+ A+ A+ A+ A+ G+ G+ A+ G+ A+ A+ A+ A+ A+ G+ A+ G+ G+ G+ A+ G+ A+
G+ G+ A+ G+ G+ G+ G+ G+ G+ G+ A+ A+ A+ G+ A+ G+ A+ G+ G+ G+ A+ A+ G+ G+ G+ G+ G+ A+ G+ G+ G+ A+ G+ G+ A+ G+ A+ G+ A+
```

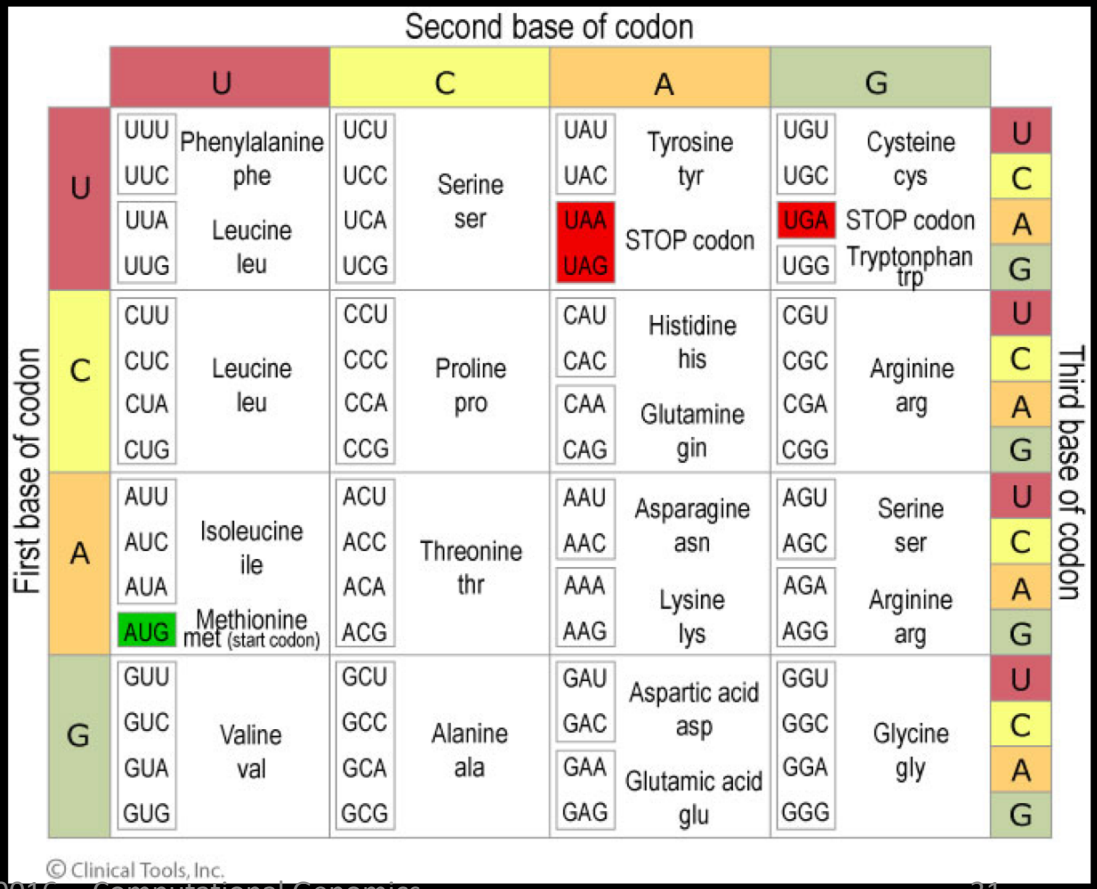
# HMMs Summary

- HMMs allow us to investigate the input as output – which sounds confusing – from a model.
- If we know of different possible states in our data, such as:
  - Cs followed by Gs are quite likely in CpG islands.
  - But not so much outside of CpG islands.
- Then we can model the probabilities of CpG islands and non-CpG islands producing (emitting) certain sequences.
- Further, if we know how many CpG islands are expected in out input and how long they should be, we can allow the model to transition between the hidden states to tell us where in the input the CpG islands are.

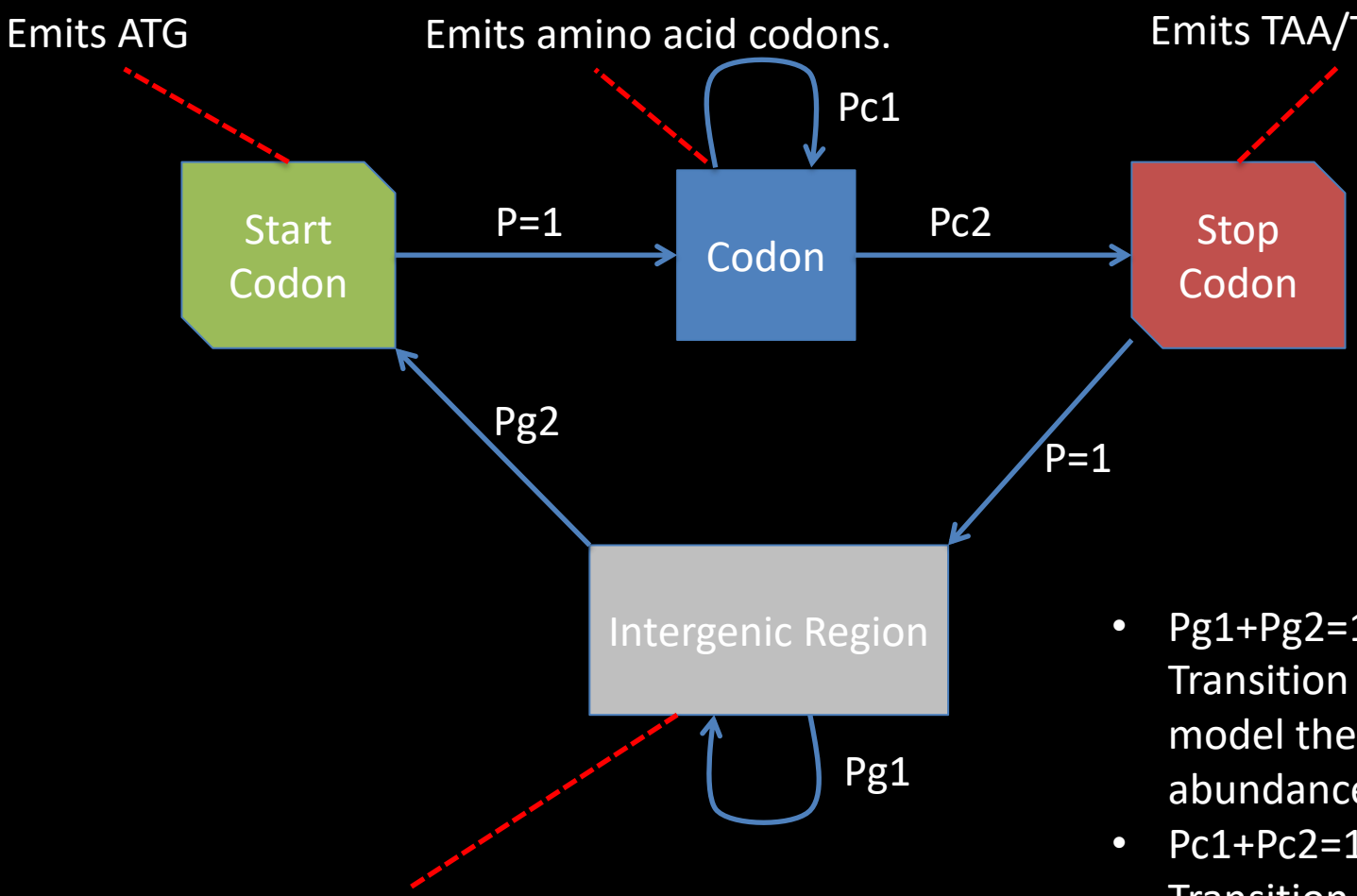
# Gene Finding

- What features set a gene apart from non-genic sequence?

Genes have a start codon.
Genes have a stop codon.
-> Open reading frames
Some codons are more common than others.
-> some tri-nucleotides unlikely to occur within a gene.

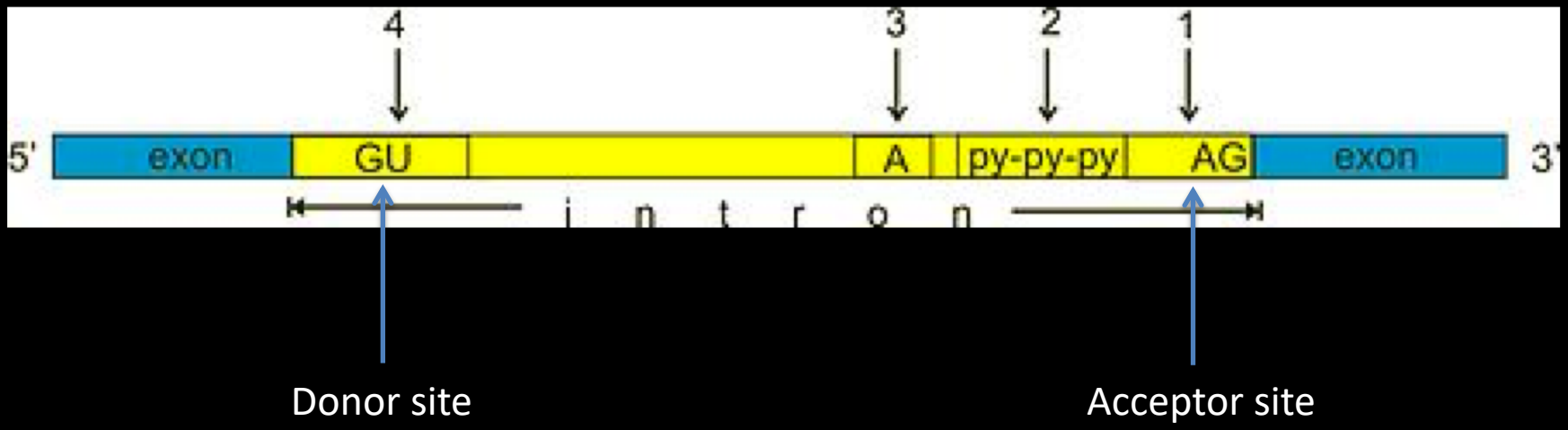


# A Simple HMM for Gene Finding




- Pg1+Pg2=1
  Transition probabilities model the general abundance of genes
- Pc1+Pc2=1
  Transition probabilities model average length of genes

# More Gene Details



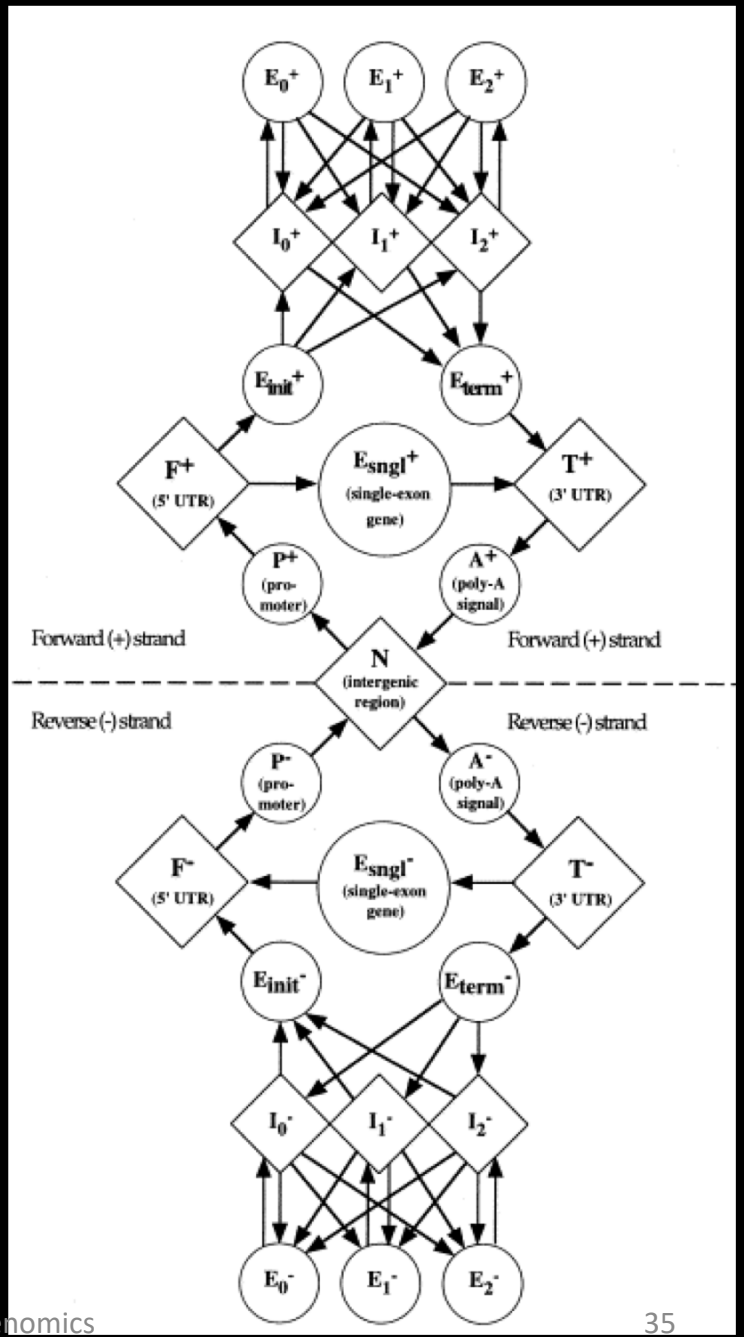

Gene splicing is a very complex mechanism.
For our purposes it is sufficient to understand that sequence in the RNA tells the cell machinery when an exon ends and an intron starts and vice versa

# Additional Features to Identify Genes

- Promotor regions
- Poly-A tail
- Knowledge about:
  - Nucleotide contents of genome
  - Gene abundance in genome

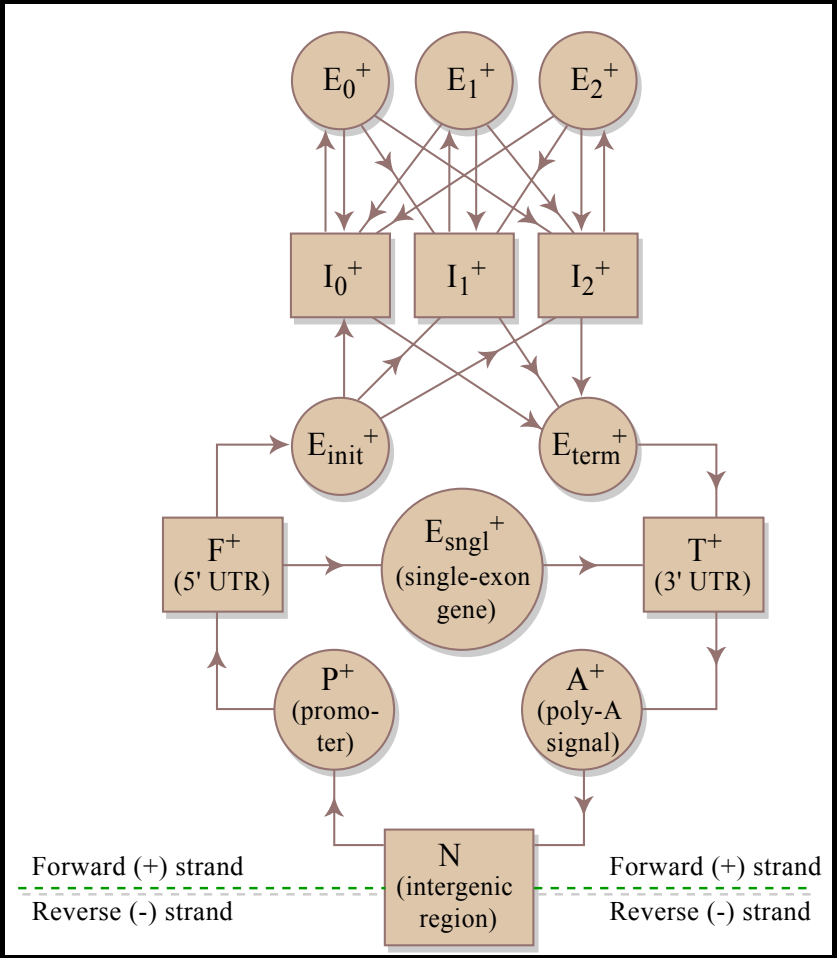
# GENSCAN HMM

*Prediction of complete gene structures in human genomic DNA1*
Chris Burge, Samuel Karlin
Journal of Molecular Biology
doi:10.1006/jmbi.1997.0951



# GENSCAN Details

- N: intergenic region
- P: promotor
- F: 5’ UTR
- $E_{sngl}$: single exon gene
- $E_{init}$: initial exon (including start codon and first donor splice site)
- $E_i$: phase i internal exon (including acceptor splice site and donor splice site)
- $E_{term}$: terminal exon (including final acceptor splice site and stop codon
- $I_j$: phase j intron (memory state of where the exon was left; first, second or last base of a codon)
- T: 3’ UTR
- A: poly-A tail of gene



# The Biggest Surprise From the HGP

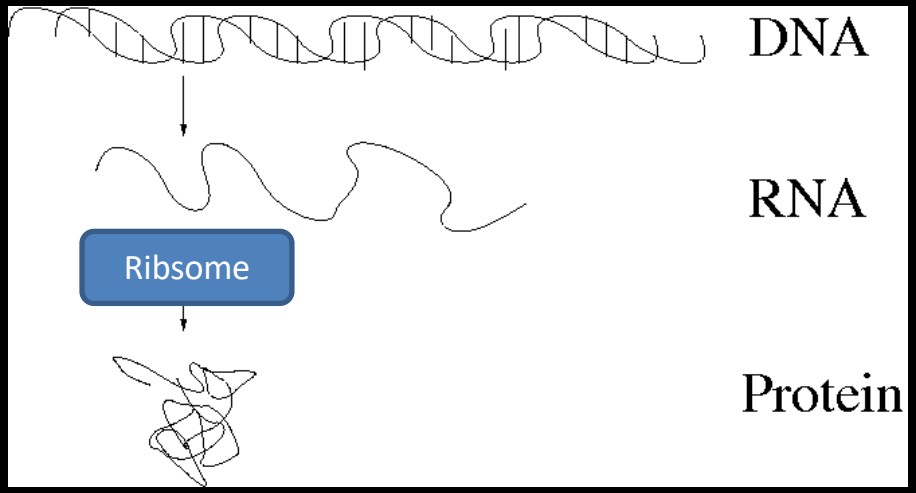
- First estimate: 30,000 genes
- Only 30,000 genes?
- 24,000 genes? 22,000?

| | |
|---|---|
| Water flea | 31000 |
| Mouse | 23000 |
| Human | 21000 |
| Worm | 18000 |
| Fruit fly | 13000 |
| Yeast | 6000 |
| E. coli | 4000 |

- Numbers still not set in stone

# DNA Information Flow and Ribosomes

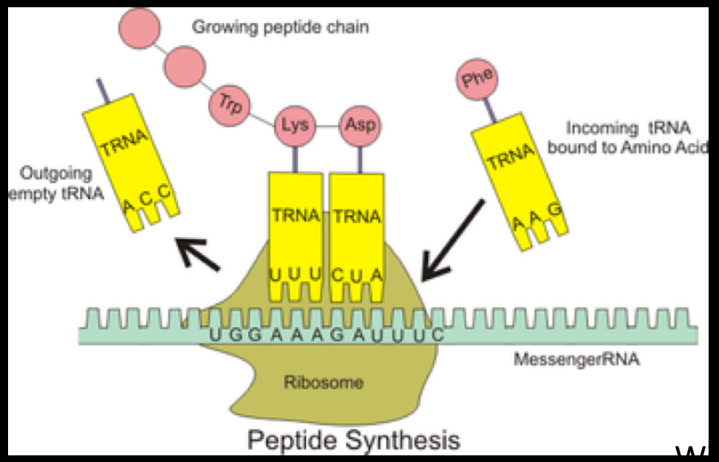
- DNA is the blueprint, proteins do the work.




- DNA→ RNA: transcription 1-to-1
- RNA → protein: translation 3-to-1

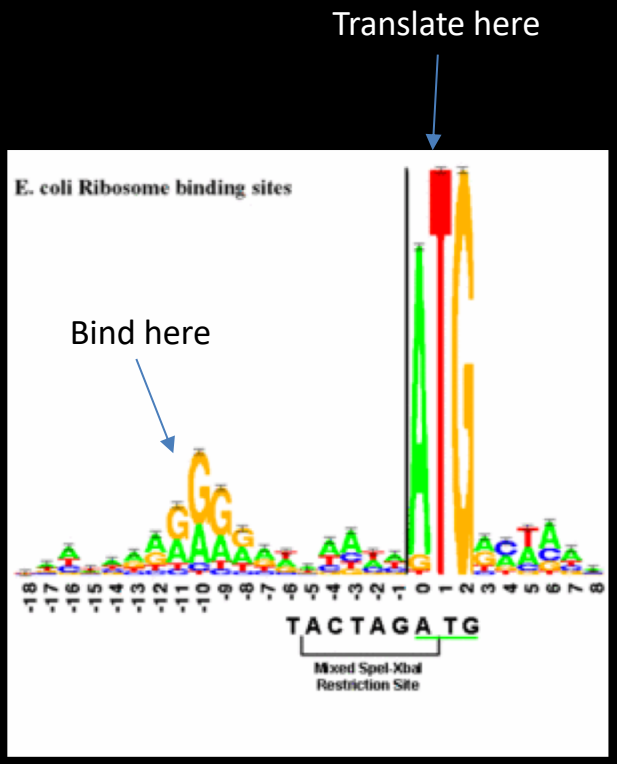
# Ribosomes at Work

- Ribosomes are complexes within a cell responsible for the translation process
- They are protein complexes made of multiple sub-units.
- mRNA is brought to the ribosome, bound to, and then translated.



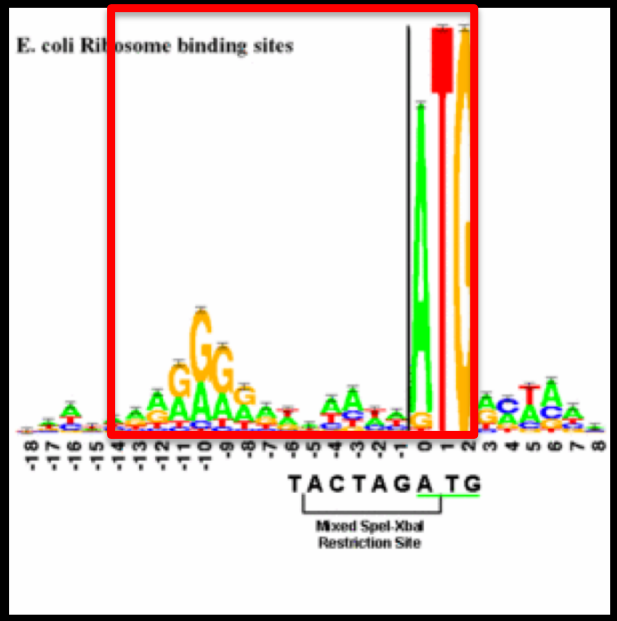
# Ribosomal Binding

- The mRNA is attached to the ribosome (recruited) by the 30S sub-unit by RNA base-pairing.
- The 30S contains single-stranded RNA of the anti-Shine-Dalgarno sequence: CCUCCU
- Therefore, the ribosome likes to bind to the Shine-Dalgarno (SD) sequence: AGGAGG.
- Genes have the SD sequence just upstream of their start codon (6-7 bases away).
- The SD sequence on the RNA affects how well the protein is translated and how well the ribosome is recruited to the RNA:
  - A more exact match leading to higher rate of translation.

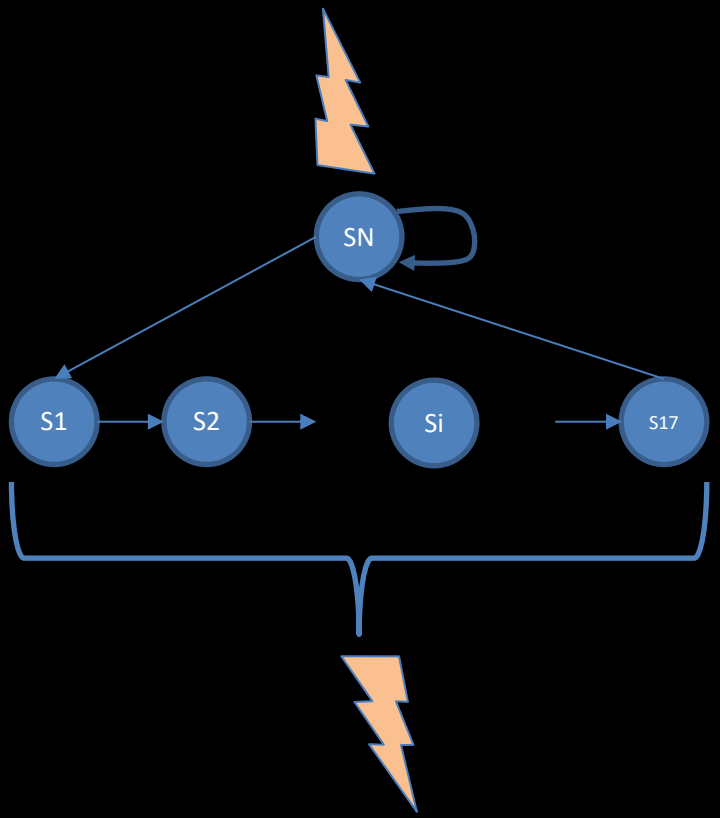



Sequence logo of ribosomal binding sites. Height of the letters indicate frequency of Nucleotides observed at position.

# HMM to Detect Ribosomal Binding



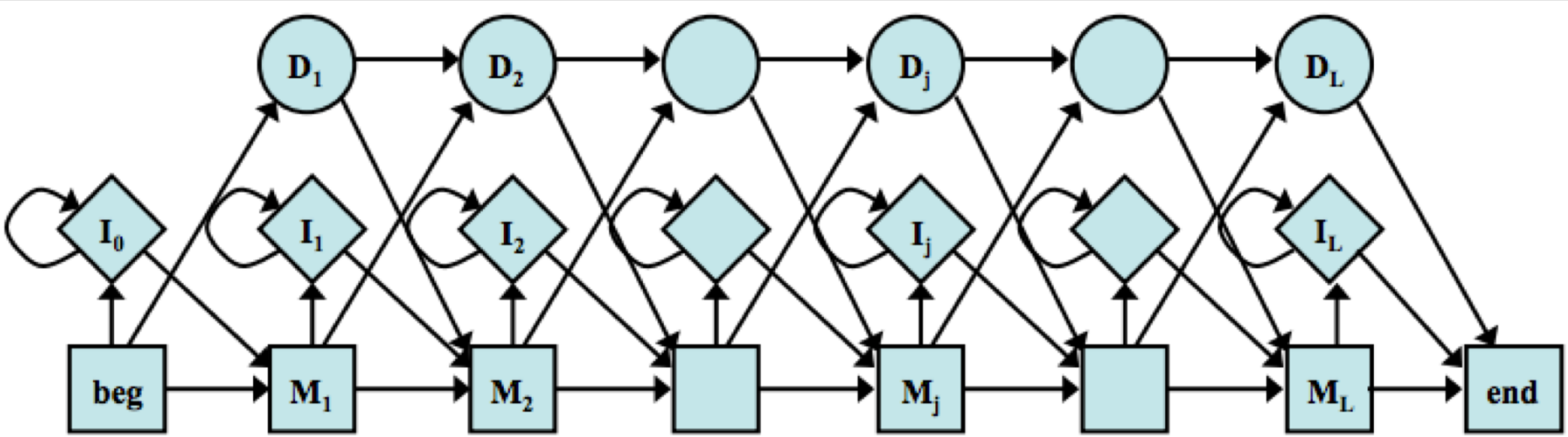

E. Coli background emissions:
25% ACGT each




E. Coli RBS binding profile:
SD sequence-gap-ATG

# Other Genomics Applications of HMMs

- Sequence alignment
  Profile HMMs
  States for matching deleting and inserting bases.




- See Richard Durbin's **Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids**

# Baum-Welch Algorithm For HMM Training

- To train a HMM effectively from testing data, the Baum-Welch algorithm, offers heuristic approximation for parameters:
  - It attempts to maximise the probability of the input data being created by the model.
  - Starting with random starting probabilities, it calculates the probability for the training data, as well as the probability for individual states to be passed through.
  - Then it updates the transition and emission probabilities to maximise for the testing data (expectation maximisation).
  - The procedure is repeated for a maximum number of iterations, or until parameter change falls below a threshold.

# Final Points on HMMs

- We looked at the Viterbi algorithm to find the highest probability path through hidden states, which generate the observations.
- There are algorithms to train HMMs (such as the Baum-Welch procedure), which determine the parameters of the model to begin with.
- HMMs are only useful when modeling the subject at hand well (training vs. real life).

# Conclusion

- HMMs are a versatile modeling technique that finds application in various genomics problems:
  - CpG islands
  - Gene finding
  - Copy number detection (next lecture)
- Its limitations are
  - Dependency on good training data
  - Scalability $O(nS^2)$ limits the number of states.