Gene prediction (finding)

Pedagogical note on algorithms [i]

- This class is practical with an emphasis on
  - Formulation of a biological problem in terms of bioinformatics approaches/tools
  - Evaluation of the best (set) application(s) / tool(s) / program(s) for any given problem
  - Deployment and execution of those tools to address the problem and do the job

- Not an algorithms course *per se*

- Useful to understand the algorithmic foundations of the various
  - Can inform choice of best applications/tools
  - Can inform parameter choice decisions
  - Can help to monitor behavior and trouble shooting of applications

Pedagogical note on algorithms [ii]

- Ongoing overview of foundational algorithms in bioinformatics

- Previously (genome assembly)
    - Sequence substrings (k-mers)
    - Graph based approaches

- Today (gene prediction)
    - Sequence substring (k-mer) indexing
    - Dynamic programming (alignment)
    - Hidden Markov Models (HMM)
    - Dynamic programming (Viterbi algorithm)

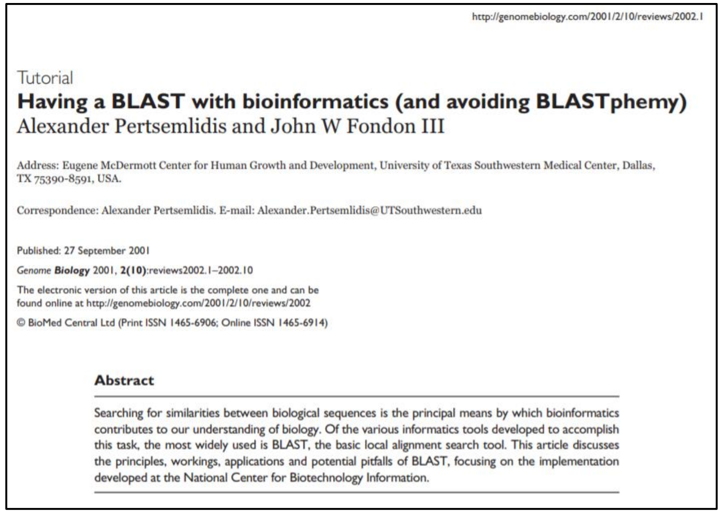
Approaches to gene prediction

- Homology-based methods
  - Find genes via comparison with sequences of know genes
  - Extrinsic information
  - Reliable for what we already know
  - Limited by what we already know (no new knowledge)
  - Can use to validate/support *ab initio*

- *Ab initio* methods
  - Find genes based on intrinsic characteristics of genome sequence
  - Prior knowledge = differences in sequence composition between protein coding and non-coding sequences
  - Not quite as robust as homology-based methods
  - Opportunity for new knowledge

Homology-based gene prediction with BLAST

- Homology-based methods
  - Find genes via comparison with sequences of know genes
  - Extrinsic information
  - Reliable for what we already know
  - Limited by what we already know (no new knowledge)



http://genomebiology.com/2001/2/10/reviews/2002.1

Tutorial

Having a BLAST with bioinformatics (and avoiding BLASTphemy)

Alexander Pertsemlidis and John W Fondon III

Address: Eugene McDermott Center for Human Growth and Development, University of Texas Southwestern Medical Center, Dallas, TX 75390-8591, USA.

Correspondence: Alexander Pertsemlidis. E-mail: Alexander.Pertsemlidis@UTSouthwestern.edu

Published: 27 September 2001

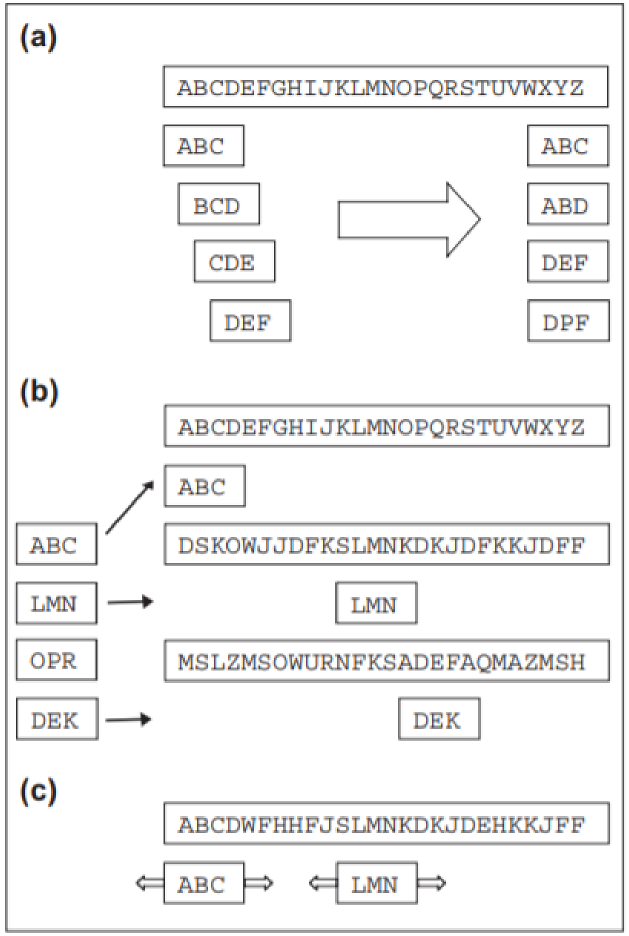
Genome ***Biology*** 2001, **2(10)**:reviews2002.1–2002.10

The electronic version of this article is the complete one and can be found online at http://genomebiology.com/2001/2/10/reviews/2002

© BioMed Central Ltd (Print ISSN 1465-6906; Online ISSN 1465-6914)

Abstract

Searching for similarities between biological sequences is the principal means by which bioinformatics contributes to our understanding of biology. Of the various informatics tools developed to accomplish this task, the most widely used is BLAST, the basic local alignment search tool. This article discusses the principles, workings, applications and potential pitfalls of BLAST, focusing on the implementation developed at the National Center for Biotechnology Information.


(a)
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABC
BCD
CDE
DEF
ABC
ABD
DEF
DPF
(b)
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABC
ABC
DSKOWJJDFKSLMNKDKJDFKKJDFF
LMN
LMN
OPR
MSLZMSOWURNFKSADEFAQMAZMSH
DEK
DEK
(c)
ABCDWFHHFJSLMNKDKJDEHKKJFF
ABC
LMN

Ab initio gene prediction

- *Ab initio* methods
  - Find genes based on intrinsic characteristics of genome sequence
  - Prior knowledge = differences in sequence composition between protein coding and non-coding sequences
  - Not quite as robust as homology based methods
  - Opportunity for new knowledge

Models and Definitions

- Markov model
  - Stochastic model of a randomly changing system
  - Future state depends only on the current state (not previous states)
  - Critical assumption that facilitates computation (tractable algorithms)

- Hidden Markov Model (HMM)
  - Markov model of a randomly changing system
  - System is made up of unobserved (hidden) states
    - Coding versus non-coding sequences
  - Hidden states ‘emit’ observed states
    - Observed sequence of DNA residues

HMMs and Machine Learning

- Machine learning algorithms are presented with ***training data*** to derive insight about unknown (hidden) parameters in the data
  - More training data generally yields more accurate parameter inferences
  - Parameters = biological knowledge

- Once an algorithm is trained, it can apply these insights to the analysis of ***test data***
  - Test data should be different from training data
  - Apply biological knowledge (parameters) with algorithm to new (test) data

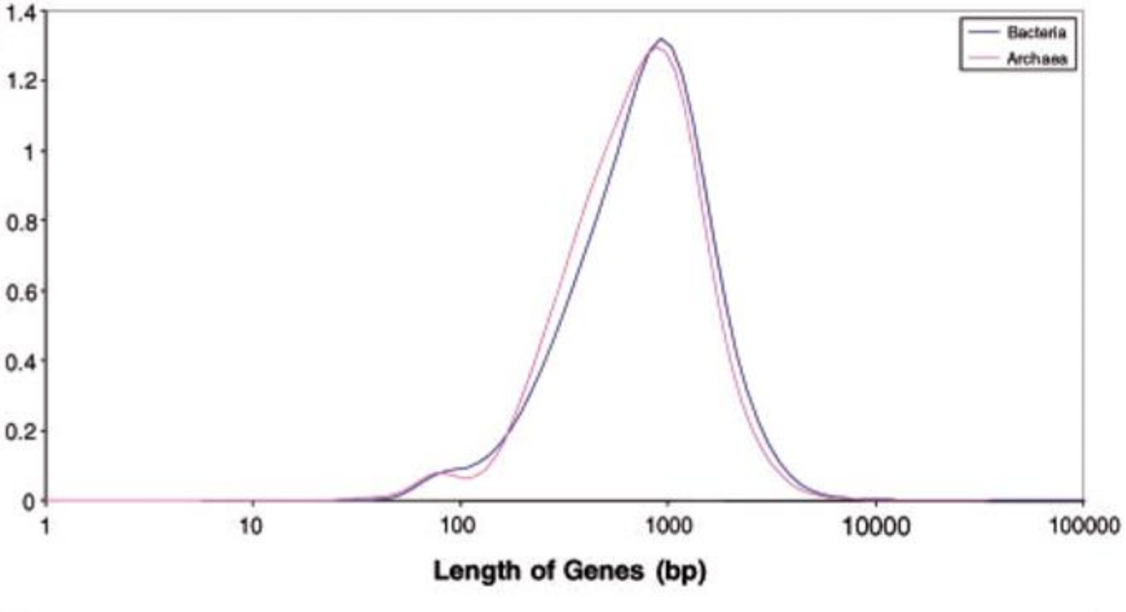
Biology of HMMs for gene prediction

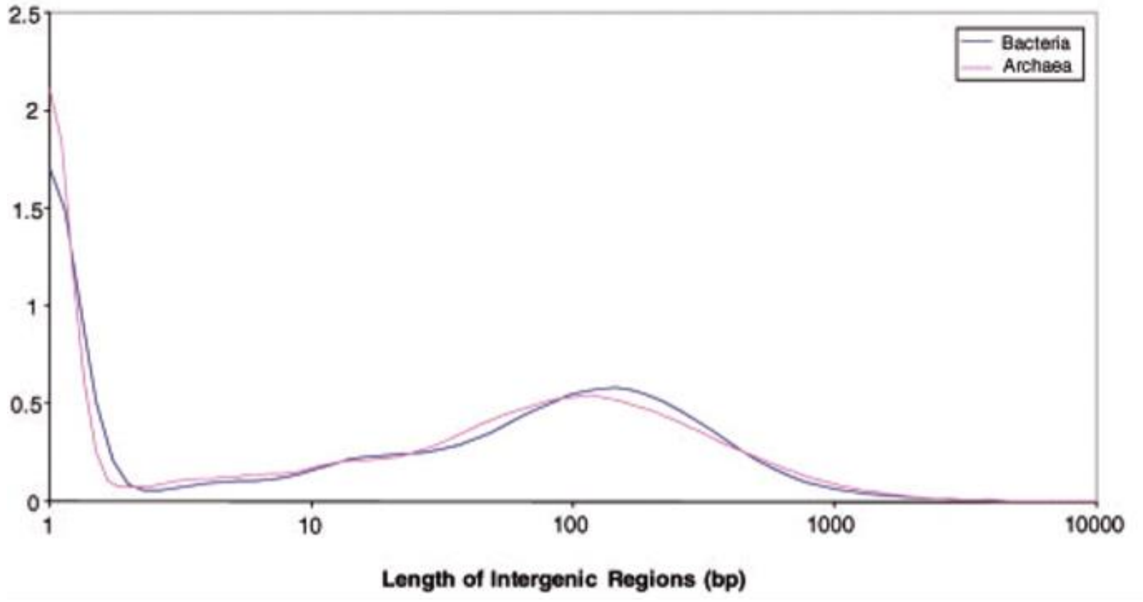
- *Ab initio* gene prediction relies on the use of intrinsic features of genome to find genes (features) in sequence
  - Distinguish protein coding (gene) regions from non-coding regions

- Biological insights underlying these intrinsic features
  - Protein coding sequences (genes) are relatively long sequences interrupted by shorter intergenic regions dispersed along the genome
    - *HMM transition probabilities*

  - Protein coding sequences have distinct sequence compositions compared to non-coding sequences
    - Owing to the degeneracy of the genetic code
    - *HMM emission probabilities*

Genic vs. intergenic length distributions





Gene length >> intergenic length

Koonin and Wolf (2008). Nucleic Acids Res. 36: 6688

Genome sequence composition: coding vs. non-coding

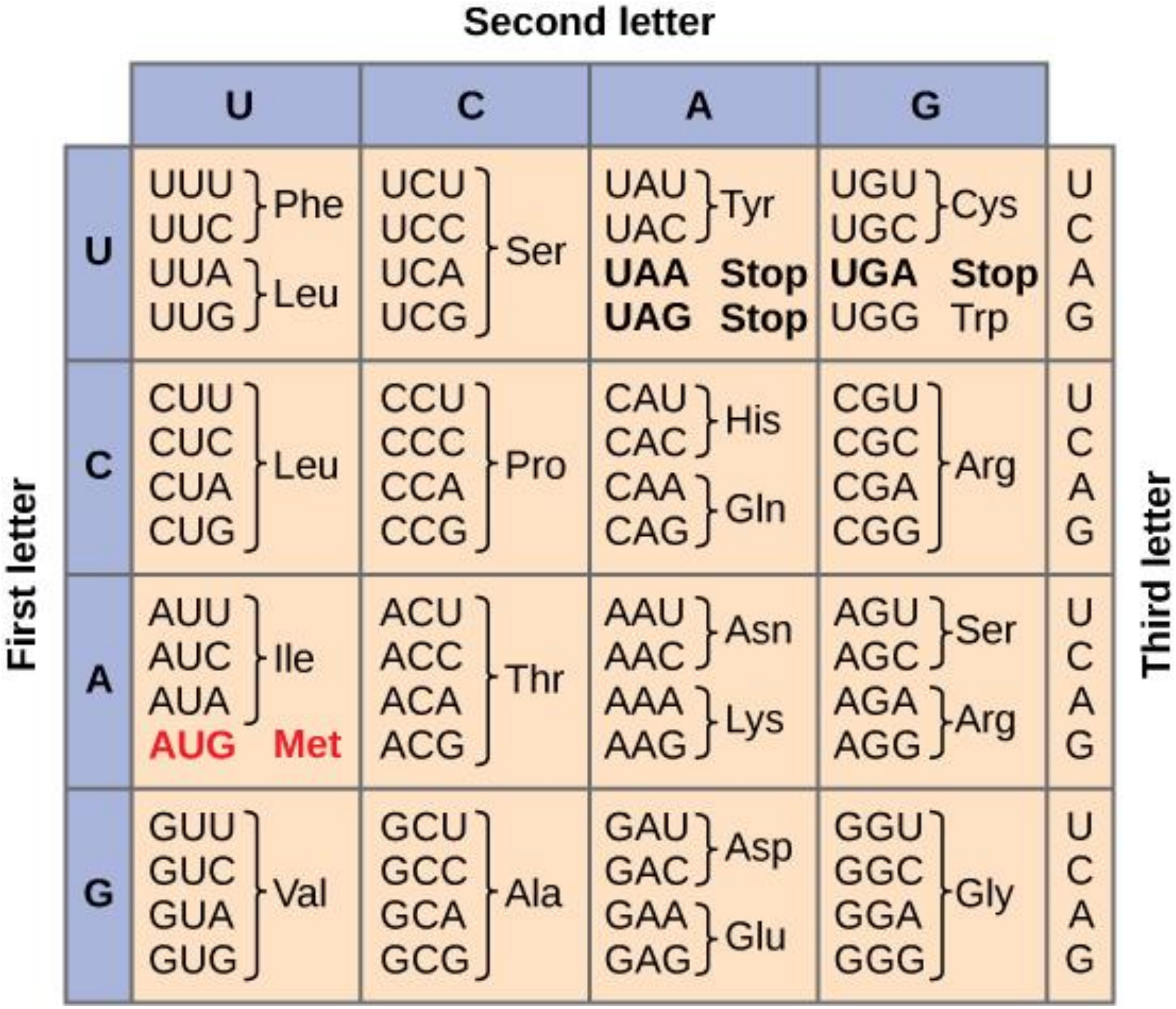
- Sequence composition (% GC content) differs across different organisms (species)

- % GC content differs between protein coding (higher) and non-coding (lower) regions

- % GC content differs among different positions of codons
  - Based on composition (availability) of tRNAs

Codon usage database
http://www.kazusa.or.jp/codon/

Genetic code



First letter	Second letter: U		C		A		G		Third letter
U	UUU	Phe	UCU	Ser	UAU	Tyr	UGU	Cys	U
	UUC		UCC		UAC		UGC		C
	UUA	Leu	UCA		**UAA**	**Stop**	**UGA**	**Stop**	A
	UUG		UCG		**UAG**	**Stop**	UGG	Trp	G
C	CUU	Leu	CCU	Pro	CAU	His	CGU	Arg	U
	CUC		CCC		CAC		CGC		C
	CUA		CCA		CAA	Gln	CGA		A
	CUG		CCG		CAG		CGG		G
A	AUU	Ile	ACU	Thr	AAU	Asn	AGU	Ser	U
	AUC		ACC		AAC		AGC		C
	AUA		ACA		AAA	Lys	AGA	Arg	A
	AUG	**Met**	ACG		AAG		AGG		G
G	GUU	Val	GCU	Ala	GAU	Asp	GGU	Gly	U
	GUC		GCC		GAC		GGC		C
	GUA		GCA		GAA	Glu	GGA		A
	GUG		GCG		GAG		GGG		G

- Code is redundant
- *Synonymous codons* = different codons (RNA triplets) encoding the same amino acid
- Constraints on overall and codon position-specific %GC content

Codon usage

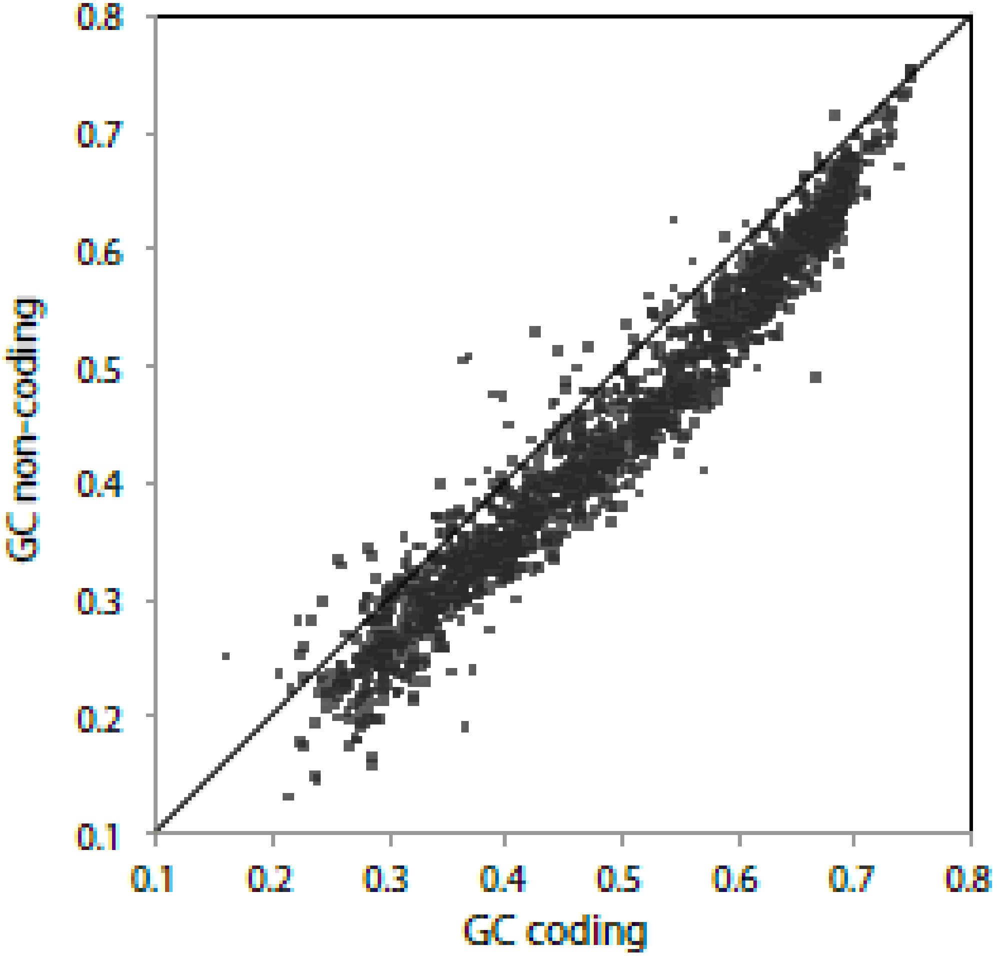
- Synonymous codons are used at different frequencies in different organisms (species)
    - Based on availability (abundance) of specific tRNAs

E. coli Leucine	
UUA	13.8%
UUG	13.0%
CUU	11.4%
CUC	10.5%
CUA	3.9%
CUG	51.1%

B. subtilis Leucine	
UUA	19.8%
UUG	15.8%
CUU	21.8%
CUC	10.7%
CUA	4.9%
CUG	23.0%

Codon usage database
http://www.kazusa.or.jp/codon/

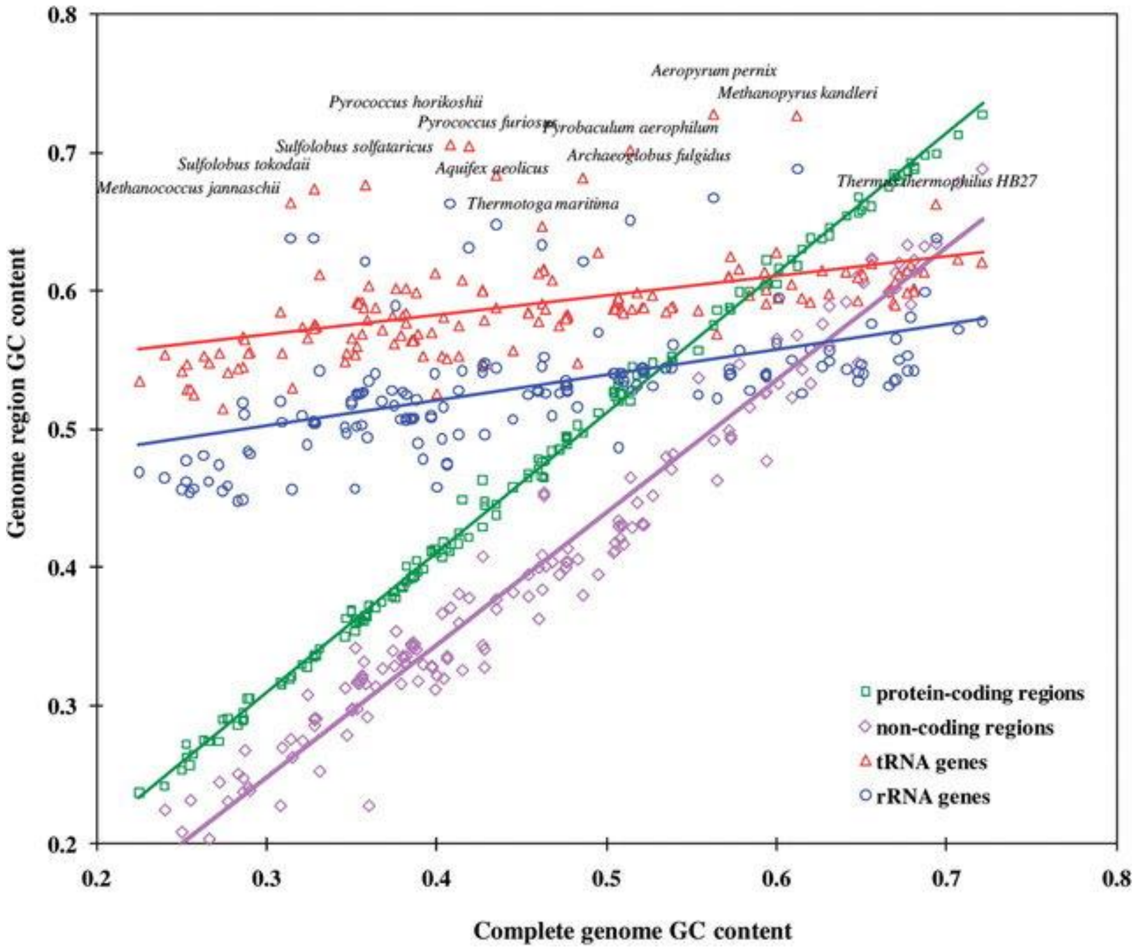
Genome sequence composition: coding vs. non-coding



- GC coding > GC non-coding

Brocchieri (2014) J Phylogenetics Evol Biol 2: e108

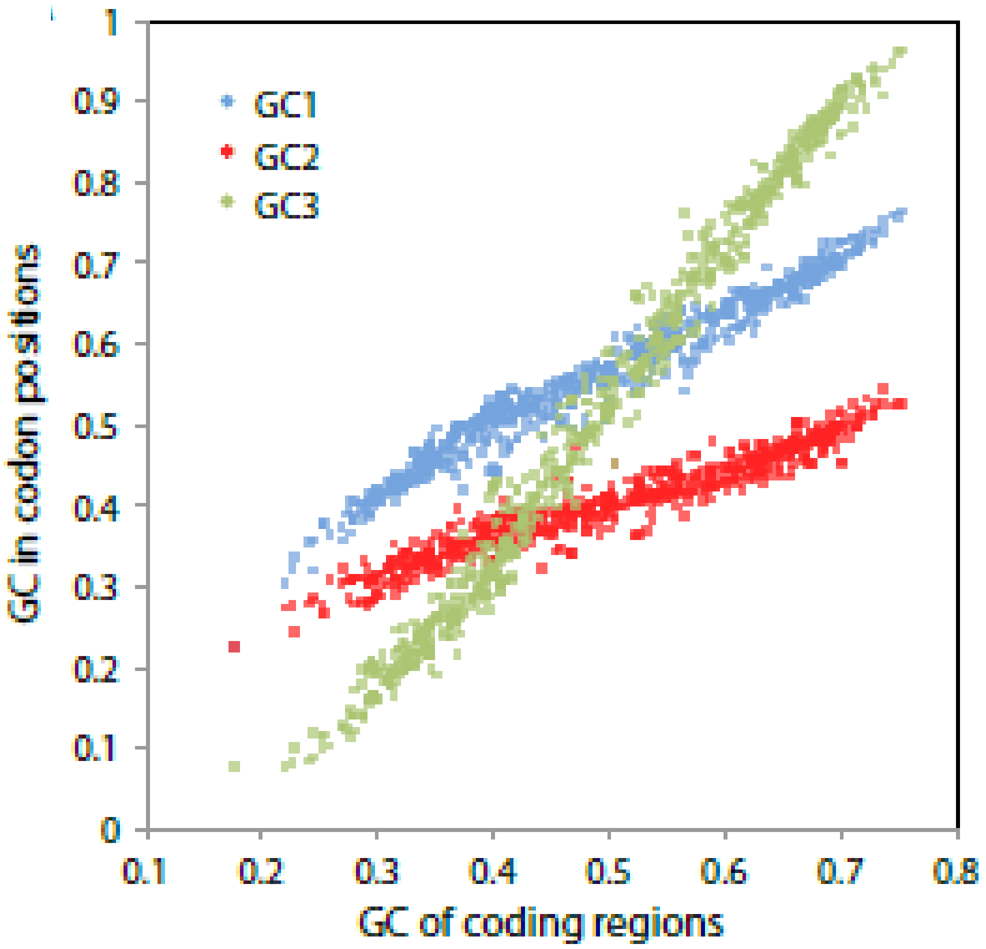
Genome sequence composition: coding vs. non-coding



- GC coding > GC non-coding

Zhu et al. (2010) Nucleic Acids Res. 38: e132

Genome sequence composition: codon positions



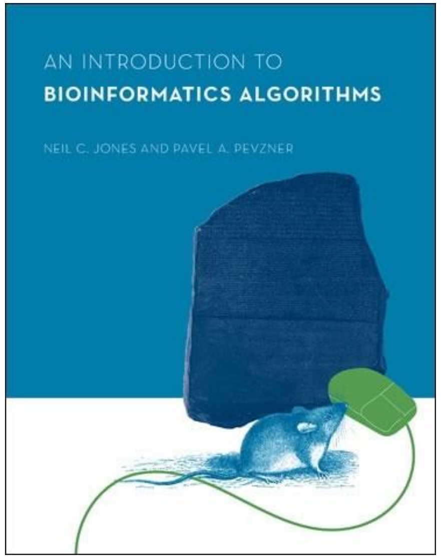
- GC1 ≅ GC2 ≅GC3 coding

Brocchieri (2014) J Phylogenetics Evol Biol 2: e108

HMMs for bacterial gene prediction (finding)

- Gene finding = distinguish protein coding from non-coding regions in a DNA sequence

1. Formulate the problem of gene finding in the context of HMMs (evaluation)
2. Use biological knowledge to parameterize (train) HMMs (learning)
3. Use dynamic programming (Viterbi) algorithm to solve problem (decoding)



HMM as a symbol emitting ‘machine’

- HMM is machine that produces output – discrete sequence of symbols

- At each step, machine is in one of k hidden states

- At each step, machine decides:

1. What state will I move to next
   - Choose from among k hidden states

2. What symbol will emit from that state
   - Choose from an alphabet Σ of symbols

HMM as a symbol (DNA) emitting ‘machine’

ATGCAATGCATTACGTGCATATGACGATTCGGCATC

Emission

Hidden State

Non-coding (N)

HMM formal definition

- Σ is an alphabet of symbols; Σ = {A, T, C, G}
- Q is a set of hidden states; $Q =$ {Coding (C), Non-coding (N)}
- A = (a_{kl}) is a matrix describing the probability of changing to state l after the HMM is in state k (learned from data)
- E = ($e_k(b)$) is a matrix describing the probability of emitting the symbol b when the HMM is in step k (learned from data)

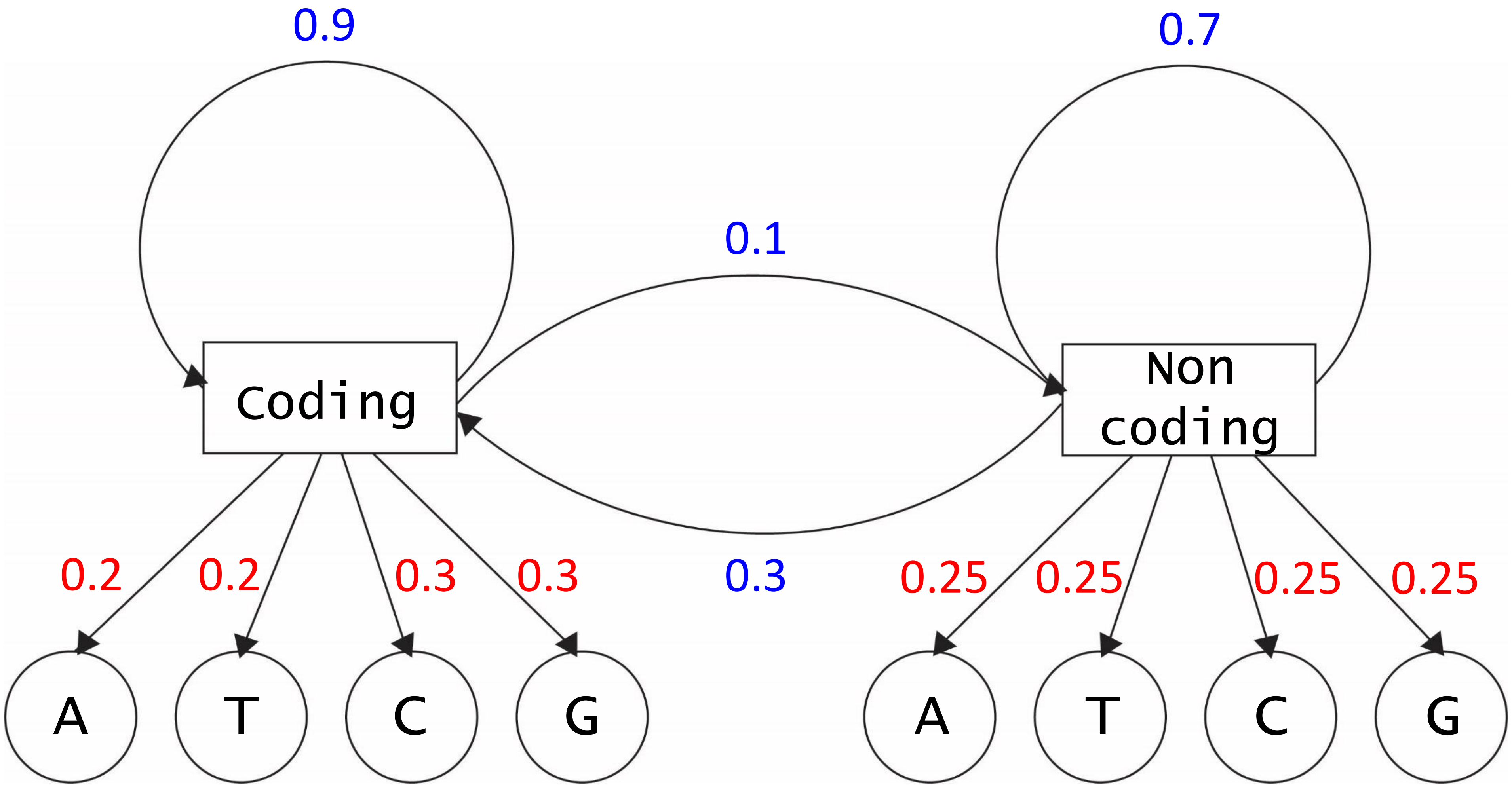
Hidden state transition matrix A - (a_{kl})

	Coding (C$_l$)	Non-coding (NC$_l$)
Coding (C$_k$)	0.9	0.1
Non-coding (NC$_k$)	0.3	0.7

Hidden state emission matrix E - $(e_k(b))$

b	Coding (C_k)	Non-coding (NC_k)
A	0.2	0.25
T	0.2	0.25
C	0.3	0.25
G	0.3	0.25

HMM for coding vs. non-coding sequence



Probability of a path through the HMM given the observed states (evaluating)

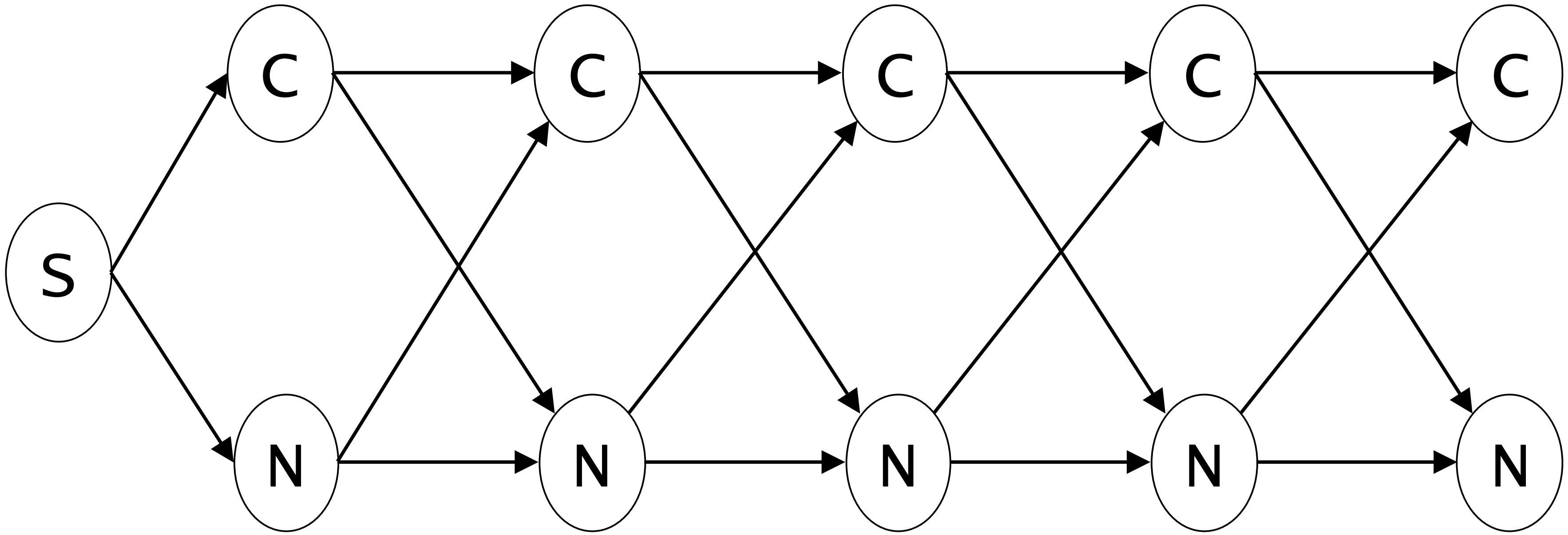
$$\begin{array}{c|cccccccccc|}
x & \text{G} & \text{C} & \text{A} & \text{C} & \text{T} & \text{A} & \text{T} & \text{G} & \text{G} & \text{C} \\
\pi & \text{Cd} & \text{Cd} & \text{Cd} & \text{Cd} & \text{Nc} & \text{Nc} & \text{Nc} & \text{Cd} & \text{Cd} & \text{Cd} \\
P(x_i|\pi_i) & 0.3 & 0.3 & 0.2 & 0.3 & 0.25 & 0.25 & 0.25 & 0.3 & 0.3 & 0.3 \\
P(\pi_{i-1} \rightarrow \pi_i) & 0.8 & 0.9 & 0.9 & 0.9 & 0.1 & 0.7 & 0.7 & 0.3 & 0.9 & 0.9
\end{array}$$

$$= \prod_{i=1}^{n} P(\pi_{i-1} \rightarrow \pi_i) P(x_i|\pi_i)$$

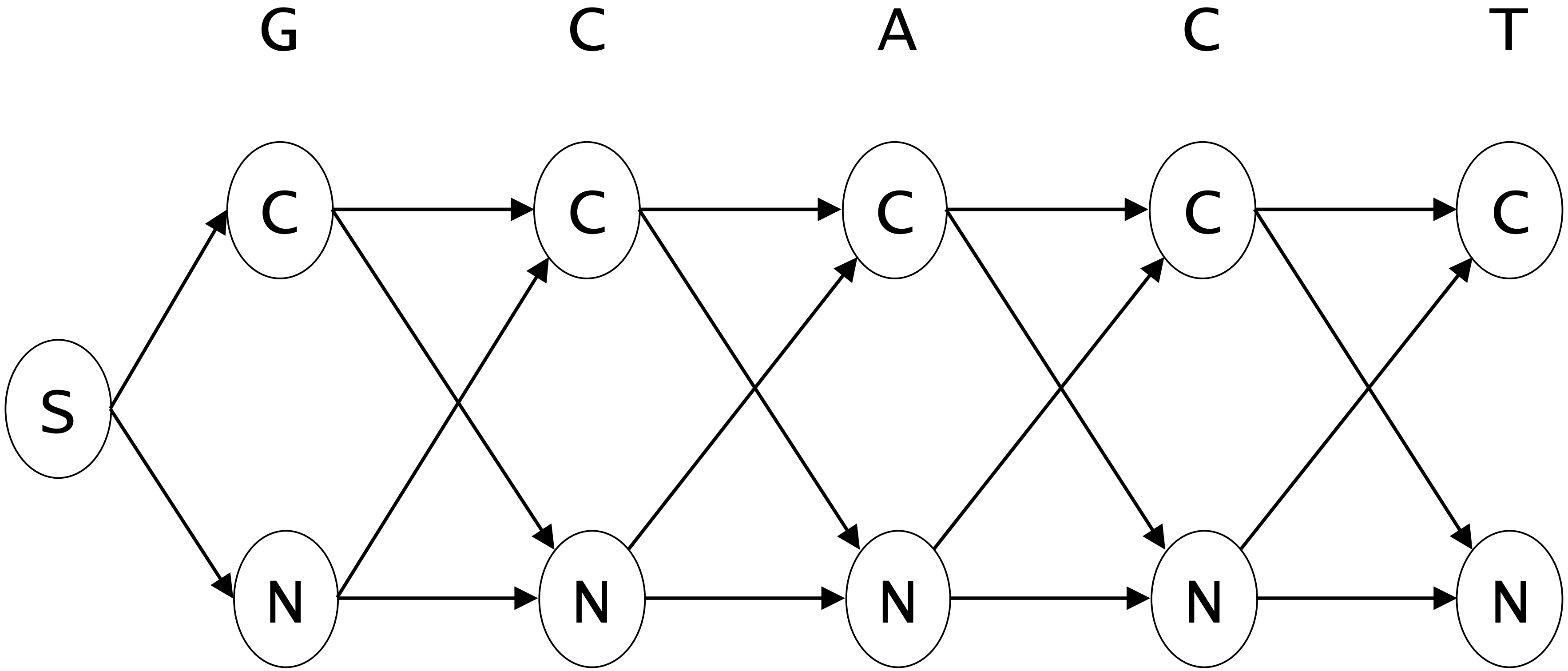
$$= (0.8*0.3)\ (0.9*0.3)\ (0.9*0.2)\ (0.9*0.3)\ (0.1*0.25)\ (0.7*0.25)\ (0.7*0.25)\ (0.3*0.3)\ (0.9*0.3)\ (0.9*0.3)$$

Note that log values are used for mathematical simplicity

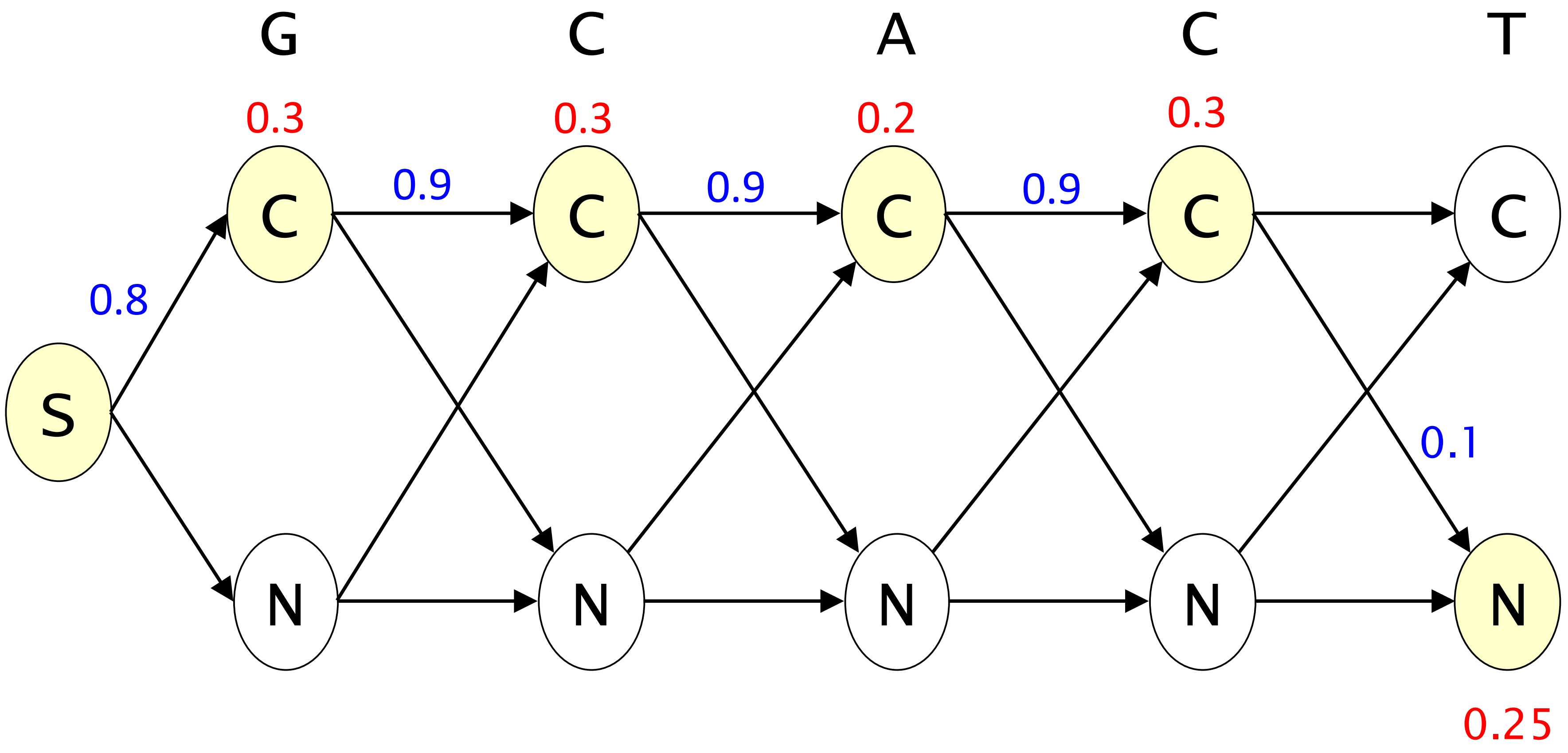
Evaluating the HMM (probability model generated output)



Evaluating the HMM (probability model generated output)



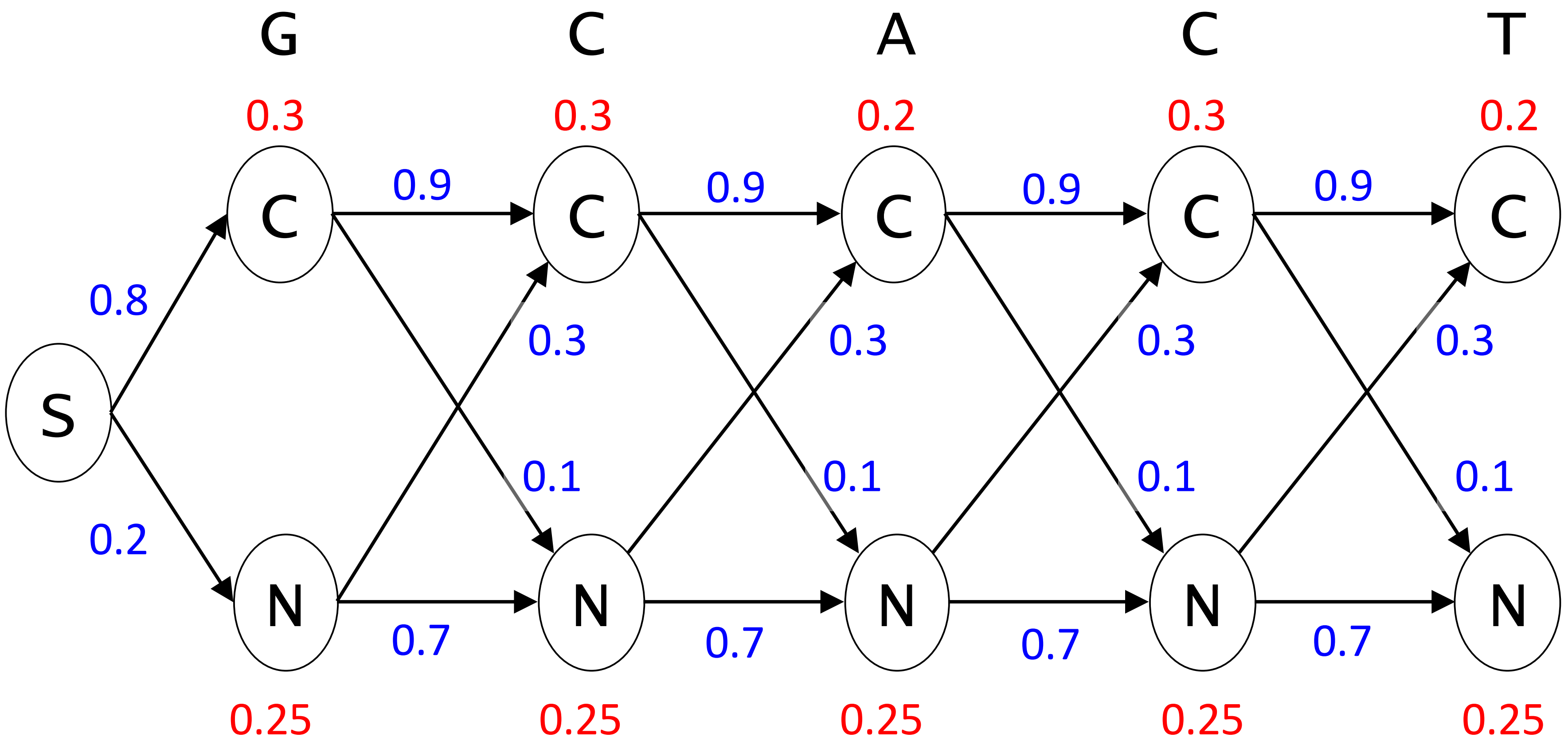
Evaluating the HMM (probability model generated output)



= (0.8*0.3) (0.9*0.3) (0.9*0.2) (0.9*0.3) (0.1*0.25)

Decoding the HMM (solving for best path)

but which is best path … form 2^n possible paths



https://www.youtube.com/watch?v=kqSzLo9fenk

log transformation for mathematical convenience

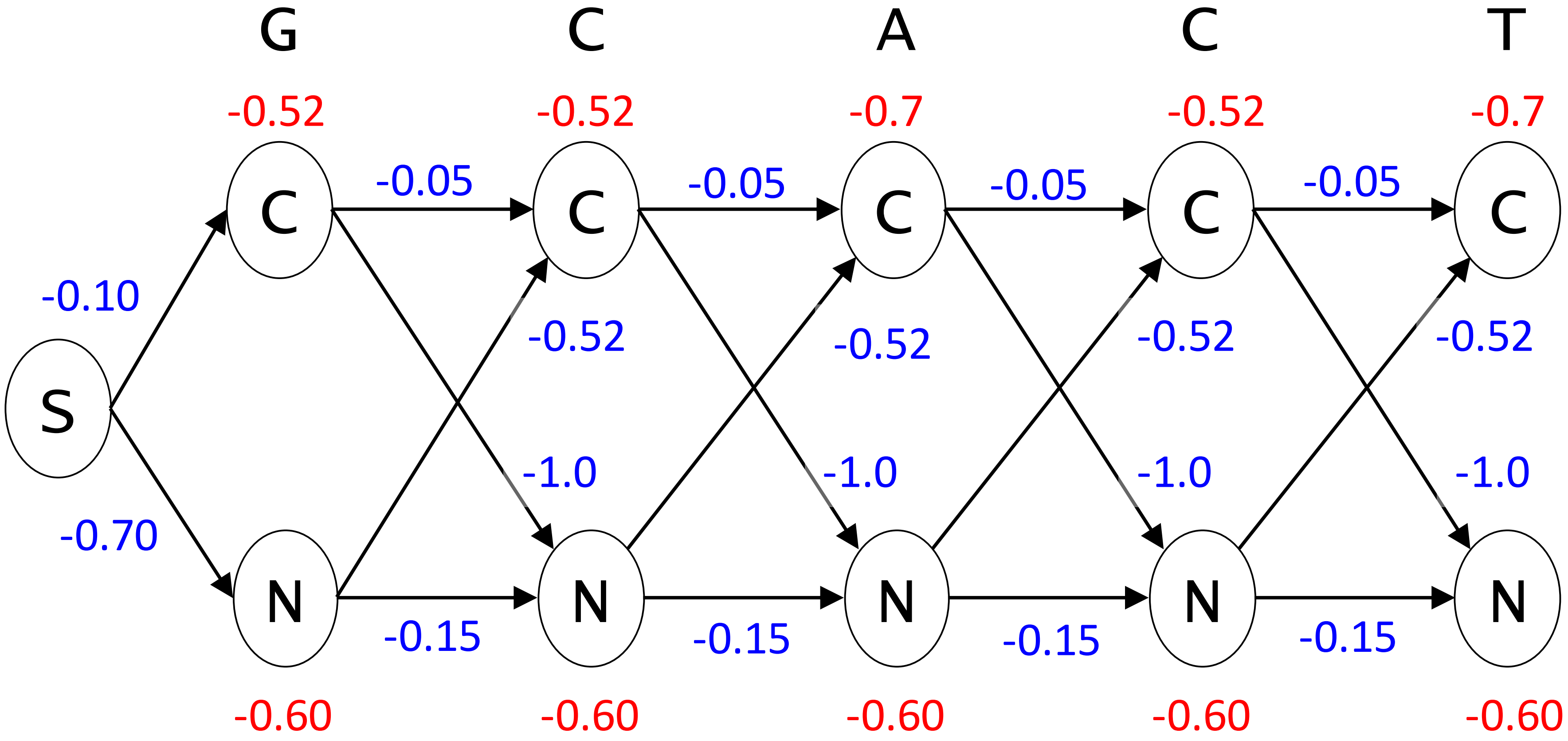
- We are multiplying probabilities (fractions) to get the best path
- Path that maximizes $P(\pi | x)$ over all possible paths π
- This quickly leads to very small fractions and overflow
- log transformed probabilities are used to avoid this problem
- Adding log transformed values is equivalent to multiplying the same values

$0.8 * 0.3 = 0.24$ $\log_{10}(0.24) = \mathbf{-0.62}$

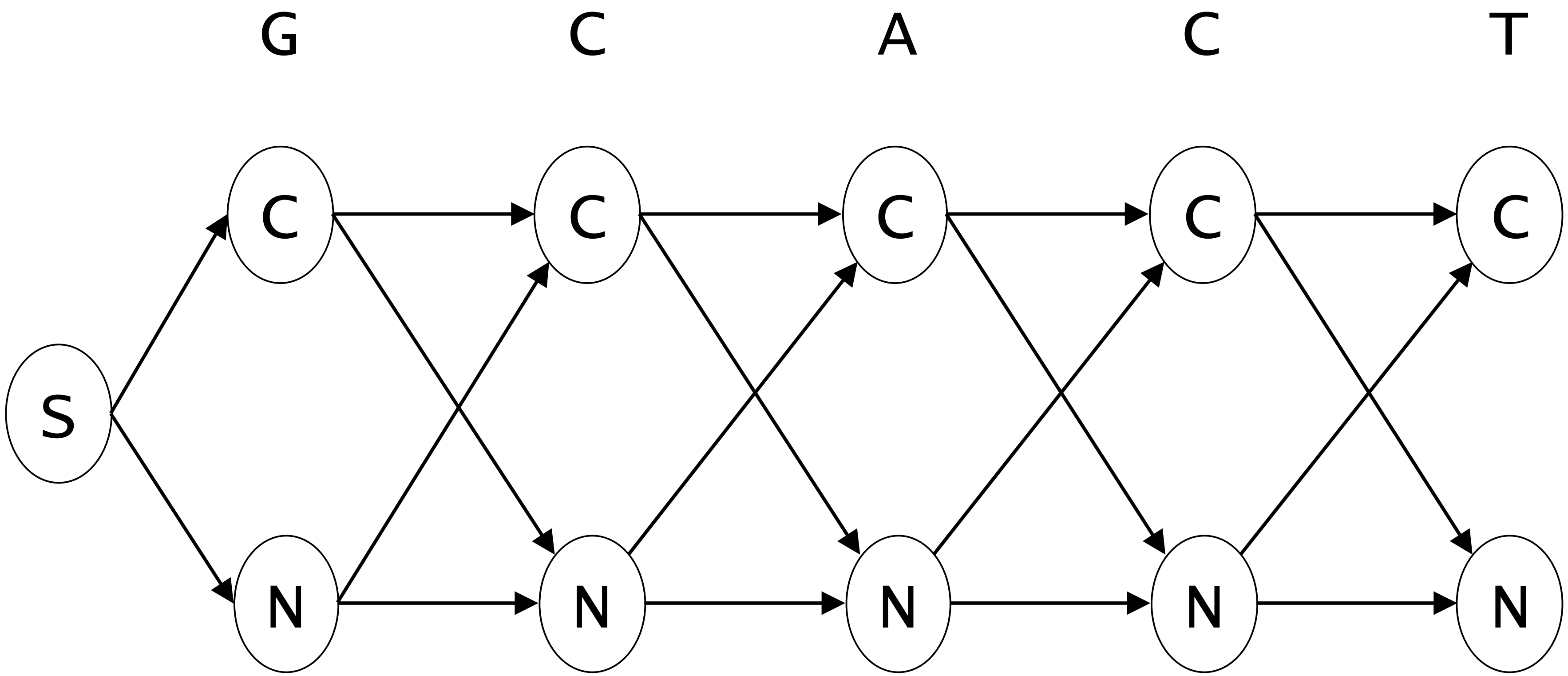
$\log_{10}(0.8) = -0.097$ $\log_{10}(0.3) = -0.52$ $-0.097 + -052 = \mathbf{-0.62}$

Decoding the HMM (solving for best path)

but which is best path … from 2^n possible paths … log transformed

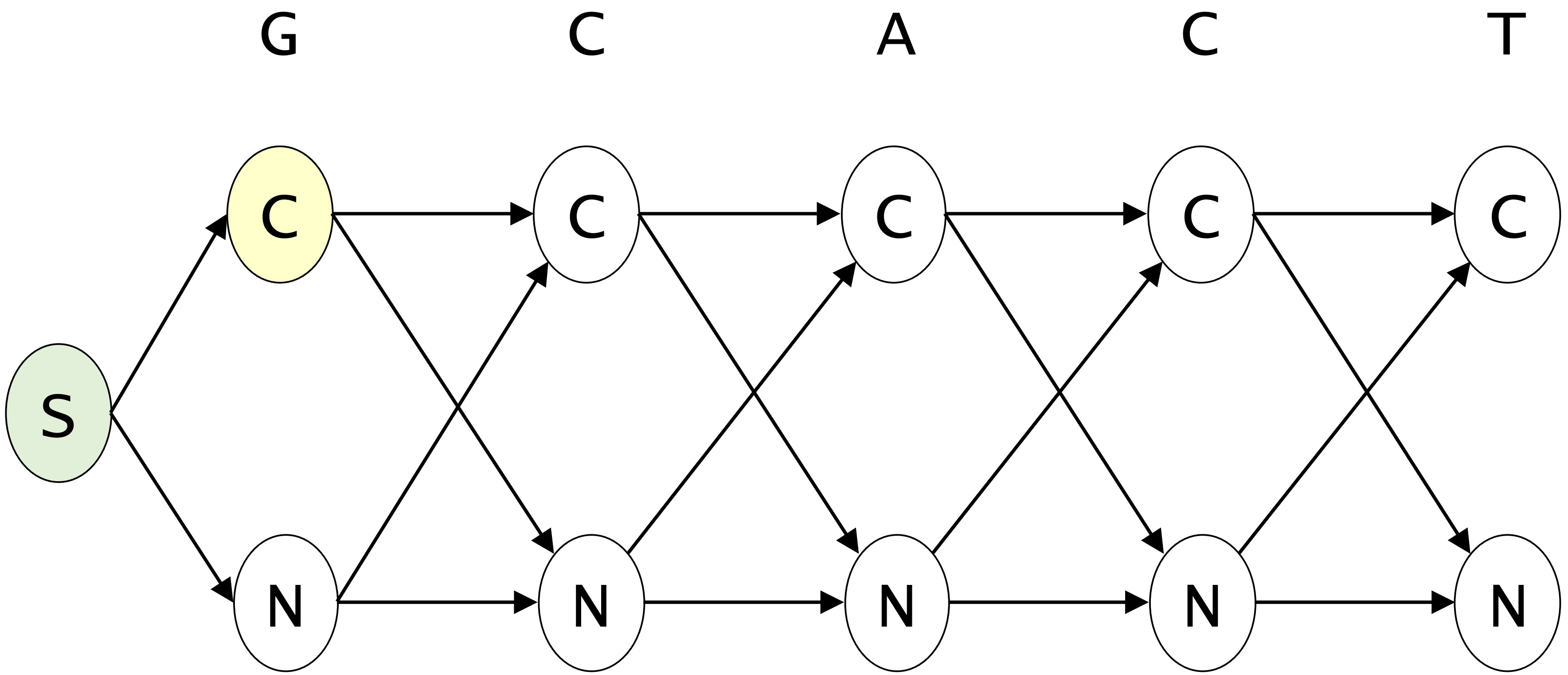


Dynamic programming with Viterbi algorithm



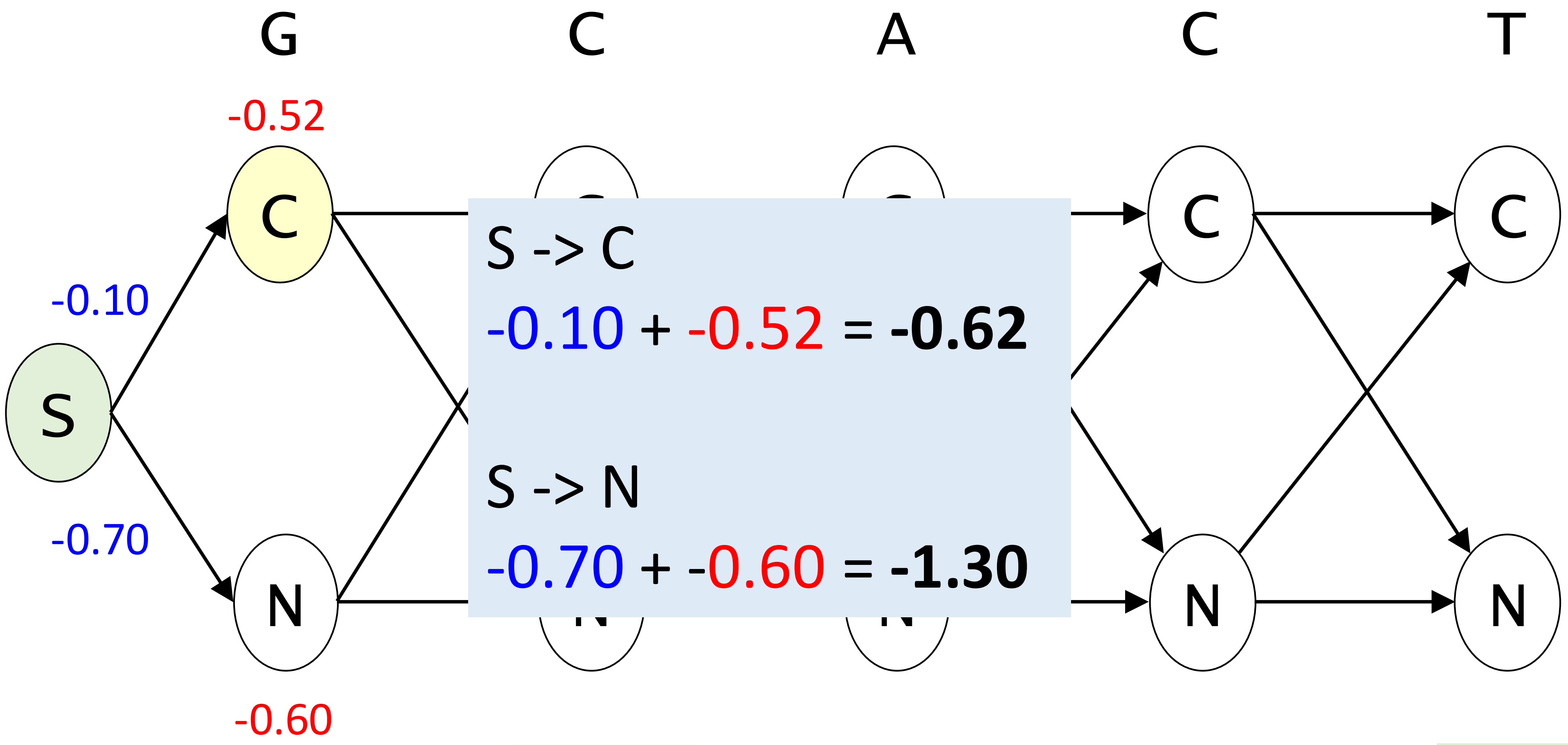
solve each sub-problem (left -> right), then trace best path

Dynamic programming with Viterbi algorithm



compute maximum state i scores for all possible paths from state i-1

Dynamic programming with Viterbi algorithm



S -> C

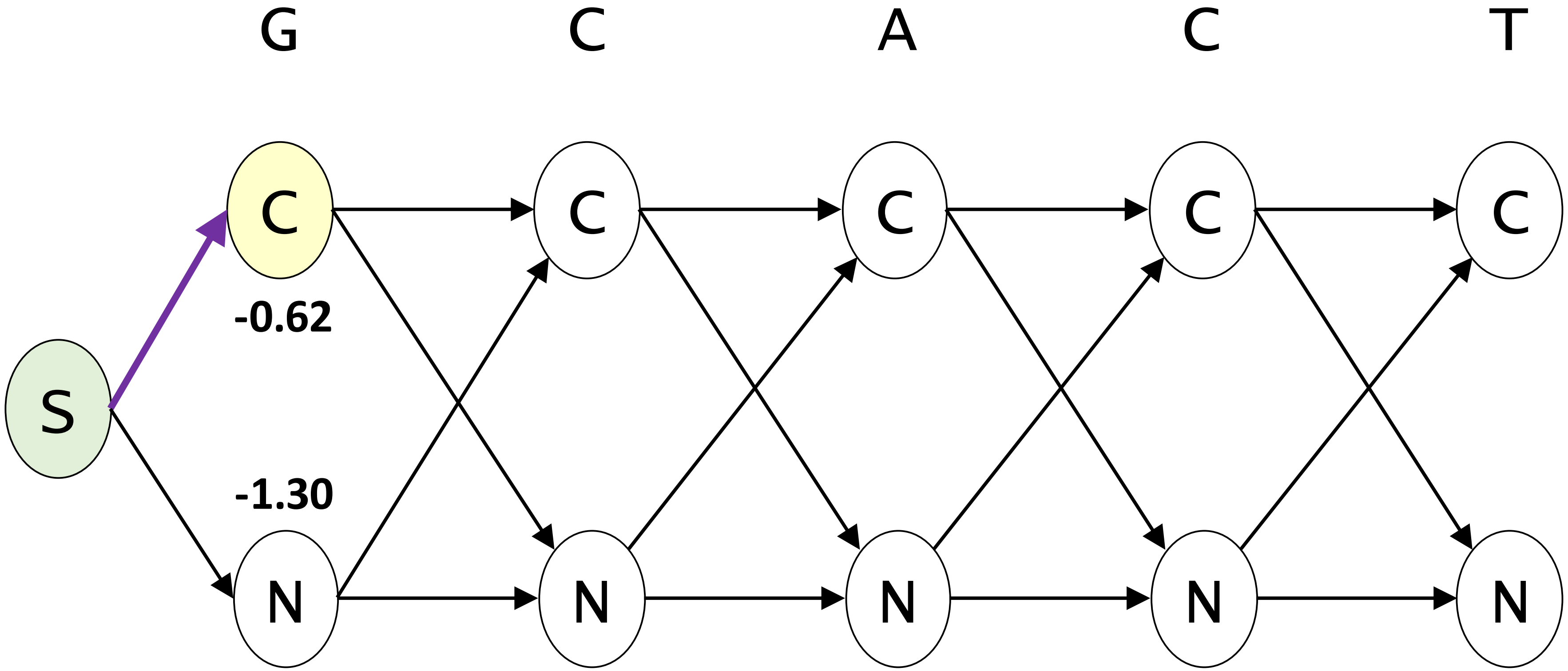
-0.10 + -0.52 = **-0.62**

S -> N

-0.70 + -0.60 = **-1.30**

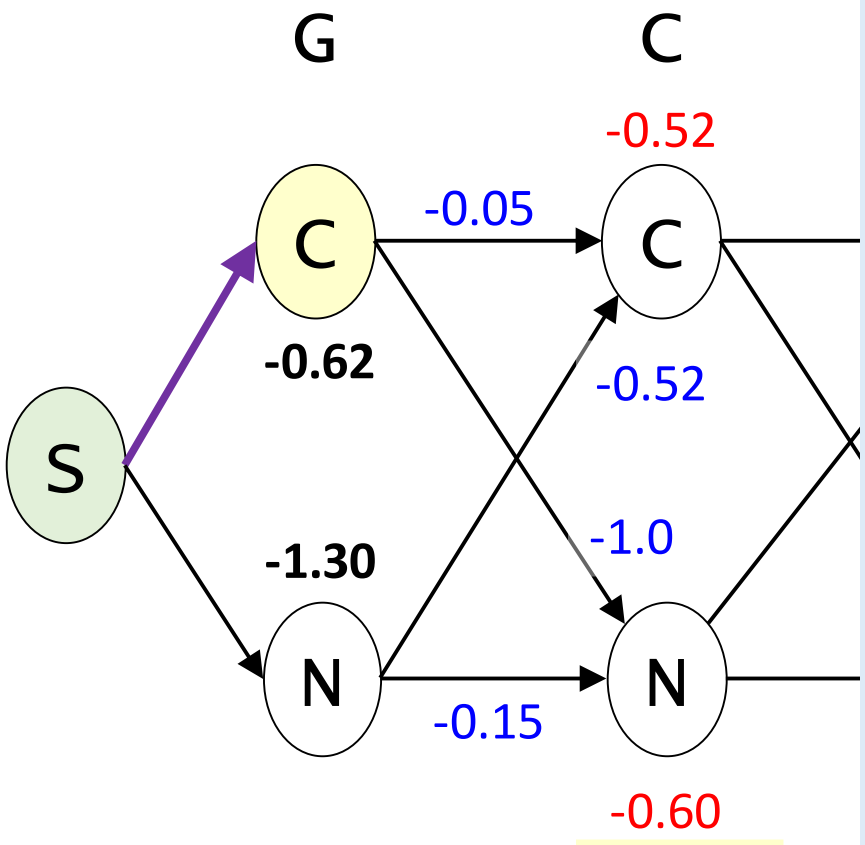
compute maximum state i scores for all possible paths from state i-1

Dynamic programming with Viterbi algorithm



compute maximum state i scores for all possible paths from state i-1

Dynamic programming with Viterbi algorithm



C -> C

-0.62 + -0.05 + -0.52 = **-1.19**

N -> C

-1.30 + -0.52 + -0.52 = -2.34

C -> N

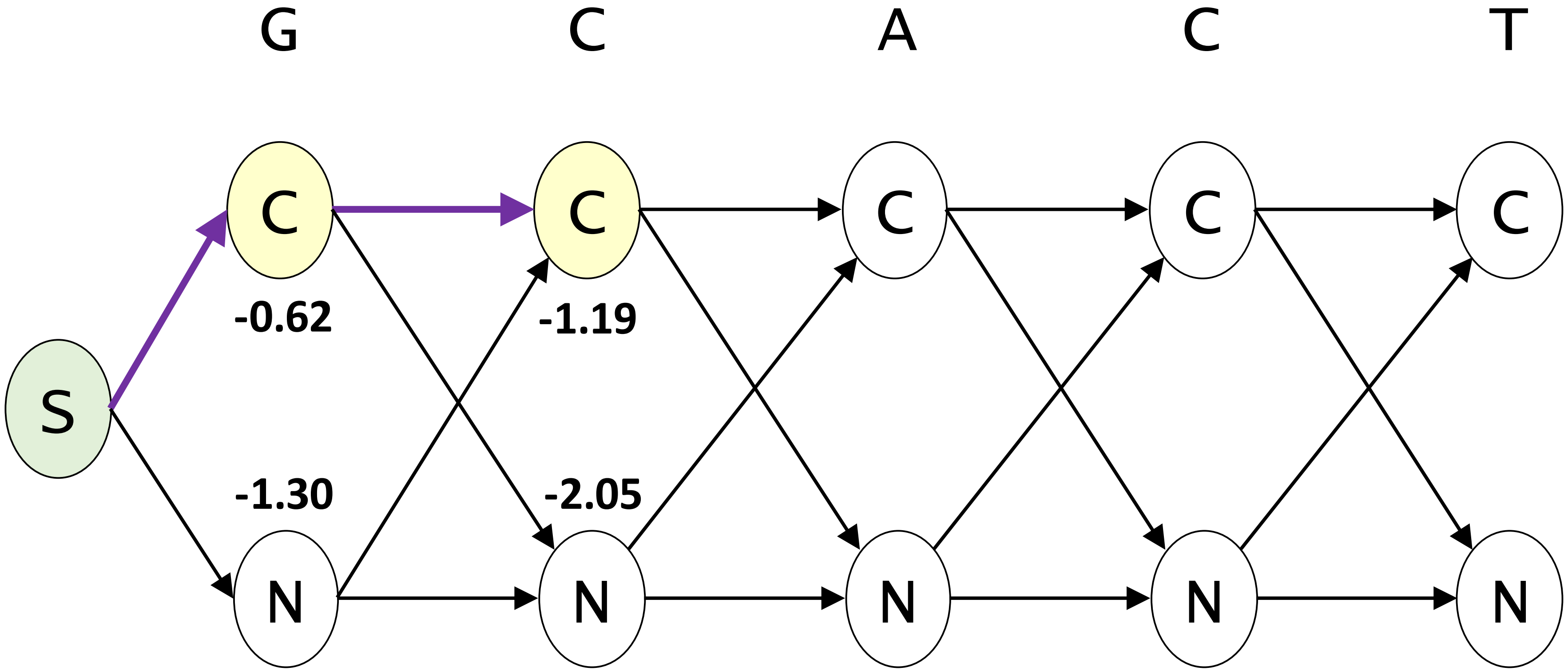
-0.62 + -1.0 + -0.60 = -2.22

N -> N

-1.30 + -0.15 + -0.60 = **-2.05**

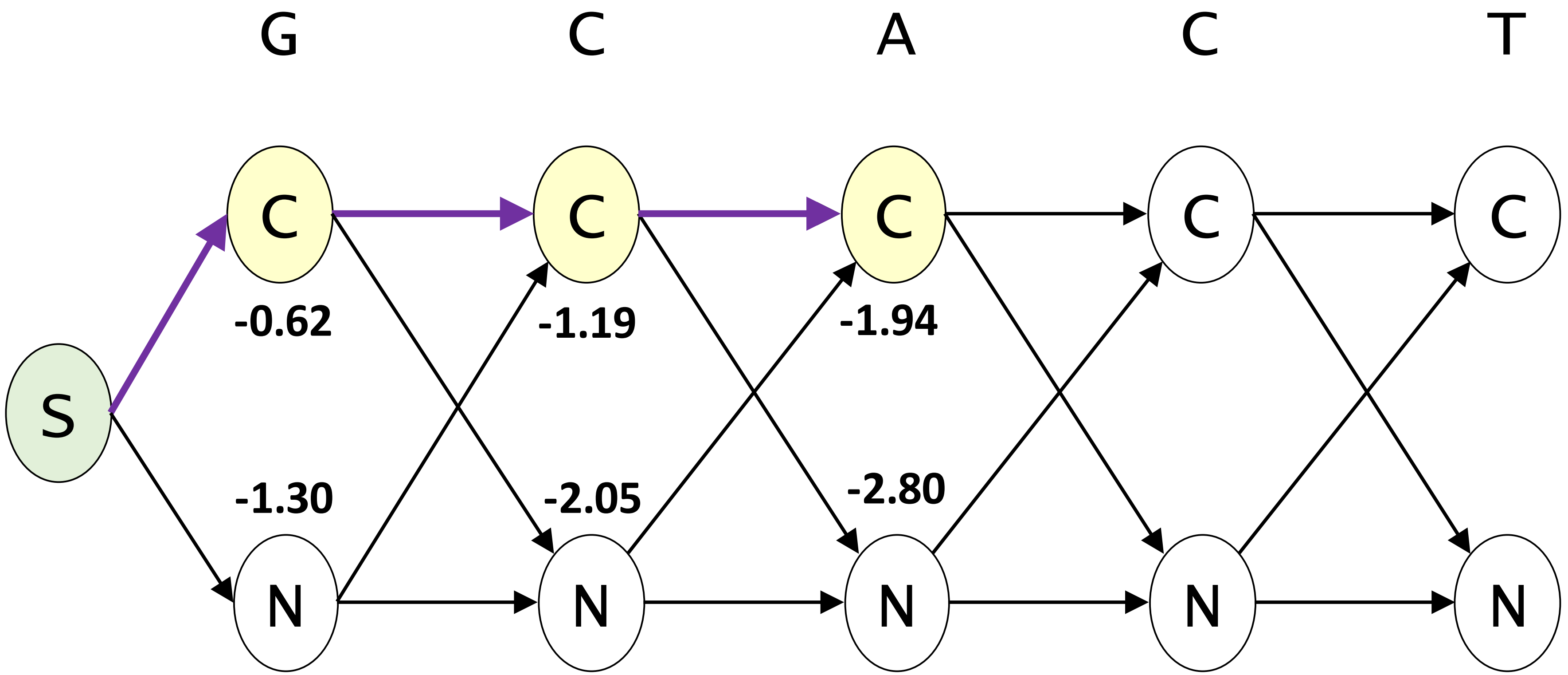
compute maximum state i sco i-1

Dynamic programming with Viterbi algorithm



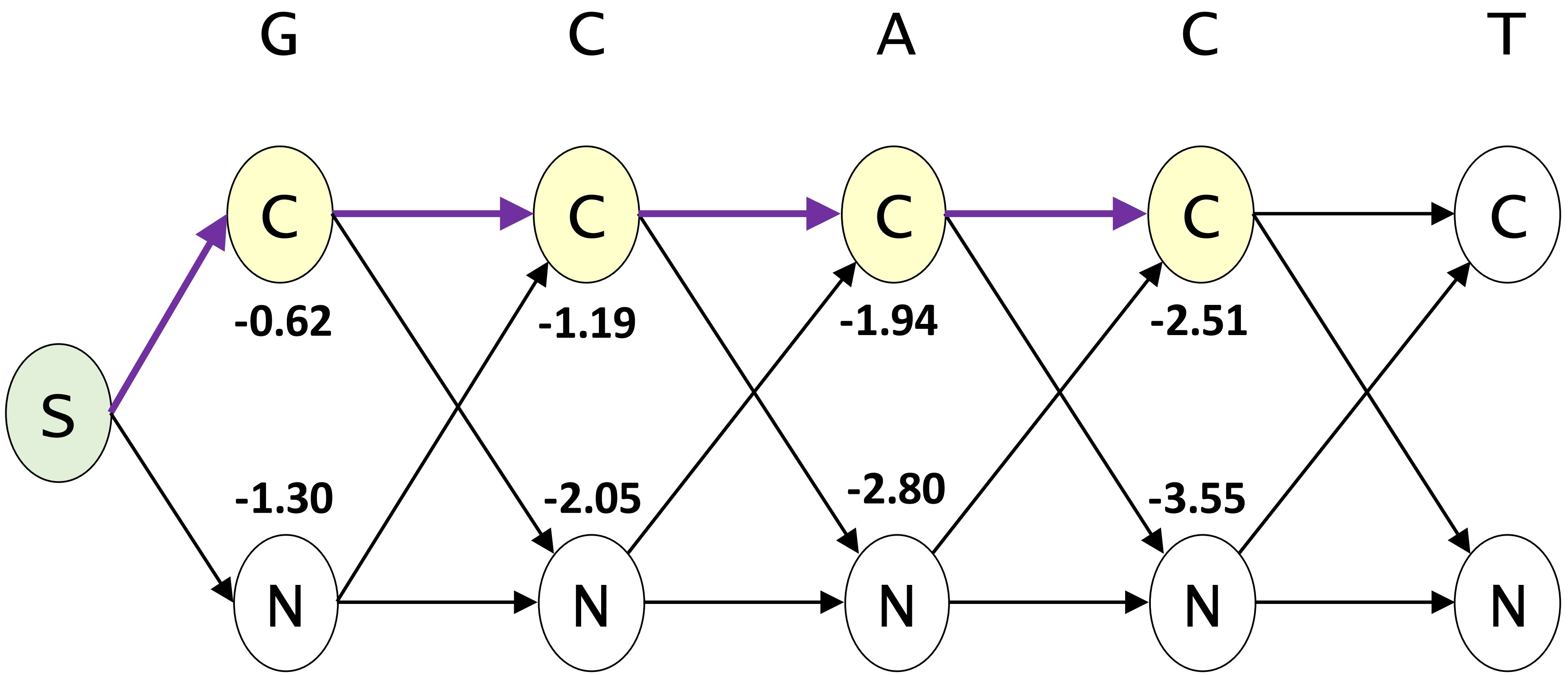
compute maximum state i scores for all possible paths from state i-1

Dynamic programming with Viterbi algorithm



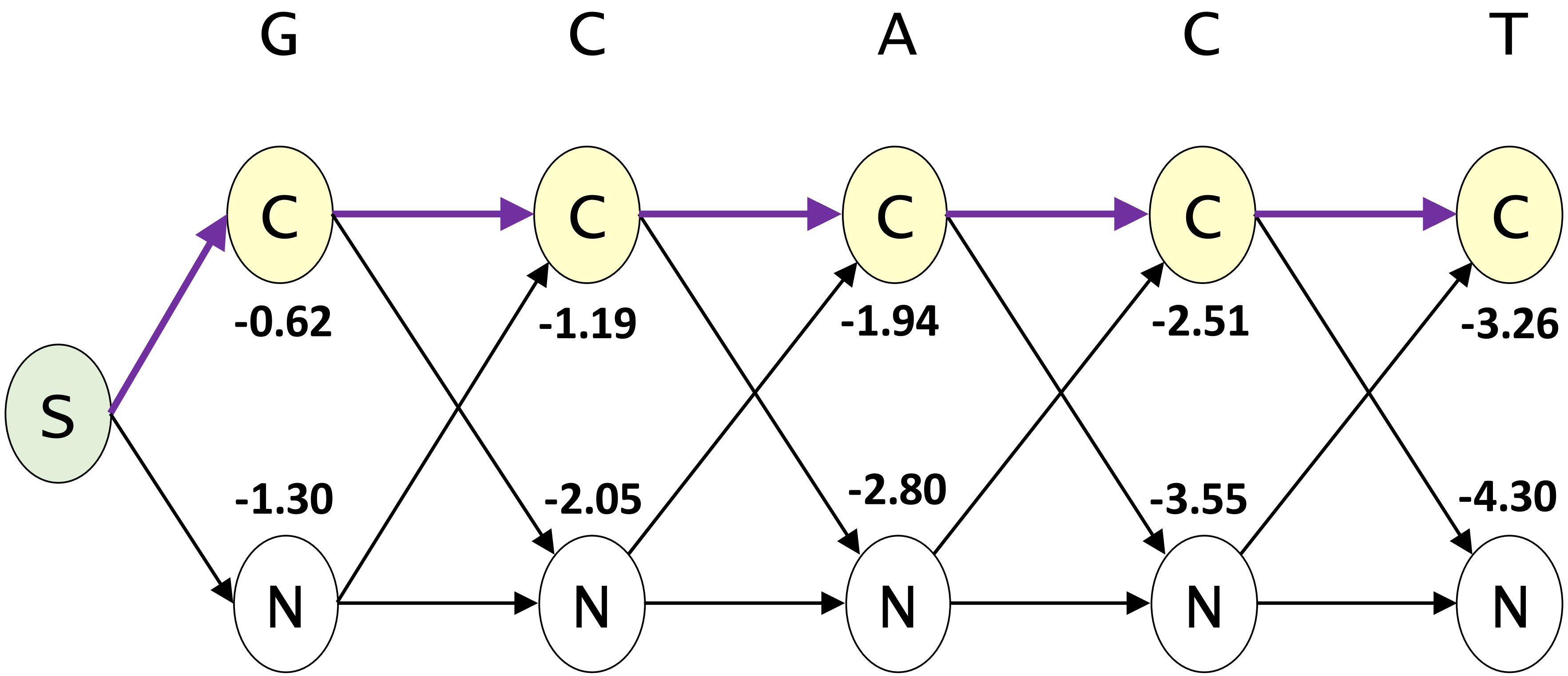
compute maximum state i scores for all possible paths from state i-1

Dynamic programming with Viterbi algorithm



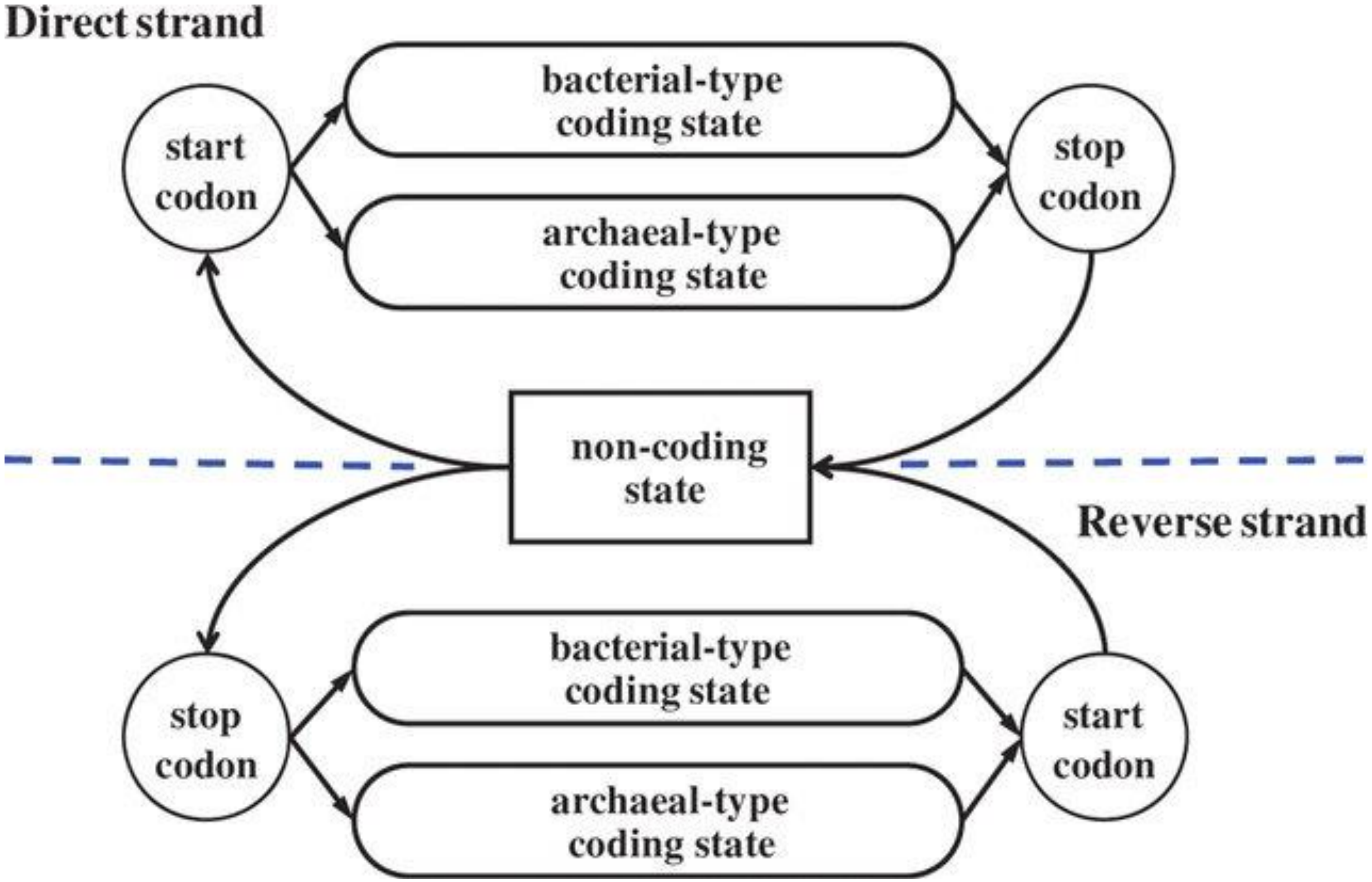
compute maximum state i scores for all possible paths from state i-1

Dynamic programming with Viterbi algorithm



compute maximum state i scores for all possible paths from state i-1

More realistic gene finding HMM


Direct strand
start codon
bacterial-type coding state
archaeal-type coding state
stop codon
non-coding state
Reverse strand
stop codon
bacterial-type coding state
archaeal-type coding state
start codon

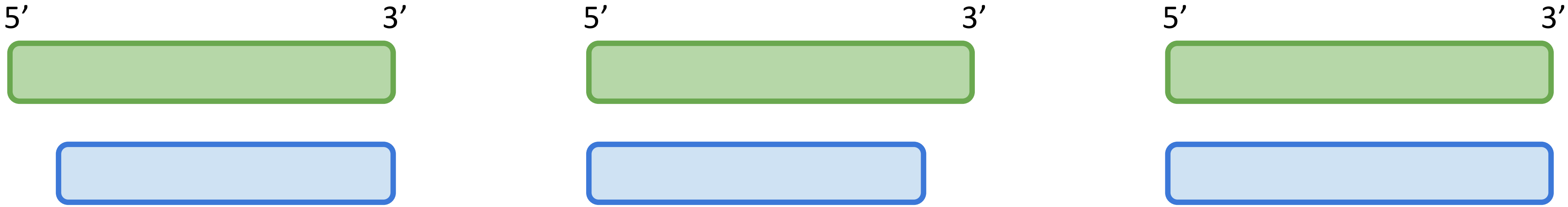
Zhu et al. (2010) Nucleic Acids Res. 38: e132

Additional complexities

- Higher order Markov models – kth order model, probability of event based on k previous events (nucleotides)
    - Previous example based on simple 1st order model
- Inhomogenous Markov models – changes probabilities based on codon position (captures periodicity of genetic code)
- Interpolated Markov models – value of k changes depending on local nucleotide context

Evaluating gene prediction accuracy

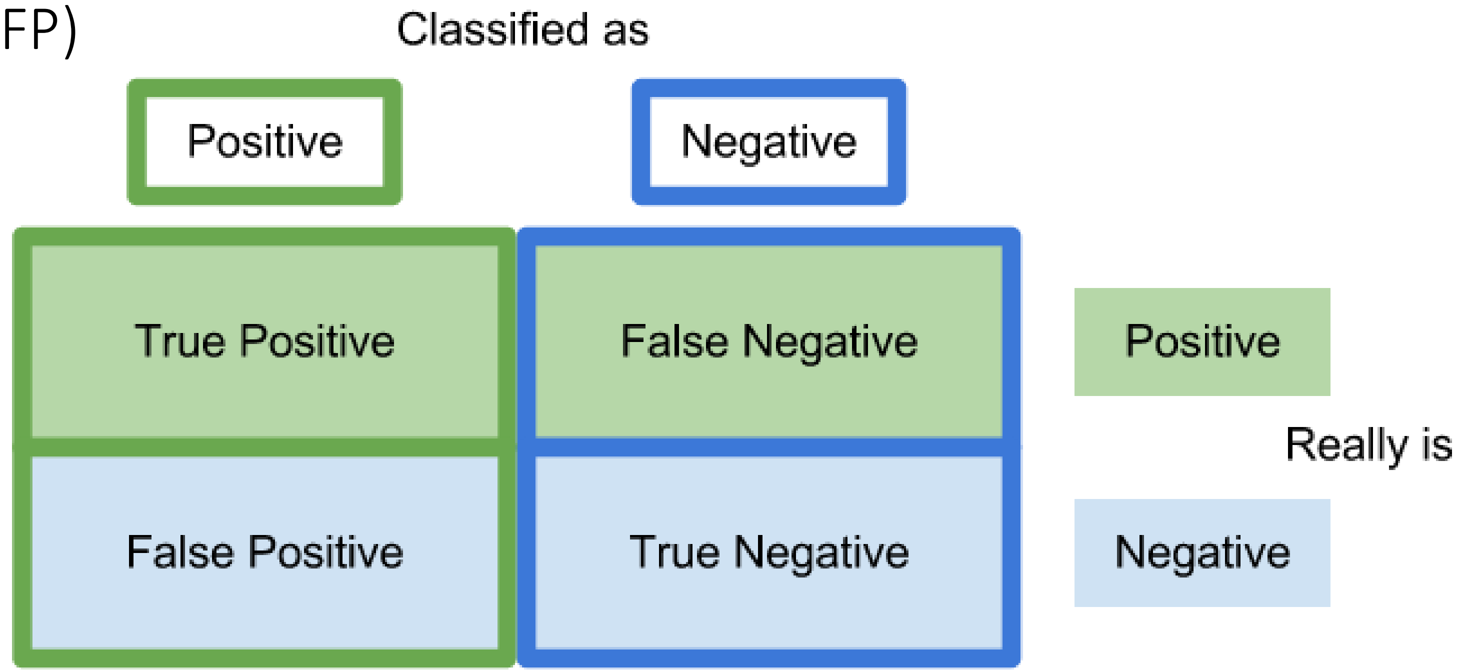
- Overlap measured according to 5’ (start) and 3’ (stop) site correspondence
- Start sites vary more often than stop sites (results will differ)



Real genes vs. Predicted genes

Evaluating gene prediction accuracy

- Sensitivity (Sn) = TP / (TP + FN)
- Specificity (Sp) = TN / (TN + FP)



Classified as

Positive	Negative	Really is
True Positive	False Negative	Positive
False Positive	True Negative	Negative

https://en.wikipedia.org/wiki/Sensitivity_and_specificity

Additional questions?