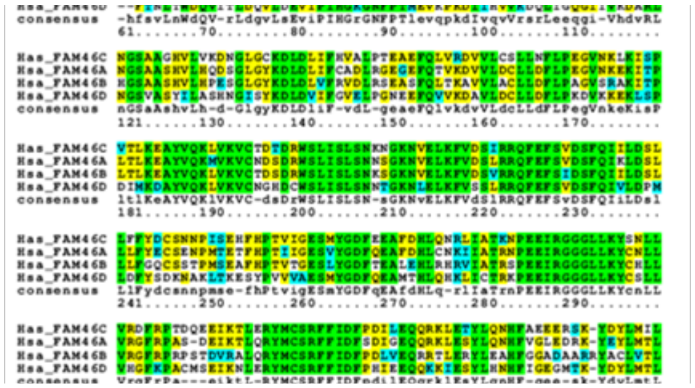
CSCI 1820/2820: An overview

Spring 2022

- Ch. 1 BLAST Algorithm and Karlin-Altschul Statistics
- Ch. 2 Genome Assembly and Haplotype Assembly Algorithms
- Ch. 3 Hidden Markov Models (HMM) Algorithms: The Learning Problem
- Ch. 4 Recombination and Ancestral Recombination Graphs (ARGs)
- Ch.5 Rigorous Clustering and Spectral Clustering Algorithms
- Ch. 6 Algorithms for Constructing Suffix Trees in Linear Time
- Ch. 7 Protein Folding Algorithms (Introduction)

Ch. 1: BLAST Algorithm



```
Hsa_FAM46D  --[illegible]
consensus   -hfsvLnWdQV-rLdgvLsEviPIHGrGNFPTlevqpkdIvqvVrsrLeeqgi-VhdvRL
            61.......70.........80.........90.........100.......110........

Has_FAM46C  NGSAAGHVLVKDNGLGCKDLDLIFHVALPTEAEFQLVRDVVLCSLLNFLPEGVNKLKISP
Hsa_FAM46A  NGSAASHVLHQDSGLGYKDLDLIFCADLRGEGEFQTVKDVVLDCLLDFLPEGVNKEKITP
Hsa_FAM46B  HGSAASHVLHPESGLGYKDLDLVFRVDLRSEASFQLTKAVVLACLLDFLPAGVSRAKITP
Hsa_FAM46D  NGSVASYILASHNGISYKDLDVIFGVELPGNEEFQVVKDAVLDCLLDFLPKDVKKEKLSP
consensus   nGSaAshvLh-d-GlgyKDLDliF-vdL-geaeFQlvkdvVLdcLLdFLPegVnkeKisP
            121......130.......140.......150.......160.......170........

Has_FAM46C  VTLKEAYVQKLVKVCTDTDRWSLISLSNKNGKNVELKFVDSIRRQFEFSVDSFQIILDSL
Hsa_FAM46A  LTLKEAYVQKMVKVCNDSDRWSLISLSNNSGKNVELKFVDSLRRQFEFSVDSFQIKLDSL
Hsa_FAM46B  LTLKEAYVQKLVKVCTDSDRWSLISLSNKSGKNVELKFVDSVRRQFEFSIDSFQIILDSL
Hsa_FAM46D  DIMKDAYVQKLVKVCNGHDCWSLISLSNNTGKNLELKFVSSLRRQFEFSVDSFQIVLDPM
consensus   ltlKeAYVQKlVKVC-dsDrWSLISLSN-sGKNvELKFVdSlRRQFEFSvDSFQIiLDsl
            181......190.......200.......210.......220.......230........

Has_FAM46C  LFFYDCSNNPISEHFHPTVIGESMYGDFEEAFDHLQNRLIATKNPEEIRGGGLLKYSNLL
Hsa_FAM46A  LLFYECSENPMTETFHPTIIGESVYGDFQEAFDHLCNKIIATRNPEEIRGGGLLKYCNLL
Hsa_FAM46B  LLFGQCSSTPMSEAFHPTVTGESLYGDFTEALEHLRHRVIATRSPEEIRGGGLLKYCHLL
Hsa_FAM46D  LDFYSDKNAKLTKESYPVVVAESMYGDFQEAMTHLQHKLICTRKPEEIRGGGLLKYCSLL
consensus   LlFydcsnnpmse-fhPtvigESmYGDFqEAfdHLq-rlIaTrnPEEIRGGGLLKYcnLL
            241......250.......260.......270.......280.......290........

Has_FAM46C  VRDFRPTDQEEIKTLERYMCSRFFIDFPDILEQQRKLETYLQNHFAEEERSK-YDYLMIL
Hsa_FAM46A  VRGFRPAS-DEIKTLQRYMCSRFFIDFSDIGEQQRKLESYLQNHFVGLEDRK-YEYLMTL
Hsa_FAM46B  VRGFRPRPSTDVRALQRYMCSRFFIDFPDLVEQRRTLERYLEAHFGGADAARRYACLVTL
Hsa_FAM46D  VHGFKPACMSEIKNLERYMCSRFFIDFPHIEEQQKKIESYLHNHFIGEGMTK-YDYLMTL
consensus   VrgFrPa---eiktL-RYMCSRFFIDFpdilEQgrklEsYLqnHF-gee-sk-YdyLmtL
```

Questions: When a DNA sequence or protein sequence is a biological sequence?
How can we computationally identify them?

Examples of problems we need to solve along the way:

Problem 1. General scoring schemes – and the max scoring subsequence

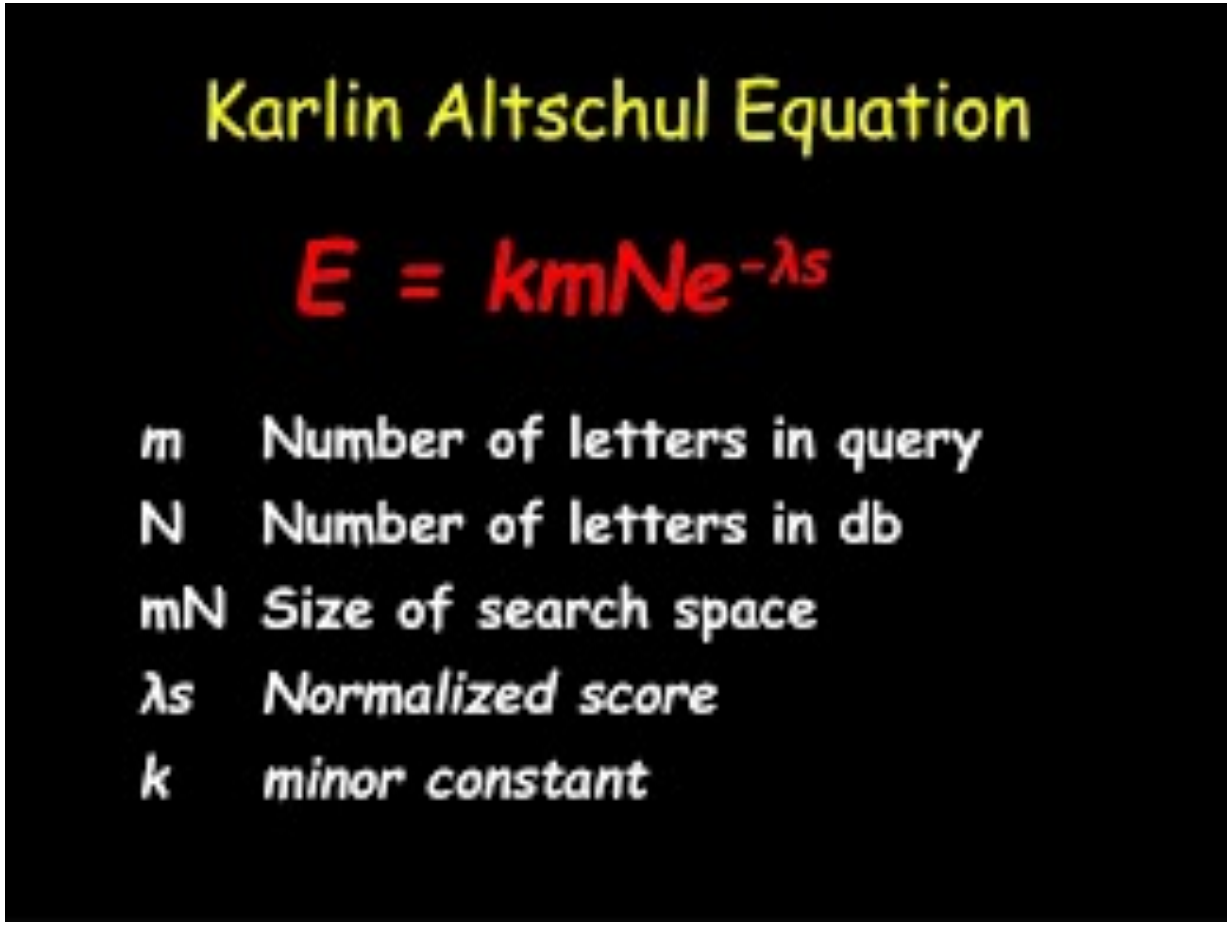
Problem 2. The Gambler's Ruin/Random Walks

The BLAST Algorithm

Authors

- Stephen Altschul
- Warren Gish
- Webb Miller
- Eugene W. Myers
- David Lipman
- "Basic Local Alignment Search Tool"

Journal of Molecular Biology (1990) 215, 403-410



Karlin Altschul Equation

$$E = kmNe^{-\lambda s}$$

m	Number of letters in query
N	Number of letters in db
mN	Size of search space
λs	*Normalized score*
k	*minor constant*

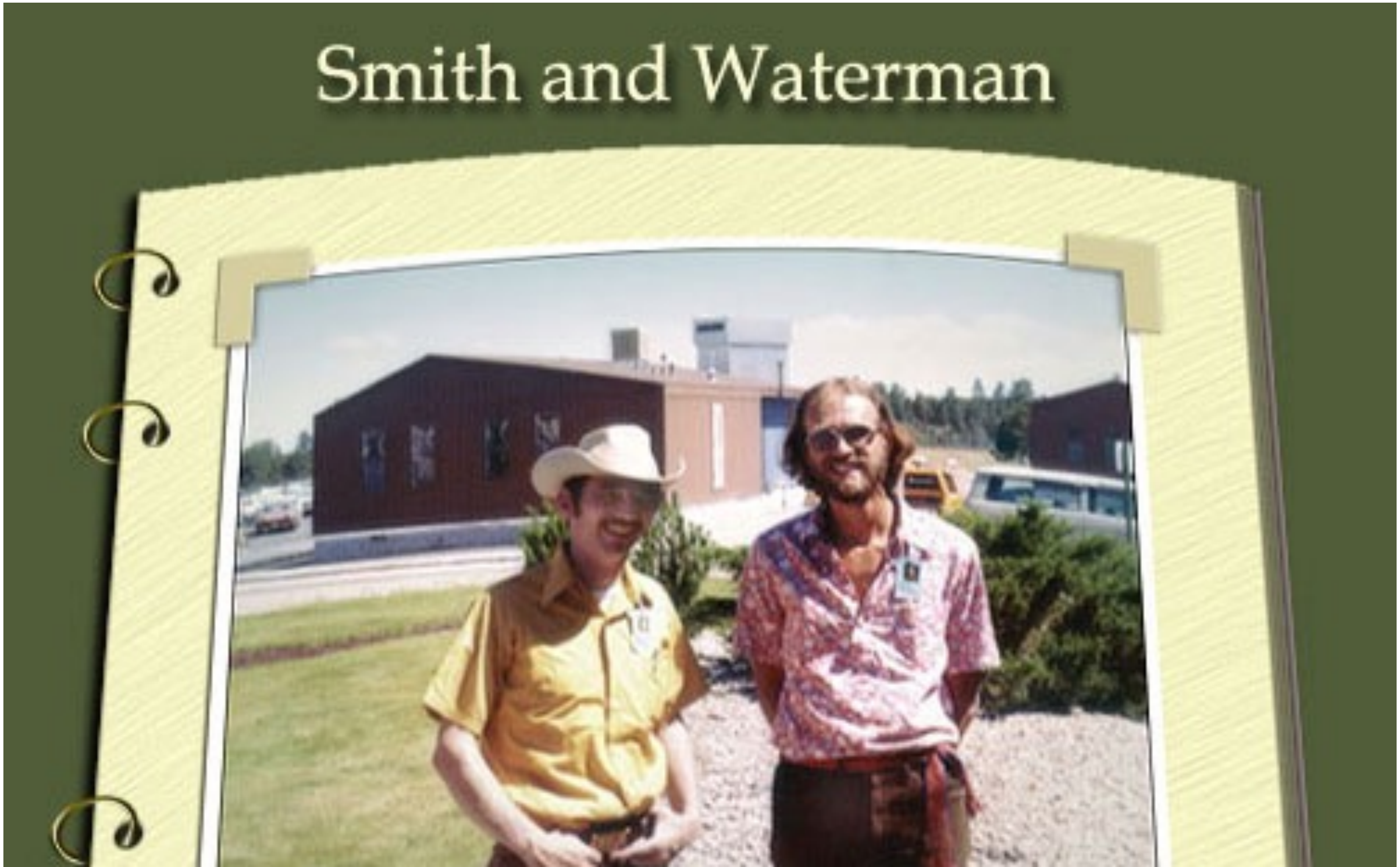
Gambler’s Ruin problem

In Sir Ronald Fisher we trust!

Dr. Margaret Oakley Dayhoff
The Mother & Father of Bioinformatics

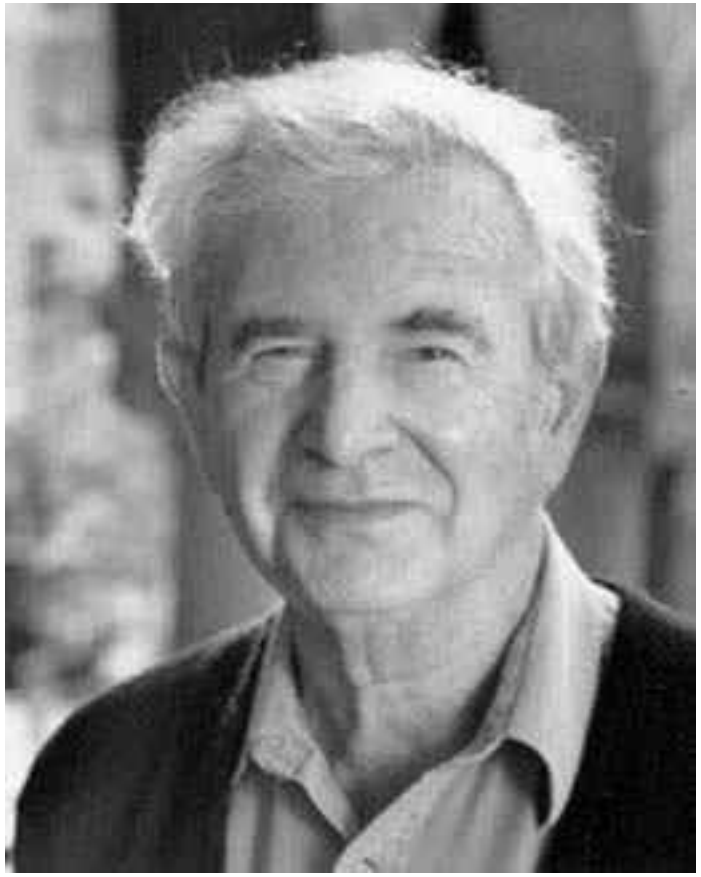
Smith and Waterman at Los Alamos, New Mexico

Photo by David Lipman, Taken Summer of 1980



Karlin-Astschul Statistics Theory

- Samuel Karlin and Stephen Altschul



Ch. 2: Genome Assembly and Haplotype Assembly Algorithms

Questions: What algorithms to use to assemble DNA pieces into a contigs?
How long are the contigs?
How much the DNA target region is covered by the contigs?

Examples of problems we need to solve along the way

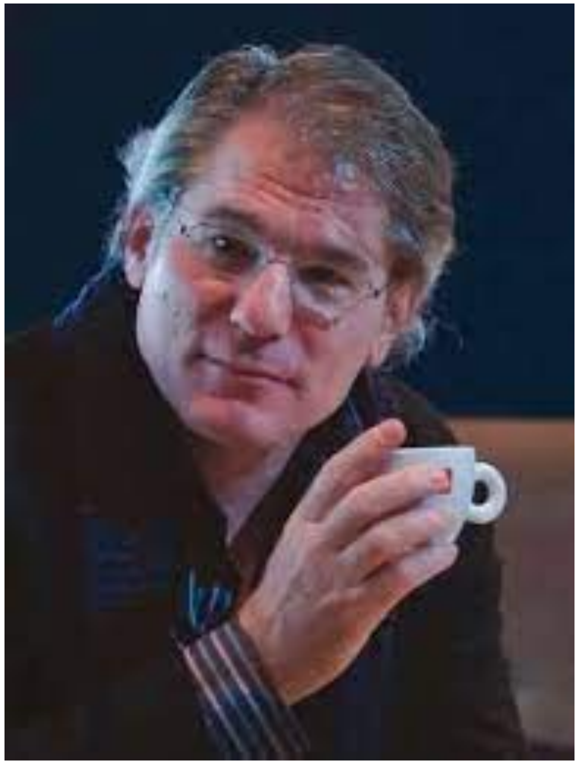
Problem 1. Poisson statistics and DNA and Assembly

Problem 2. Ham Smith's DNA breaking in a Lab with no windows

Hamiltonian Paths Algorithms for Genome Assembly

Gene Myers

Craig Venter



Eulerian Paths Algorithms for Genome Assembly

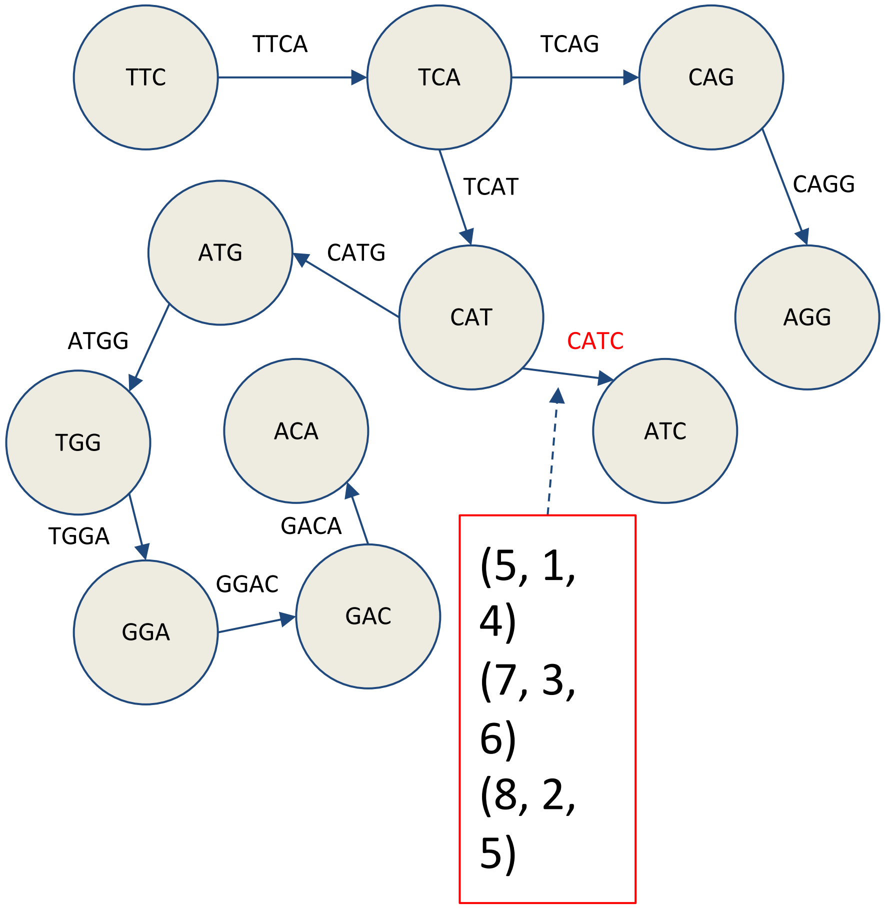
Pavel Pevzner

Michael Waterman

Construct the sequence graph on (k-1)-mers

f_1	TTCAGG
f_2	TTCATGG
f_3	ATGGACA
f_4	TTCAT
f_5	CATCGAC
f_6	TCGAC
f_7	GACATC
f_8	ACATCGA

TTCA
TCAG
CAGG
TCAT
CATG
ATGG
TGGA
GGAC
GACA
CATC
ATCG
TCGA
CGAC
ACAT



(5, 1, 4)
(7, 3, 6)
(8, 2, 5)

Construct the sequence graph on (k-1)-mers

f_1	TTCAGG
f_2	TTCATGG
f_3	ATGGACA
f_4	TTCAT
f_5	CATCGAC
f_6	TCGAC
f_7	GACATC
f_8	ACATCGA

TTCA
TCAG
CAGG
TCAT
CATG
ATGG
TGGA
GGAC
GACA
CATC
ATCG
TCGA
CGAC
ACAT

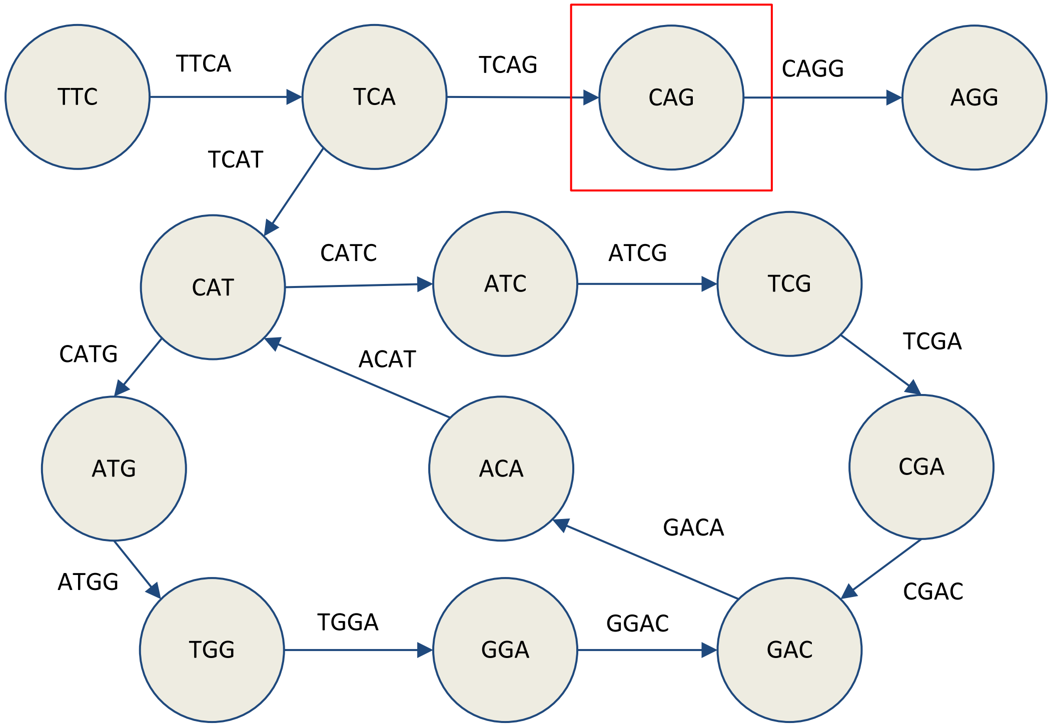
For each k-mer ($a_1 \ldots a_k$), we create an edge between nodes labeled $a_1 \ldots a_{k-1}$ and $a_2 \ldots a_k$.

If those nodes do not exist yet, we add them to the graph.

We label the edge by its k-mer, $a_1 \ldots a_k$.

We also store the set of position values (f, i, j) in each edge, which identify all occurrences of that k-mer by (fragment index, start position, end position)*

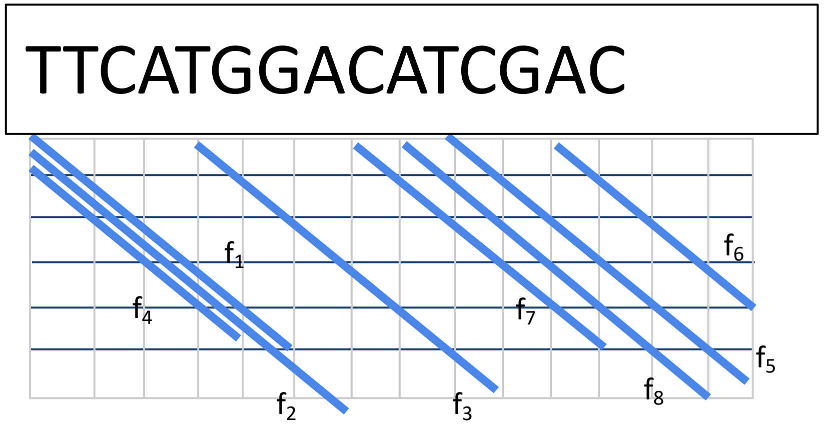
Graph reductions: singletons



Align the reads to the assembled sequence

f_1	TTCAGG
f_2	TTCATGG
f_3	ATGGACA
f_4	TTCAT
f_5	CATCGAC
f_6	TCGAC
f_7	GACATC
f_8	ACATCGA

TTCATGGACATCGAC



First, we apply hashing methods to identify where each fragment might align well to the sequence.

This will produce “candidate diagonals.”

We can then perform alignment along those diagonals, which is more efficient than using the entire edit graph.

Statistics of Sequence Graphs: vertices

$$\mathbb{E}(True) = L' \sum_{i=1}^{\infty} (1 - R^i)\mathbb{P}(X = i)$$

pmf of Poissor $\Pr(X{=}k) = \frac{\lambda^k e^{-\lambda}}{k!}$

$$= L' \sum_{i=1}^{\infty} \left(\frac{e^{-c} c^i}{i!} - \frac{e^{-c} (cR)^i}{i!} \right)$$

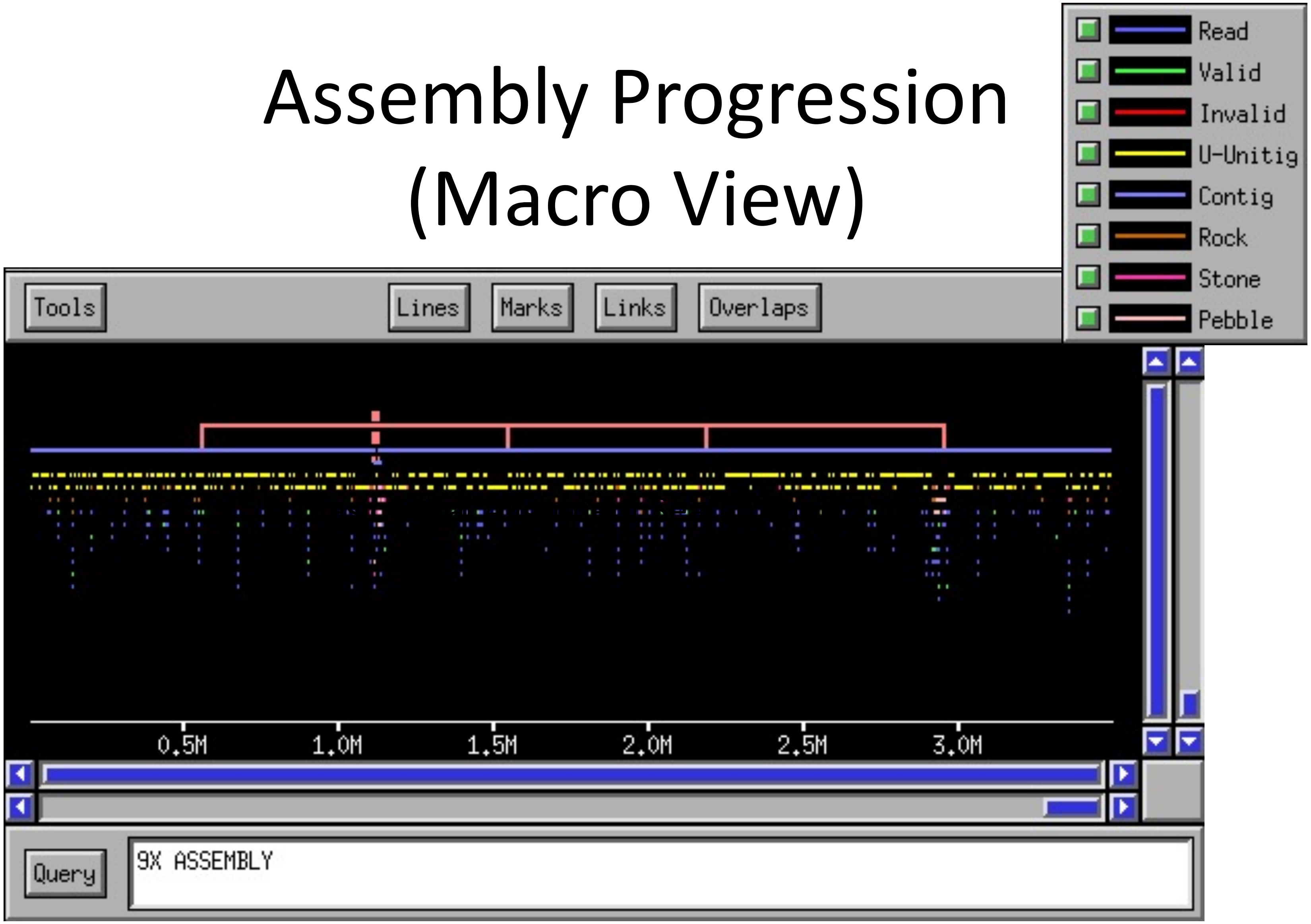
using Taylor expansion for e: $e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$

$$= L'(1 - e^{-c(1-R)}).$$

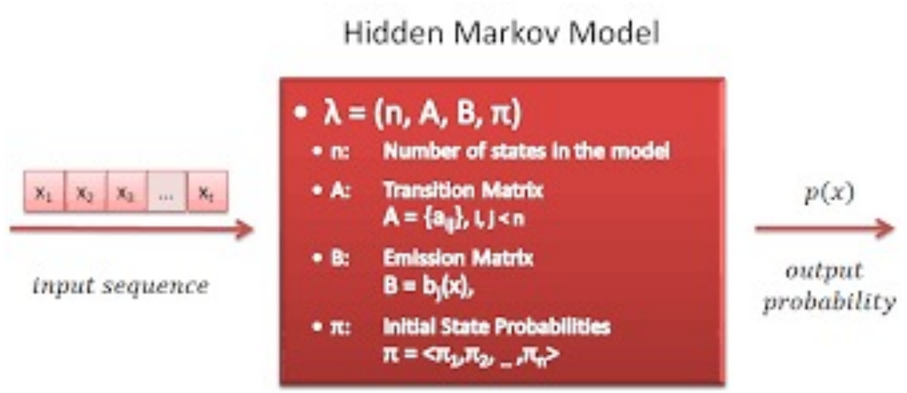
Summing the number of false vertices and true ve

The expected number of vertices $\mathbb{E}(|V|) = RT + [1 - e^{-c(1-R)}]L'$.

Assembly Progression (Macro View)



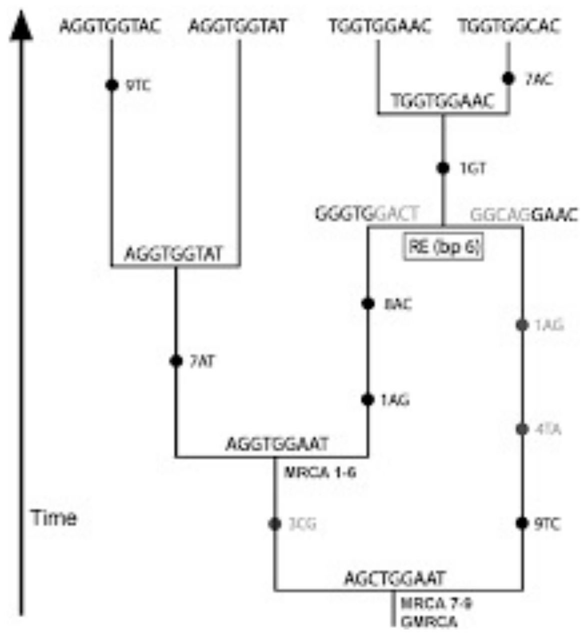
Ch. 3: HMM - the Learning Problem



What does machine learning an HMM model mean?

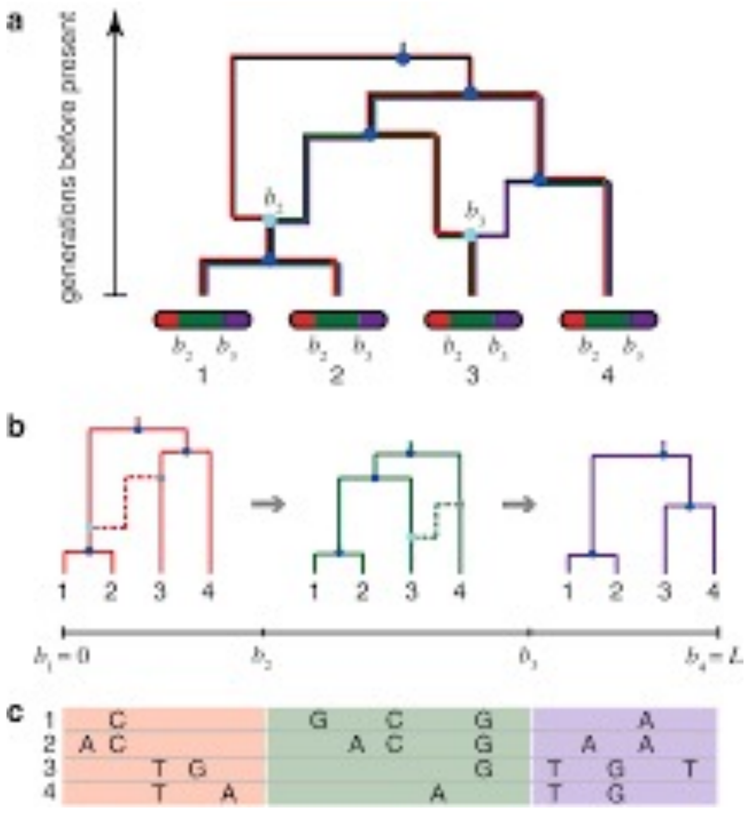
Maximum Likelihood and the Expectation-Maximization problem

Ch. 4 Recombination and Ancestral Recombination Graphs (ARG) Algorithms

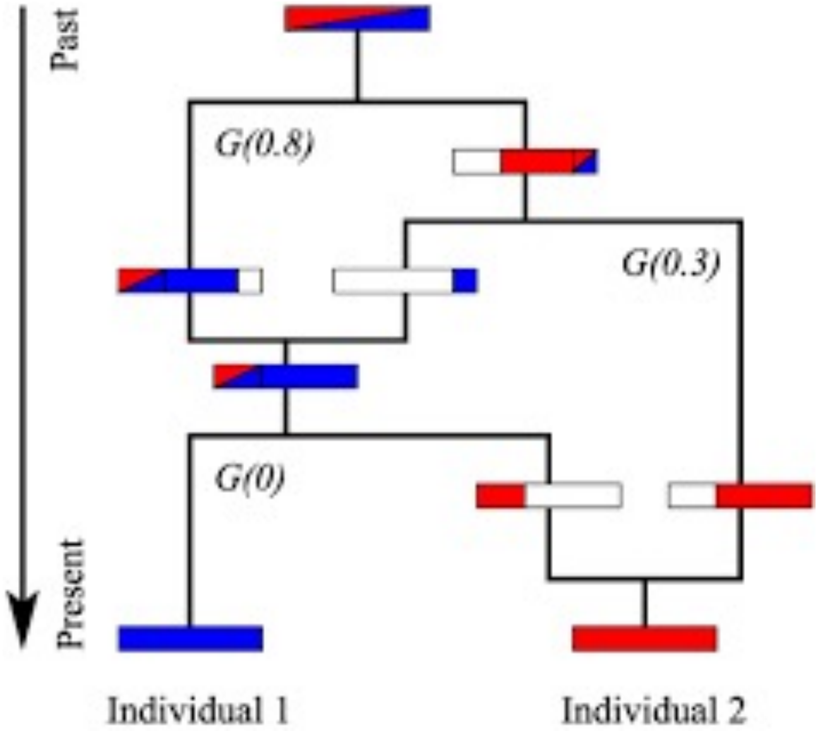


How do we reconstruct genealogies of a sample of individuals incorporating past mutations and recombinations?

Recombination + Phylogenetic Trees = ARG



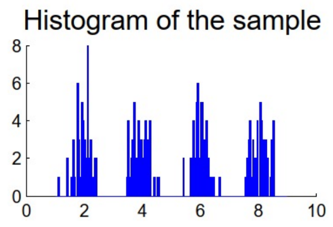
1	C	G C G	A
2	A C	A C G	A A
3	T G	G	T G T
4	T A	A	T G

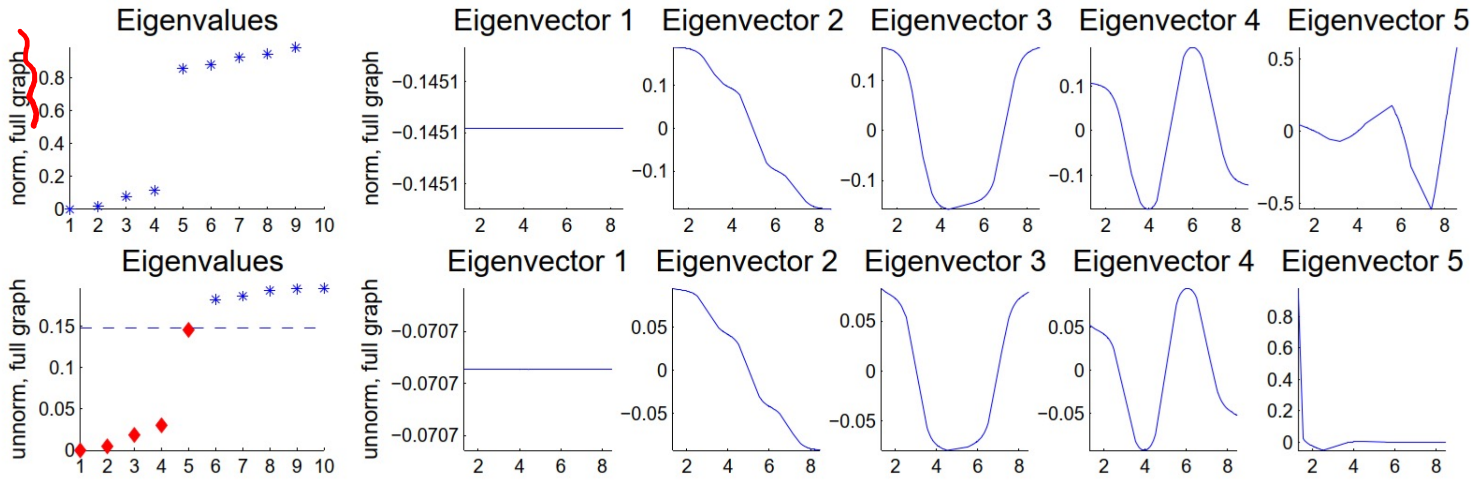


Ch. 3: Spectral Clustering

GRAPH LAPLACIANS

- Quick example

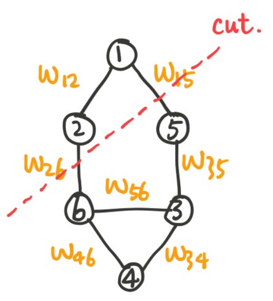




Sentences in red and graphs are cited from A Tutorial on Spectral Clustering (Ulrike von Luxburg). See reference list at the enfor detail.

GRAPH CUT POINT OF VIEW

Graph-based



$G = \{V, E\}$ $\rightarrow$ $\begin{cases} V = \bigcup_{k=1}^{k} A_k \\ A_i \cap A_j = \phi, \forall i,j \in \{1, \cdots, k\} \end{cases}$

$V = \{1, 2, 3, \cdots, N\}$

$E : W = [w_{ij}], 1 \le i,j \le N$

W: similarity matrix (affinity matrix).

$$w_{ij} = \begin{cases} K(x_i, x_j) : \exp\left\{-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right\}, & (i,j) \in E \\ 0 & (i,j) \notin E \end{cases}$$

$A \subset V,\ B \subset V,\ A \cap B = \phi$

$$W(A, B) = \sum_{\substack{i \in A \\ j \in B}} w_{ij}$$

Sentences in red and graphs are cited from A Tutorial on Spectral Clustering (Ulrike von Luxburg). See reference list at the enfor detail.

RANDOM WALK POINT OF VIEW

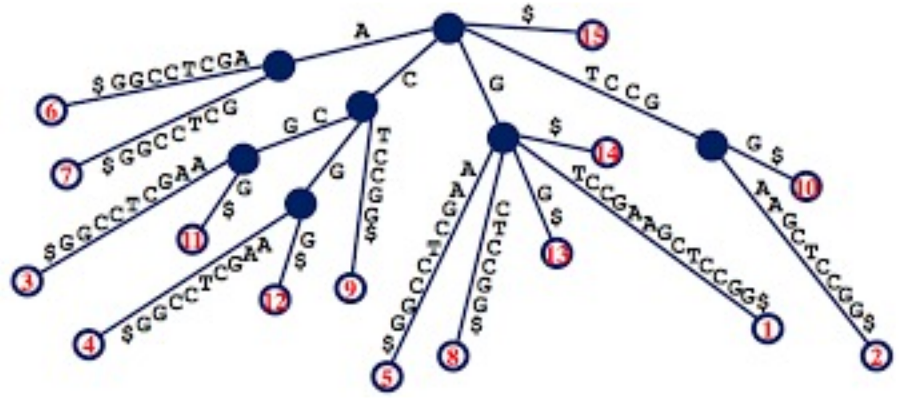
- What is random walk?
  - A random walk on a graph is a stochastic process which randomly jumps from vertex to vertex.
- How does it walk?
  - Formally, the transition probability of jumping in one step from vertex **vi** to vertex **vj** is proportional to the edge weight **wij** and is given by **pij := wij/di.**
  - The transition matrix P = (pij)i,j=1,...,n of the random walk is thus defined by

$$P = D^{-1}W.$$

- Initial condition?
  - a unique stationary distribution **π = (π1, . . . , πn)′** , where **πi = di/ vol(V).**
- Clustering in random walk?
  - Finding a partition of the graph, such that the random walk stays along within the same cluster and seldom jumps between clusters.
  - Intuitively, it is the same as the graph cut.

Sentences in red and graphs are cited from A Tutorial on Spectral Clustering (Ulrike von Luxburg). See reference list at the enfor detail.

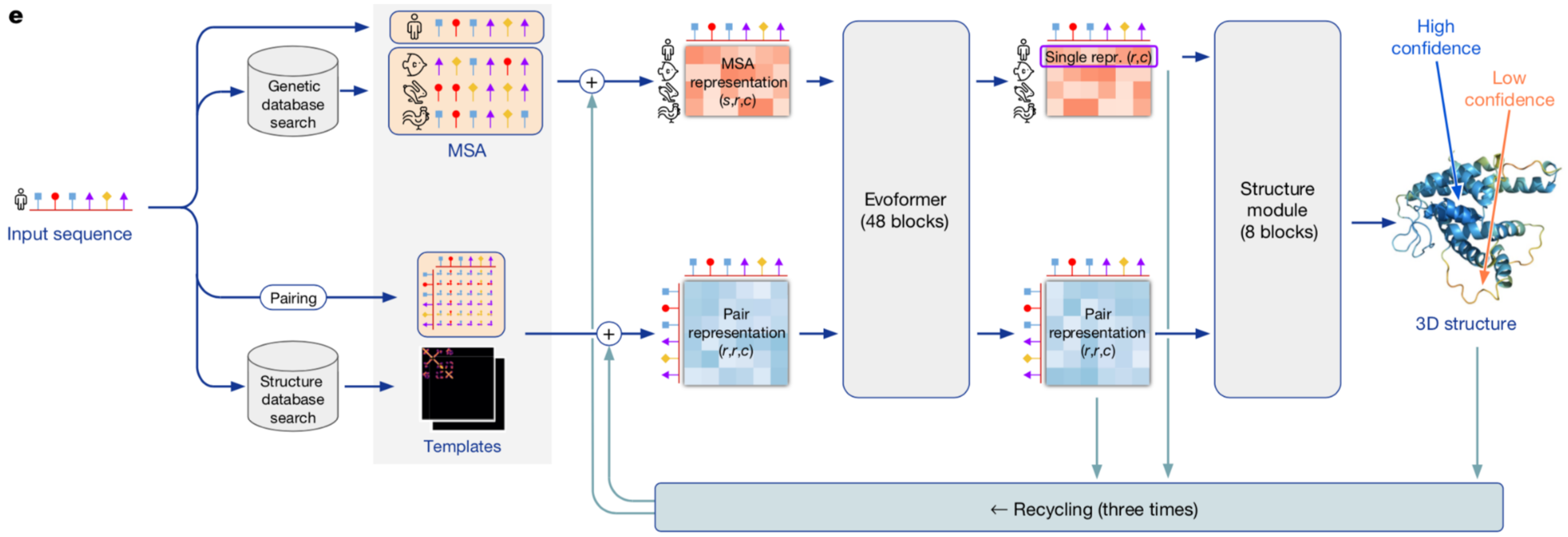
Ch. 6 Suffix Trees in Linear Time



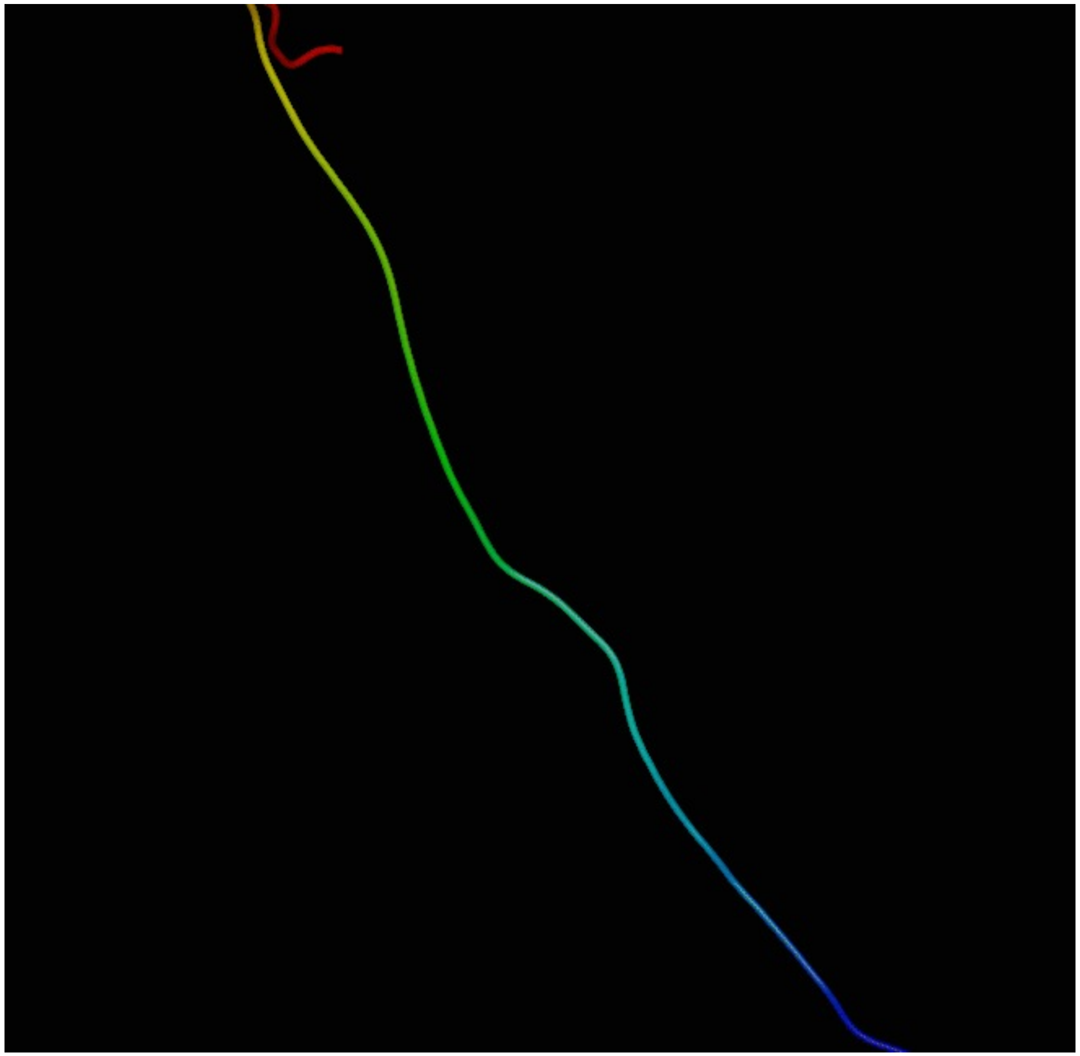
Ch. 7 Protein Folding Algorithms (Intro)

- Protein Folding on Lattice Models
- AlphaFold and Deep Learning

High-level Overview of Architecture of AlphaFold


e
Input sequence
Genetic database search
MSA
Pairing
Structure database search
Templates
MSA representation (s,r,c)
Pair representation (r,r,c)
Evoformer (48 blocks)
Single repr. (r,c)
Pair representation (r,r,c)
Structure module (8 blocks)
High confidence
Low confidence
3D structure
← Recycling (three times)

Deep learning uses sequential modules (layers) to progressively extract information (learn) from the input data.



The Protein Folding Problem

Statistical Mechanics
models

Mixed character of the problem :

continuous mathematics -- geometry of surfaces &
discrete mathematics -- combinatorics of folds

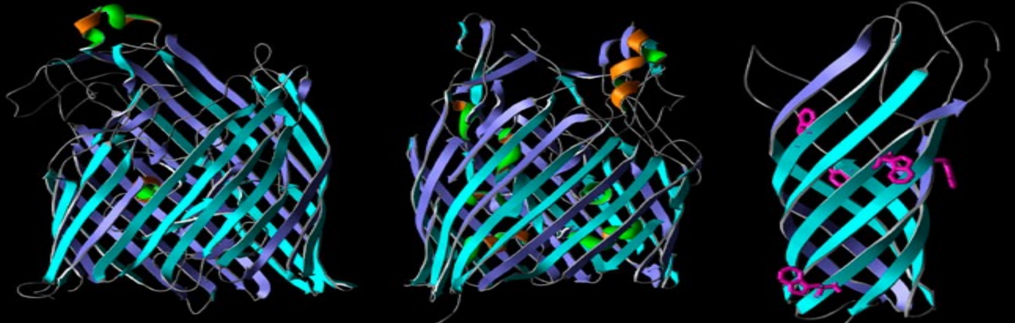

FhuA (Ferguson et al., 1998) FepA (Buchanan et al., 1999) OmpA (Pautsch & Schulz, 1998)

Illustration © 1999 JHK