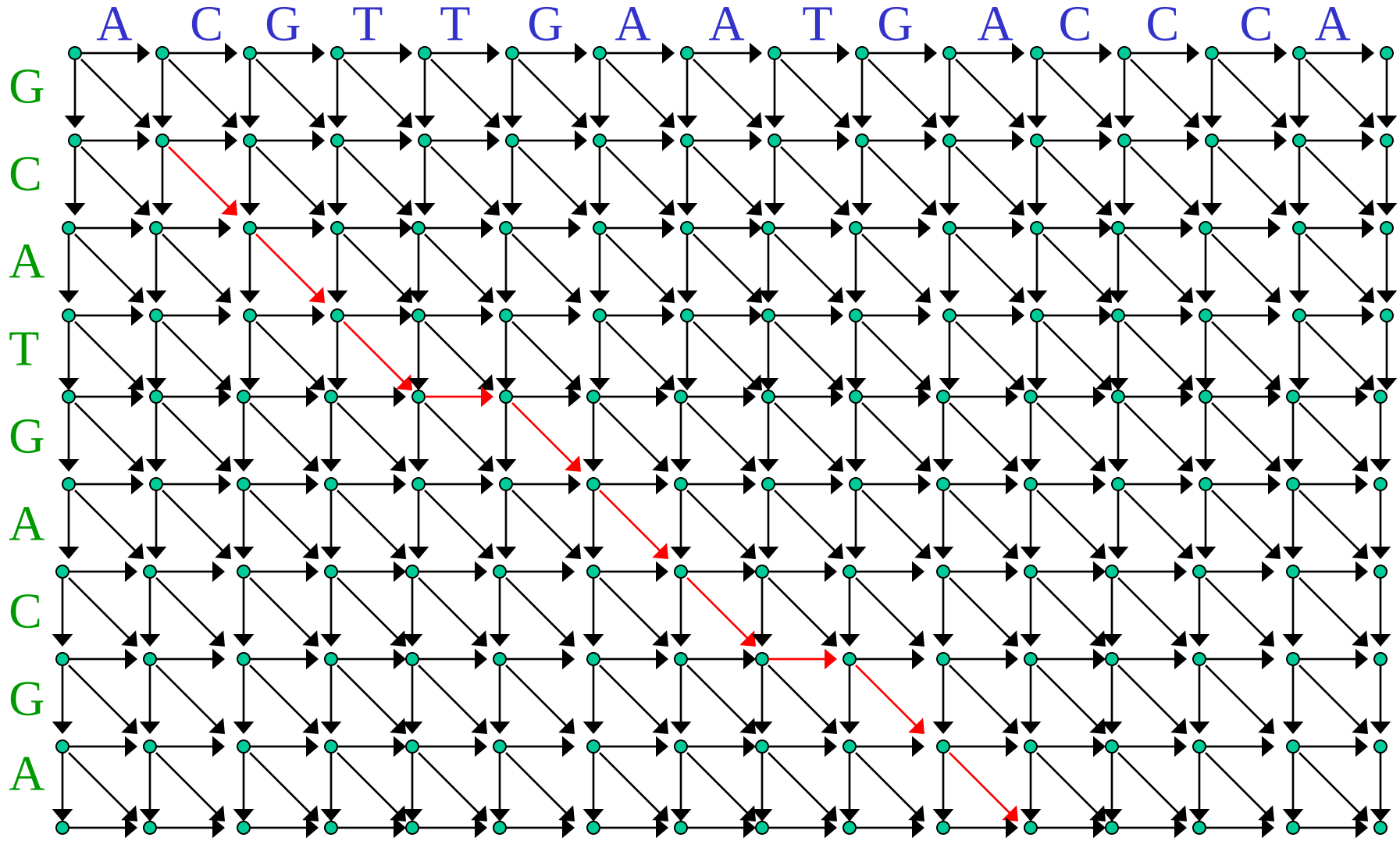
Lecture 9

- Improved scoring
  - Affine gap penalties
  - Profiles
- Statistical significance
- Reducing time
  - Word nucleation algorithms



Above path corresponds to following alignment (w/ lower case letters considered unaligned):

```
aCGTTGAATGAccca
gCAT-GAC-GA
```

BLOSUM62 Score Matrix

GAP -12 -2

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0	-2	-1	0	-4
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3	-1	0	-1	-4
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3	3	0	-1	-4
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3	4	1	-1	-4
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1	-3	-3	-2	-4
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2	0	3	-1	-4
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2	1	4	-1	-4
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3	-1	-2	-1	-4
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3	0	0	-1	-4
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3	-3	-3	-1	-4
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1	-4	-3	-1	-4
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2	0	1	-1	-4
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1	-3	-1	-1	-4
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1	-3	-3	-1	-4
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2	-2	-1	-2	-4
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2	0	0	0	-4
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0	-1	-1	0	-4
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3	-4	-3	-2	-4
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1	-3	-2	-1	-4
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4	-3	-2	-1	-4
B	-2	-1	3	4	-3	0	1	-1	0	-3	-4	0	-3	-3	-2	0	-1	-4	-3	-3	4	1	-1	-4
Z	-1	0	0	1	-3	3	4	-2	0	-3	-3	1	-1	-3	-1	0	-1	-3	-2	-2	1	4	-1	-4
X	0	-1	-1	-1	-2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-2	0	0	-2	-1	-1	-1	-1	-1	-4
*	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	1

Better Alignment Scoring

- Optimal alignment scoring depends on probabilistic modelling (e.g. LLR scores)

- Limitations of our current approach:
  1. each alignment column (edge in WDAG) is scored independently
     → an independence assumption for probability model
  2. Score depends only on the residues that are present (via a BLOSUM-type score matrix) – i.e. independently of position within sequence

- Ways to allow ***partial*** non-independence while preserving dynamic programming framework:

1. Enhance graph
   - Allows ‘memory’ of preceding columns

2. Allow scores to depend on position within the sequence
   - so some substitutions (of same residues) or gaps penalized more heavily than others
   - like a site model!

Gap Penalties

```
TNAVAHVD-----DMPNAL
YEAAIQLQVTGVVVTDATL
```

- Usual scoring scheme assigns same penalty g to each gap edge, so
  - weights on extended gaps of size s are *linear* in s, i.e.
  - total gap penalty $gap(s) = s \times g$.
  - e.g. in above example, if each g = -6, total penalty on gap would be

$$gap(5) = 5 \times -6 = -30$$

- Would like more flexible gap penalties:
- In proteins, insertions & deletions are rare;
  - but when occur, often consist of several residues, because
    - they are in regions (loops) tolerant of length changes
  - at DNA level, indels in protein coding sequence usually a multiple of 3 nucleotides
    - otherwise, would change reading frame
- In noncoding DNA sequence,
  - the most common indel size is 1
  - *but* larger indels occur much more frequently than multiple independent single-base indels

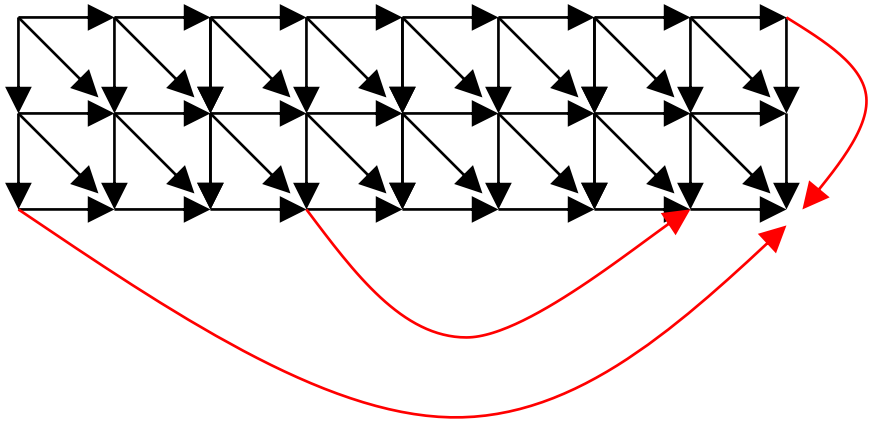
- Can allow arbitrary *convex* gap penalties
  - $gap(s+t) \geq gap(s) + gap(t)$, where s and t are (integer) gap sizes

by extending edit graph:

- add edges corresponding to *arbitrary length* gaps from each vertex to each horizontally or vertically downstream vertex
- (convexity condition prevents favoring two adjacent short gaps over a single long gap).

Time complexity now $O(MN(M+N))$

- often unacceptable for moderate M, N.
- Also: how to choose appropriate weights? (need data to estimate!)



Affine Gap Penalties

- *Affine* gap penalties:
  - less general than arbitrary convex penalties, but
  - more general than linear penalties.
- Two parameters:
  - *gap opening* penalty g_o
  - *gap extension* penalty g_e
- *gap*(*n*) (penalty for size *n* gap) is then

$$g_o + n\, g_e \;\; = g_i + (n - 1)\, g_e$$

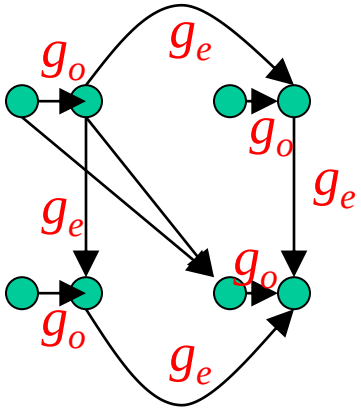
where the gap *initiating* penalty $g_i = g_o + g_e$

- Example: for BLOSUM62, good penalties are
  - g_i = -12,
  - g_e = -2

  These perform *much* better than linear penalty
  - (e.g. g = -6)
- N.B. Durbin *et al.* reverse g_i and g_o
  - g_i is called the 'gap opening' penalty
- Can obtain affine penalties using extension of edit graph, retaining complexity $O(MN)$:

Edit Graph for Affine Gap Penalties

Double # vertices, creating left-right pair in place of each original vertex. Each cell looks like this:



each left vertex has out-degree and in-degree = 2

each right vertex has out-degree and in-degree = 3

• gap-opening edges from left vertex to right vertex of each pair : weight g_o

• gap extension edges going horizontally or vertically between right vertices : weight g_e

• diagonal edges originate from either left or right vertex, but always go to a left vertex.

- Paths in the augmented graph still correspond to alignments
  - can ∃ more than one path for same alignment
  - but highest scoring paths still give best alignments
- Score assigned to size n gap is $g_o + n\, g_e$
  - *i.e.* affine penalty
- ‘Smith-Waterman-Gotoh algorithm’

Finding values for gap penalties

- Direct definition as LLR seems problematic
  - what are ‘random alignments’?
- *Empirical approach*: Given a score matrix (e.g. BLOSUM62), for various (g_o , g_e) choices
  - Align real sequences to known homologues & simulated sequences
  - Measure score discrimination (E-values of homologue alignments)
  - Find (g_o , g_e) giving best discrimination

Profiles (position-specific scoring)

- Different parts of sequence may evolve at different rates
- In proteins
  - conserved functional motifs
  - structural constraints:
    - internal core region of tightly packed residues, or active sites of enzyme, are more highly conserved;
    - surface residues, particularly in loops, often less conserved.

Conserved Domain in RecR and Class I Topisomerases

```
RecR   RLAEEKITEVILATNPTVEGEATANYIAELC
RecM   RLQDDQVTEVILATNPNIEGEATAMYISRLL
RecR   RVDDVGITEVIIATDPNTEGEATATYLVRMV
TrsI   IFKENKIDEVIIATDPARGGENIAYKILNQL
TOP1   KQLAEKADHIYLATDLDREGEAIAWRLREVI
ORF1   AELLKQANTIIVATDSDREGENIAWSIIHKA
TOP1   KDALKDADELILATDEDREGKVISWHLLQLL
TOP1   TIFDKRVKTIILATDAAAEGEYIGRNILYRL
TOP3   KREARNADYLMIWTDCDREGEYIGWEIWQEA
TOP3   KRFLHEASEIVHAGDPDREGQLLVDEVLDYL
RGYR   RNLAVEADEVLIGTDPDTEGEKIAWDLYLAL
```

CONSENSUS **xxxxxxxxxxU&uatDxxxEGexxxxxUxxxu**

Consensus key:

Uppercase: all residues chemically similar

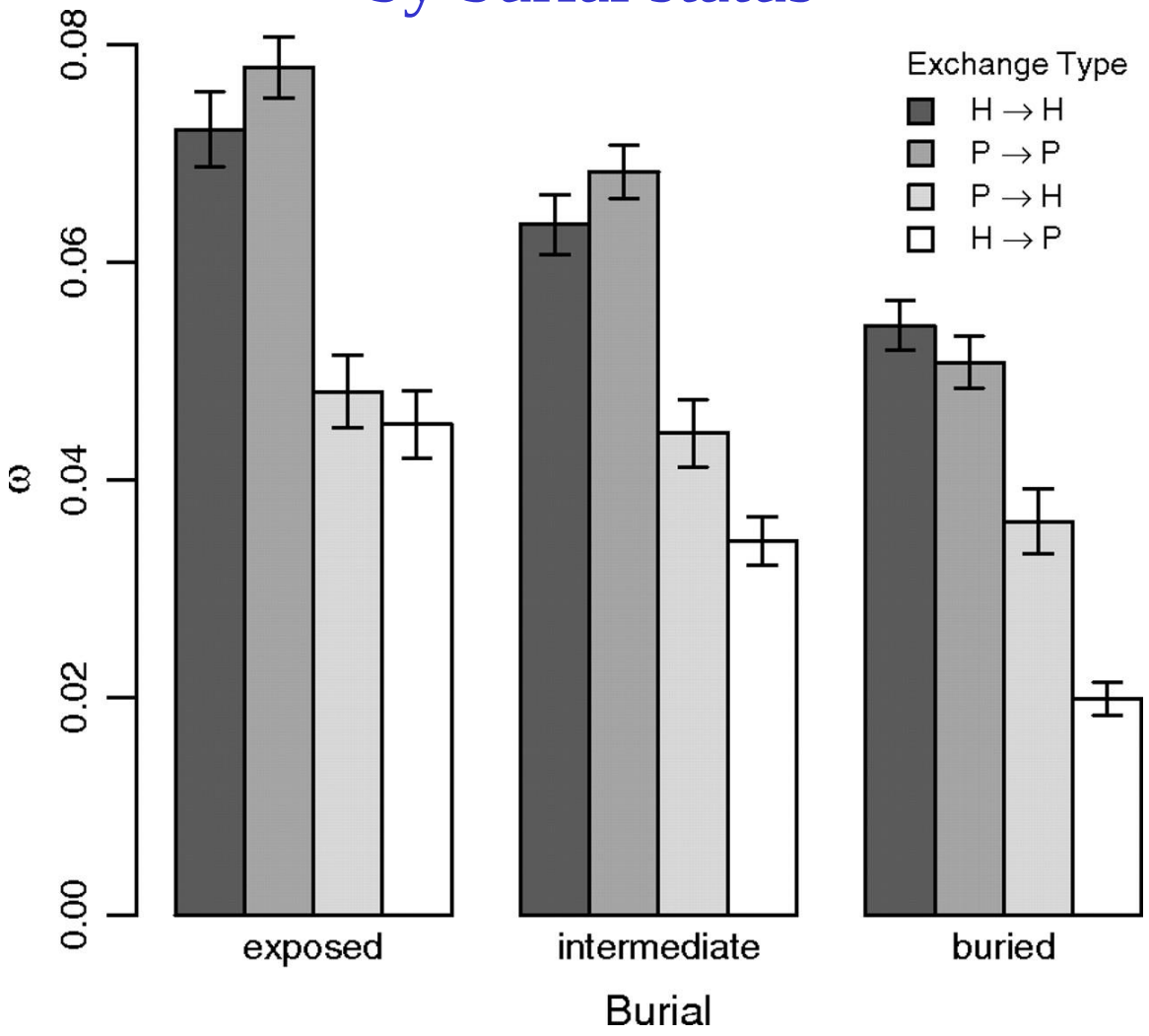
lowercase: most are

U,u: bulky aliphatic (I,L,V)

&: bulky hydrophobic (I,L,V,M,F,Y,W)

From RL Tatusov, SF Altschul, and EV Koonin, PNAS 91: 12091-12095

Rates of amino acid exchange in mammalian proteins by burial status



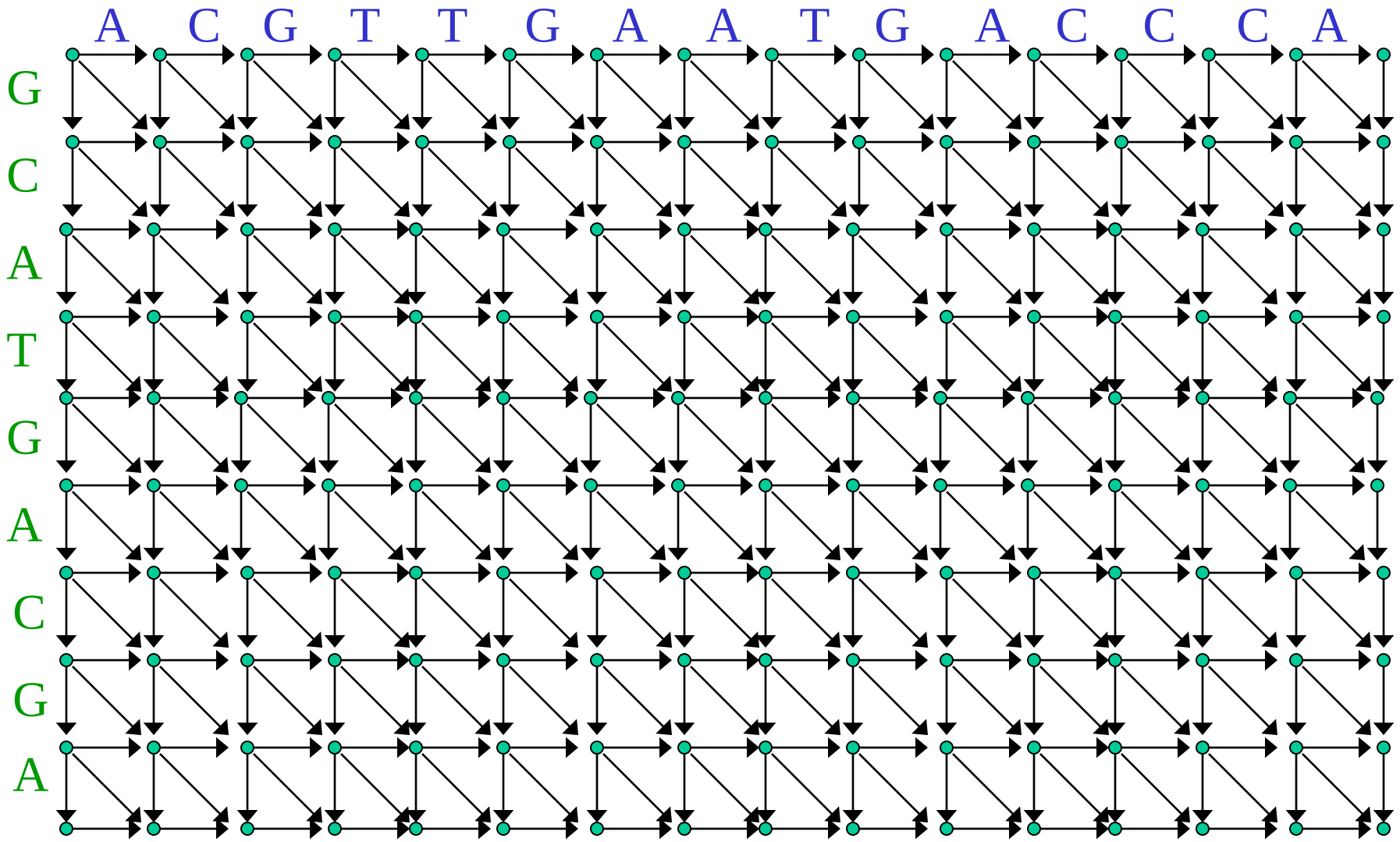
H: hydrophobic

P: polar

Saunders & Green Mol Biol Evol 2007 24:2632-2647; doi:10.1093/molbev/msm190

Molecular Biology and Evolution

The *Edit Graph* for a Pair of Sequences



- *Profiles*: *Position-specific* scoring scheme specifying score of each possible substitution at each position of a sequence

C →

Cons	A	C	D	E	...	T	V	W	Y	Open	Ext
G	7	-14	-1	-5	...	6	4	-34	-22	28	28
P	5	-26	4	1	...	1	-4	-48	-31	28	28
L	-18	-31	-40	-35	...	-16	13	-31	-9	100	100
T	7	-21	-4	-6	...	10	-3	-38	-28	100	100
E	6	-37	11	12	...	2	-10	-61	-38	100	100
A	5	-34	3	4	...	1	-8	-48	-34	100	100
E	0	-53	26	31	...	-5	-29	-60	-42	100	100
R	-11	-45	-11	-13	...	-3	-21	-2	-33	100	100
T	4	-28	-2	-1	...	8	7	-51	-24	100	100
M	-7	-47	-6	-6	...	-3	-6	-35	-26	100	100
V	0	-20	-22	-36	...	2	41	-56	-27	100	100
K	-9	-44	-11	-11	...	0	-5	-29	-31	100	100
N	5	-27	7	6	...	8	-11	-40	-32	100	100
A	7	-27	-4	-6	...	4	5	-46	-31	100	100
W	-47	-69	-58	-60	...	-40	-49	139	-6	100	100
G	11	-31	5	1	...	3	-5	-65	-43	100	100
K	-2	-46	5	8	...	-1	-23	-49	-45	100	100
V	-4	-23	-27	-45	...	-2	34	-48	-18	100	100
L	-3	-9	-6	-5	...	-3	3	-3	-1	26	26
N	-4	-26	3	2	...	-4	-19	-31	-9	26	26
A	4	-16	0	1	...	2	-12	-40	-10	26	26
H	0	-30	14	10	...	3	-15	-41	-21	100	100
I	-2	-20	-18	-23	...	-1	17	-50	-11	100	100
....											

***From* R. Luthy, I. Xenarios and P. Bucher, Improving the sensitivity of the sequence profile method *Protein Sci.* 3: 139-146 (1994)**

- The *scores* are ***position-specific*** LLRs:
- *Instead of*

  $M(r, s) = \log_a(h_{r,s} / b_{r,s})$ where

  $h_{r,s}$ = freq of $\begin{matrix} r \\ s \end{matrix}$ in homologous seq alignments

  $b_{r,s}$ = freq of $\begin{matrix} r \\ s \end{matrix}$ in 'background' (random) alignments
- *take, for i-th row* (*with residue* r_i)
  - $M_i(s) = \log_a(h_{i,s} / b_{i,s})$ where

    $h_{i,s}$ = freq of s aligned to r_i in homologue alignments

    $b_{i,s}$ = freq of s in random alignments

- PSIBLAST approach:
  1. initially compare query sequence to database sequences (using BLOSUM-type scoring matrix),
  2. build profile using matches
  3. rescan database using profile
  4. iterate 2 & 3 until …

Karlin / Altschul for sequence alignments

- For LLR-based alignment scoring
  - *i.e.* $s(r) = \log_a(t_r / b_r)$, where r is an alignment column,

the expected # local alignments of score $\geq S$ for (random) seqs of length M , N is

$$\approx MNK\, a^{-S}$$

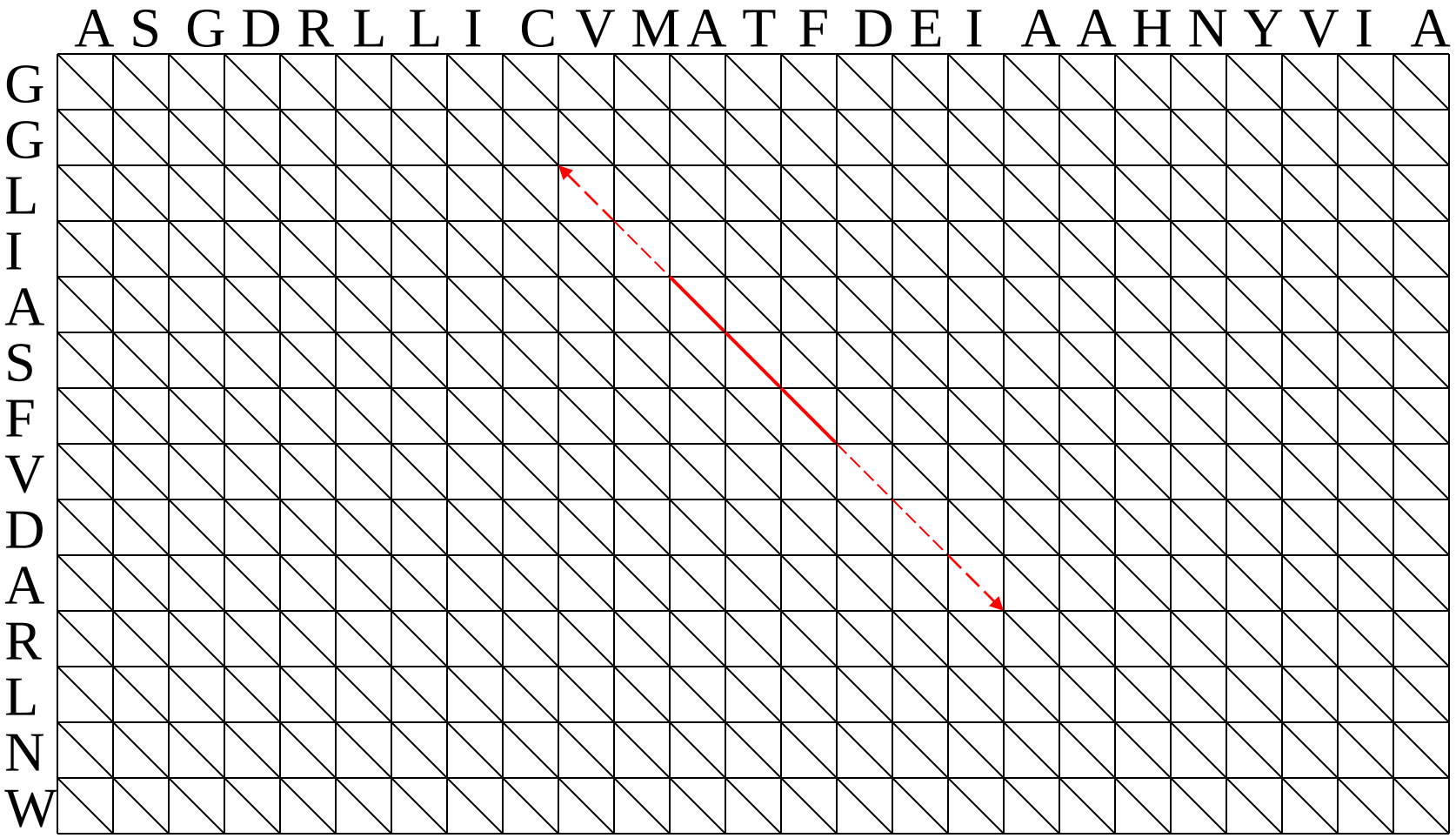
for some constant K (not depending on S)

- Note that $a^{-S} = a^{-LLR} = 1 / LR$
- K-A developed theory for *ungapped* alignments, but empirical studies suggest it applies more broadly
  - Estimate K from alignments to random sequence

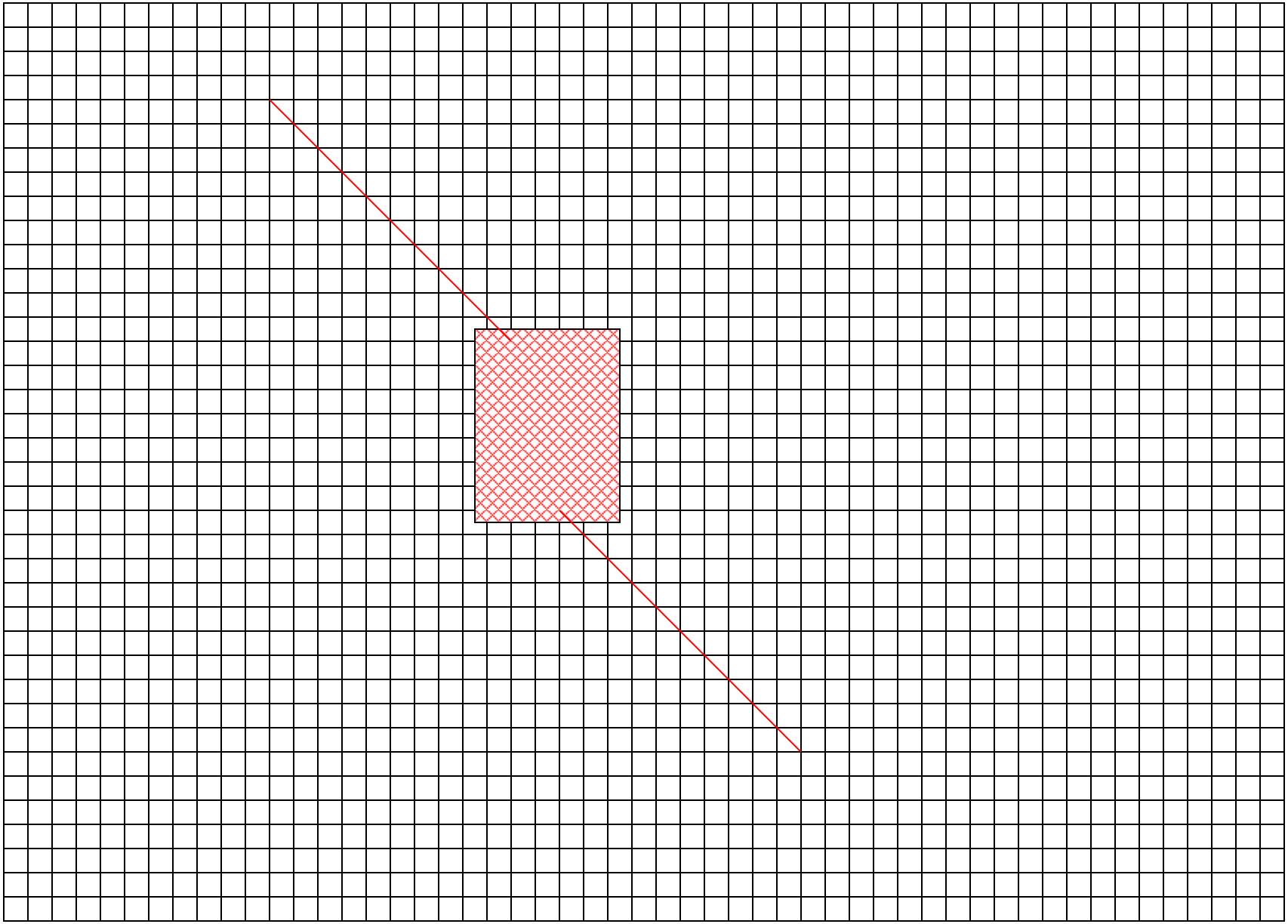
Word Nucleation Algorithms

- Idea: find short (perfect or imperfect) word matches to 'nucleate' graph search
  - Each such match defines short *diagonal* path
  - Only search part of graph 'surrounding' this path
- BLAST: allow *imperfect* short (e.g. length 3) matches.
  - "*Neighbors*": set of 3-residue sequences having ≥ min score T against some 3-residue sequence of query
  - Scan database seqs until hit word in neighbor list
  - then do ungapped extension (along diagonal defined by word match)
    - 'significant' matches are those with scores ≥ a threshold S
    - Ungapped matches are effective for detecting related proteins:
      - **true protein alignments usually include substantial gap-free regions.**

BLAST: Word Nucleating Alignment



– If find ≥ 2 significant ungapped matches in same seq, expand search to connecting region of matrix, allowing gaps:



Other Word Nucleation Programs

- FASTA:
  - look for clusters of short exact matches, on nearby diagonals;
  - when found, extend to gapped alignment
- *cross_match*:
  - do full search of *bands* around exact matches
- These all still time complexity $O(MN)$
  - because # word matches proportional to MN

but with much smaller constant.

- In database searches, most seqs unrelated to query
- suggests following strategy:
  - Initial rapid pass through database using fast algorithm
    - e.g. just looking for gap-free matches

    to get (approximate) score,
  - identify sequences having scores above a threshold
  - use full Smith-Waterman on latter
  - for appropriate (low) threshold can get sensitivity nearly as good as full Smith-Waterman search.