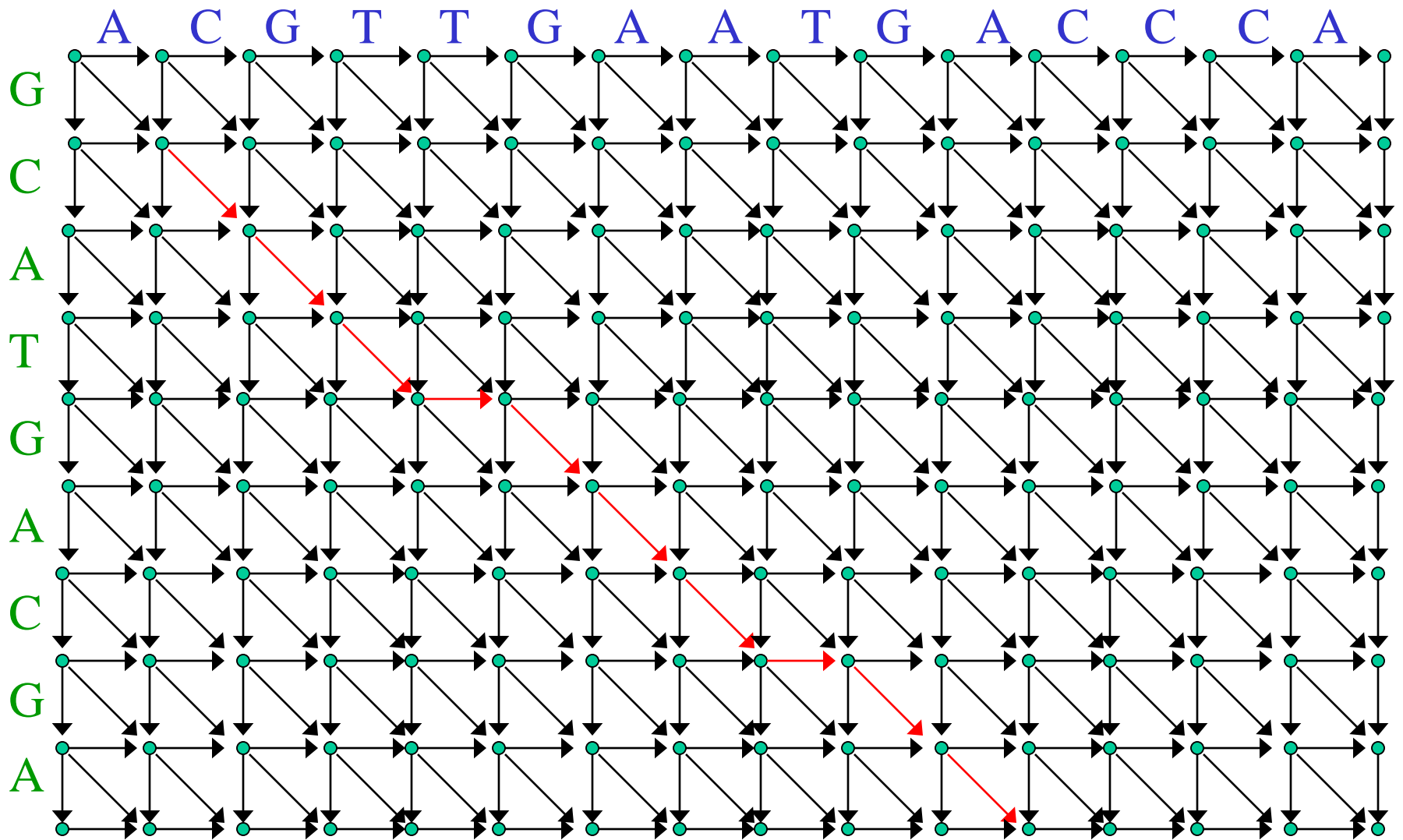


Today's Lecture

- Improved scoring of pairwise alignments
 - Affine gap penalties
 - Profiles
- Smith-Waterman special cases



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

aCGTTGAATGAccca
gCAT-GAC-GA

- *Two strategies to allow* allow partial non-independence while preserving dynamic programming framework:
 - Enhance graph
 - Allow scores to depend on position within the sequence (i.e. *not* just on a BLOSUM-type score matrix)
 - so some substitutions (of same residues) or gaps penalized more heavily than others

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

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

Gap Penalties

- 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 sequence,
 - the most common indel size is 1
 - *but* larger indels occur much more frequently than multiple independent single-base indels

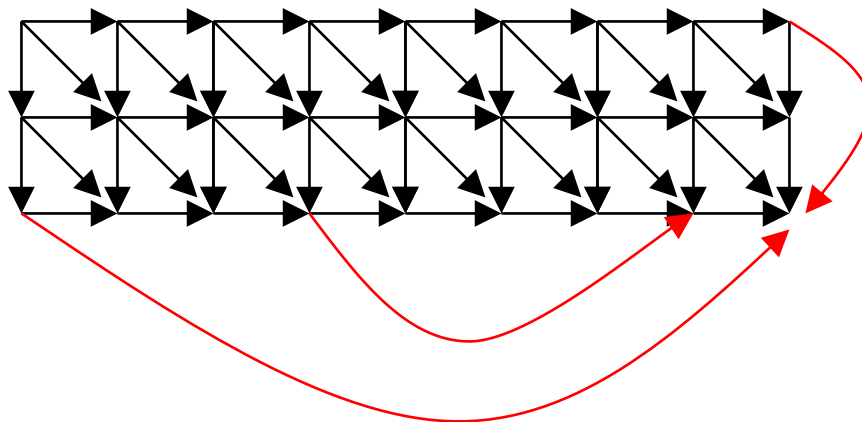
- Can allow arbitrary *convex* gap penalties
 - $gap(s+t) \geq gap(s) + gap(t)$, where s and t are (integer) gap sizes

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 $g_i = g_o + g_e$

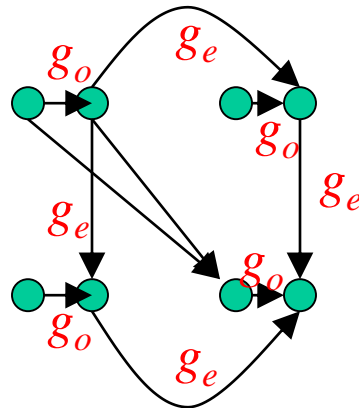
- Example: for BLOSUM62, good penalties are
 - $g_i = -12$,
 - $g_e = -2$

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*

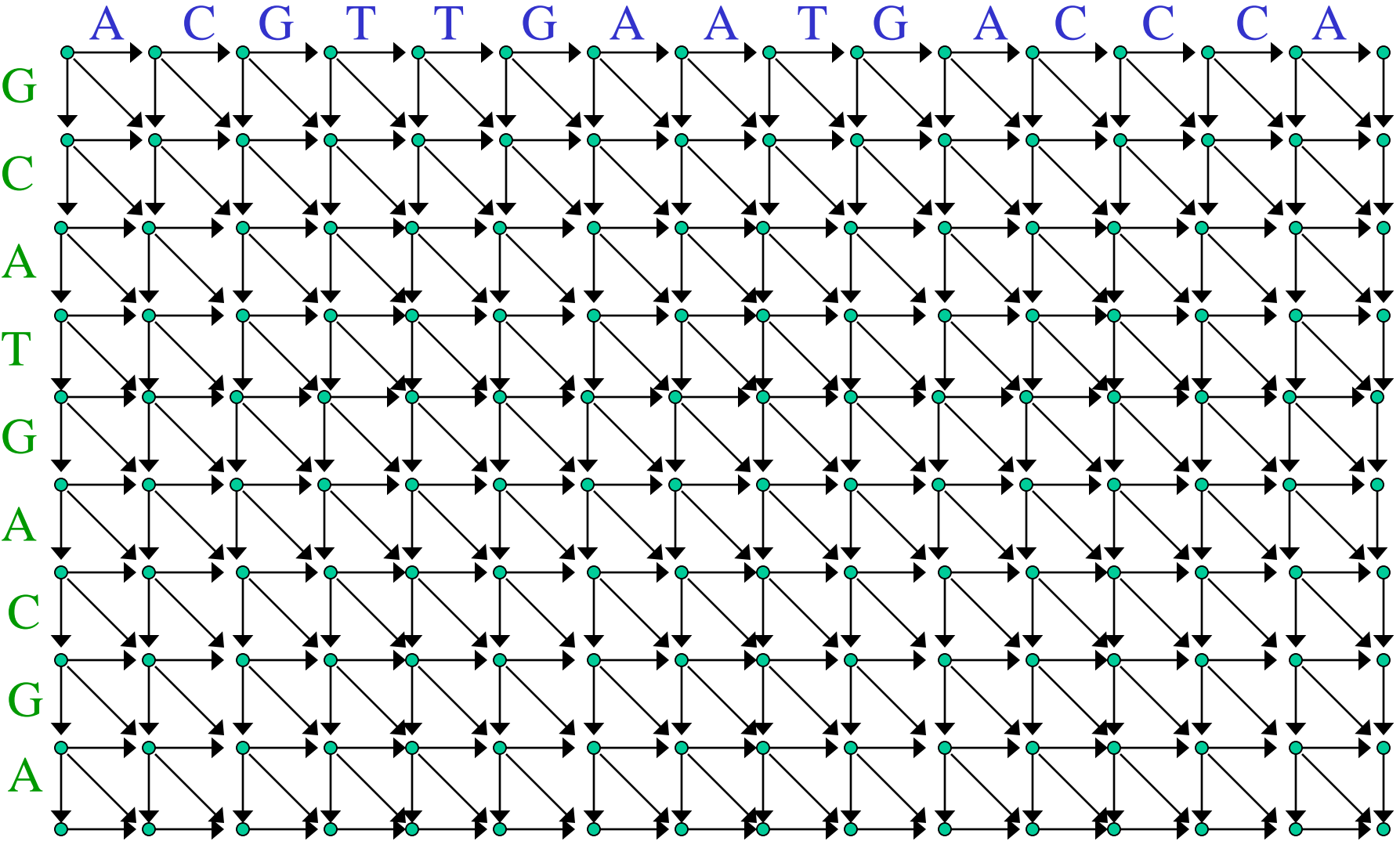
*each right vertex has out-degree
and in-degree = 3*

- 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 \exists 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

Profiles (position-specific scoring)

The *Edit Graph* for a Pair of Sequences



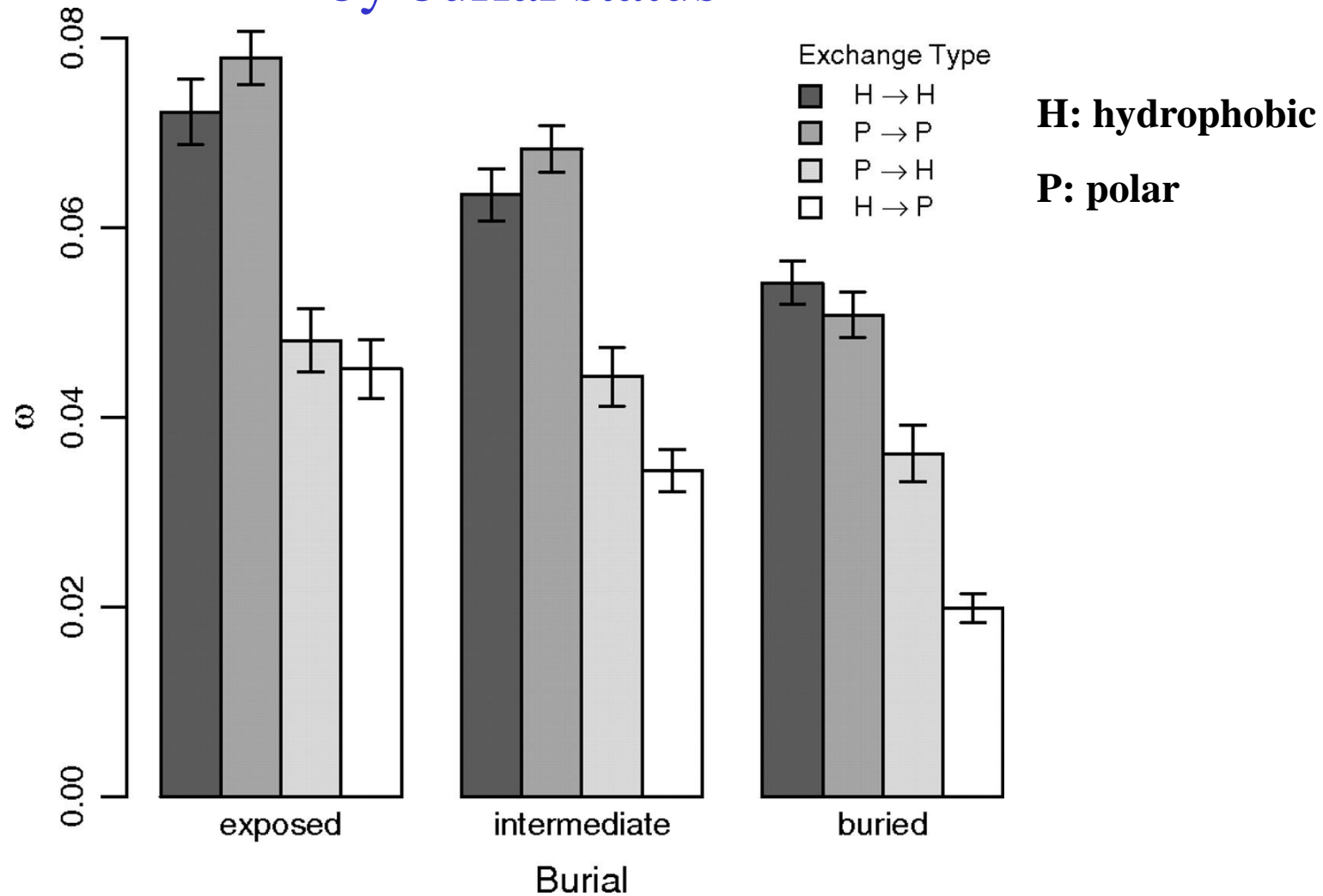
- *Profiles: Position-specific* scoring scheme specifying score of each possible substitution at each position of a sequence

	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)

- This is an important improvement!
 - reflects fact that different parts of sequence may evolve at different rates
- e.g. in proteins,
 - internal core region of tightly packed residues, or active sites of enzyme, are more highly conserved;
 - surface residues, particularly in loops, often less conserved.
 - so scores tend to be correlated (high scores in core, lower on surface)

Rates of amino acid exchange in mammalian proteins by burial status



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

Molecular Biology
and Evolution

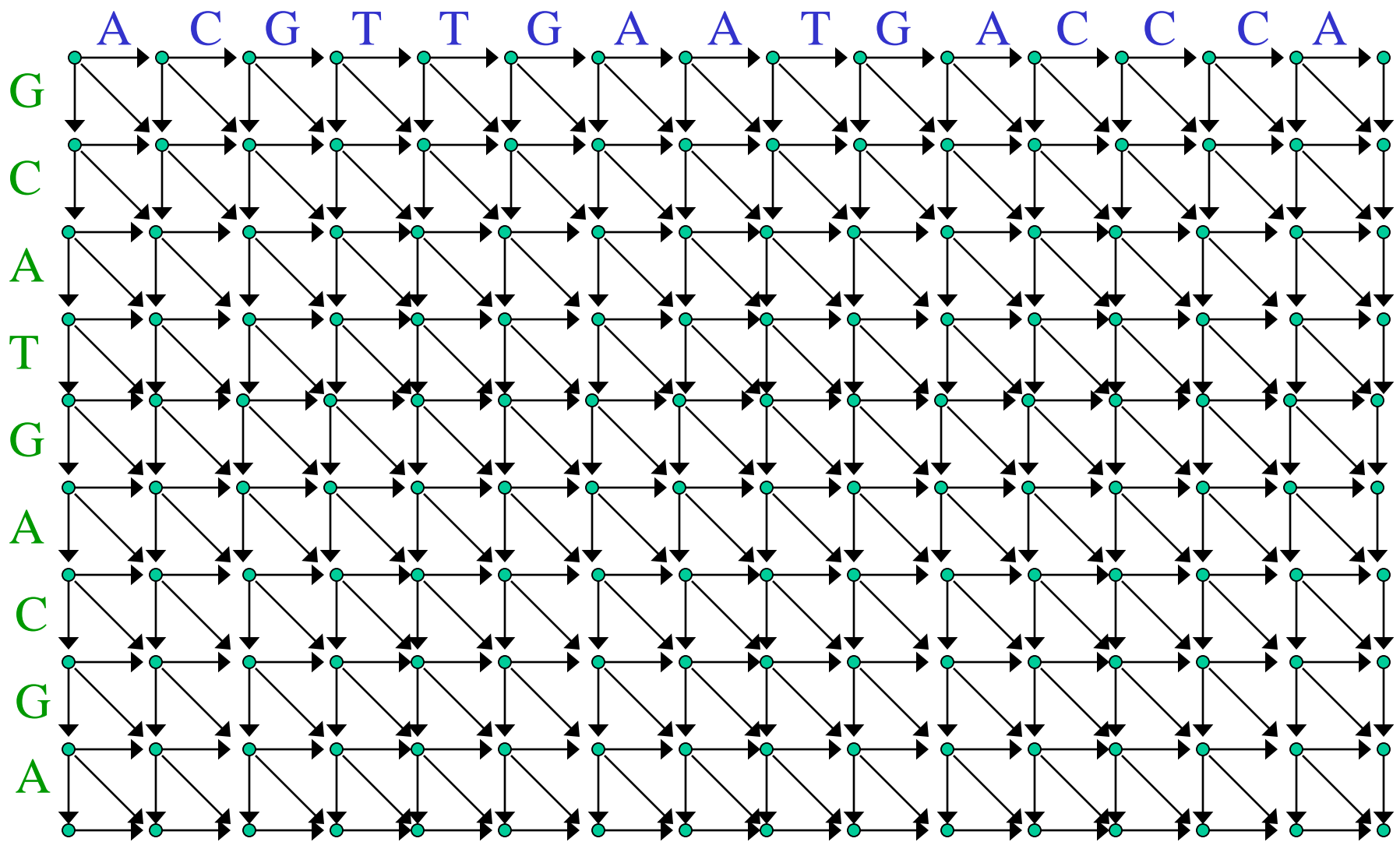
- PSIBLAST approach:
 - initially compare query sequence to database sequences (using BLOSUM-type scoring matrix),
 - build profile using initial matches
 - rescan database using profile
- Optimal choice of
 - substitution matrix,
 - gap penalties, or
 - profiles

comes from LLR based on alignment data
(target vs background)

Smith-Waterman special cases

- Various special cases are optimal path problems for *subgraphs* of edit graph:
 - *Gap-free* alignments correspond to paths confined to a diagonal of edit graph
 - (i.e. subgraph without horizontal & vertical edges).
 - Find *perfectly* matching segments using weights
 - +1 for identical residue pair,
 - $-\infty$ (or large negative penalty) for mismatches or gaps.
- Less efficient than “sorting pointers” method from lecture 1 / HW1.

The Edit Graph for a Pair of Sequences



- Find *imperfect internal repeats* by searching edit graph of sequence against itself
 - i.e. the same sequence labels columns and rows

above (& not including) the main diagonal:

 - if include main diagonal, best path will be identity match to self
 - complexity = $O(N^2)$ where N = sequence length.

Graph for finding imperfect internal repeats:

