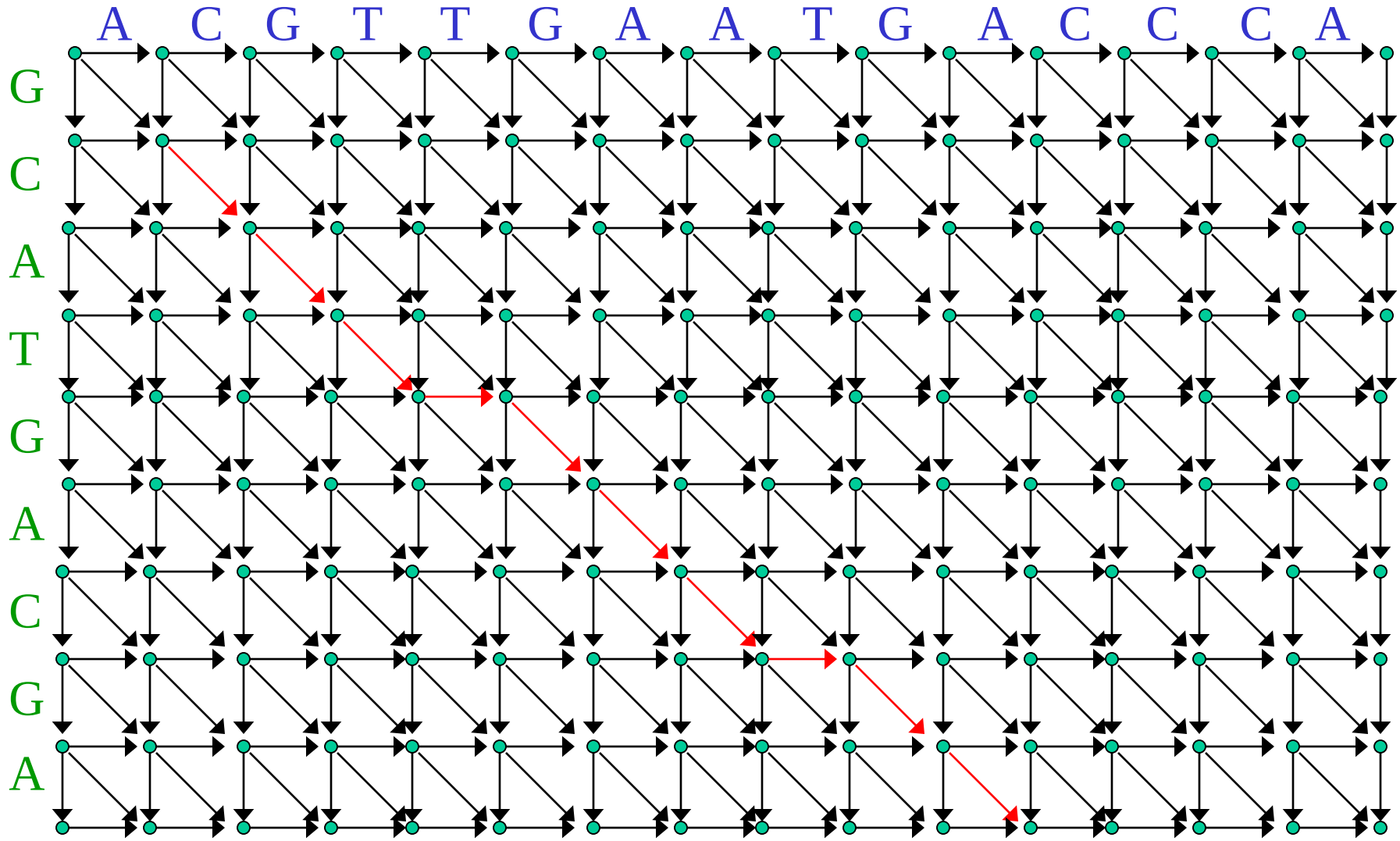
Today's Lecture

- Alignment algorithms
  - Smith-Waterman, Needleman-Wunsch
- Local vs global
- Computational complexity of pairwise alignment
- Multiple sequence alignment



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

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

Alignment algorithms

- *Smith-Waterman* algorithm to find highest scoring alignment

  = dynamic programming algorithm to find highest-weight path

  – Is a *local* alignment algorithm:
    - finds alignment of subsequences rather than the full sequences.
- Can process nodes in any order in which parents precede children. Commonly used alternatives are
  – depth order
  – row order
  – column order

- If constrain path to
  - start at upper-left corner node and
  - extend to lower-right corner node,

  get a *global* alignment instead
- This sometimes called *Needleman-Wunsch algorithm*
  - (altho original N-W alg treated gaps differently)
- ∃ variants which constrain path to
  - start on the left or top boundary,
  - extend to the right or bottom boundary.

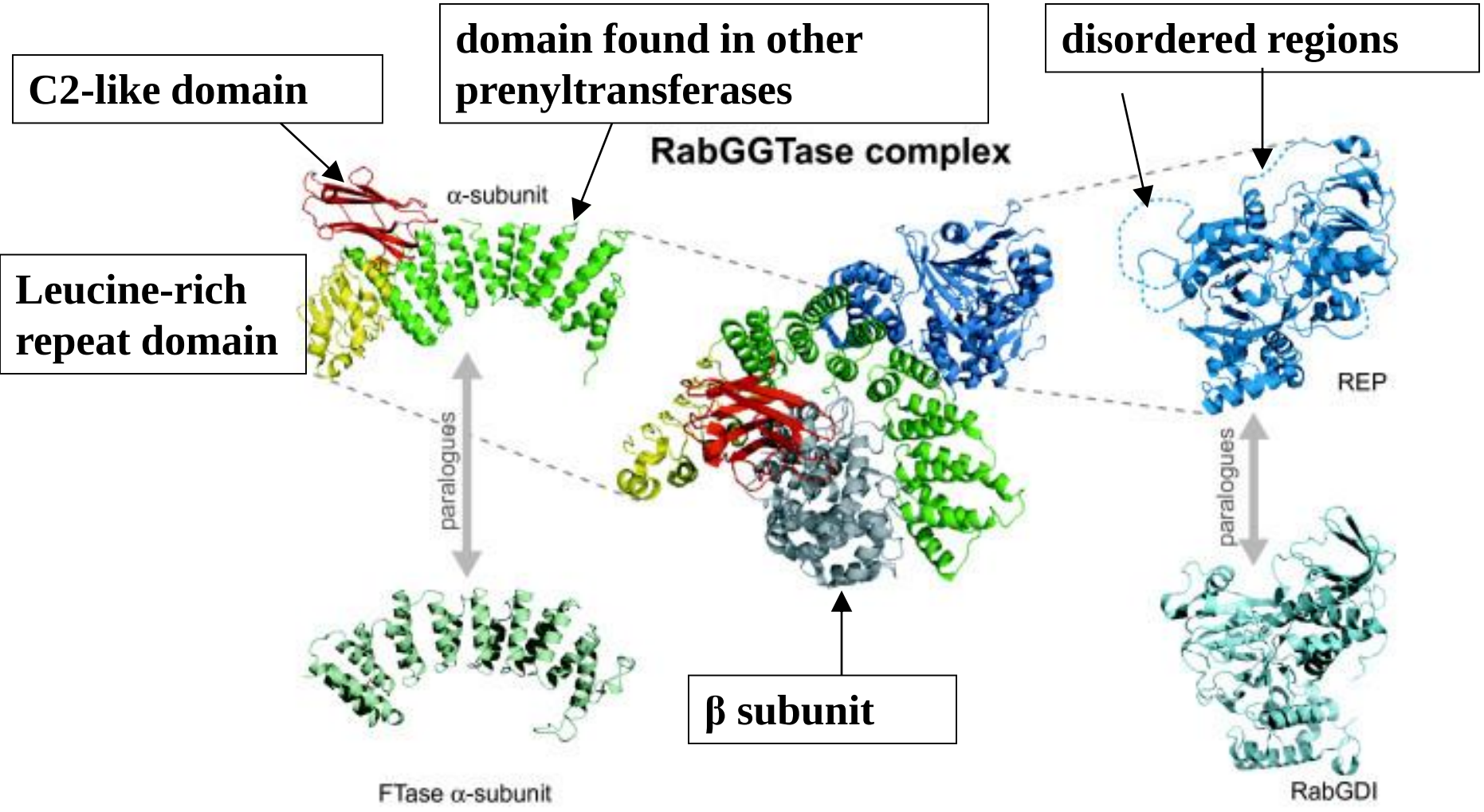
Local vs. Global Alignments: Biological Considerations

- Many proteins consist of multiple 'domains' (modules), some of which may be present
  - with similar, but not identical sequence

  in many other proteins
  - e.g. ATP binding domains, DNA binding domains, protein-protein interaction domains ...

  Need *local alignment* to detect presence of similar regions in otherwise dissimilar proteins.
- Other proteins consist of single domain evolving as a unit
  - e.g. many enzymes, globins.

  Global alignment sometimes best in such cases
  - ... but even here, some regions are more highly conserved (more slowly evolving) than others, and most sensitive similarity detection may be local alignment.

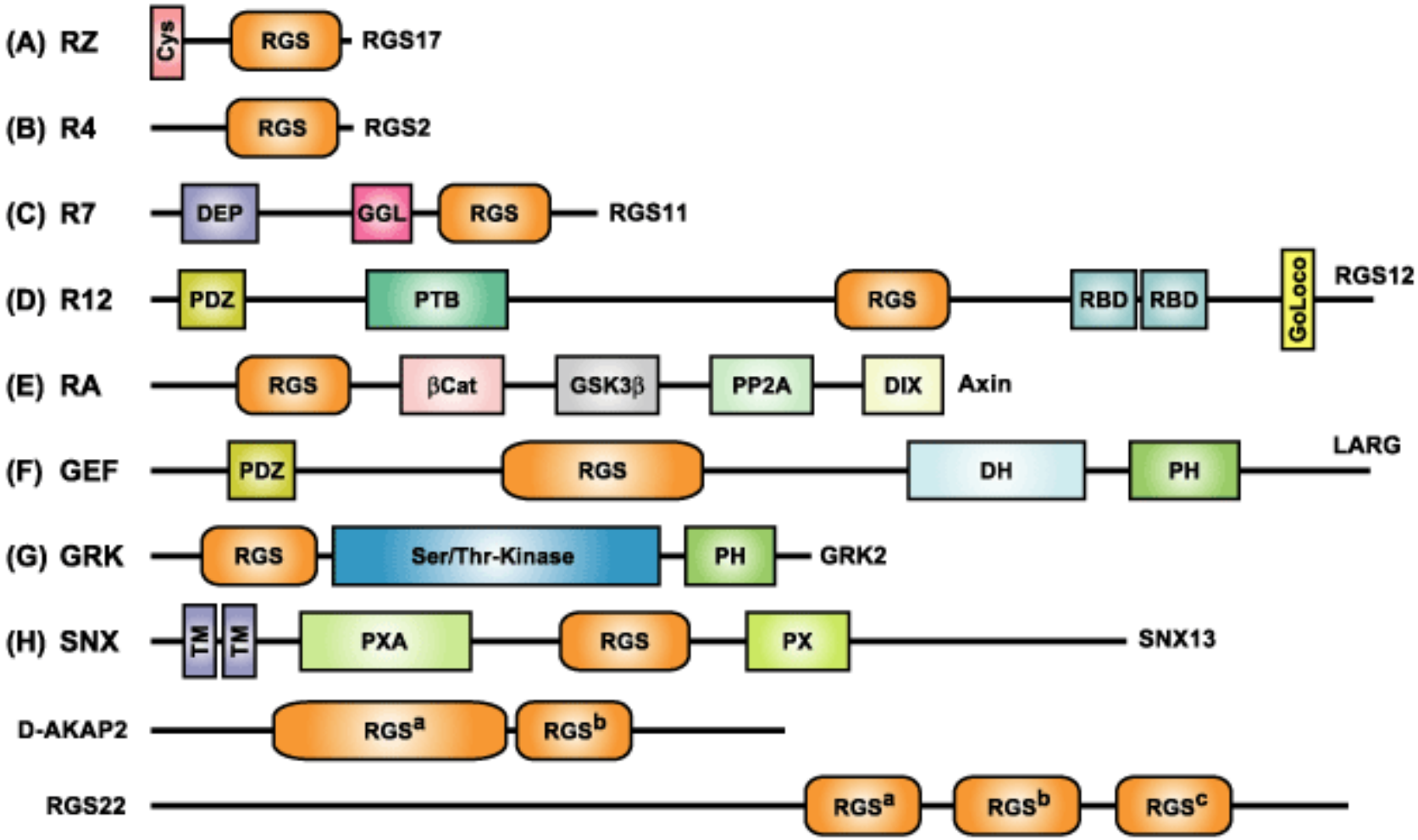


3-D structures of rat Rab Geranylgeranyl Transferase complexed with REP-1, + paralogs.

adapted from **Rasteiro and Pereira-Leal *BMC Evolutionary Biology* 2007 7:140**

Multidomain architecture of representative members from all subfamilies of the mammalian RGS protein superfamily.

from www.unc.edu/~dsiderov/page2.htm



Similar considerations apply to aligning DNA sequences:

- (semi-)global alignment may be preferred for aligning
  - cDNA to genome
  - recently diverged genomic sequences (e.g. human / chimp)

  but local alignment often gives same result!
- between more highly diverged sequences, have
  - rearrangements (or large indels) in one sequence vs the other,
  - variable distribution of sequence conservation,

  & these usually make local alignments preferable.

Complexity

- For two sequences of lengths M and N, edit graph has
  - $(M+1)(N+1)$ nodes,
  - $3MN+M+N$ edges,
- time complexity: $O(MN)$
- space complexity to find

  highest score and beginning & end of alignment

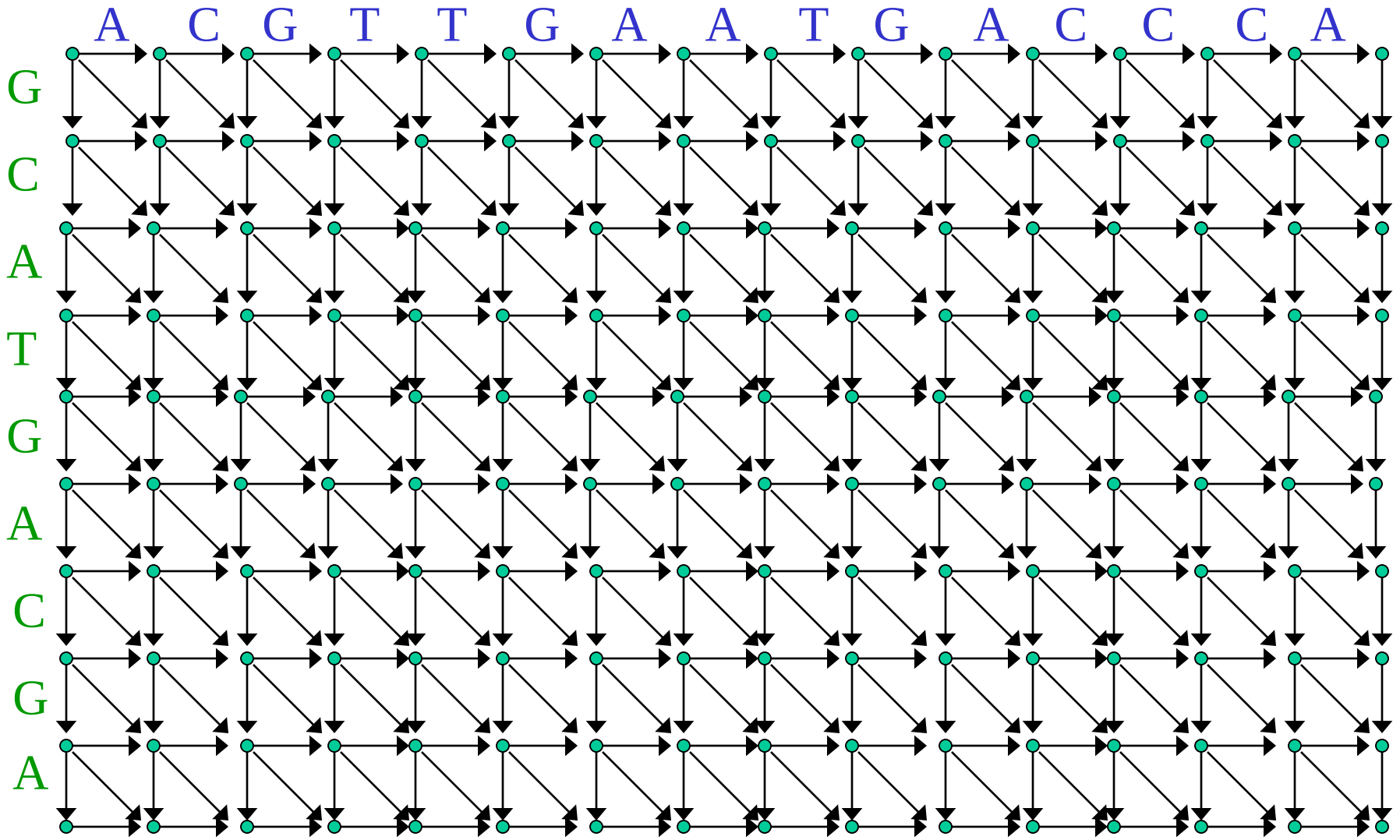
  is $O(\min(M,N))$

  (since only need store node's values until children processed)
- space complexity to reconstruct highest-scoring alignment: $O(MN)$

- For genomic comparisons may have
  - $M, N \approx 10^6$ (if comparing two large genomic segments), or
  - $M \approx 10^3$, $N \approx 10^9$ (if searching gene sequence against entire genome);

  in either case $MN \approx 10^{12}$.
- Time complexity 10^{12} is (marginally) acceptable.
- $\exists$ speedups which reduce constant by
  - reducing calculations per matrix cell, using fact that score often 0
    - (our program *swat*).
    - still guaranteed to find highest-scoring alignment.
  - reducing # cells considered, using nucleating word matches
    - (*BLAST*, or *cross_match*).
    - Lose guarantee to find highest-scoring alignment.

The *Edit Graph* for a Pair of Sequences

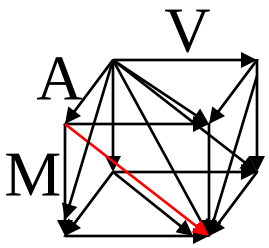


Multiple Alignment via Dynamic Programming

- Higher dimension edit graph
  - each dimension corresponds to a sequence; co-ordinates labelled by residues
  - Each edge corresponds to aligned column of residues (with gaps).
  - Can put arbitrary weights on edges; in particular,
    - can make these correspond to probabilities under an evolutionary model (Sankoff 1975).
  - implicitly assumes independence of columns
- Highest weight path through graph again gives optimal alignment

Generalization to Higher Dimension

Each “cell” in 3-dimensional case looks like this:



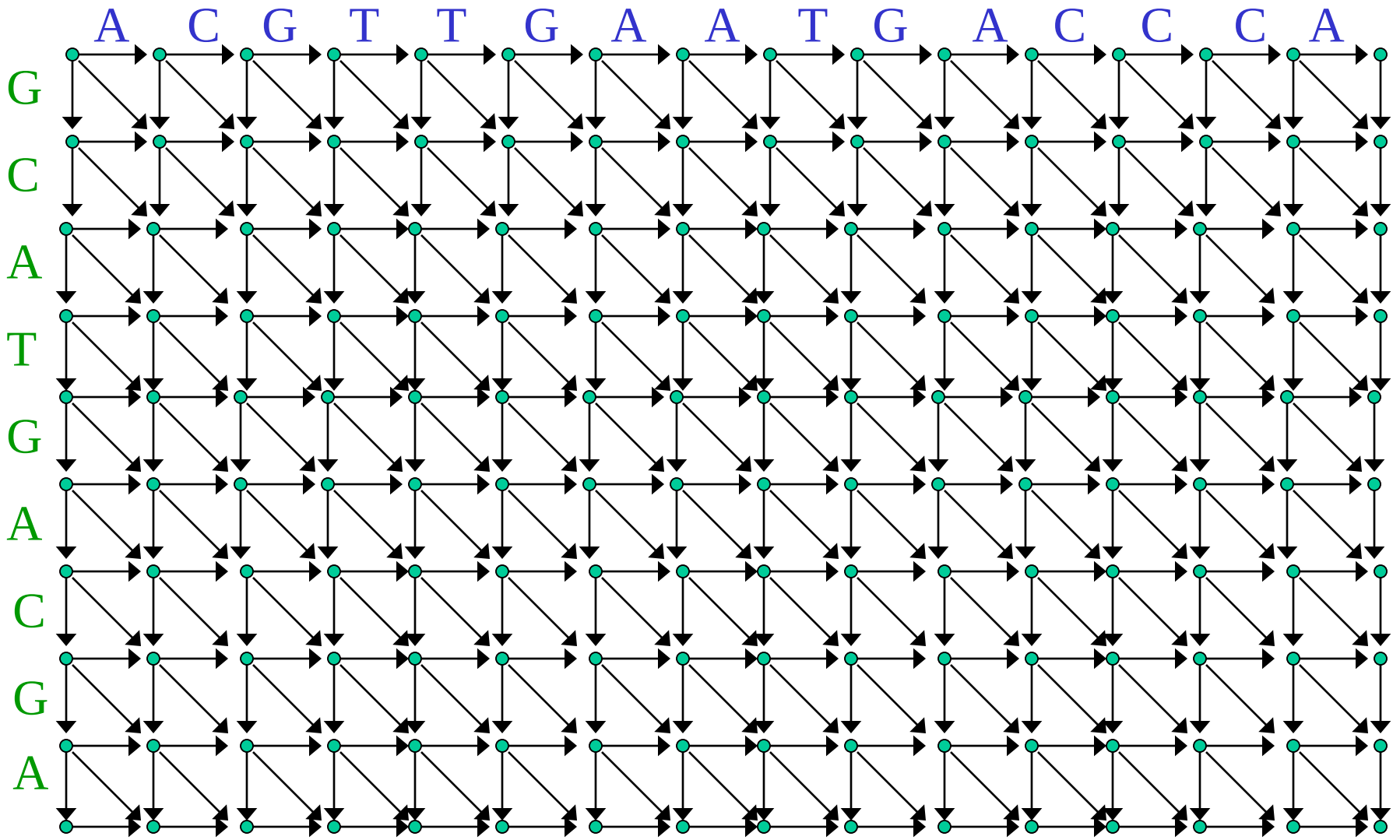
Each edge projects onto a gap or residue in each dimension, defining an alignment column; e.g. red edge defines

V

–

M

The *Edit Graph* for a Pair of Sequences



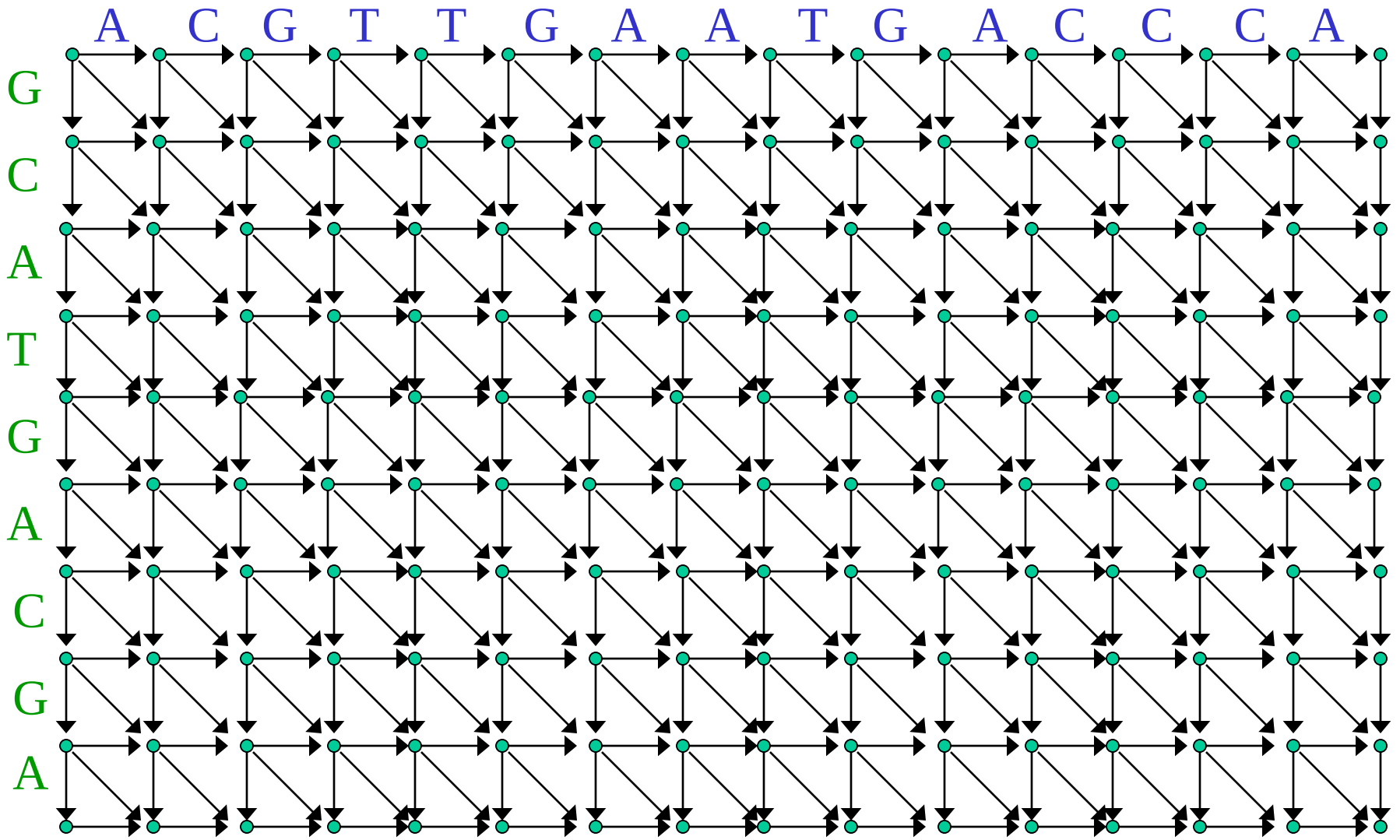
- # edges & # vertices are proportional to product of sequence lengths.
  - For k sequences of size N, is of order $O(N^k)$
    - impractical even for proteins (N ~ 300 to 500 residues) if $k > 5$:

$$300^5 = 2.4\ 10^{12}$$

Multiple alignments: paths in huge WDAGs

- To find high-scoring paths, need to
  - reduce size of graph
  - restrict allowed weighting schemes, and/or
  - sacrifice optimality guarantees
- Durbin *et al.* discuss methods implementing these ideas:
  - Hein
  - Carillo-Lipman
  - progressive alignment (e.g. Clustal)
- HMMs provide nice (but not guaranteed optimal) approach for constructing multiple alignments

The *Edit Graph* for a Pair of Sequences



Better Scoring Models

- Optimal alignment scoring depends on probabilistic modelling (e.g. LLR scores).
- Inherent limitation of dynamic programming: each alignment column (edge in WDAG) scored independently
  - biologically unrealistic, but
  - required for dynamic programming to work!

- *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