Lecture 9: Sequence Alignment

- Sequence alignment and evolution
  - mutations
- Edit graph & alignment algorithms
- Multiple sequence alignment
  - Higher-dimensional edit graphs
  - Progressive alignment

Aligning sequences: Major uses in genome analysis

- To find relationship between sequences from "same" genome, e.g.
  - finding gene structure by aligning cDNA to genome
  - assembling sequence reads in genome sequencing project
  - NextGen applications: "Resequencing", ChIPSeq, etc

  Still need to allow for discrepancies
  - due to basecalling errors & polymorphisms, introns

  but exact match methods (hashtables, suffix arrays) do most of the work

- To detect evolutionary relationships among sequences:
  - illuminating protein structure and function via distant matches
  - illuminating ***mutation*** and ***selection*** in genomes
    - helps find non-neutrally evolving (functional) regions

Here, frequent discrepancies make finding the alignment more challenging

- Often we're interested in details of alignment
  - (i.e. precisely which residues are aligned),

but

- sometimes only care whether alignment score is large enough to imply sequences are related

Sequences & evolution

- Similar sequences of sufficient length usually have a common evolutionary origin
  - i.e. are ***homologous***
- For a pair of sequences
  - "% similarity" makes sense
  - "% homology" doesn't
- In alignment of two homologous sequences
  - differences mostly represent *mutations* that occurred in one or both lineages, but
  - Not all mutations are inferrable from the alignment

Mutation types

- single-base substitution error by DNA polymerase
  - most common type?
- strand slippage error by polymerase, inserting or deleting one or more bases
- DNA damage (radiation, or chemical) + error-prone repair, possibly altering more than one nucleotide, e.g.
  - CpG (hydrolytic deamination of methyl C)
  - dinucleotide changes, perhaps UV-induced dipyrimidine lesions (*Science* 287: 1283-1286)

- *Rearrangements* (break and rejoin)
  - Inversion (2 breaks on same chromosome)
  - Translocation (2 breaks on different chromosomes)
  - More complex (> 2 breaks)
- *Duplication* of a segment
- *Deletion* of a segment
- *Insertion/excision* of transposable element
- Acquisition of DNA from another organism ("*horizontal transfer*")

Mutation ***rates*** may depend on:

- lineage (organism): no universal "molecular clock"
- sex: e.g. in mammals, mut rate higher in males than females
- type of change – e.g.
  - replacement ("substitution") of one nucleotide by another more freq than indels (insertions or deletions)
  - *transition* replacements
    - pyrimidine → pyrimidine (T ↔ C), or purine → purine (A ↔ G)

    more freq than *transversion* replacements
    - pyrimidine → purine, or purine → pyrimidine
  - GC or AT bias in some organisms
    - e.g. G→A more freq than A→G in most eukaryotes
      - causes most genomes to be relatively A+T rich
  - (small) deletions generally more frequent than (small) insertions

- sequence context (e.g. CpG effect)
- position in sequence – some sites more slowly changing than others, due to
  - selection – e.g. in coding sequences,
    - indels strongly selected against because would disrupt reading frame;
    - non-synonymous changes less freq than synonymous
  - variation in underlying mutation rate (cf. mouse genome paper)
    - may in part depend on replication timing (late replication less accurate)

- typical per base subst rates in non-coding DNA:
  - ~1 x 10^{-9} per base per year (order of magnitude)
  - in humans, about 10^{-9} / base / year, $\Rightarrow$ 2 x 10^{-8} / base / generation

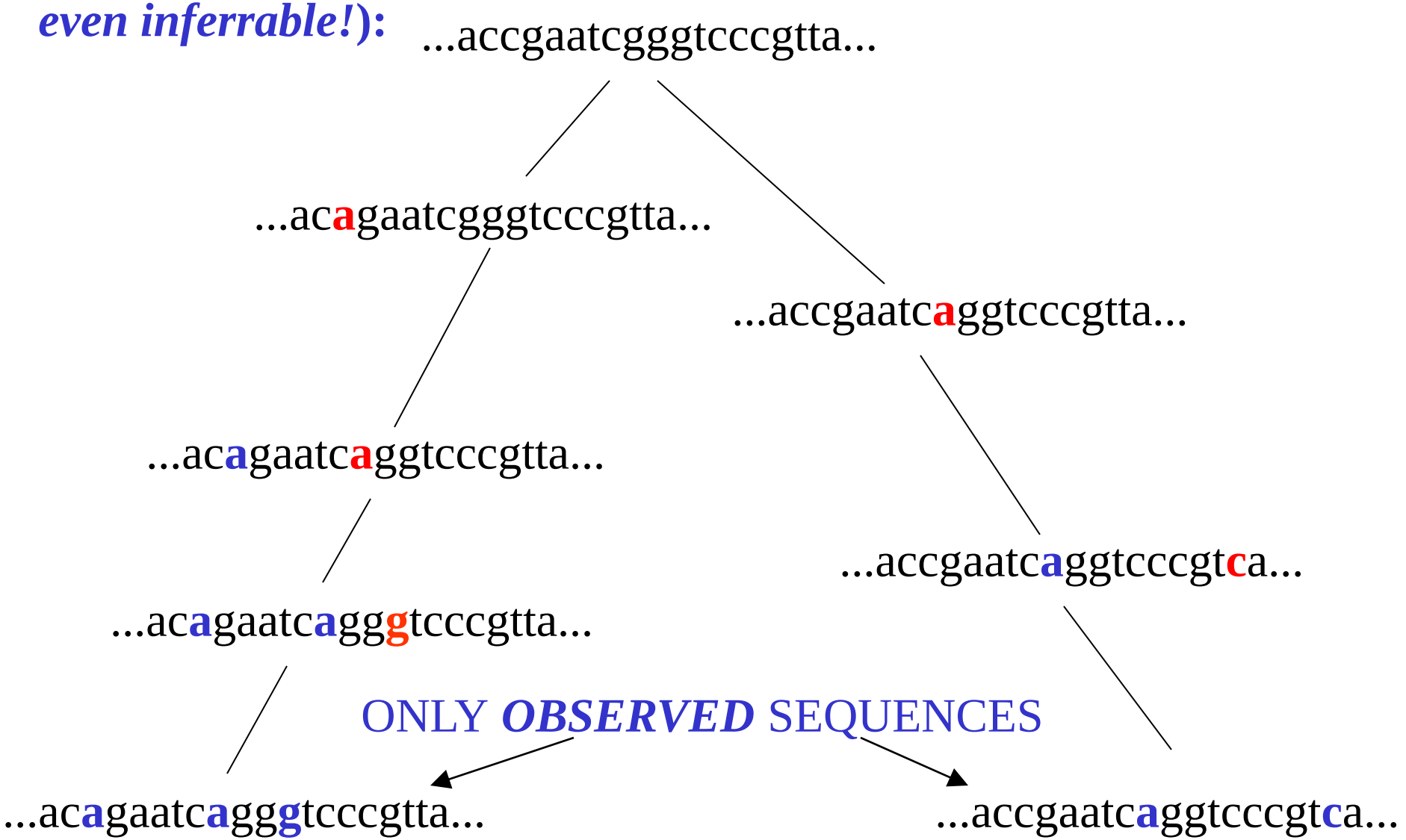
    $\Rightarrow$ 120 / diploid genome / generation

    (recent de novo estimates are lower!)
- freq of gene duplication is ~ 10^{-8} per gene per year (*Science* 290: 1151-1155)
- freq of simultaneous dinuc substitutions is ~ 10^{-10} per dinuc site per year (*Science* 287: 1283-1286)
- freq of CpG $\Rightarrow$ TpG or CpA changes is ~10-fold higher (per CpG) than other substs in mammalian DNA;
  - may account for ~20% of all substitutions.

(Observed) ALIGNMENT:
(*may not be unique*!)

```
...acagaatcagggtcccgtta...
...accgaatcagg-tcccgtca...
```

(Unobserved) MUTATION HISTORY (*in general, this is not even inferrable!*):



Complications

- Parallel & back mutations

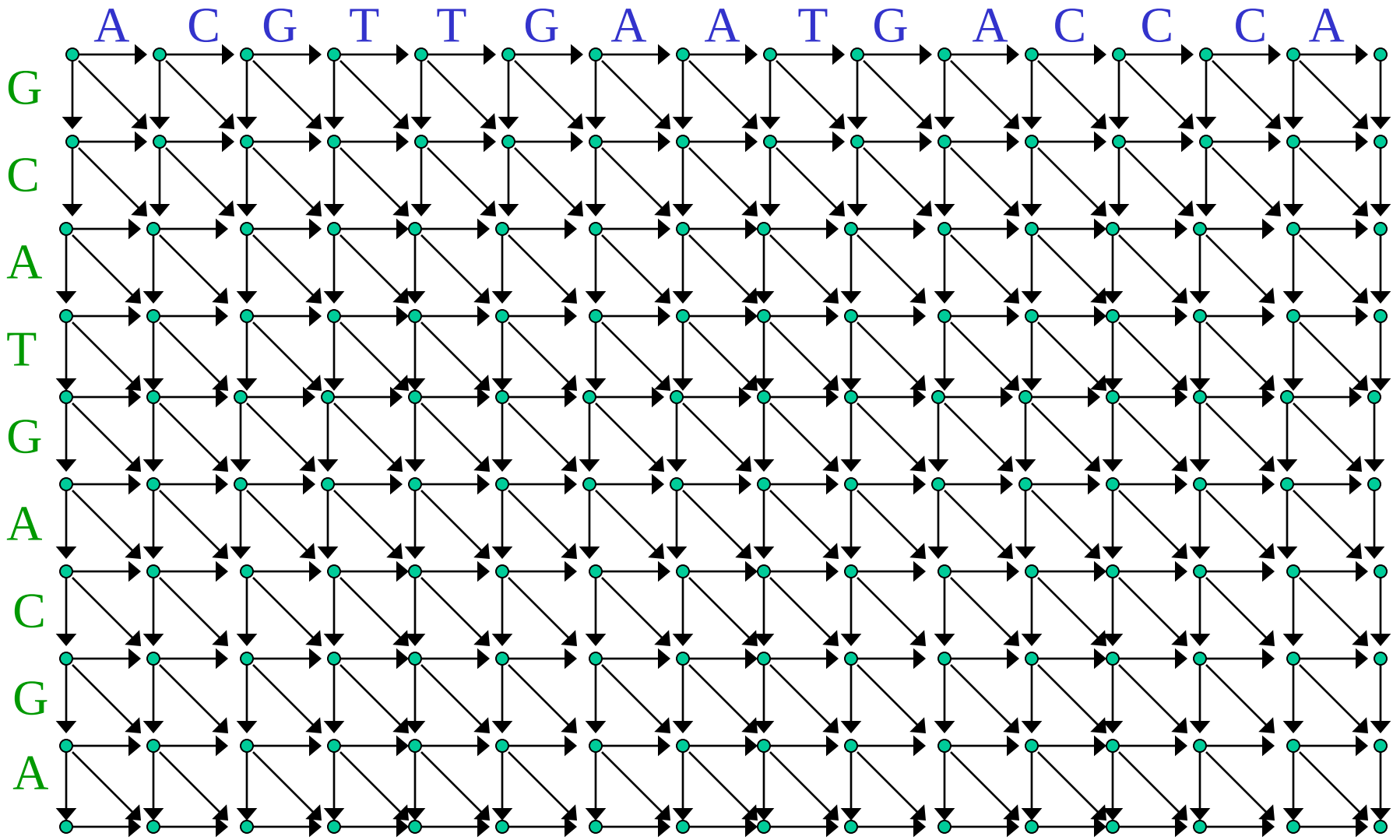
  $\Rightarrow$ estimating total # of mutations requires statistical modelling

- Segmental mutations
  - duplications & other large indels
  - inversions

  are not well modelled by alignments
  - genome-scale alignments usually done ‘in pieces’

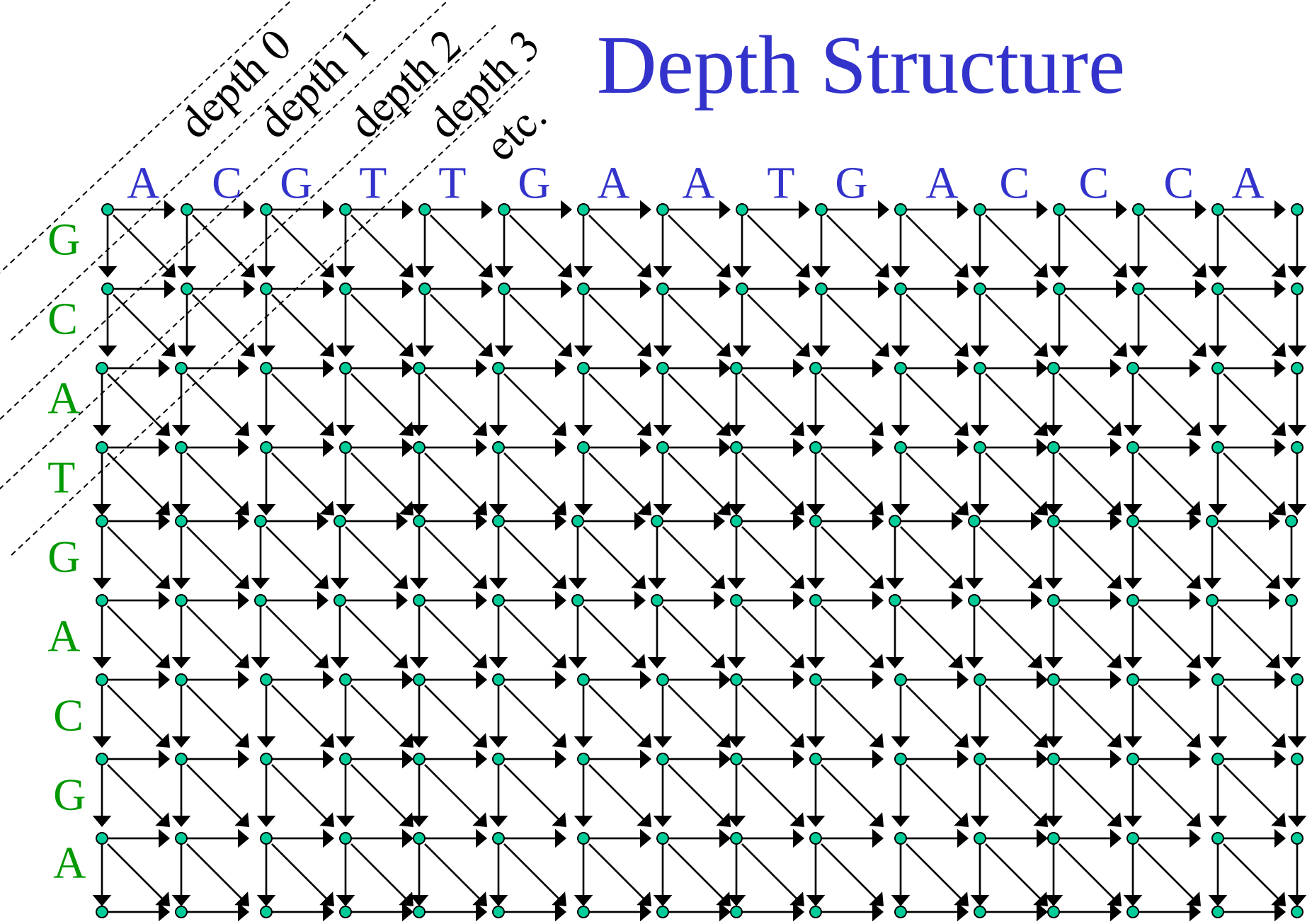
Sequence alignments correspond to *paths* in a *DAG*!

The *Edit Graph* for a Pair of Sequences

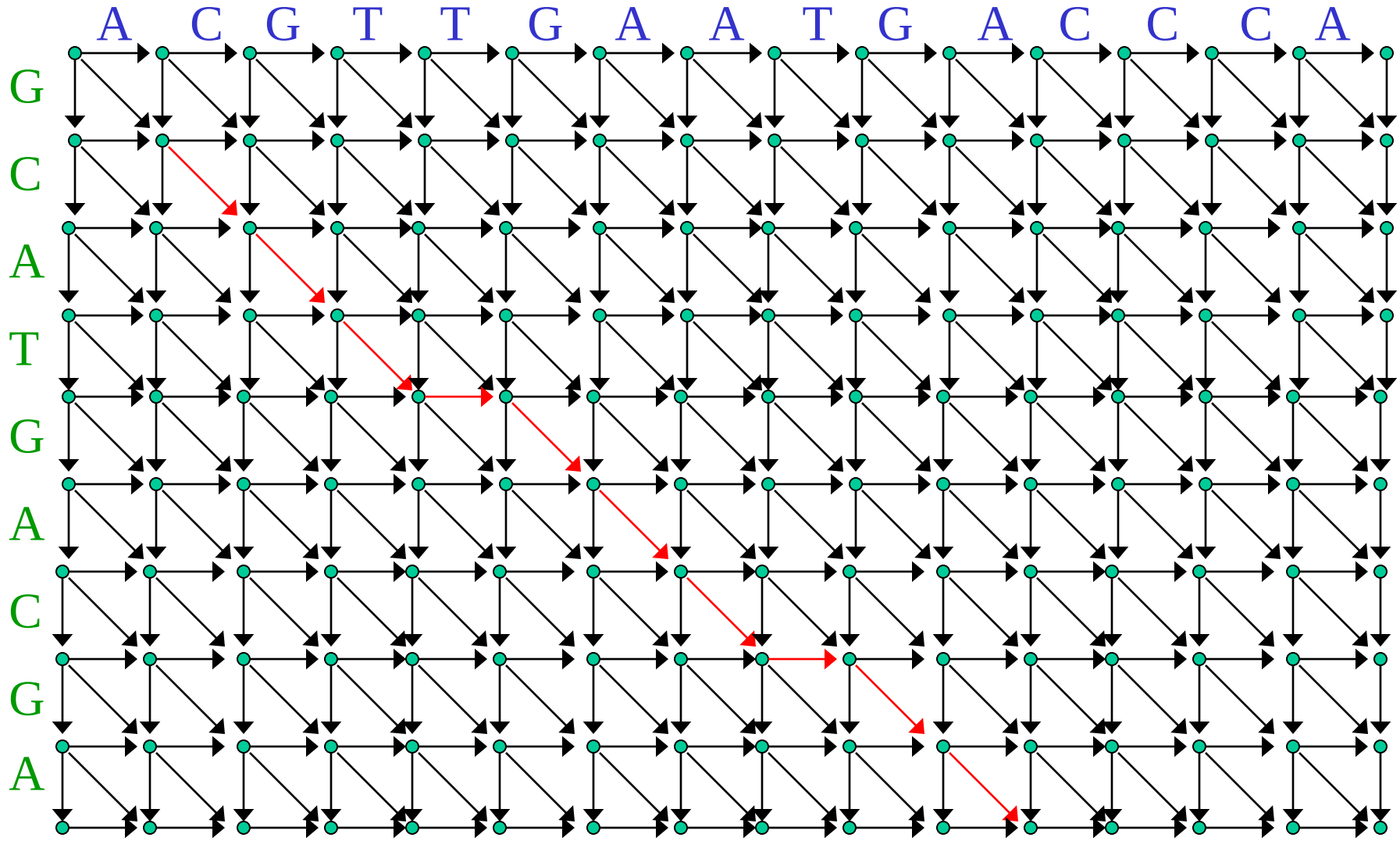


- The edit graph is a DAG.
  - Except on the boundaries, the nodes have in-degree and out-degree both 3.
- The depth structure is as shown on the next slide. Child of node of depth n always has
  - depth $n + 1$ (for a horizontal or vertical edge), or
  - depth $n + 2$ (for a diagonal edge).

Depth Structure



- *Paths* in edit graph correspond to *alignments* of subsequences
  - each edge on path corresponds to an alignment column
  - diagonal edges correspond to column of two aligned residues
  - horizontal edges correspond to column with
    - residue in 1st (top, horizontal) sequence
    - gap in the 2d (vertical) sequence
  - vertical edges correspond to column with
    - residue in 2d sequence
    - gap in 1st sequence



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

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

Weights on Edit Graphs

- Edge weights correspond to scores on alignment columns.
- Highest weight path corresponds to highest-scoring alignment for that scoring system.
- Weights may be assigned using
  - a *substitution score matrix*
    - assigns a score to each possible pair of residues occurring as alignment column
  - or *profile*
    - scores specific to a particular sequence

  and
  - a *gap penalty*
    - assigns a score to column consisting of residue opposite a gap.

based on appropriate probability models (next lecture!)

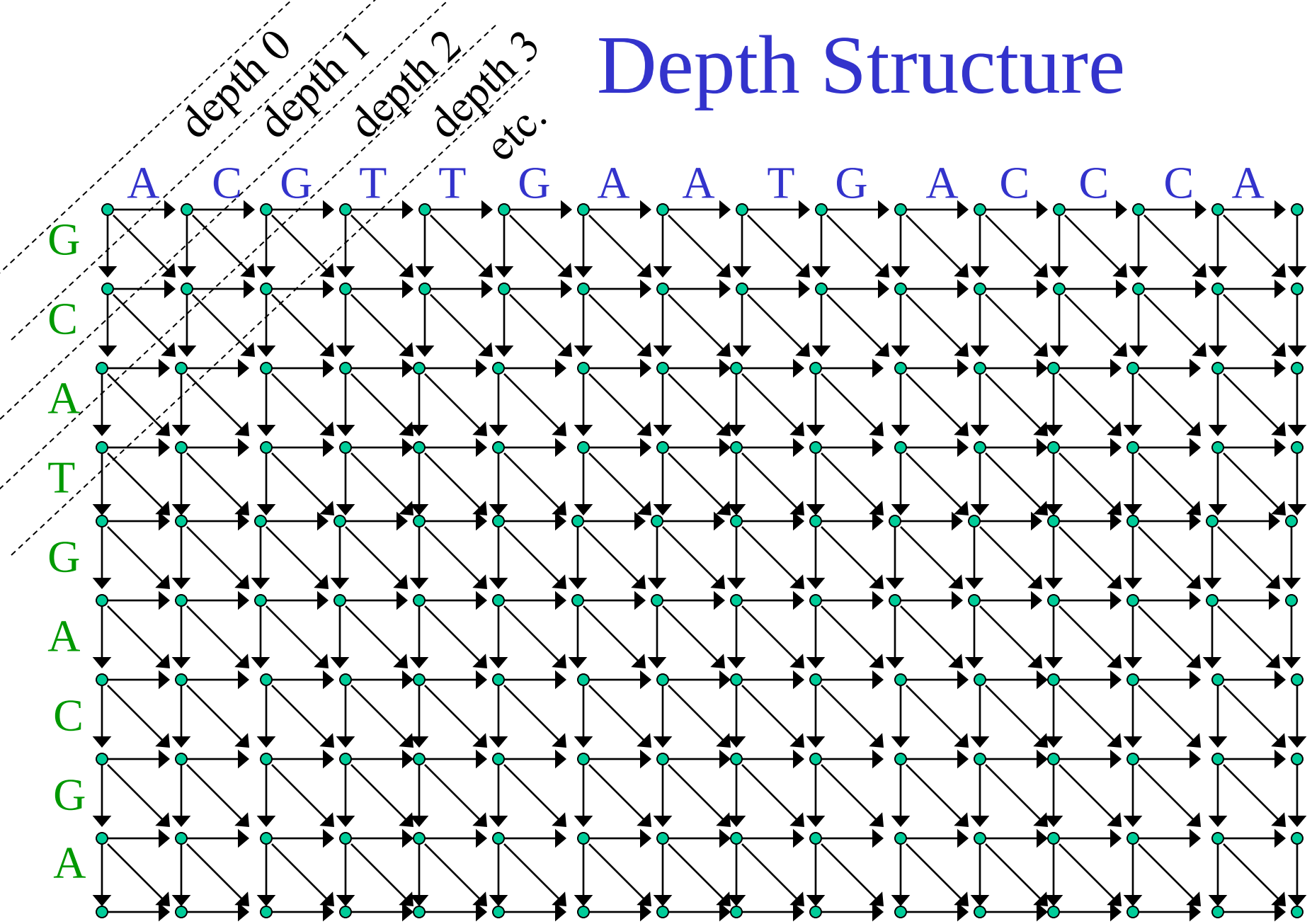
Alignment algorithms

- *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

Depth Structure



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: $O(MN)$

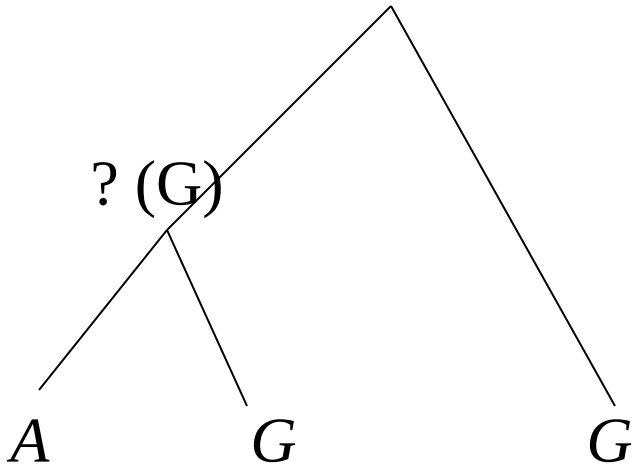
- For genomic comparisons may have
  - $M, N \approx 10^6$ (if comparing two large genomic segments), or
  - $M \approx 10^3$, $N \approx 10^9$ (if searching gene sequence against entire genome);

  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.

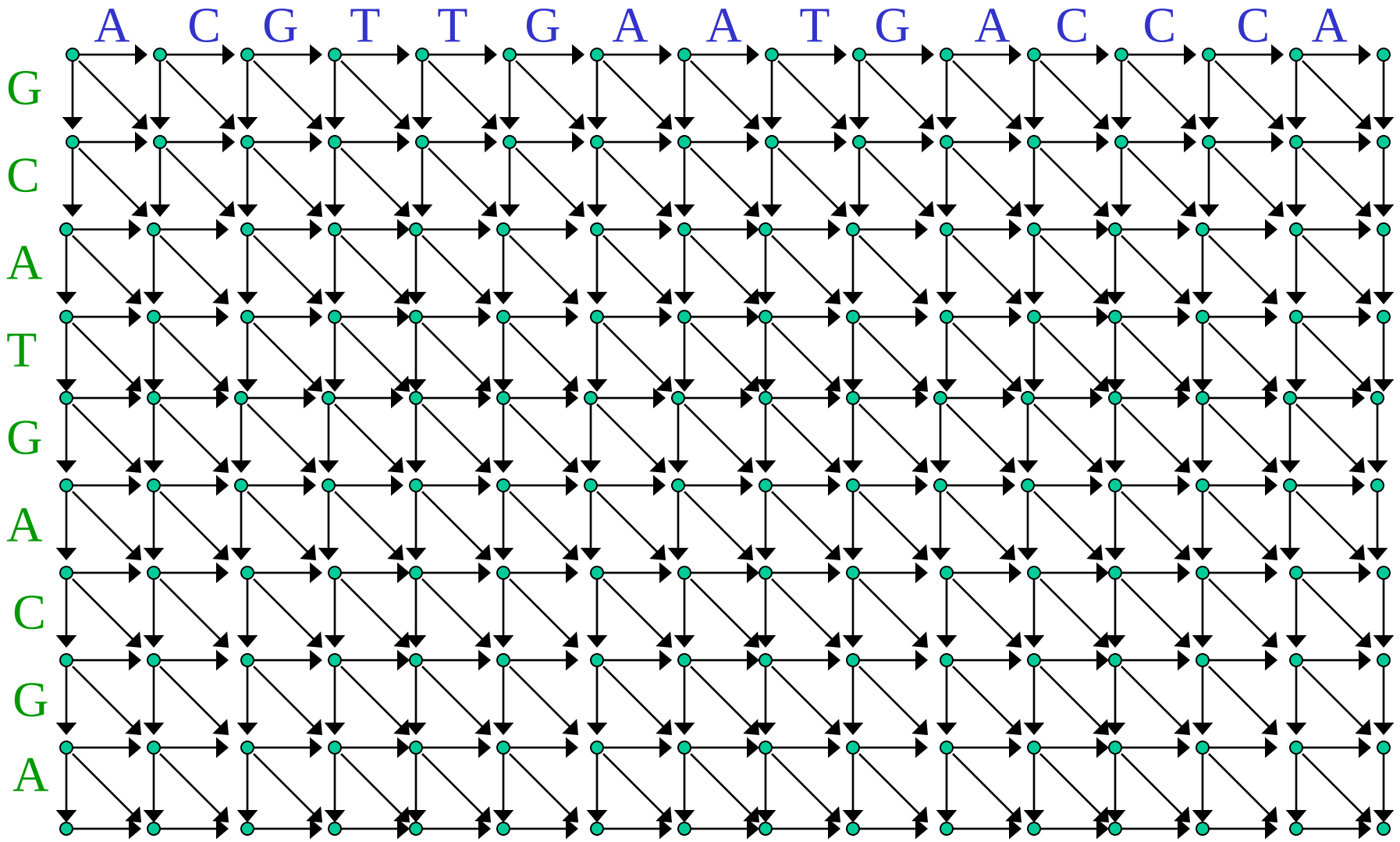
Multiple sequence alignment

- More sequences =>
  - (potentially) more accurate alignments
  - better *resolution* of mutations, selection
- Need > 2 sequences to ***polarize*** mutations

- An evolutionary *tree* relates the sequences!


? (G)
A
G
G

The *Edit Graph* for a Pair of Sequences

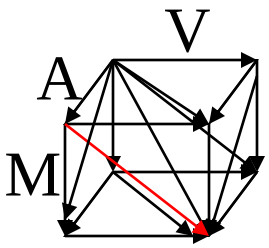


Multiple Alignment via Dynamic Programming

- 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

- # 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 ($N \sim 300$ to 500 residues) if $k > 5$:

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

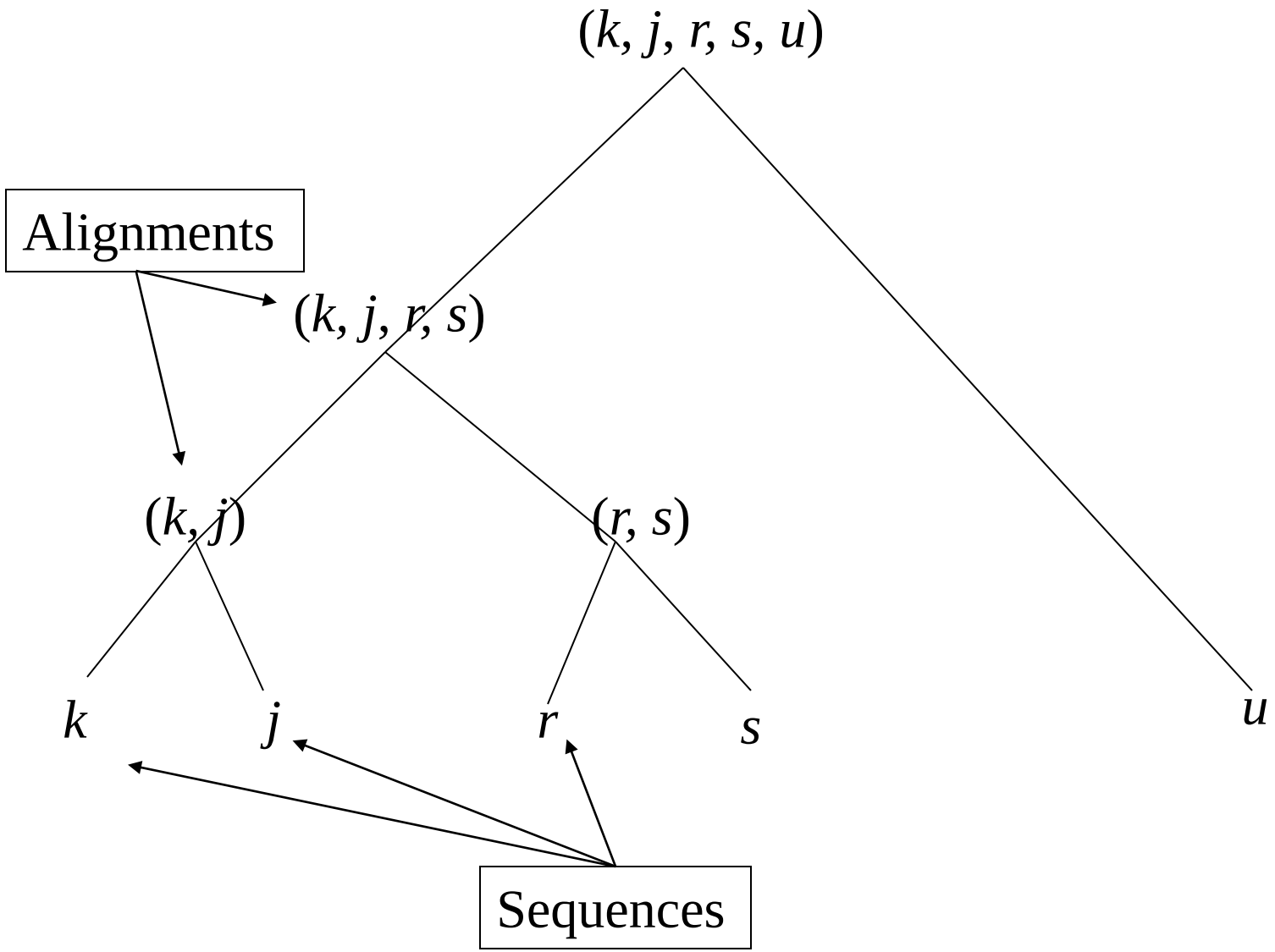
Multiple alignments: paths in huge WDAGs

- 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

Progressive alignment

- Simplest version: align one sequence (the reference) to each of the others, pairwise; construct multiple alignment from that.
- More generally, progressively align *pairs* of (*sequences or*) *alignments*, using a *guide tree*
  - Tree may reflect evolution, or sequence quality
  - Will tend to be more accurate
- Revise gaps
  - correct errors due to gap placement & gap attraction

Guide Tree



- Complexity: $N^2 \times (n - 1)$ where
  - N = seq length, n = # seqs

  instead of N^n
- (does not count gap correction)