Lecture 8

- Sequence alignment and evolution
  - mutations
- Edit graph & alignment algorithms
  - Smith-Waterman, Needleman-Wunsch
- Local vs global

Aligning sequences

- Major uses in genome analysis:
  - To find relationship between sequences from “same” genome
    - (still need to allow for discrepancies – due to errors/polymorphisms)

    E.g.
    - finding gene structure by aligning cDNA to genome
    - assembling sequence reads in genome sequencing project
    - NextGen applications: “Resequencing”, ChIPSeq, etc
  - To detect evolutionary relationships:
    - illuminates function of distantly related sequences under selection
    - finds corresponding positions in neutrally evolving sequence
      - to illuminate mutation process
      - helps find non-neutrally evolving (functional) regions

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

  but

- sometimes only interested in whether alignment score is large enough to imply that sequences are likely to be 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 – not understood! (cf. mouse genome paper)

- 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, ⇒ 2 x 10^{-8} / base / generation

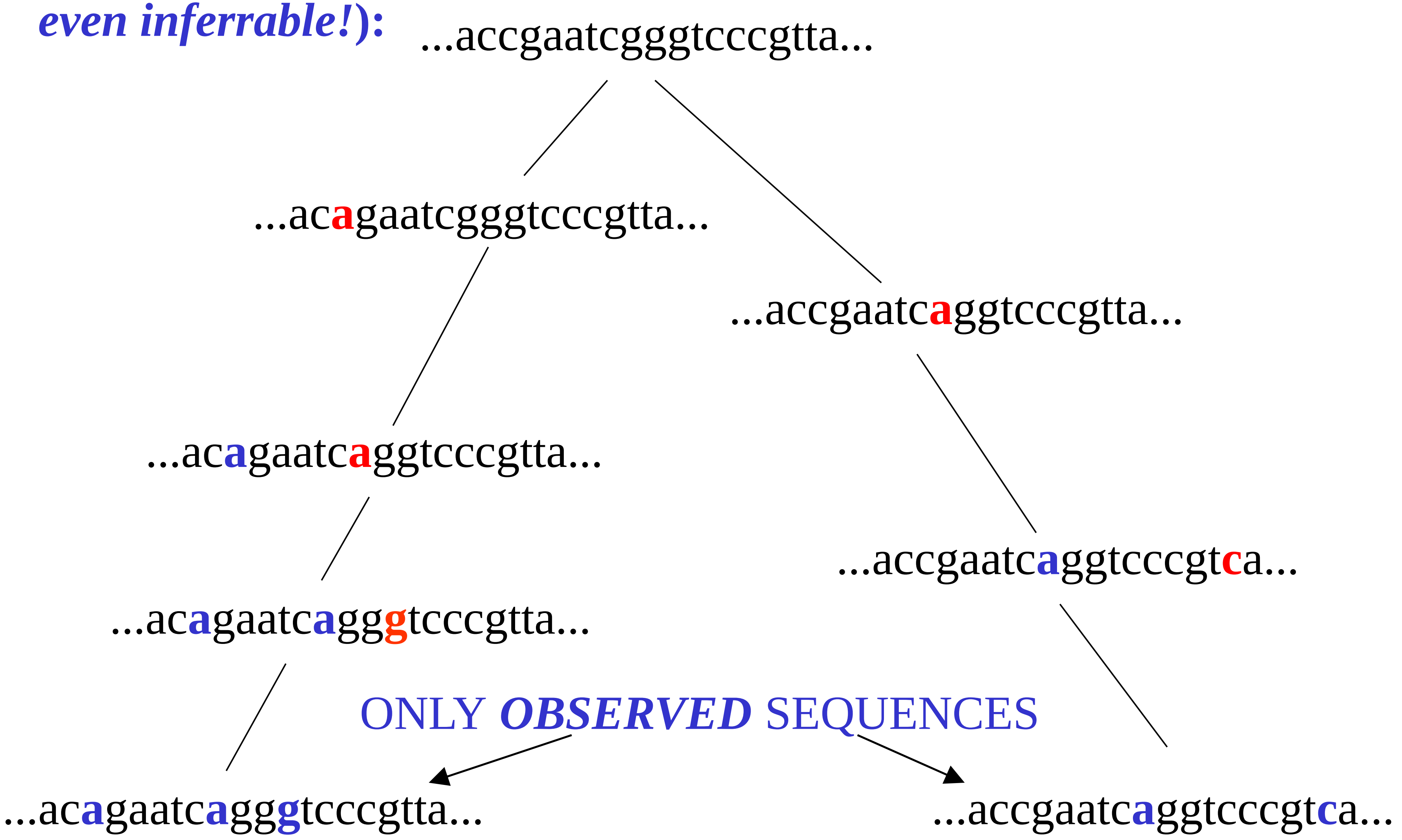
    ⇒ 120 / diploid genome / generation

    (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 ⇒ 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!*):



...accgaatcgggtcccgtta...

...ac**a**gaatcgggtcccgtta...

...accgaatc**a**ggtcccgtta...

...ac**a**gaatc**a**ggtcccgtta...

...accgaatc**a**ggtcccgt**c**a...

...ac**a**gaatc**a**gg**g**tcccgtta...

ONLY ***OBSERVED*** SEQUENCES

...ac**a**gaatc**a**gg**g**tcccgtta...

...accgaatc**a**ggtcccgt**c**a...

Complications

- Parallel & back mutations

  ⇒ estimating total # of mutations requires statistical modelling

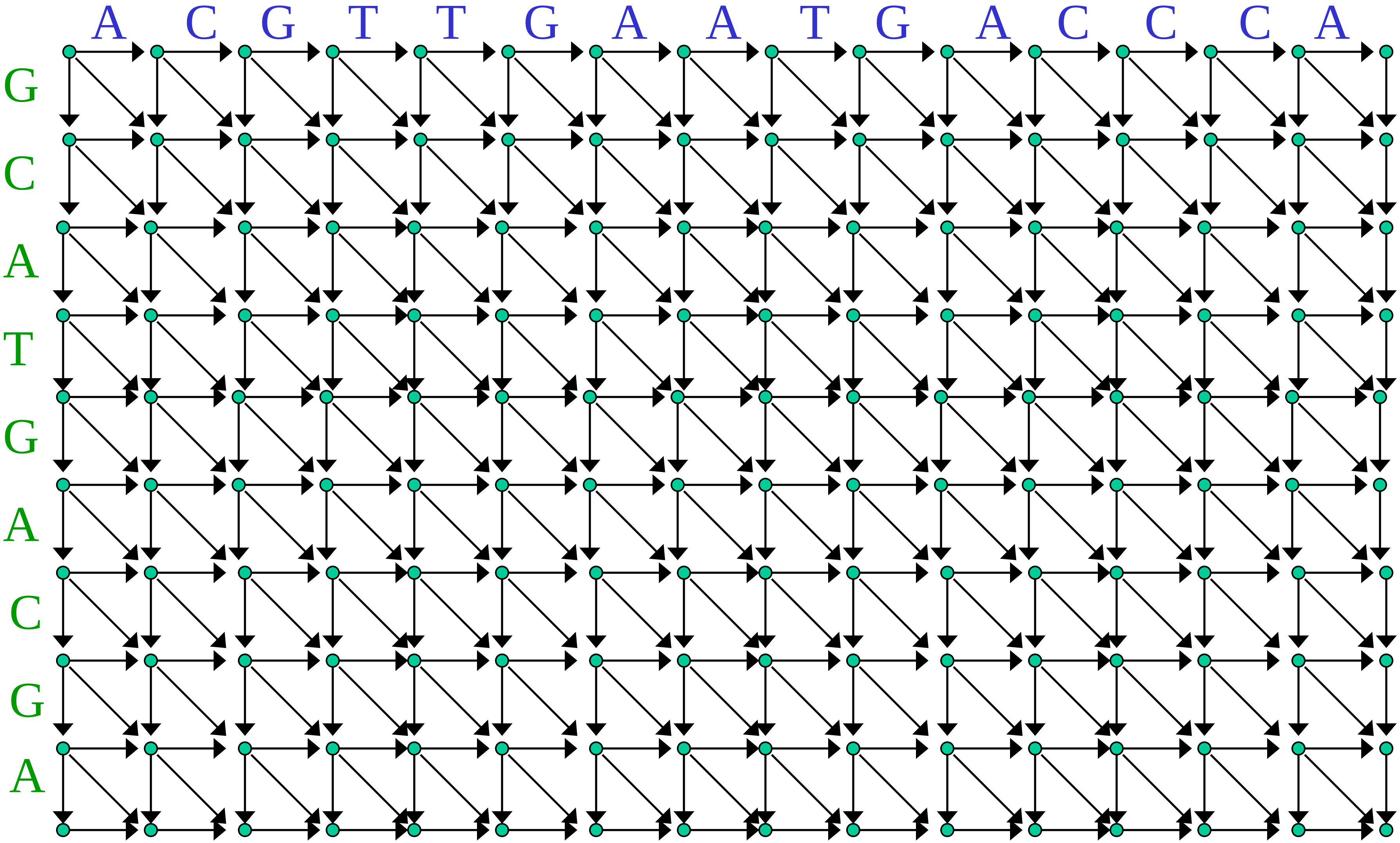
- Insertion/deletion, & segmental mutations

  ⇒ finding the correct alignment can be problematic (‘gap attraction’)

  -- even in closely related sequences!

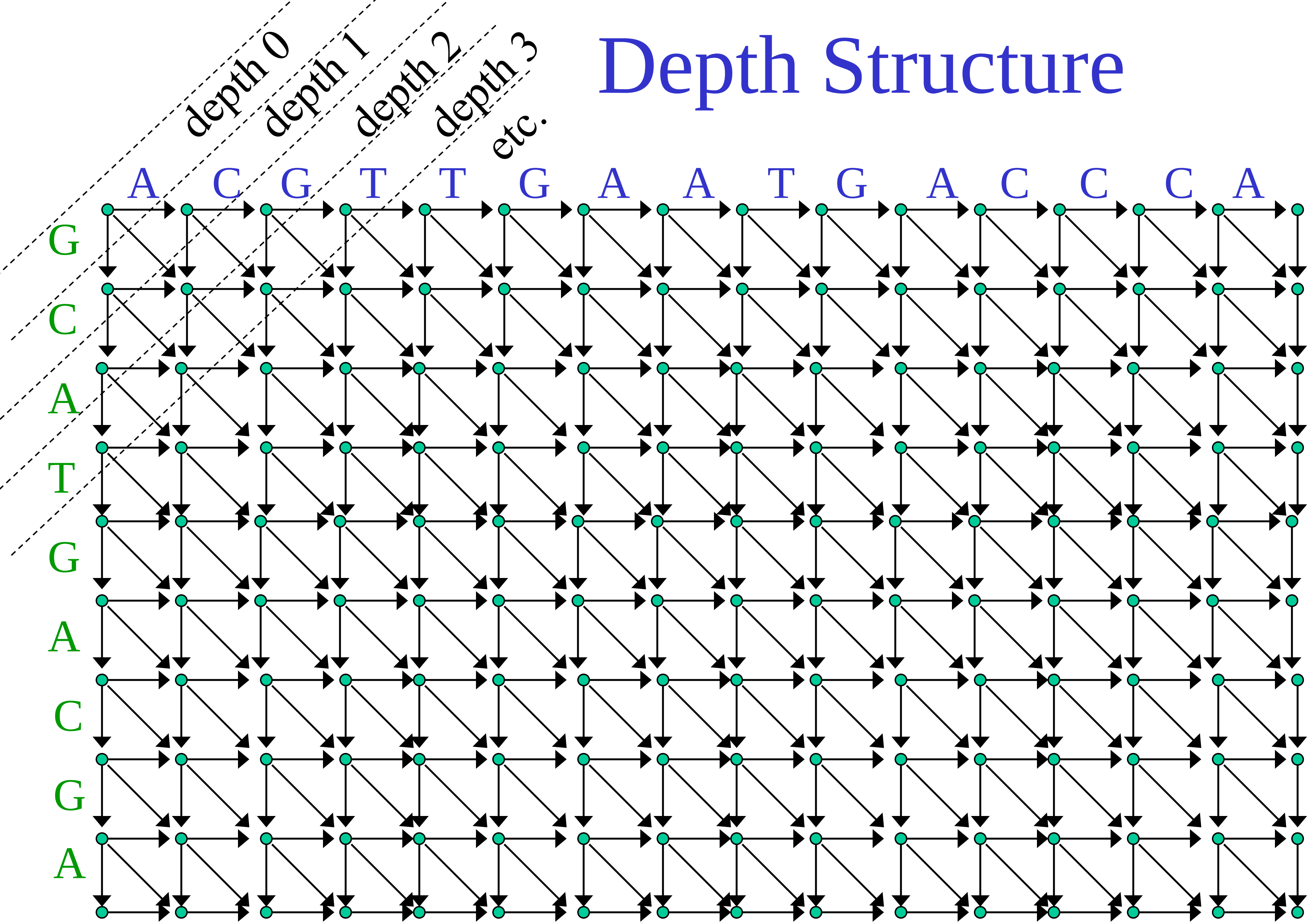
Sequence alignments correspond to *paths* in a *DAG*!

The *Edit Graph* for a Pair of Sequences

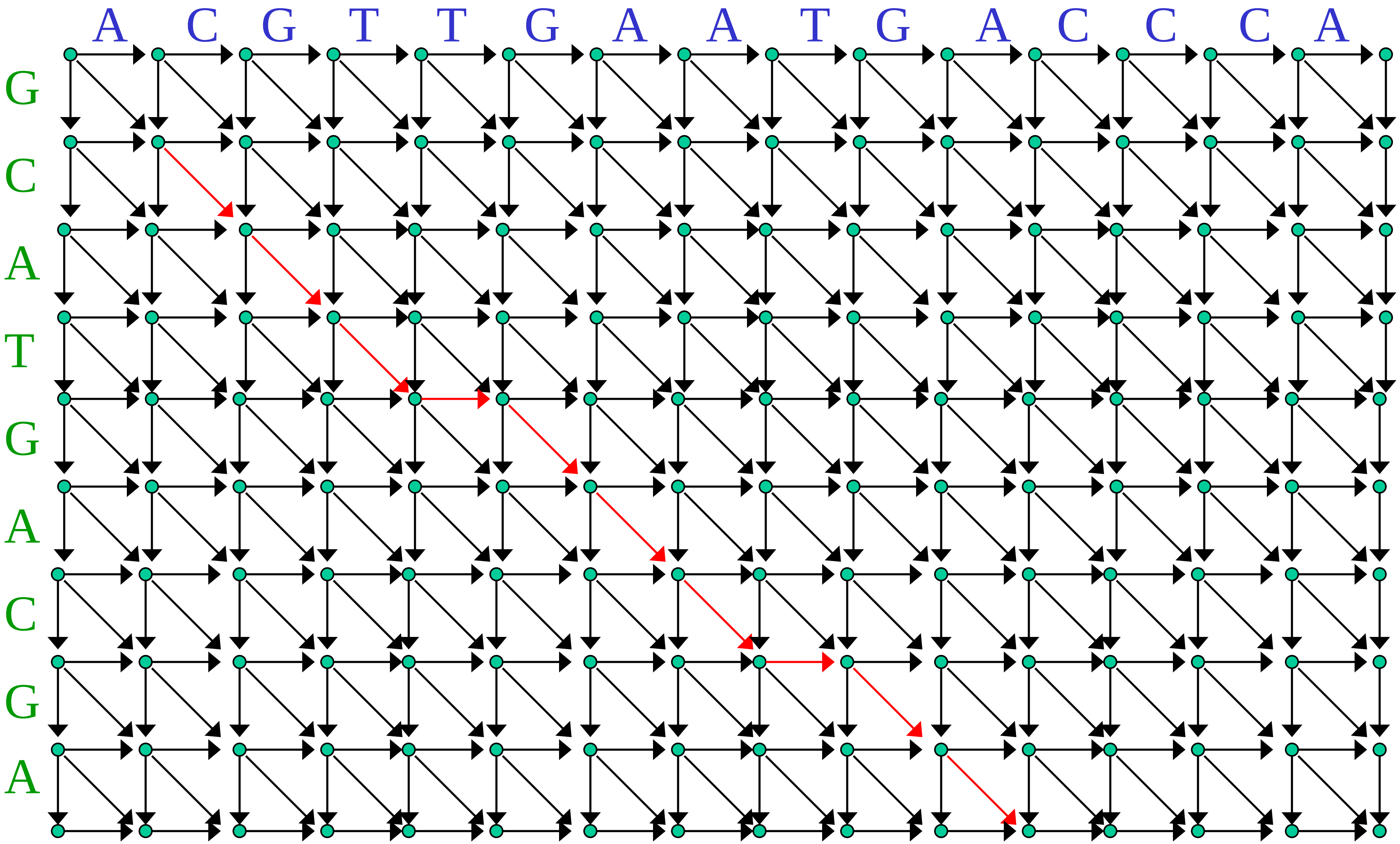


- The edit graph is a DAG.
  - Except on the boundaries, the nodes have in-degree and out-degree both 3.
- The depth structure is as shown on the next slide. Child of node of depth *n* always has
  - depth $n + 1$ (for a horizontal or vertical edge), or
  - depth $n + 2$ (for a diagonal edge).

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

    and
  - a *gap penalty*
    - assigns a score to column consisting of residue opposite a gap.
  - Example for protein sequences: BLOSUM62

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
```

- Matrix entries are of form

  $M(r, s) = \log_a(\mathrm{h}_{r,s} / \mathrm{b}_{r,s})$ (rounded to int) where

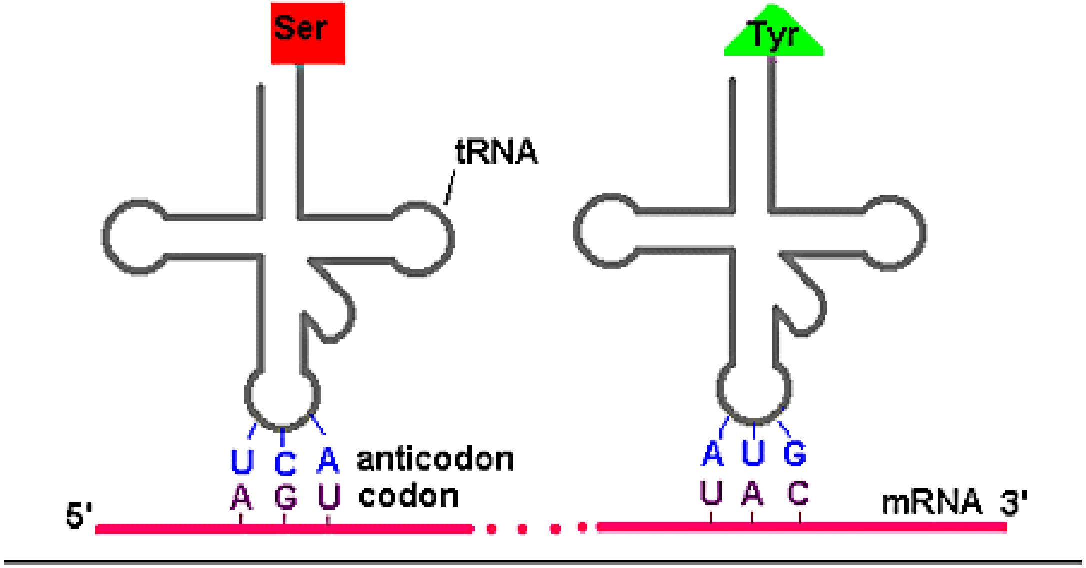
  $\mathrm{h}_{r,s}$ = freq of $\begin{matrix} r \\ s \end{matrix}$ in homologous* seq alignments

  * ’62’ refers to specific set of homologue alignments

  $\mathrm{b}_{r,s}$ = freq of $\begin{matrix} r \\ s \end{matrix}$ in ‘background’ (random) alignments

  a (the logarithm base) = $\sqrt{2}$ (‘half bits’)

- amino acid pairs with positive scores tend to be
  - *chemically similar*
  - in *same row or col* of genetic code table



2nd base in codon

1st base in codon

	U	C	A	G	
U	Phe	Ser	Tyr	Cys	U
	Phe	Ser	Tyr	Cys	C
	Leu	Ser	STOP	STOP	A
	Leu	Ser	STOP	Trp	G
C	Leu	Pro	His	Arg	U
	Leu	Pro	His	Arg	C
	Leu	Pro	Gln	Arg	A
	Leu	Pro	Gln	Arg	G
A	Ile	Thr	Asn	Ser	U
	Ile	Thr	Asn	Ser	C
	Ile	Thr	Lys	Arg	A
	Met	Thr	Lys	Arg	G
G	Val	Ala	Asp	Gly	U
	Val	Ala	Asp	Gly	C
	Val	Ala	Glu	Gly	A
	Val	Ala	Glu	Gly	G

3rd base in codon

The Genetic Code

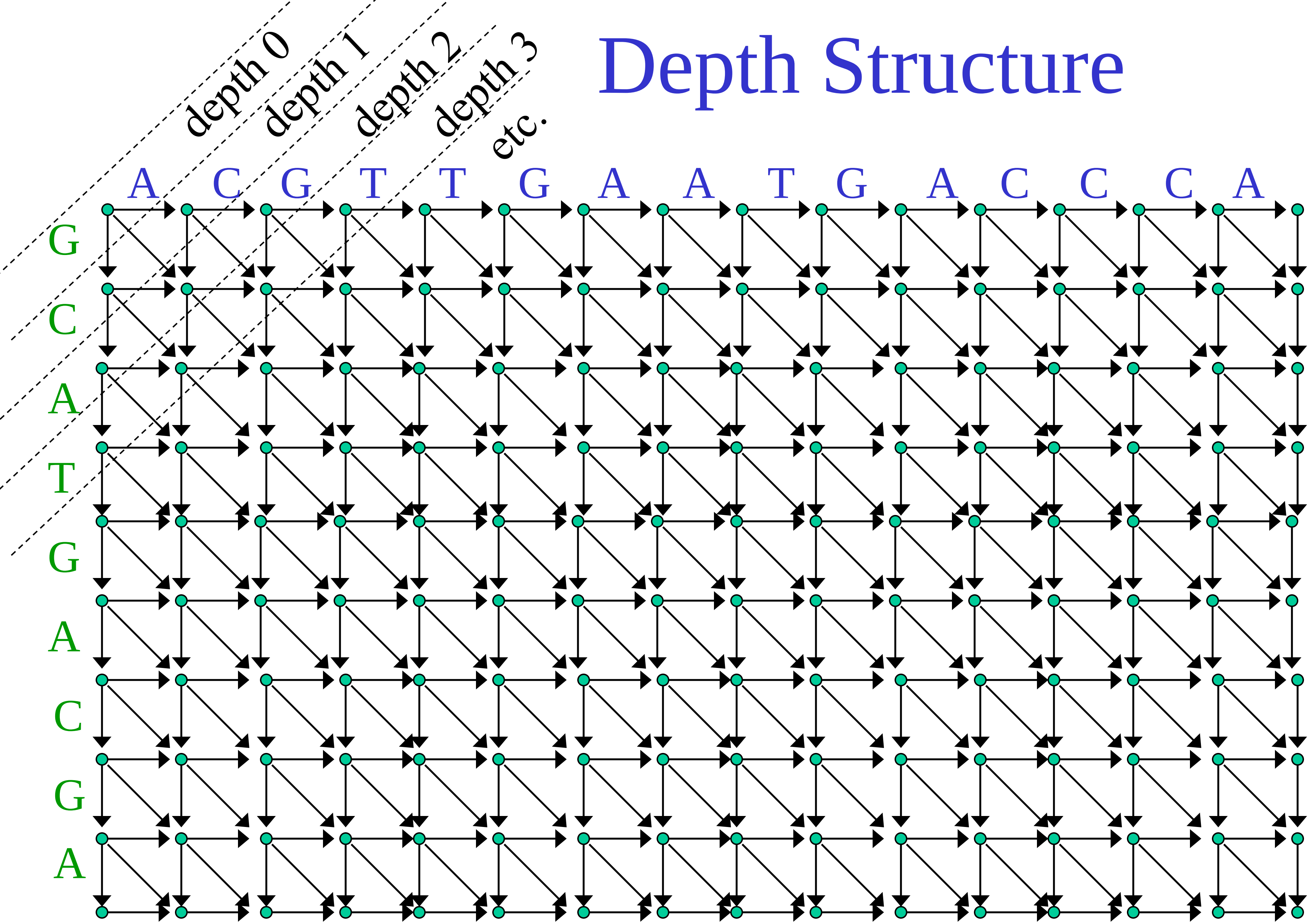
Alignment algorithms

- *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



- 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.

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.
- ∃ 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.

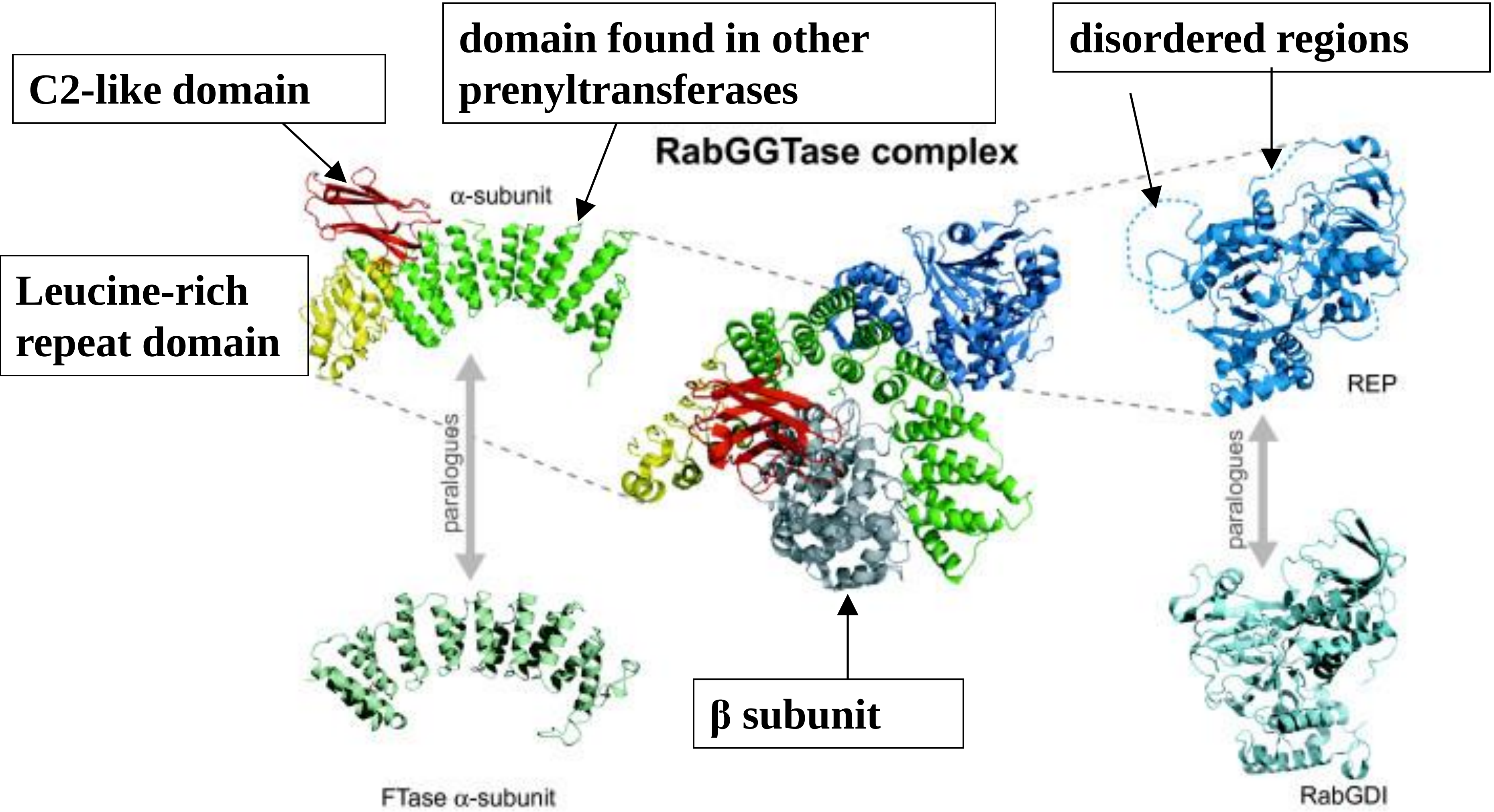
Local vs. Global Alignments: Biological Considerations

- 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.

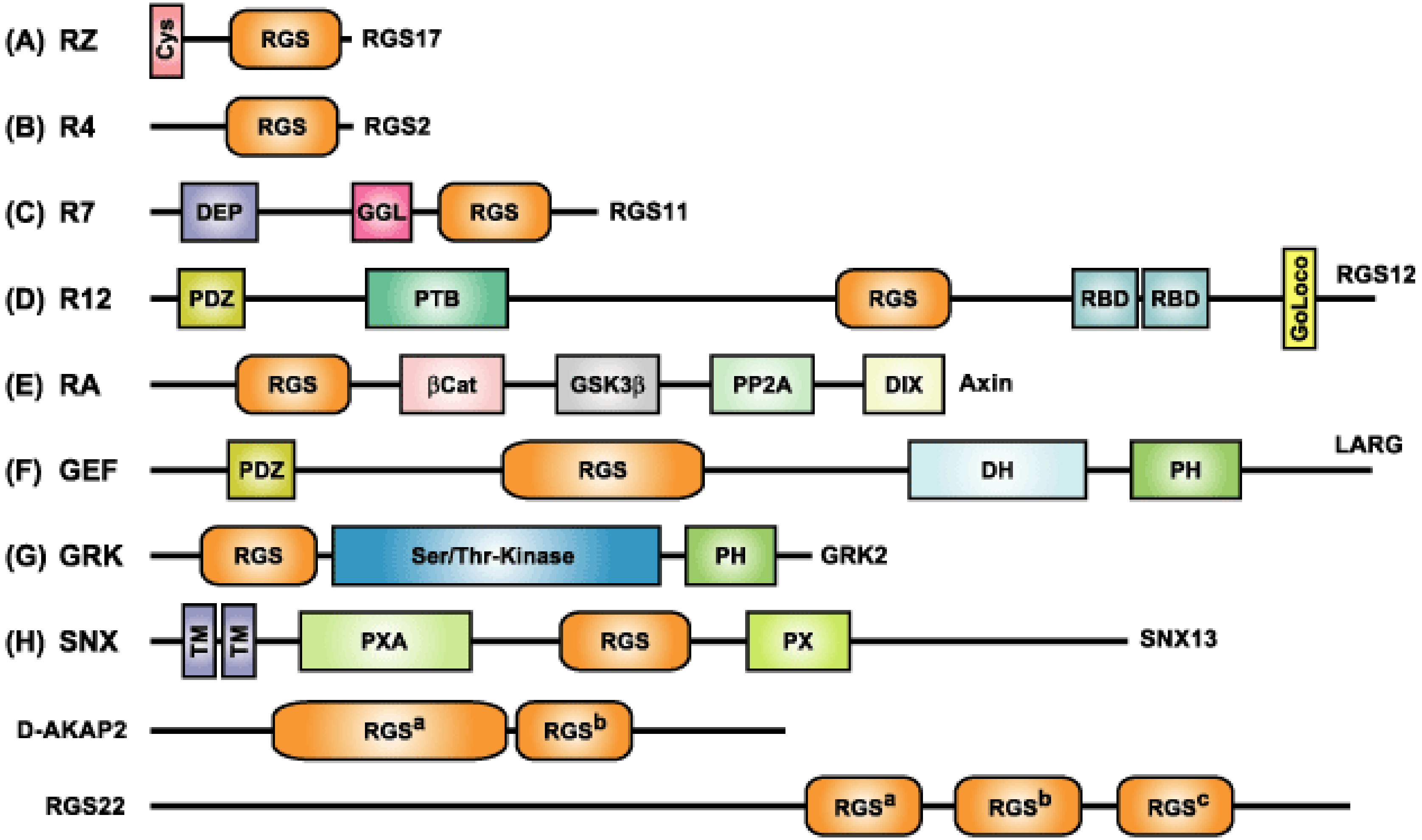


3-D structures of rat Rab Geranylgeranyl Transferase complexed with REP-1, + paralogs.

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



(c) 2004 Siderovski & Willard

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.