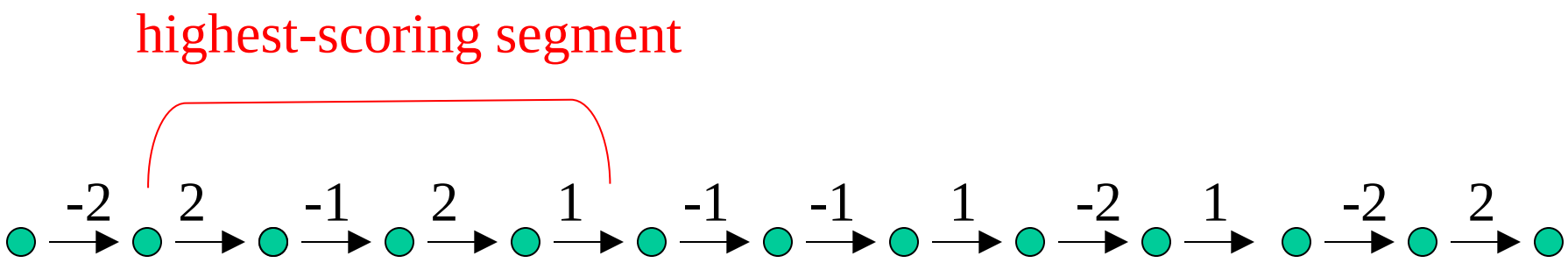
Today's Lecture

- Weighted linked lists
  - Applications: Sequence graphs, "motif clusters", numerical data
  - Statistical issues
  - Finding multiple high-scoring paths/segments

Weighted Linked Lists (WLLs)

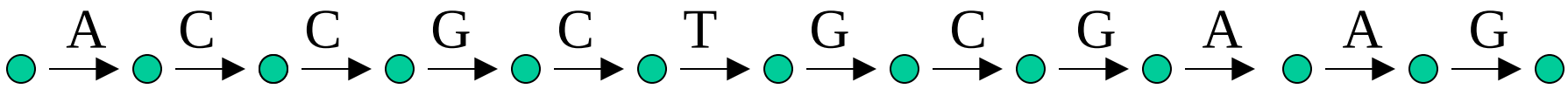
- *WLL* is linked list with weights on each edge
  - simplest kind of WDAG.
- Highest weight paths correspond to highest-scoring segments of WLL.



- Find these segments by dynamic programming
  - Much better than “brute force” algorithm!
- Beginning & end of best path determine path uniquely, so
  - traceback is unnecessary
  - single pass through list suffices to find best path.

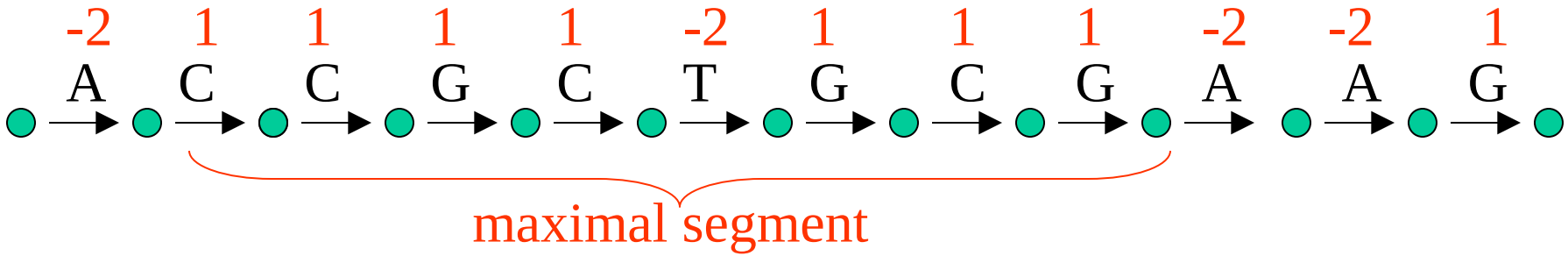
Applications to Sequences

- A *sequence graph* of a sequence is linked list whose edges are labelled by sequence residues (in order):
- e.g. graph for sequence ACCGCTGCGAAG is:



Weighted Sequence Graphs

- If attach weight to each residue, sequence graph becomes a WLL.



- Highest weight paths correspond to highest-scoring segments of sequence.
- Useful for identifying segments with "atypical composition"

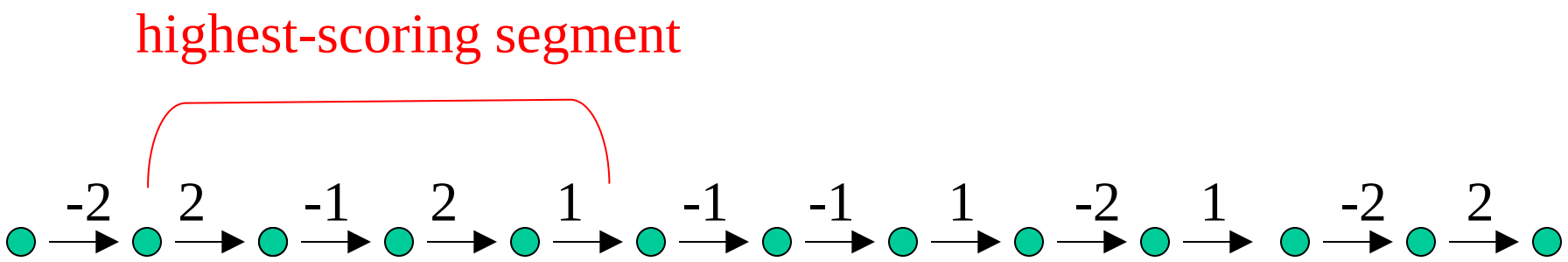
- For example:
  - Gives good way to find GC-rich regions in AT-rich thermophile genomes
    - generally correspond to RNA genes (Rob Klein & Sean Eddy)
  - AT-rich, purine-rich, pyrimidine-rich regions
  - Hydrophobic, acidic, or basic regions in protein sequences

- More broadly, can find regions enriched for sequence *motifs*:
  - CpG islands in mammalian genomes
    - positive weight (e.g. +17) to the first C of each CpG, and
    - negative weight (e.g. –1) to every other base

    (This approach was used in *Nature* human genome paper).
  - *horizontally transferred* regions
  - Regions rich in (known) transcription-factor motifs

Weighted Linked Lists (WLLs)

- *WLL* is linked list with weights on each edge
  - simplest kind of WDAG.
- Highest weight paths correspond to highest-scoring segments of WLL.



WLLs with non-sequence-based scores

- Can also assign scores to each genomic position based on other quantitative info:
  - Next-gen read frequency, e.g.
    - CNVs (Homework 3)
    - Hypersensitive sites
    - CHIP-seq
  - Other measurements?
- Attach scores to *columns* in sequence *alignments*

Important issues!

- What is best scoring system to detect the ‘target regions’?
  - Short answer: $s(r) = \log(t_r / b_r)$ where
    - t_r , b_r are freqs of residue (or motif) r in target and background
    - (if unknown, can sometimes estimate iteratively)
- When is the score of a segment ‘significant’?
  - ∃ theory (due to Karlin & Altschul) for score dist’n for highest-scoring segments in a random sequence

- Will revisit both issues later.

Finding *multiple* high-scoring segments

- In general, expect several regions of particular type in a given sequence – not just one!
- So want to find multiple high-weight paths in a WDAG
- But not interested in slight perturbations of previously found paths
- One strategy:
  - Find highest-weight path
  - 'Mask it' (remove its edges from graph)
  - Repeat above two steps until scores no longer 'interesting'

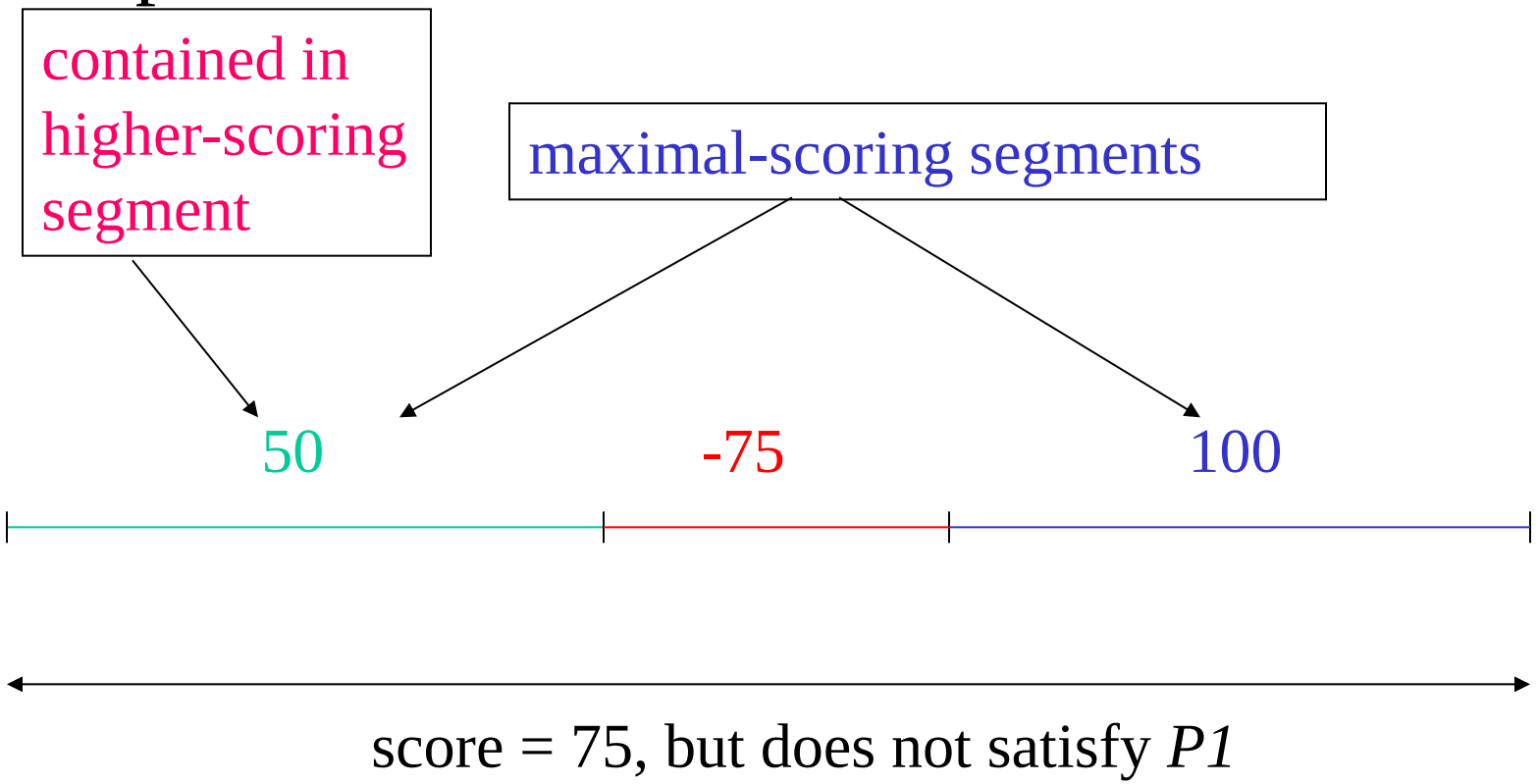
- ∃ more efficient algorithm not requiring repeated scans?
  - Ruzzo & Tompa solved for WLLs
  - ∃ solution for arbitrary WDAGs?

Maximal Segment Analysis – Definitions

- let $\{s_i\}$, $i = 1,\ldots,N$ be sequence of real nos.
  - e.g. scores assigned to
    - residues in a DNA or protein sequence, or
    - columns in an alignment
- *segment* is set of integers of the form

  $[d,e] = \{i \mid d \leq i \leq e\}$ where $1 \leq d \leq e \leq N$.
- *score* of $[d,e]$ is $\Sigma^{e}_{i=d} s_i$

- A *maximal(-scoring) segment* I is one such that
  - *P1:* no subsegment of I has a higher score than I
  - *P2:* no segment properly containing I satisfies *P1*
- Example:



- *Problem*: given $S > 0$, find all maximal segs of score $\geq S$
- Segments are *paths* in a linked-list WDAG with *N+1* vertices and *N* edges
- *Highest weight path* is found by dynamic programming;

  in (pseudo-)pseudocode:

```
cumul = max = 0;  start = 1;
for (i = 1; i ≤ N; i++)  {
    cumul += s[i];
    if (cumul ≤ 0)
        {cumul = 0;  start = i + 1;}  /* NOTE RESET TO ZERO */
    else if (cumul ≥ max)
        {max = cumul;  best_end = i;  best_start = start;}
}
if (max ≥ S) print best_start, best_end, max
```

Maximal segments – from cumulative score plot (*without* 0 resets)