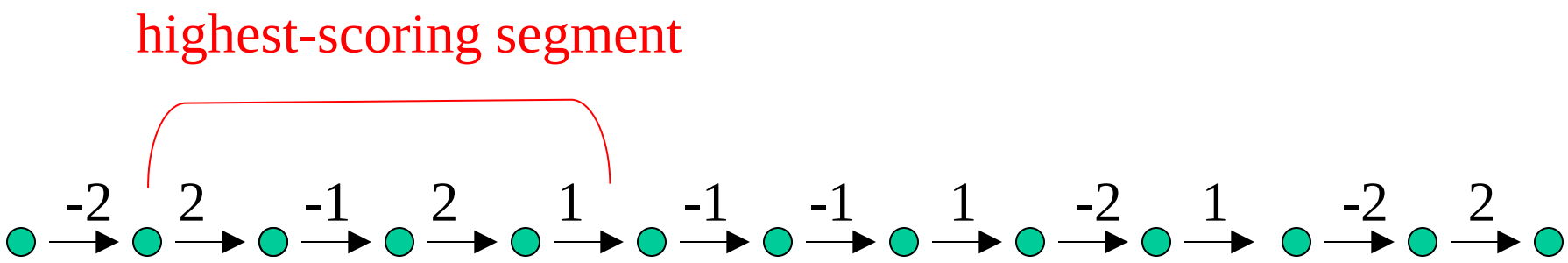
Today's Lecture

- WLLs for numerical data
- Statistical issues
- Finding multiple high-scoring 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.



Non-sequence-based scoring

- 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?

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'

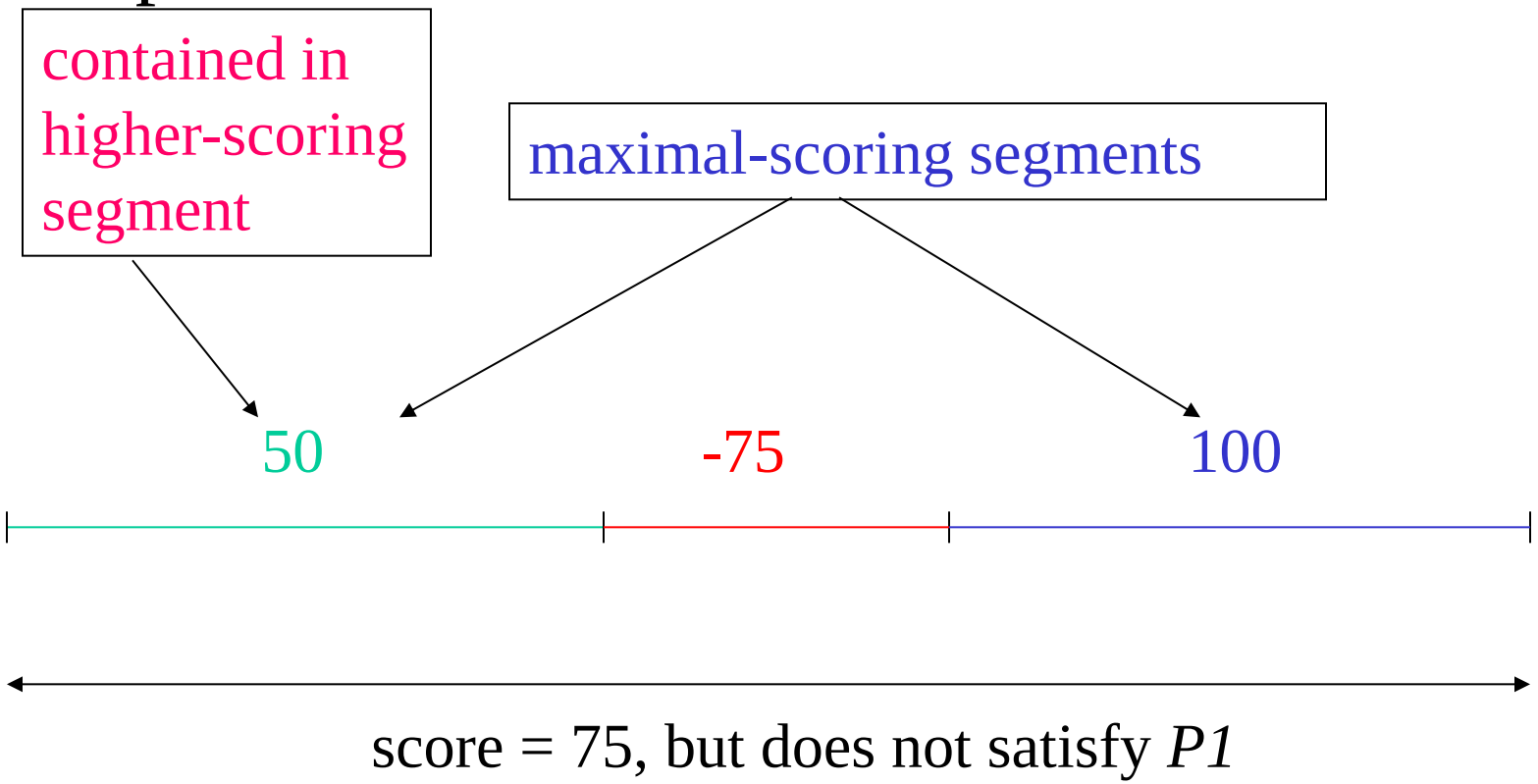
- ∃ more efficient algorithm not requiring repeated scans?
  - Ruzzo & Tompa solved for WLLs
  - ∃ solution for arbitrary WDAGs?

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
```