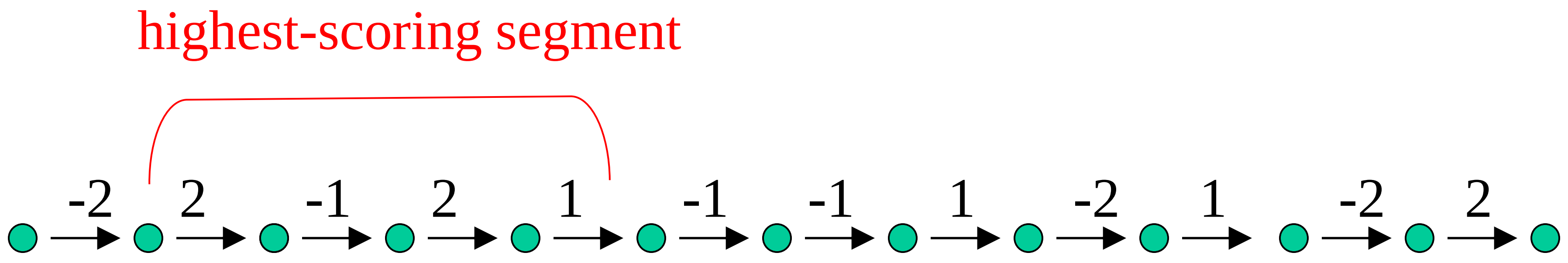
# Lecture 8: Weighted linked lists

- Simplifications to general WDAG algorithm
- Sequence graphs & regions of atypical residue composition
- Defining & interpreting scores
- Other applications
  - motif clusters
  - read count data
  - coding sequence

# Weighted Linked Lists (WLLs)

- *WLL* is linked list with weights on each edge
  - – simplest kind of WDAG.

- Paths = ‘segments’ or ‘regions’



- Find highest-scoring 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.

## *from lecture 7 :*

- To reconstruct best path, need "traceback" pointer to immediate predecessor of *v* in best path:

$$T(v) = \begin{cases} v & w(v) = 0 \\ \arg\max_{u \in \text{parents}(v)} \left(w(u) + w((u,v))\right) & w(v) \neq 0 \end{cases}$$

  - in preceding graph, *T*(*v*) is the *parent* on *red edge* coming into *v*
    - if more than one such edge, pick one at random;
    - if no such edge, *T*(*v*) = *v*
- Sometimes useful to record *beginning* of best path:

$$B(v) = \begin{cases} v & w(v) = 0 \\ B(T(v)) & w(v) \neq 0 \end{cases}$$

# Implementing Dynamic Programming in a Computer Program

- Storing entire graph has space complexity = $O(|V|+|E|)$
- If graph has regular structure, can often "create" and process vertices and edges on the fly, without storing in memory
  - cf. edit graph (to be defined later) for aligning sequences

- ***Highest weight path*** via dynamic programming (no explicit graph):

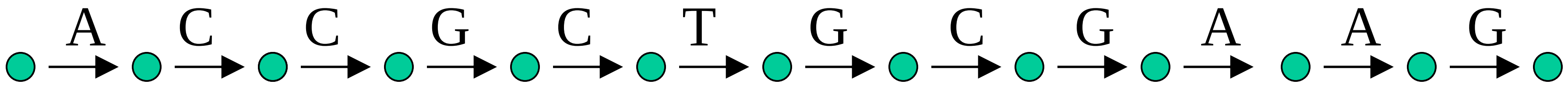
  in (pseudo-)pseudocode:

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

- Correspondence to (implicit) WLL
  - i labels *edges*
  - cumul = w(v) (where v is vertex at end of edge i)
  - max = best w(v) so far
  - best_end = i corresponding to edge ending at best w(v) so far
  - start = edge following B(v)

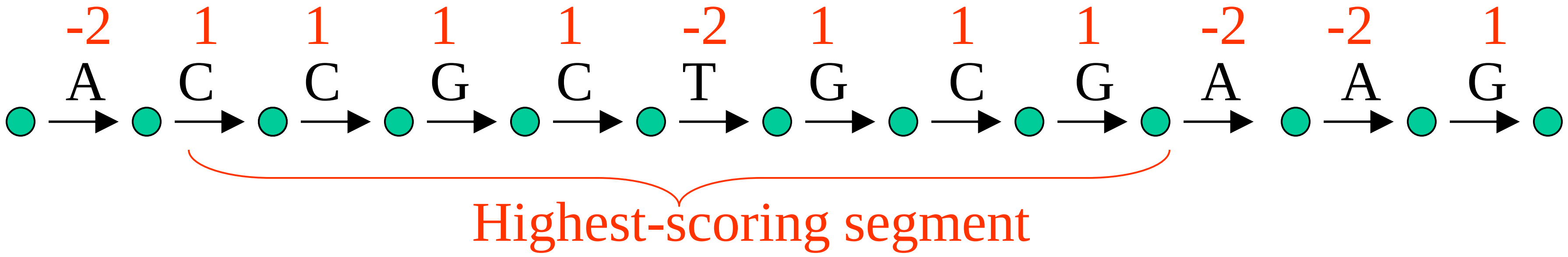
# Applications to Sequences

- 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.



- Useful for identifying sequence regions (‘target regions’) with atypical composition:

- In DNA:
  - GC-rich regions in AT-rich thermophile genomes
    - generally correspond to RNA genes (Rob Klein & Sean Eddy)
  - *horizontally transferred* regions
  - isochores (mammalian DNA)
- In proteins:
  - hydrophobic regions
    - transmembrane segments
  - hydrophilic regions
    - loops, intrinsically disordered regions
  - acidic or basic regions

## ‘Optimal’ scores

- *Assume* sequence consists of
  - *target regions* with residue freqs $t_r$
  - *background regions* with residue freqs $b_r$
  - *independence assumption* applies in both
- *Then* ‘best’ scoring system to detect the target regions uses LLRs:

$$s(r) = \log_a(t_r / b_r)$$

- if residue freqs are unknown, can usually estimate iteratively

# Karlin / Altschul approximation

- for $s(r) = \log_a(t_r / b_r)$, expected # segments of score $\geq S$ in (random) backgd seq of length N

$$\approx NK a^{-S}$$

- for some constant $K$ (not depending on $S$)
- Note that $a^{-S} = a^{-LLR} = 1 / LR$

so (apart from $K$) this is essentially the observation in lecture 6:

# *(from lecture 6)* *Average* likelihood ratios

- ***average LR*** (for sites) ≈ ***average spacing*** between occurrences of 'site-like' sequences ***in background***
- So e.g. for 3' splice sites
  - if the average *LR* is 1000, then one expects 'splice-site-like' sequences to occur on average once per kb *in background sequence*
  - ***N.B.*** This says nothing about the frequency of *actual* splice sites! (which could be greater or smaller than 1 per kb), and so doesn't by itself provide the probability that an *apparent* splice site is an *actual* site.

## Example: Finding 60% G+C regions in 40% G+C genomes

- Score system (LLR) :
  - $s(\mathrm{C}) = s(\mathrm{G}) = \log_{10}(.3 / .2) = .176$
  - $s(\mathrm{A}) = s(\mathrm{T}) = \log_{10}(.2 / .3) = -.176$
- Avg score (per position) in hiGC region:

  $.3\ .176 + .3\ .176 + .2\ (-.176) + .2(-.176) = .0352$
- 100-base hiGC region:

  Score ~3.52; occurs once per $\sim 10^{3.52} \approx 3.3$ kb in backgd
- 300-base hiGC region:

  Score ~10.5; occurs once per $\sim 10^{10.5} \approx 36$ Gb

- So:
  - 300-base hiGC regions are unexpected in background
    - (likely have biological ‘significance’)
  - but 100-base such regions are frequent
- Caveats to preceding:
  - Applies to a specific assumed hiGC composition
  - we ignored K
    - Can get more precise results by simulation

## Can use *non-residue-based* scores to find:

- Regions enriched in particular 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).
  - Regions rich in (known) transcription-factor motifs
  - Optimal scores are LLRs, but now based on 'symbol frequencies' (where symbol = presence/absence of motif)

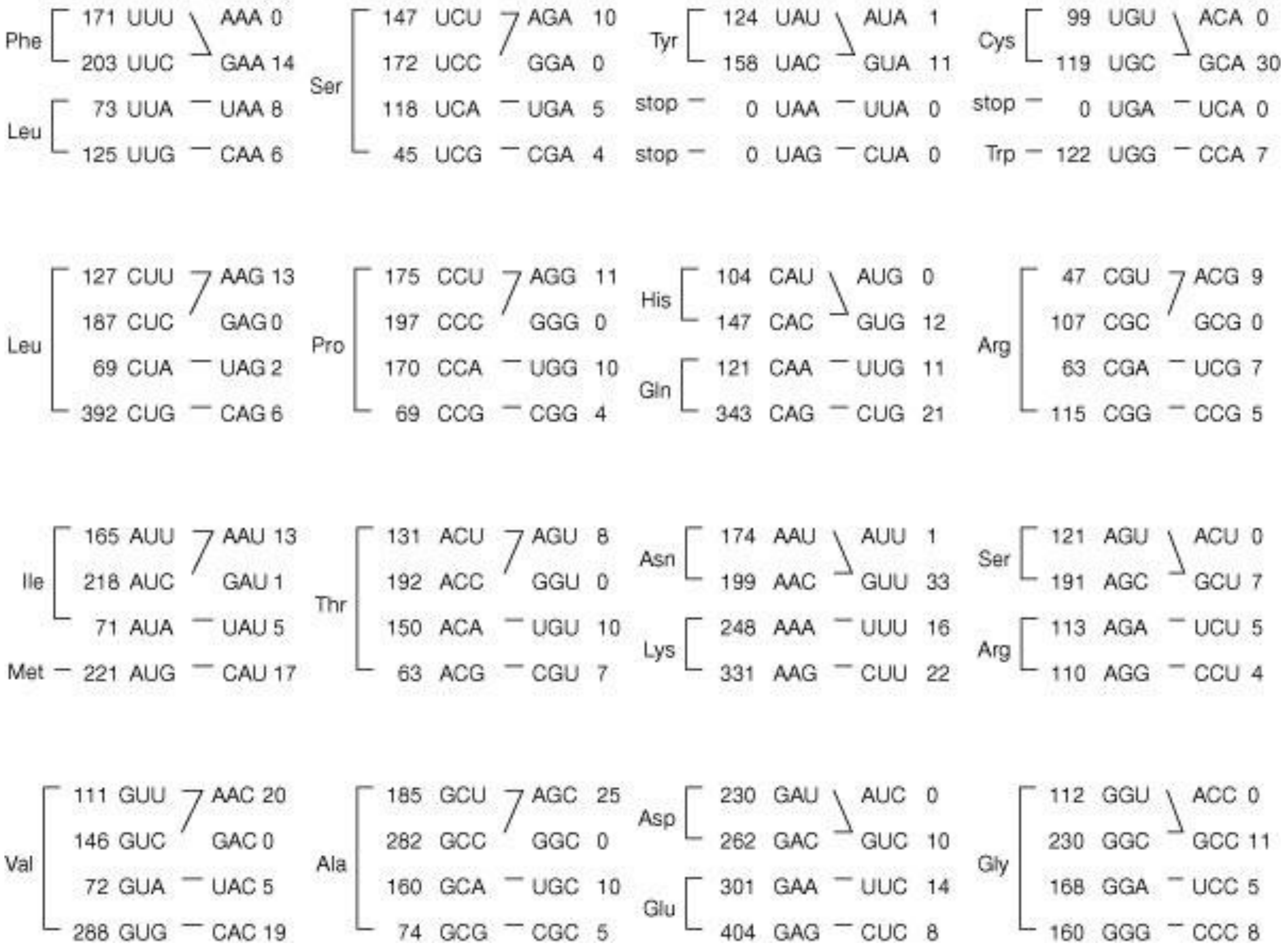
## CpG Islands

- Regions in mammalian genomes where CpGs are *not* significantly underrepresented
  - likely not methylated in the germ line
- Found at 5’ ends of ~60% of protein-coding genes (& in some RNA genes);
  - frequently extends into first exon & even coding sequence
- More likely to be associated with “housekeeping” genes (expressed in all or most cells),
  - less frequently with tissue-specific genes
- May play regulatory role (methylation of island can shut down gene)
- Substantial length variation: 75% are less than 850bp, but longest (in human) is 37kb

- Regions targeted by *next-gen read experiments* (symbols = *read start counts*)
  - CNVs (Homework 6)
  - Hypersensitive sites
  - CHIP-seq
- Conserved regions in *sequence alignments* (symbols = *alignment columns*)

- Coding sequences (symbols = *successive trinucleotides within a reading frame*)
  - Target freqs: codon freqs in known coding seq
  - Background: trinuc freqs in background

From *Initial sequencing and analysis of the human genome*, International Human Genome Sequencing Consortium, *Nature* 409, 860-921 (2001)



| | | | | |
|---|---|---|---|---|
| Phe | 171 | UUU | AAA | 0 |
| | 203 | UUC | GAA | 14 |
| Leu | 73 | UUA | UAA | 8 |
| | 125 | UUG | CAA | 6 |
| Ser | 147 | UCU | AGA | 10 |
| | 172 | UCC | GGA | 0 |
| | 118 | UCA | UGA | 5 |
| | 45 | UCG | CGA | 4 |
| Tyr | 124 | UAU | AUA | 1 |
| | 158 | UAC | GUA | 11 |
| stop | 0 | UAA | UUA | 0 |
| stop | 0 | UAG | CUA | 0 |
| Cys | 99 | UGU | ACA | 0 |
| | 119 | UGC | GCA | 30 |
| stop | 0 | UGA | UCA | 0 |
| Trp | 122 | UGG | CCA | 7 |
| Leu | 127 | CUU | AAG | 13 |
| | 187 | CUC | GAG | 0 |
| | 69 | CUA | UAG | 2 |
| | 392 | CUG | CAG | 6 |
| Pro | 175 | CCU | AGG | 11 |
| | 197 | CCC | GGG | 0 |
| | 170 | CCA | UGG | 10 |
| | 69 | CCG | CGG | 4 |
| His | 104 | CAU | AUG | 0 |
| | 147 | CAC | GUG | 12 |
| Gln | 121 | CAA | UUG | 11 |
| | 343 | CAG | CUG | 21 |
| Arg | 47 | CGU | ACG | 9 |
| | 107 | CGC | GCG | 0 |
| | 63 | CGA | UCG | 7 |
| | 115 | CGG | CCG | 5 |
| Ile | 165 | AUU | AAU | 13 |
| | 218 | AUC | GAU | 1 |
| | 71 | AUA | UAU | 5 |
| Met | 221 | AUG | CAU | 17 |
| Thr | 131 | ACU | AGU | 8 |
| | 192 | ACC | GGU | 0 |
| | 150 | ACA | UGU | 10 |
| | 63 | ACG | CGU | 7 |
| Asn | 174 | AAU | AUU | 1 |
| | 199 | AAC | GUU | 33 |
| Lys | 248 | AAA | UUU | 16 |
| | 331 | AAG | CUU | 22 |
| Ser | 121 | AGU | ACU | 0 |
| | 191 | AGC | GCU | 7 |
| Arg | 113 | AGA | UCU | 5 |
| | 110 | AGG | CCU | 4 |
| Val | 111 | GUU | AAC | 20 |
| | 146 | GUC | GAC | 0 |
| | 72 | GUA | UAC | 5 |
| | 288 | GUG | CAC | 19 |
| Ala | 185 | GCU | AGC | 25 |
| | 282 | GCC | GGC | 0 |
| | 160 | GCA | UGC | 10 |
| | 74 | GCG | CGC | 5 |
| Asp | 230 | GAU | AUC | 0 |
| | 262 | GAC | GUC | 10 |
| Glu | 301 | GAA | UUC | 14 |
| | 404 | GAG | CUC | 8 |
| Gly | 112 | GGU | ACC | 0 |
| | 230 | GGC | GCC | 11 |
| | 168 | GGA | UCC | 5 |
| | 160 | GGG | CCC | 8 |

# Coding sequences in prokaryotes

- Starting model:
  - codon freqs 1 / 61
  - background freqs 1 / 64 (equal freq assumption)
  - score per codon: $\log_{10}(64 / 61) = .021$
  - times 300 codons (typical gene len): 6.25
  - $10^{6.25} = 1.8$ Mb
  - Requirement for starting ATG – factor of 64
  - Can detect typical genes without any amino acid or codon bias info!
- Get better freqs from these predictions; iterate