Lecture 13
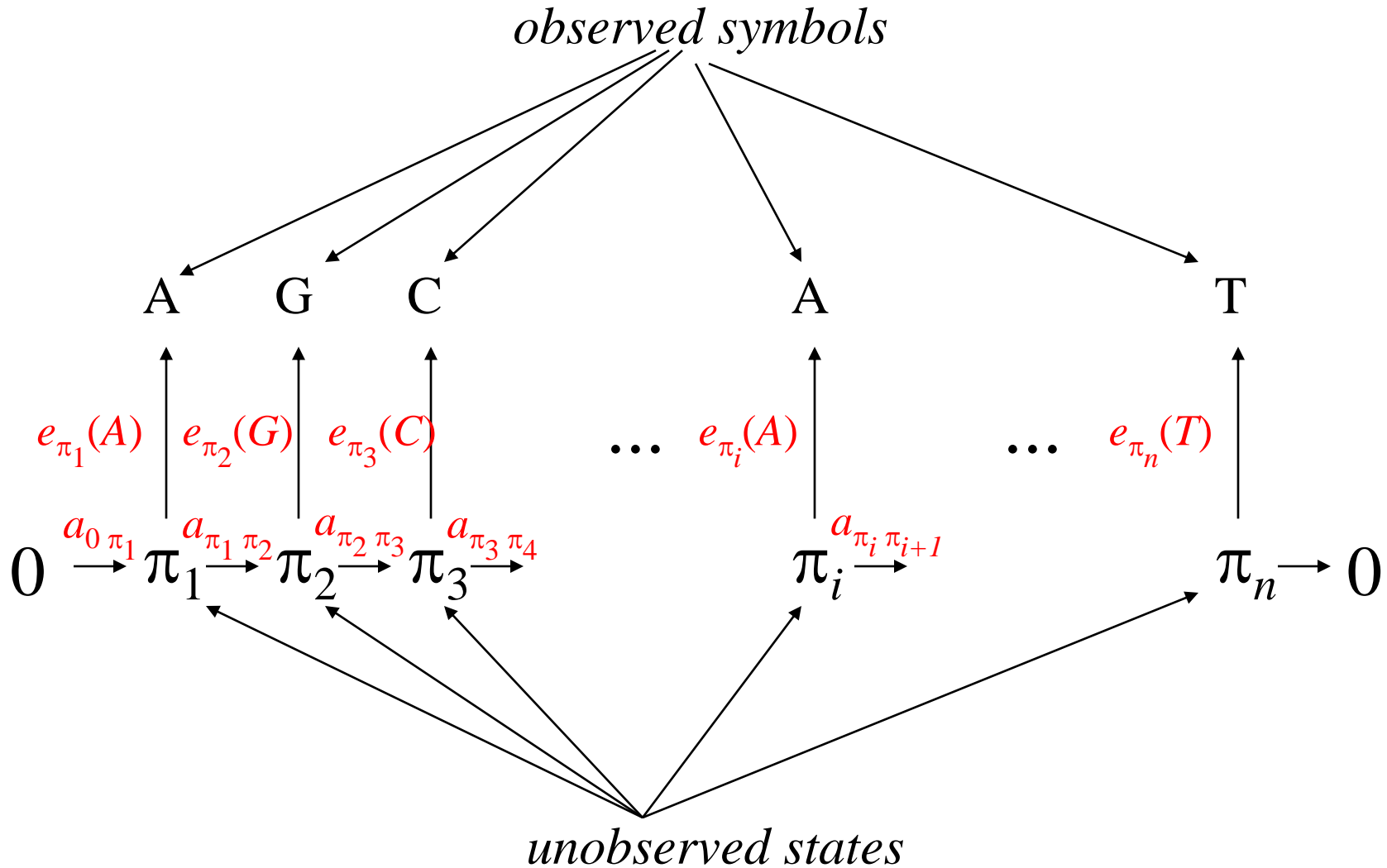
- HMM probability calculations
  - WDAG
  - Viterbi algorithm

- 2-state HMMs & D-segments

Hidden Markov Model

observed symbols

$e_{\pi_1}(A)$ $e_{\pi_2}(G)$ $e_{\pi_3}(C)$ ... $e_{\pi_i}(A)$ ... $e_{\pi_n}(T)$

$0 \xrightarrow{a_{0\,\pi_1}} \pi_1 \xrightarrow{a_{\pi_1\,\pi_2}} \pi_2 \xrightarrow{a_{\pi_2\,\pi_3}} \pi_3 \xrightarrow{a_{\pi_3\,\pi_4}} \quad \pi_i \xrightarrow{a_{\pi_i\,\pi_{i+1}}} \quad \pi_n \to 0$

unobserved states

HMM Probabilities of Sequences

- Prob of sequence of states $\pi_1 \pi_2 \pi_3 \dots \pi_n$ is

  $a_{0\pi_1} a_{\pi_1 \pi_2} a_{\pi_2 \pi_3} a_{\pi_3 \pi_4} \dots a_{\pi_{n-1} \pi_n}.$

- Prob of seq of observed symbols $b_1 b_2 b_3 \dots b_n$,

  conditional on state sequence is

  $e_{\pi_1}(b_1) e_{\pi_2}(b_2) e_{\pi_3}(b_3) \dots e_{\pi_n}(b_n)$

- Joint probability $= a_{0\pi_1} \prod_{i=1}^{n} a_{\pi_i \pi_{i+1}} e_{\pi_i}(b_i)$

  (define $a_{\pi_n \pi_{n+1}}$ to be 1)

- (Unconditional) prob of observed sequence

  $=$ sum (of joint probs) over all possible state paths

  – not practical to compute directly, by 'brute force'! We will use dynamic programming.
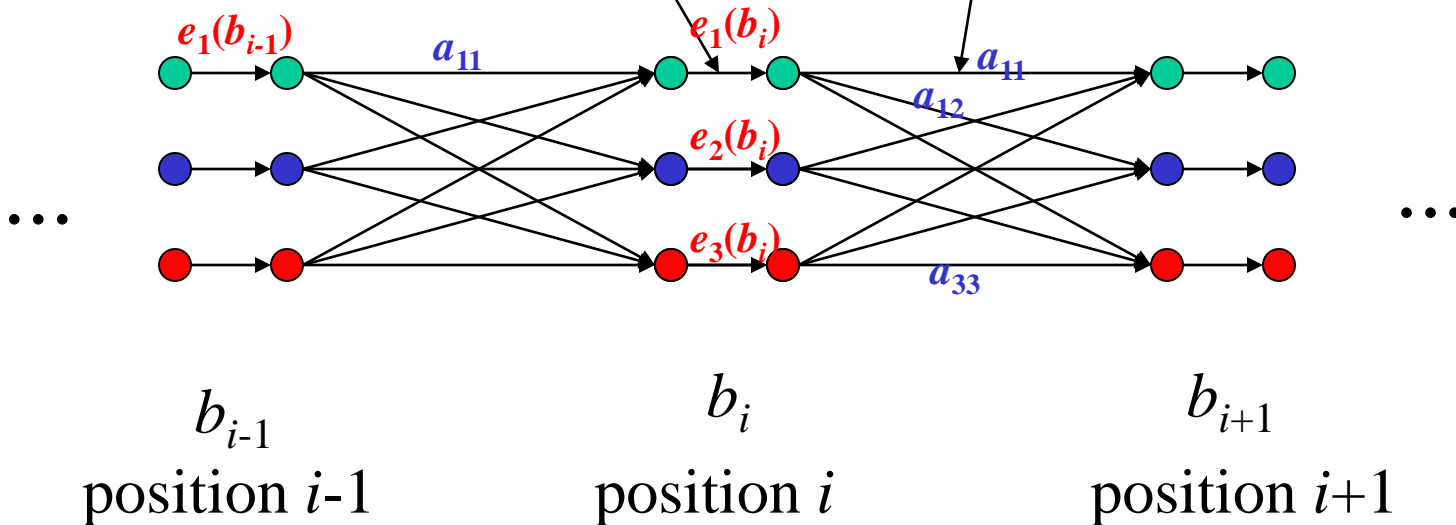
Computing HMM Probabilities

- WDAG structure for sequence HMMs:
  - for i^{th} position in seq ($i = 1, \ldots n$), have 2 nodes for each state:
    - total # nodes = $2ns + 2$, where n = seq length, s = # states
  - Pair of nodes for a given state at i^{th} position is connected by an *emission* edge
    - Weight is the emission prob for i^{th} observed residue
    - Can omit node pair if emission prob = 0
  - Have *transition* edges connecting (right-hand) state nodes at position i with (left-hand) state nodes at position $i+1$
    - Weights are transition probs
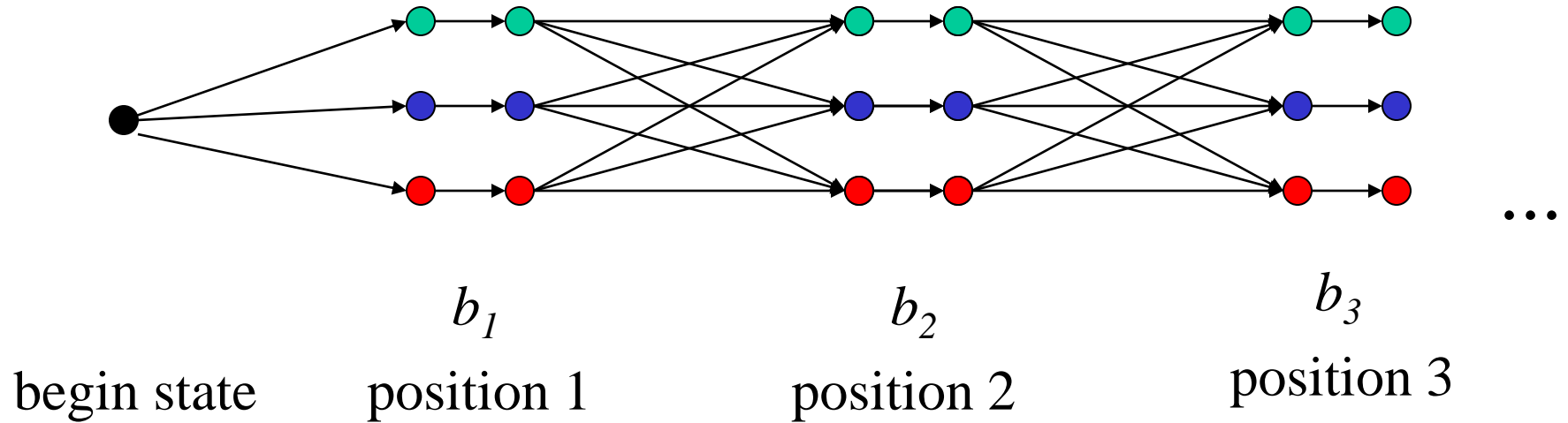   - Can omit edges with transition prob = 0

WDAG for 3-state HMM, length n sequence

weights are emission probabilities $e_k(b_i)$ for i^{th} residue b_i

weights are transition probabilities a_{kl}

$e_1(b_{i-1})$ a_{11} $e_1(b_i)$ a_{11}

a_{12}

$e_2(b_i)$

$e_3(b_i)$ a_{33}

$\cdots$ $\cdots$

b_{i-1}
position i-1

b_i
position i

b_{i+1}
position i+1

Beginning of Graph

b_1

b_2

b_3

begin state

position 1

position 2

position 3

- ***Paths*** through graph from begin node to end node correspond to ***sequences of states***
- ***Product weight*** along path
  = ***joint probability*** of state sequence & observed symbol sequence
- ***Highest-weight path*** = ***highest probability state sequence***
- ***Sum*** *of (product) path weights,* ***over all paths***,
  = ***probability of observed sequence***
- ***Sum*** *of (product) path weights* ***over***
  – all paths *going through a particular node*, or
  – all paths *that include a particular edge*,
  divided by prob of observed sequence,
    = ***posterior probability*** of that edge or node

Path Weights

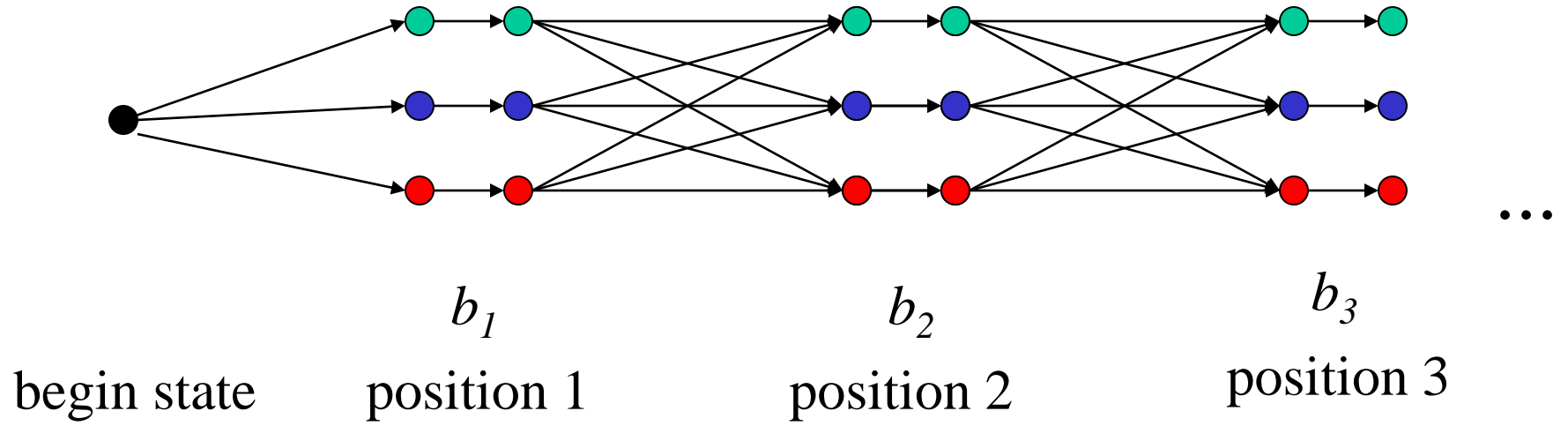position *i*-1   position *i*   position *i*+1
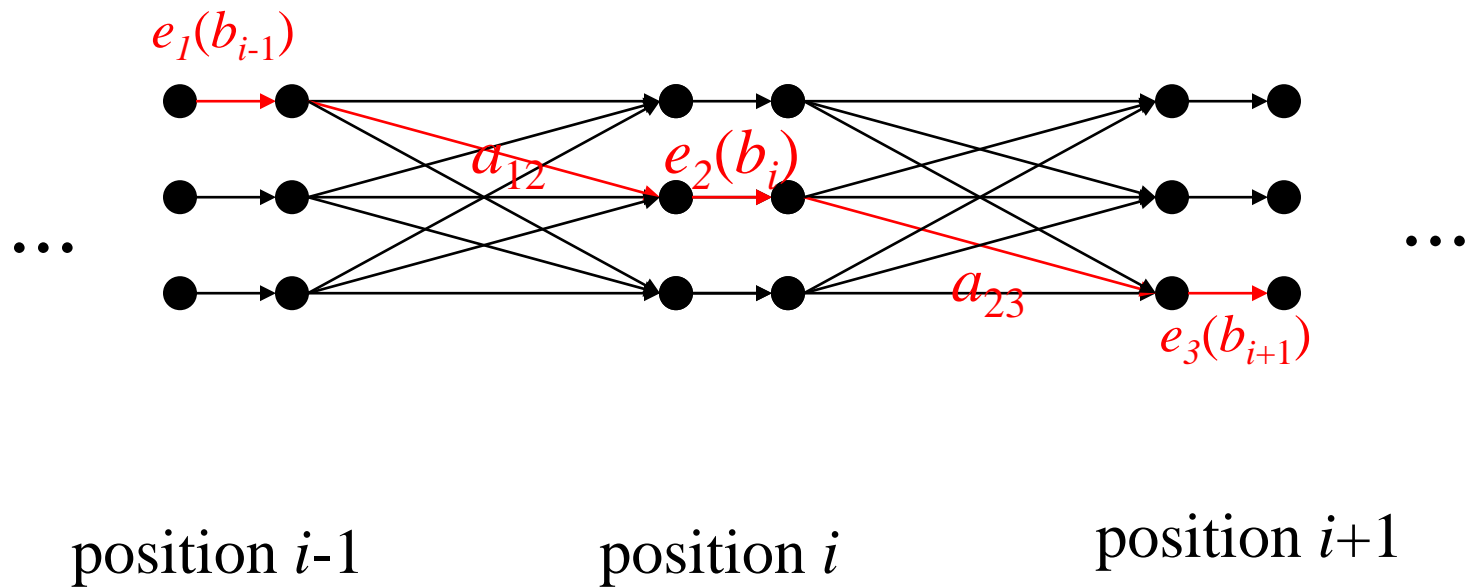
- By general results on WDAGs, can use dynamic programming to find highest weight path:

  = "Viterbi algorithm" to find highest probability path (most probable "parse")

  – in this case can use log probabilities & sum weights

  – (N.B. paths are constrained to begin at the begin node!)

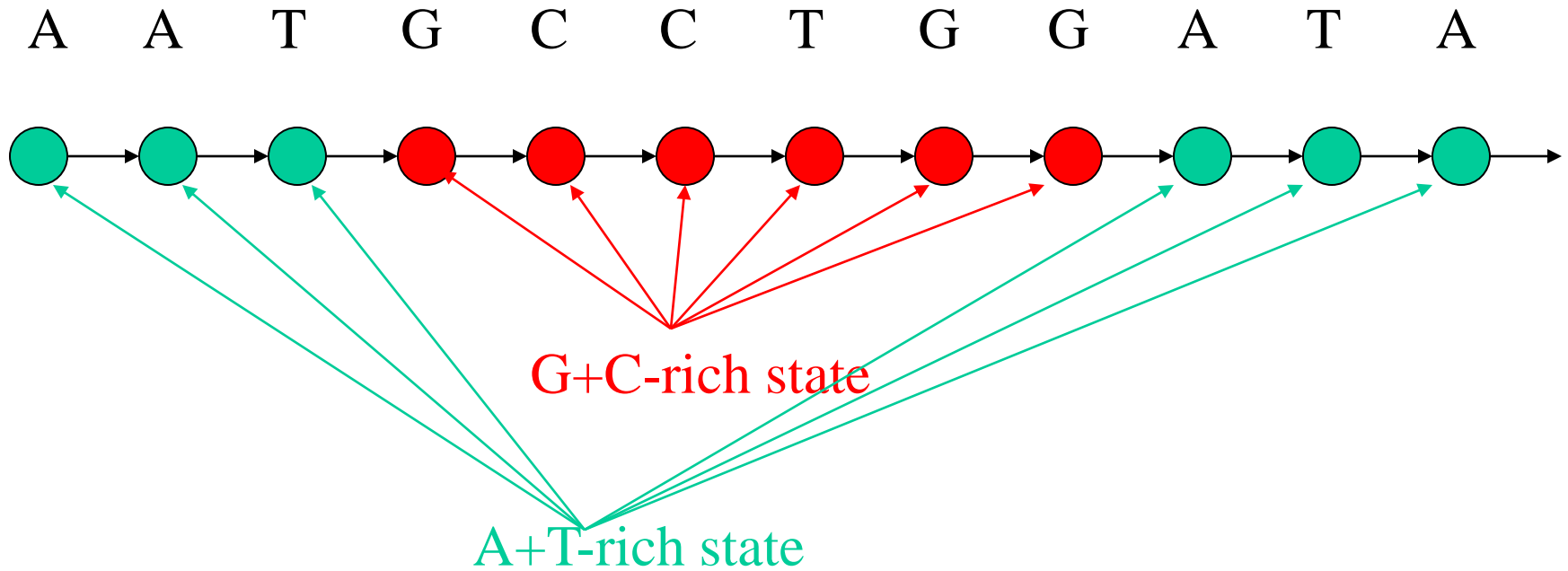# The Viterbi path is
the *most probable parse*!

Complexity

- $= O(|V|+|E|)$, i.e. total # nodes and edges.
- # nodes $= 2ns + 2$
  - where $n =$ sequence length,
  - $s = $ # states.
- # edges $= (n - 1)s^2 + ns + 2s$

- So overall complexity is $O(ns^2)$
  - (actually s^2 can be reduced to # 'allowed' transitions between states – depends on model topology).
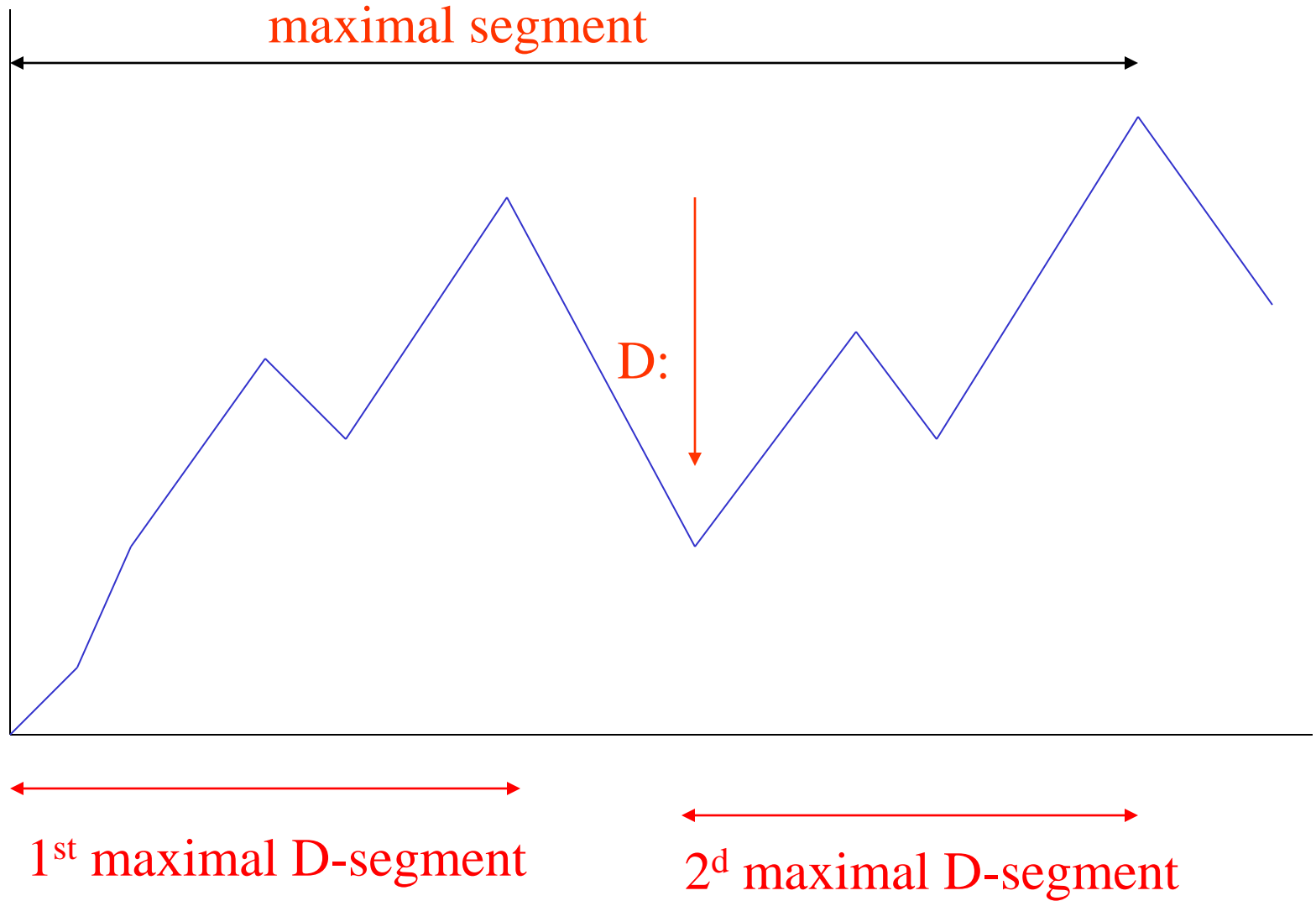
b_1

b_2

b_3

begin state

position 1

position 2

position 3

...

2-state HMMs & D-segments

A   A   T   G   C   C   T   G   G   A   T   A

G+C-rich state

A+T-rich state

from lecture 7

maximal segment

D:

1ˢᵗ maximal D-segment

2ᵈ maximal D-segment

- $O(N)$ algorithm to find all maximal D-segs:

```
cumul = max = 0; start = 1;
for (i = 1; i ≤ N; i++) {
    cumul += s[i];
    if (cumul ≥ max)
        {max = cumul; end = i;}
    if (cumul ≤ 0 or cumul ≤ max + D or i == N) {
        if (max ≥ S)
            {print start, end, max; }
        max = cumul = 0; start = end = i + 1; /* NO BACKTRACKING
            NEEDED! */
    }
}
```

D-segments ≈ 2-state HMMs

- Consider 2-state HMM
  - states 1 & 2, transition probs $a_{11}, a_{12}, a_{21}, a_{22}$
  - observed symbols $\{r\}$, emission probs $\{e_1(r)\}, \{e_2(r)\}$
- Define

  scores $s(r) = \log(e_2(r) \, a_{22}/(e_1(r) \, a_{11}))$

  $S = -D = \log(a_{11} a_{22}/(a_{21} a_{12}))$

- Then if $S > 0$, the maximal D-segments in a sequence

  $(r_i)_{i=1,n}$ are the state-2 segments in the Viterbi parse

- (can allow for non-.5 initiation probs by starting cumul at non-zero value)

D-segments vs HMMs

- D-segments
  - are very *easy to program*!
  - give Viterbi parse in *just one pass* through the sequence
  - somewhat more flexible (S, D settings)
- HMMs
  - allow more powerful parameter *estimation*
  - can attach *probabilities* to alternative decompositions
  - easily generalize to *> 2 types* of segments– just allow more states