String Alignment

CMSC423 Fall 2015

Hector Corrada Bravo

For today

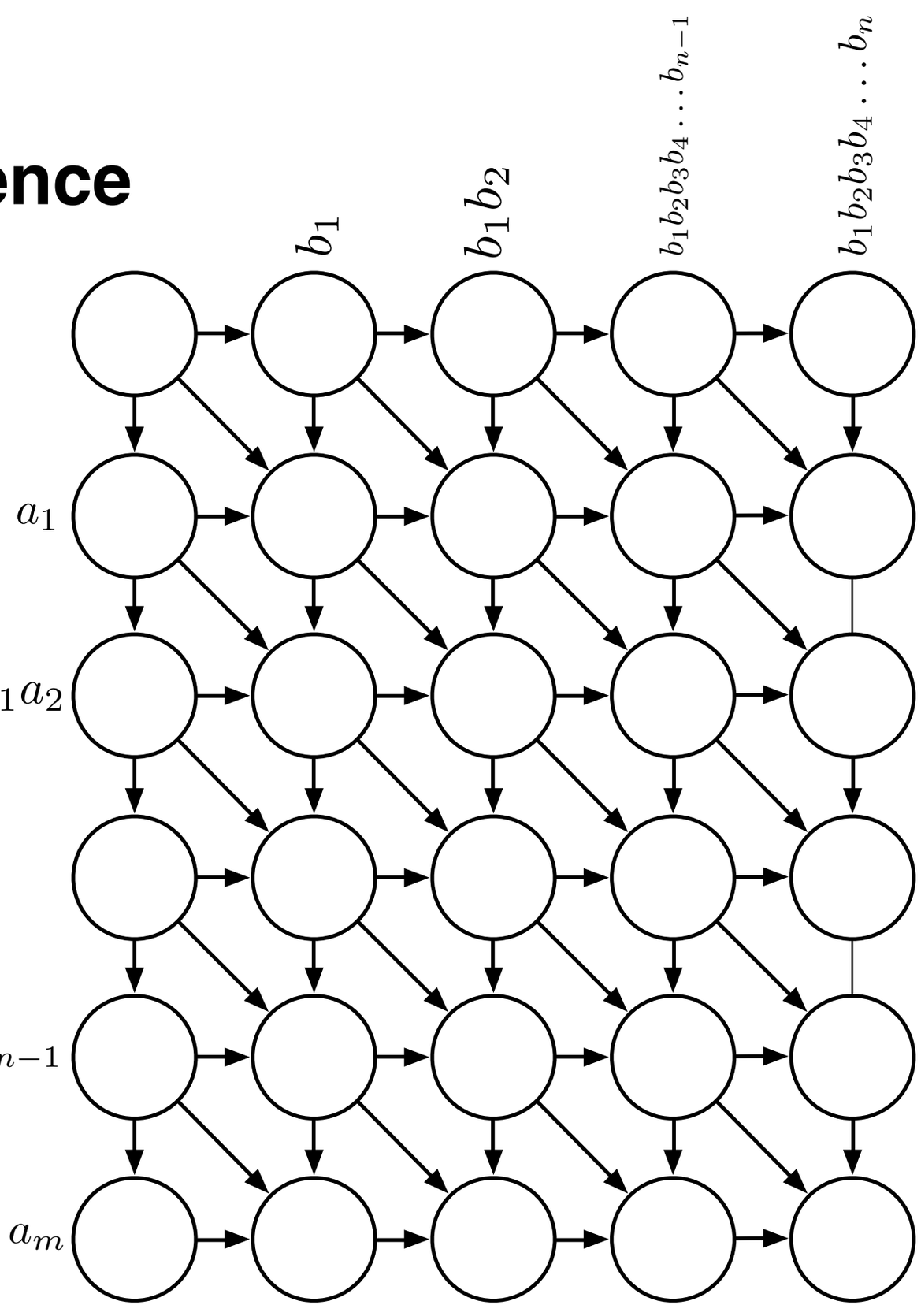
- Scoring matrices
- Local alignment
- Affine gap penalties

Recursive Solution for Longest Common Subsequence

$$a = a_1 a_2 a_3 a_4 \ldots a_m$$

$$b = b_1 b_2 b_3 b_4 \ldots b_n$$

$$s_{i,j} = \max \begin{cases} s_{i-1,j} & \text{char in } a \text{ unmatched} \\ s_{i,j-1} & \text{char in } b \text{ unmatched} \\ s_{i-1,j-1} + 1, \text{ if } a_i = b_j & \text{match!} \end{cases}$$

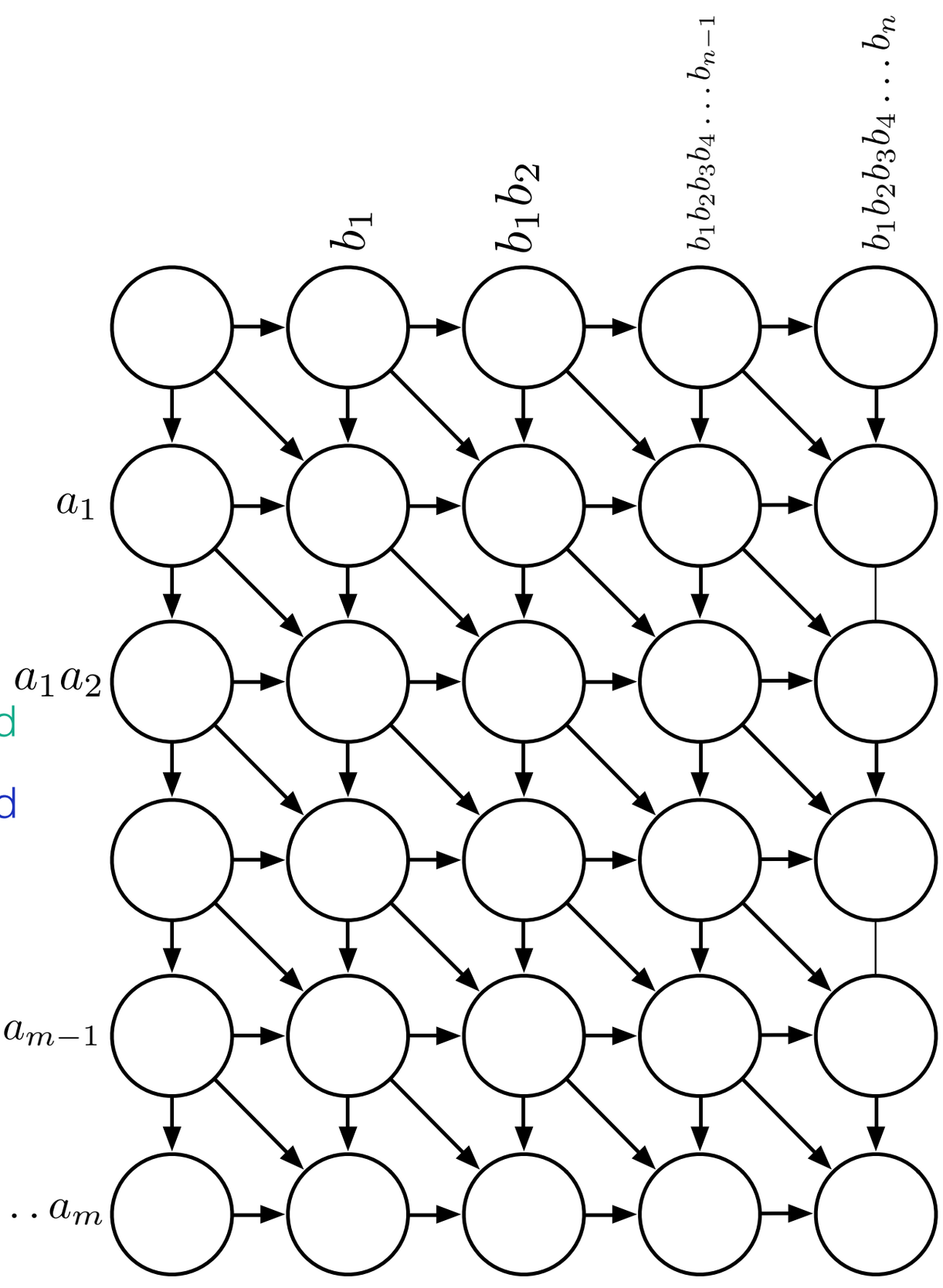


Global Alignment

Generalizing LCS:

- *scoring alignments*
- *gap penalty*

$$s_{i,j} = \max \begin{cases} s_{i-1,j} - \sigma & \text{char in } a \text{ unmatched} \\ s_{i,j-1} - \sigma & \text{char in } b \text{ unmatched} \\ s_{i-1,j-1} + \text{SCORE}(a_i, b_j), & \text{align!} \end{cases}$$



Guiding principles of scores in alignments

- Sequence is said to have diverged from a common ancestor through mutations
  - Substitutions
  - Insertions and deletions (gaps)

- Score evolutionarily close alignments higher than those that are not

- That is we compute the **likelihood ratio** of an alignment given the two sequences are related versus not related

Log odds score

- Let X be a random variable representing an alignment
- Let M_1 and M_2 be two probabilistic models for X
- Log odds score $S(X)$

$$S(X) = \log \frac{P(X|M_1)}{P(X|M_2)}$$

- If $S(X)>0$, X is more likely to come from model M_1
- If $S(X)<0$, X is more likely to come from model M_2

What are M_1 and M_2 in our sequence alignment problem

- M_1: foreground model, that is the sequences are "related by evolution".

- M_2: background model, that is the sequences are unrelated

- Need to compute the probability of an alignment X, under the two models M_1 and M_2

- Assume alignments on **protein sequences** with no gaps.

M_1: foreground model

- Assume each pair of aligned positions evolved from a common ancestor
- Let p_{ab} be the probability of observing a pair $\{a,b\}$
- Probability of an alignment between x and y is

$$P(x, y|M_1) = \prod_{i=1}^{n} p_{x_i y_i}$$

M_2: background model

- Assume the individual amino acids at a position are independent of the amino acid in another position.

- Let q_a be the probability of amino acid a

- The probability of an n-character alignment of x and y is

$$P(x, y|M_2) = \prod_{i=1}^{n} q_{x_i} \prod_{i=1}^{n} q_{y_i}$$

Computing the log odds ratio to score an alignment

- The score of an alignment is the log odds ratio of the two sequences from M_1 and M_2

$$S = \log \frac{P(x, y|M_1)}{P(x, y|M_2)}$$

$$S = \log \frac{\prod_{i=1}^{n} p_{x_i y_i}}{\prod_{i=1}^{n} q_{x_i} q_{y_i}}$$

Computing the log odds ratio to score an alignment

$$S = \sum_{i=1}^{n} \log \frac{p_{x_i y_i}}{q_{x_i} q_{y_i}}$$

Score of an alignment

$$s(a,b) = \log \frac{p_{a,b}}{q_a q_b}$$

Substitution matrix entry

Some common substitution matrices

- BLOSUM matrices [Henikoff and Henikoff, 1992]
  - BLOSUM45
  - BLOSUM50
  - BLOSUM62
    - Number represents percent identity of sequences used to construct substitution matrices
- PAM [Dayhoff et al, 1978]
- Empirically, BLOSUM62 works the best

How to estimate the probabilities?

- Need a good set of confirmed alignments
- Depends upon what we know about when the two sequences might have diverged
  - p_{ab} for closely related species is likely to be low if $a \; != b$
  - p_{ab} for species that have diverged a long time ago is likely close to the background.

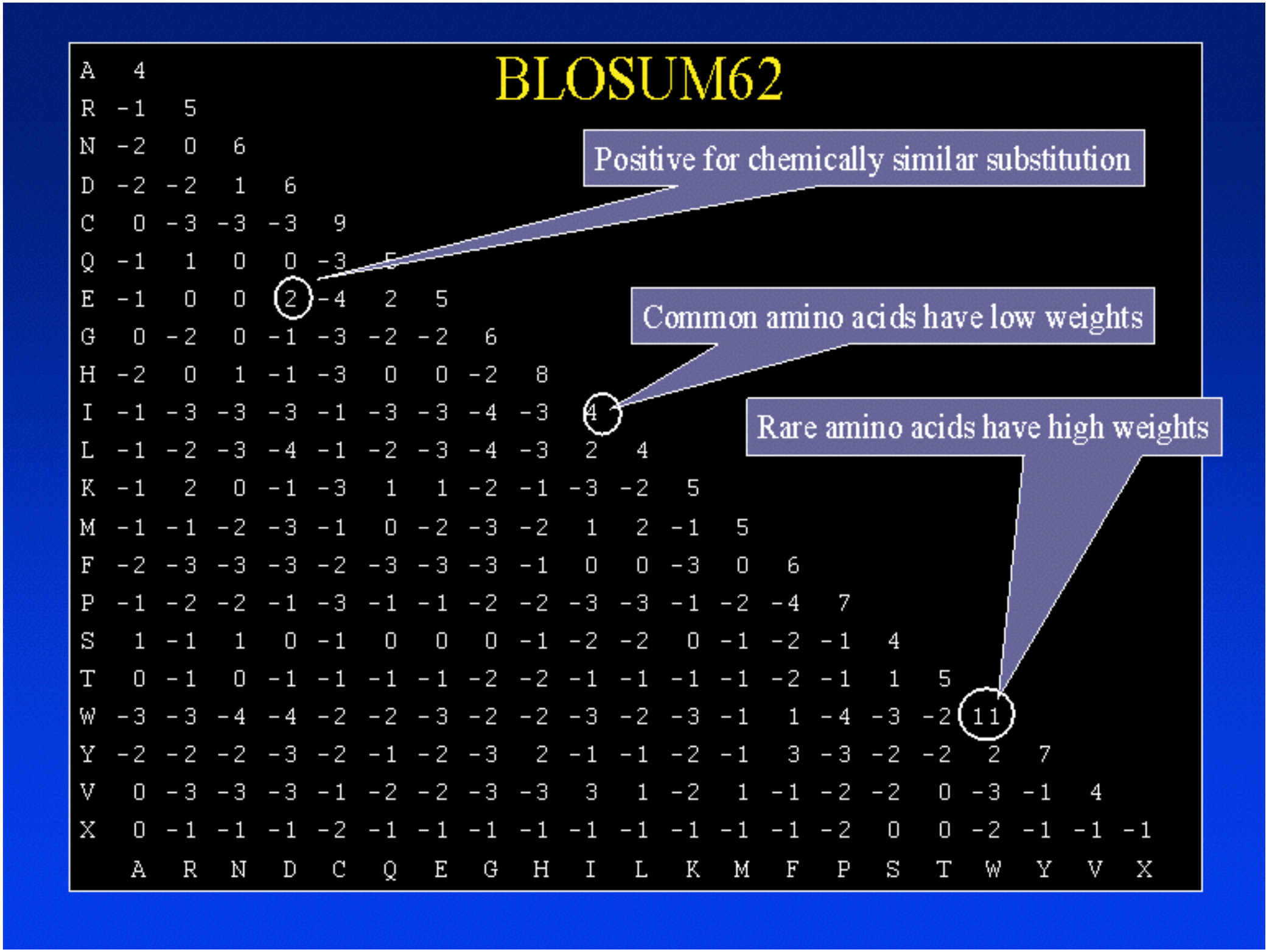
BLOSUM matrices

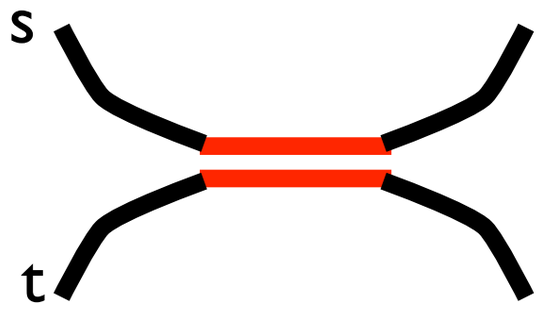
- BLOck Substitution Matrix
- Derived from a set of aligned ungapped regions from protein families called BLOCKS
- Cluster proteins such that they have no less than L % of similarity

Different BLOSUM matrices

- BLOSUM50
  - Proteins >50% similarity are in the same group
- BLOSUM62
  - Proteins >62% similarity are in the same group

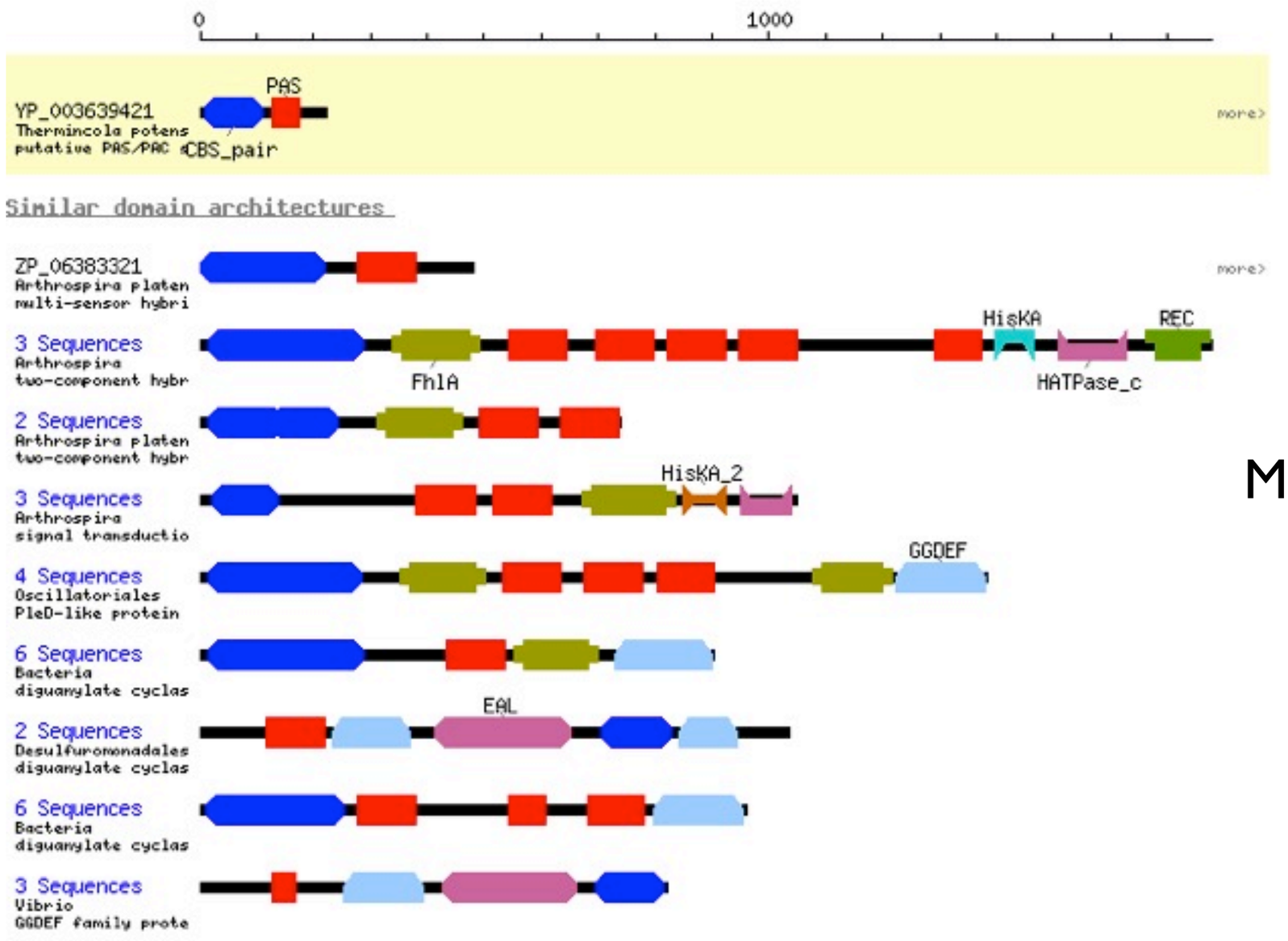
Example substitution scoring matrix (BLOSUM62)





Local Alignment

Local alignment between s and t: Best alignment between a subsequence of s and a subsequence of t.



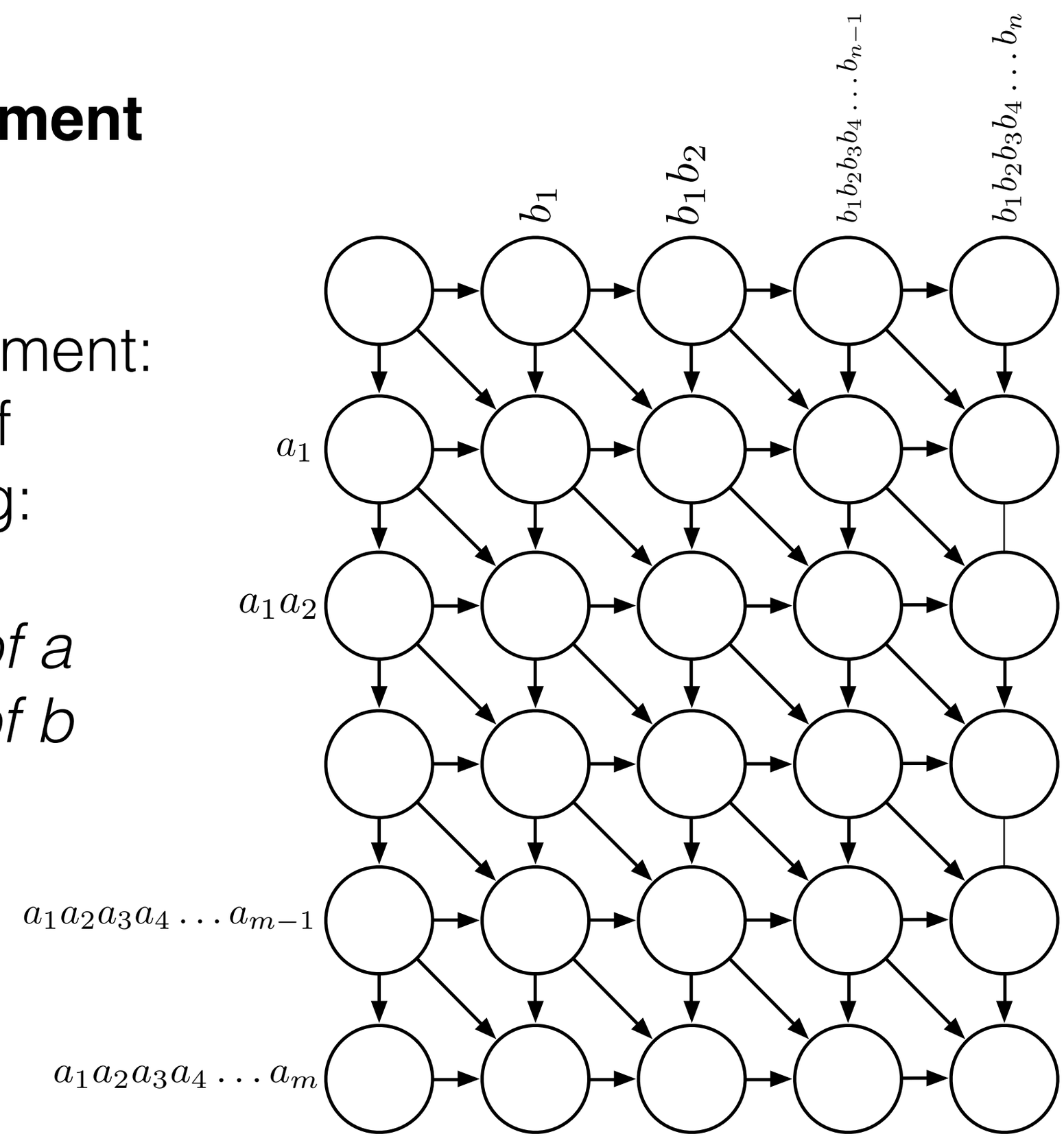
Motivation:

Many genes are composed of *domains*, which are subsequences that perform a particular function.

Local Alignment

Recall in **global** alignment:
$s_{i,j}$ is the score of
optimally aligning:

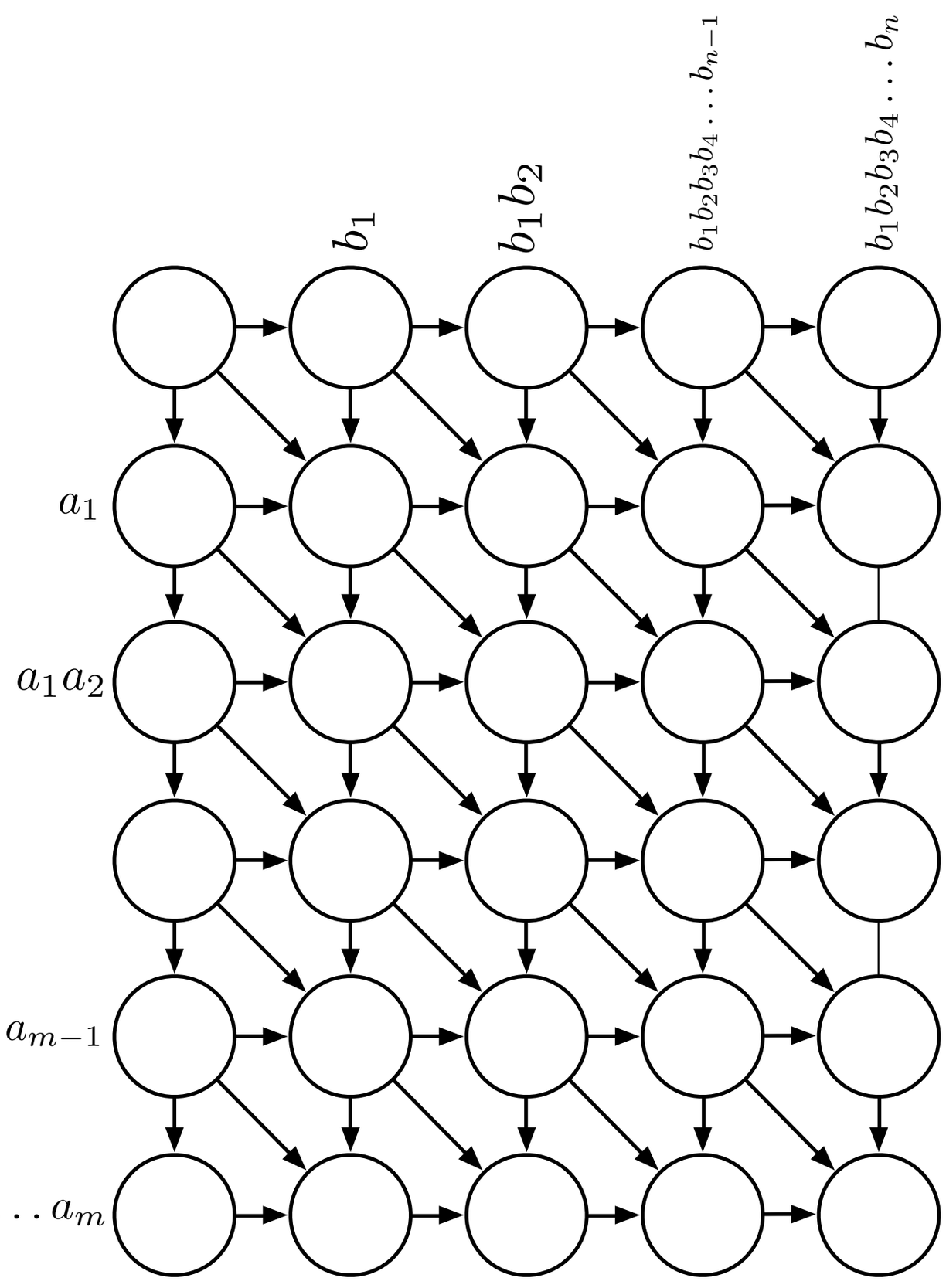
prefix *of length i of a*
prefix *of length j of b*



Local Alignment

In **local** alignment:
$s_{i,j}$ is the score of optimally aligning:

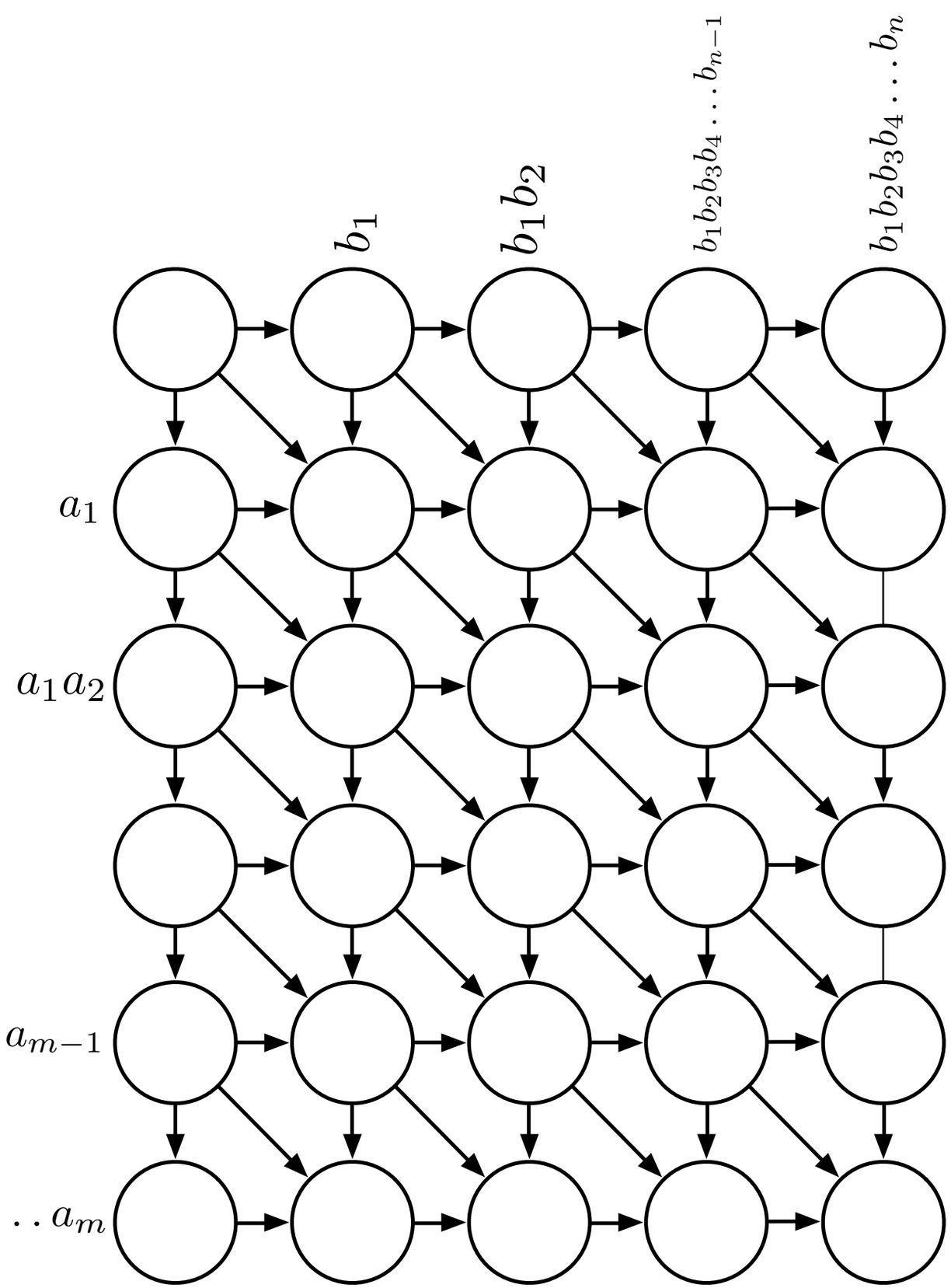
*some **suffix** of the **prefix** of length i of a*
*some **suffix** of the **prefix** of length j of b*



Local Alignment

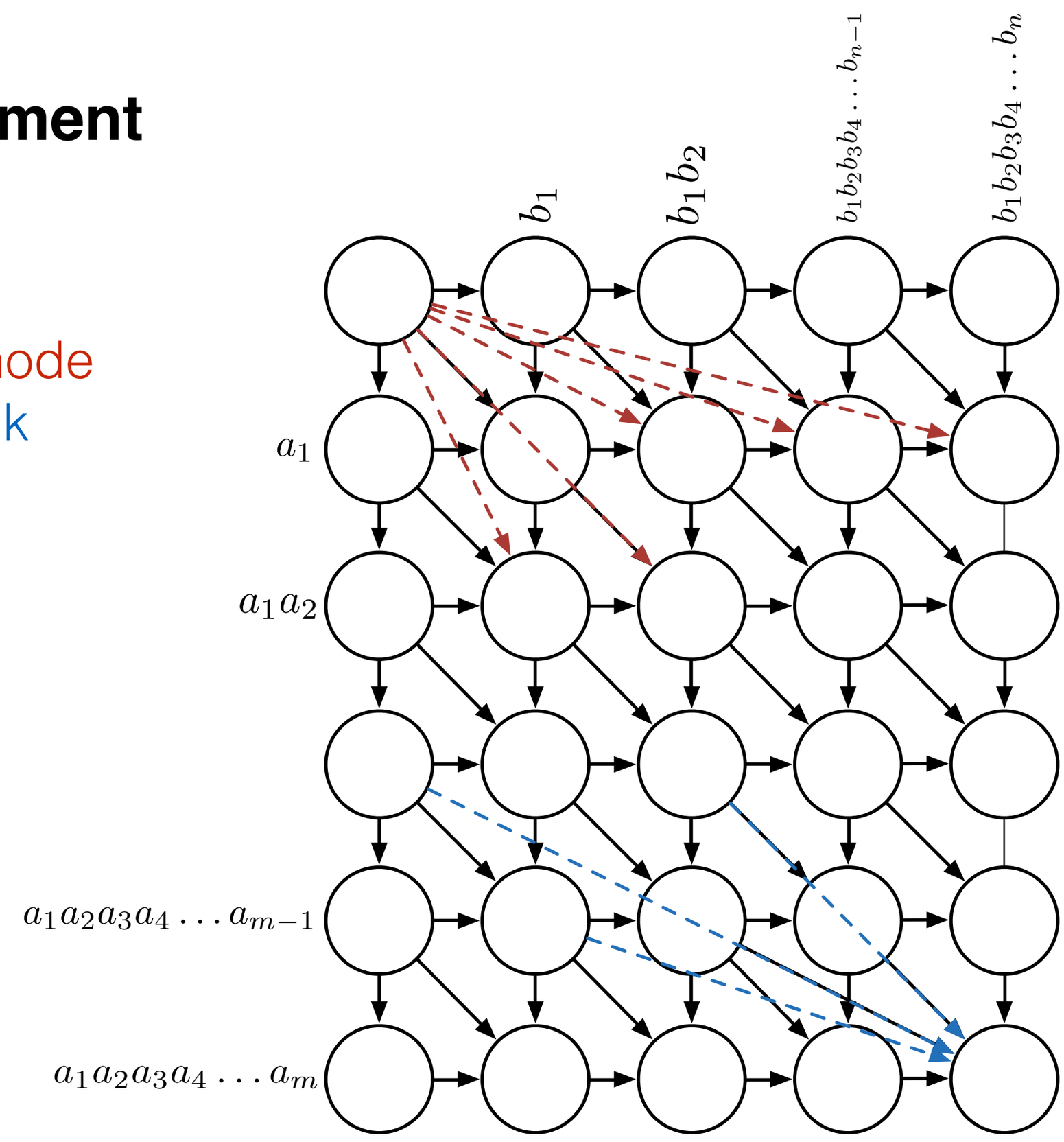
In **local** alignment:
$s_{i,j}$ is the score of
optimally aligning:

*some **substring** ending at position i of a*
*some **substring** ending at position j of b*



Local Alignment

Conceptually:
connect source to every node
connect every node to sink

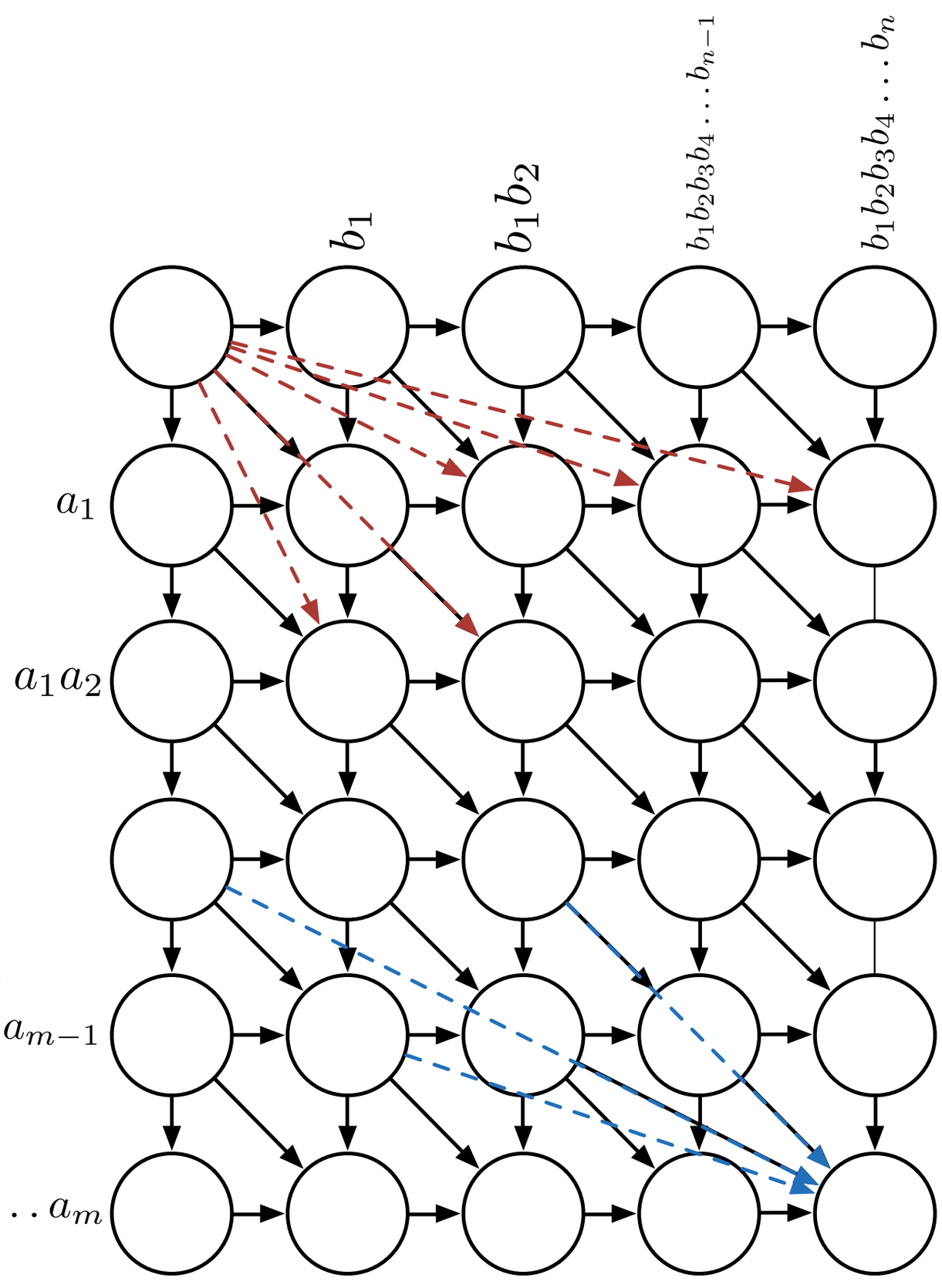


Local Alignment

Conceptually:
connect source to every node
connect every node to sink

Implementation:
connect source to every node

$$s_{i,j} = \max \begin{cases} 0 \\ s_{i-1,j} - \sigma \\ s_{i,j-1} - \sigma \\ s_{i-1,j-1} + \text{SCORE}(a_i, b_j), \end{cases}$$



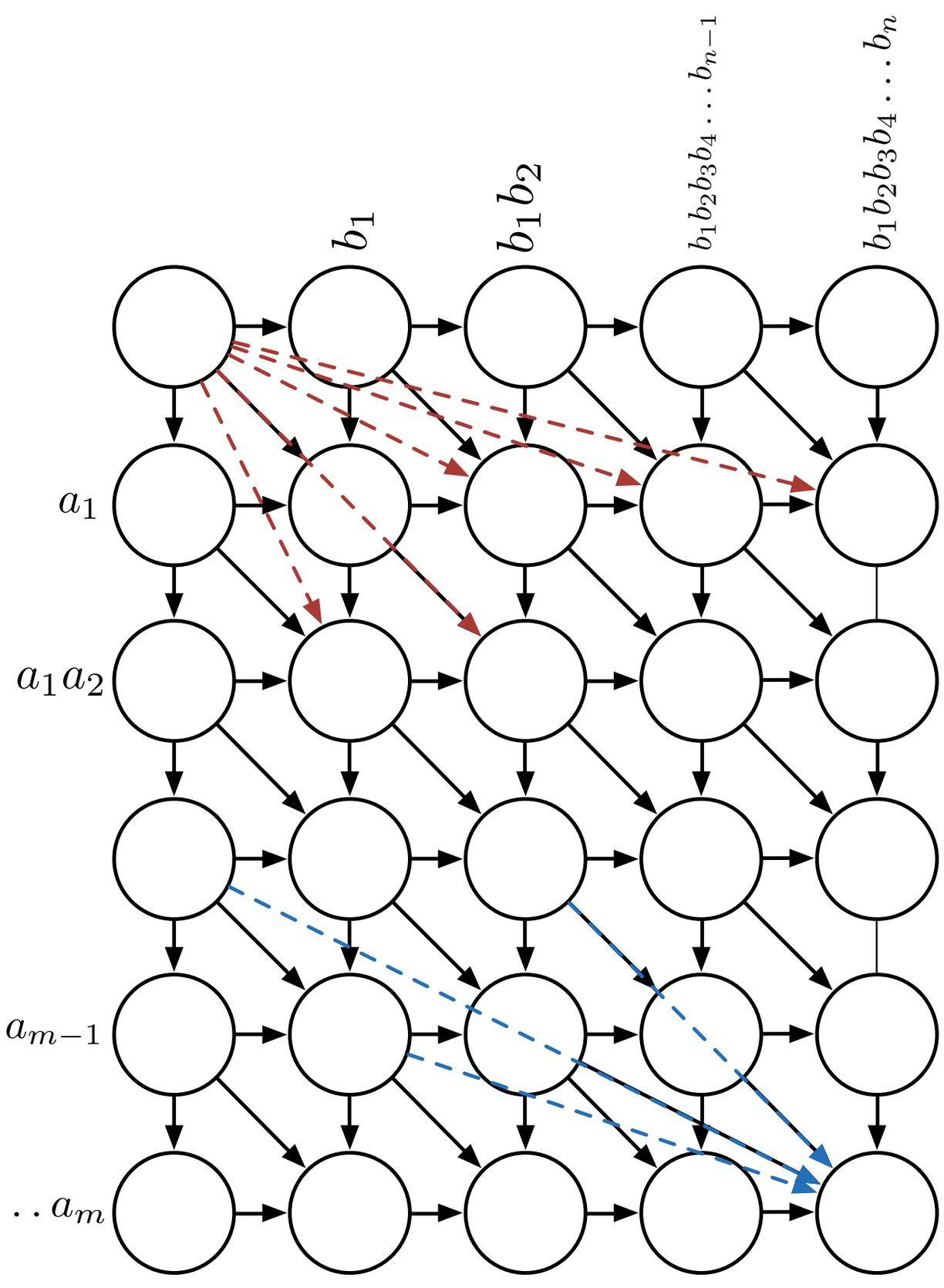
Local Alignment

Conceptually:
connect source to every node
connect every node to sink

Implementation:
connect every node to sink

start backtrack at node
with max score *anywhere*
in the graph

stop backtrack if *0* option
taken



Global/Local Alignment Recap

- Scoring matrices: based on probabilistic models of amino acid evolution

- Algorithm for **global** alignment sometimes called "Needleman-Wunsch"

- Algorithm for **local** alignment sometimes called "Smith-Waterman"

- Same basic algorithmic framework