Global Sequence Alignment

Stephen Altschul

National Center for Biotechnology Information
National Library of Medicine
National Institutes of Health

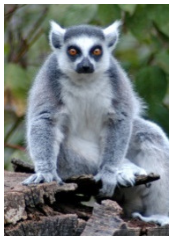
DNA and Protein Sequence Comparison

Mutations in a DNA sequence cause the protein sequences it encodes to change over evolutionary time. Individual amino acids may be inserted, deleted or change into other amino acids.

By comparing related sequences, we can learn about species relationships, about which positions in proteins are the most important, and about the causes of certain diseases.

Given two DNA or protein sequences, how can we define the *distance* between the sequences, or alternatively their *similarity*?

Alignments of Human Beta-Globin to Globins from Other Species



```
Human beta-globin                  VHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGN
                                     LTPEE   VT LWGKVNV  VGGEALGRLLVVYPWTQRFFESFGDLS PDA MGN
Ring-tailed lemur beta-globin      TFLTPEENGHVTSLWGKVNVEKVGGEALGRLLVVYPWTQRFFESFGDLSSPDAIMGN

PKVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANALAHKYH
PKVKAHGKKVL AFS GL HLDNLKGTFA LSELHC  LHVDPENF LLGNVLV VLAHHFG  F P  QAA QKVV GVANALAHKYH
PKVKAHGKKVLSAFSEGLHHLDNLKGTFAQLSELHCVALHVDPENFKLLGNVLVIVLAHHFGNDFSPQTQAAFQKVVIGVANALAHKYH
```

```
Human beta-globin                  VHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNP
                                   V  T   E SA   LWGK N DE G  AL R L VYPWTQR F  FG LS P A MGNP
Goldfish beta-globin               VEWTDAERSAIIGLWGKLNPDELGPQALARCLIVYPWTQRYFATFGNLSSPAAIMGNP

KVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFG-KEFTPPVQAAYQKVVAGVANALAHKYH
KV AHG  V G         DN K T A LS  H  KLHVDP NFRLL        A  FG   F   VQ A QK    V  AL   YH
KVAAHGRTVMGGLERAIKNMDNIKATYAPLSVMHSEKLHVDPDNFRLLADCITVCAAMKFGPSGFNADVQEAWQKFLSVVVSALCRQYH
```

```
Human beta-globin          VHLTPEEKSAVTALW----GKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKVKA
                             L       V   W    G  N   VG E    L         F  F   S        P V
Bloodworm globin IV        MGLSAAQRQVVASTWKDIAGSDNGAGVGKECFTKFLSAHHDIAAVF-GFSGAS-------DPGVAD

HGKKVLGAFSDGLAHL-DNLKGTFATLSELHCDK----LHVDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANALAHKYH
 G KVL    D   HL D  K             K     H   E F  LG  L     H  G   T     A     A    AL
LGAKVLAQIGVAVSHLGDEGKMVAEMKAVGVRHKGYGYKHIKAEYFEPLGASLLSAMEHRIGGKMTAAAKDAWAAAYADISGALISGLQ
```

```
Human beta-globin              VHLTPEEKSAVTALWG--KVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK
                               V  T      V       K N            L   P     F        P     NPK
Soybean leghemoglobin          VAFTEKQDALVSSSFEAFKANIPQYSVVFYTSILEKAPAAKDLFSFLANGVDPT----NPK

VKAHGKKVLGAFSDGLAHLDNLKGTFA--TLSELHCDKLHVDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANALAHKYH
   H  K      D    L       A   L   H  K   DP  F      L        G         A        A A
LTGHAEKLFALVRDSAGQLKASGTVVADAALGSVHAQKAVTDPQ-FVVVKEALLKTIKAAVGDKWSDELSREWEVAYDELAAAIKKA--
```

A Formal Problem

Given: Two protein or DNA sequences

$$X \equiv x_1 x_2 x_3 \ldots x_m$$
$$Y \equiv y_1 y_2 y_3 \ldots y_n$$

where the x_i and y_i are chosen from a finite alphabet $\boldsymbol{\mathcal{A}}$, e.g. $\{A, C, G, T\}$.

How can one define the *distance* between the sequences X and Y, or alternatively their *similarity*?

We shall adopt the somewhat more flexible formalism of *similarity*, with higher values considered better.

Although there are other possibilities, similarity is generally defined with reference to a *sequence alignment*, in which individual letters from each sequence are placed into correspondence.

Examples of Sequence Alignment

```
groan      colo-r          theatre     theatre
|||:|      |||| |          |||||::     |||||✕
grown      colour          theater     theater
```

```
elephant          vermiform     vermiform-----
|||: |||          ::||:::::          ||||
eleg-ant          formation     -----formation
```

```
disestablishment       disestablishment
|||        |  |||       |||          :|||
dis-------s--ent       dis---------sent
```

Applications

Sequence alignment arises in many fields:

- Molecular biology
- Inexact text matching (e.g. spell checkers; web page search)
- Speech recognition

In general:

The precise definition of what constitutes an alignment may vary by field, and even within a field.

Many different alignments of two sequences are possible, so to select among them one requires an objective (score) function on alignments.

The number of possible alignments of two sequences grows exponentially with the length of the sequences. Good algorithms are required.

Central Issues in Biological Sequence Comparison

Definitions: What are you trying to find or optimize?

Algorithms: Can you find the proposed object optimally and in reasonable time?

Statistics: Can your results be explained by chance?

In general there is a tension between questions. A definition that is too simple may allow efficient algorithms, but may not yield results of biological interest. However, a definition that includes most of the relevant biology may entail intractable algorithms and statistics. The most successful approaches find a balance between these considerations.

What Makes a Good Alignment?

What Makes a Good Alignment?

Lots of matching letters

Few mismatching letters

Few insertions or deletions (*indels*)

What Makes a Good Alignment?

Lots of matching letters

Few mismatching letters

Few insertions or deletions

Idea:

Each *match* gets a score: $+A$

Each *mismatch* gets a score: $-B$

Each *indel* gets a score: $-C$

Define the score of an alignment to be the sum of its match, mismatch and indel scores.

What Makes a Good Alignment?

Lots of matching letters

Few mismatching letters

Few insertions or deletions

Idea:

Each *match* gets a score: $+A$

Each *mismatch* gets a score: $-B$

Each *indel* gets a score: $-C$

Define the score of an alignment to be the sum of its match, mismatch and indel scores.

Define the *similarity* of two sequences to be the score of their *best* alignment.

Formal Elements of Global Sequence Alignment

No crossings allowed. For algorithmic reasons, it is fortunate that, although there are natural mechanisms (mutations) that lead to amino acid or nucleotide substitutions, insertions and deletions, there are none that yield transpositions, unlike with keyboard-produced text. In contrast, when analyzing RNA folding, one may choose for algorithmic reasons to exclude "pseudoknots", which do in fact occur naturally.

Gaps. An arbitrary number of *null* characters (represented by dashes) may be placed into either sequence, and aligned with letters in the other sequence. Two nulls may not be aligned. Depending upon one's perspective, the alignment of a letter with a null may be understood as the *insertion* of a letter into one sequence, or the *deletion* of a letter from the other. Therefore, a letter aligned with a null is sometimes called an *indel*.

Alignment scores. The score for an alignment is taken to be the sum of scores for aligned pairs of letters, and scores for letters aligned with nulls. Each such pairing is called an *alignment column*.

Substitution scores. Scores for aligned pairs of letters are called *substitution scores*, whether the letter aligned are identical or not. Most simply, substitution scores may take the form of *match* scores and *mismatch* scores.

Gap scores. The score for a letter aligned with a null is called a *gap score*. Usually gap scores are letter-independent.

Global alignment. All letters and nulls in each sequence must be aligned.

Sequence Similarity

Define the *similarity* of two sequences as the score of their highest-scoring (optimal) alignment.

How do we find the an optimal alignment of two sequence, and its score?

Brute force enumeration is impractical, because the number of possible alignments becomes astronomically large for even fairly short sequences.

Fortunately, the problem is soluble efficiently using a technique called *dynamic programming*.

What Makes a Good Alignment?

Lots of matching letters

Few mismatching letters

Few insertions or deletions

Idea:

Each *match* gets a score: $+A$

Each *mismatch* gets a score: $-B$

Each *indel* gets a score: $-C$

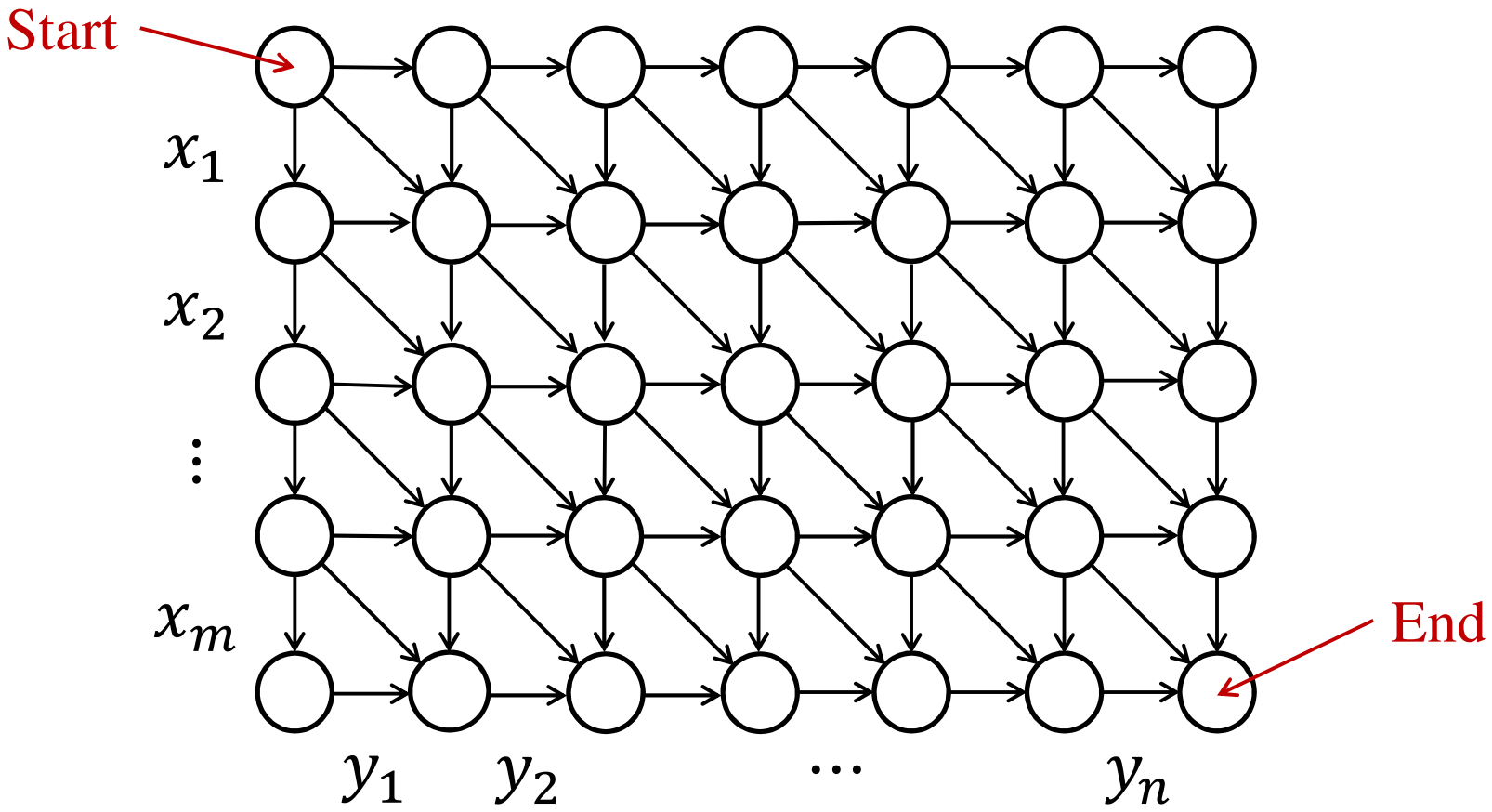
Define the score of an alignment to be the sum of its match, mismatch and indel scores.

Define the *similarity* of two sequences to be the score of their *best* alignment.

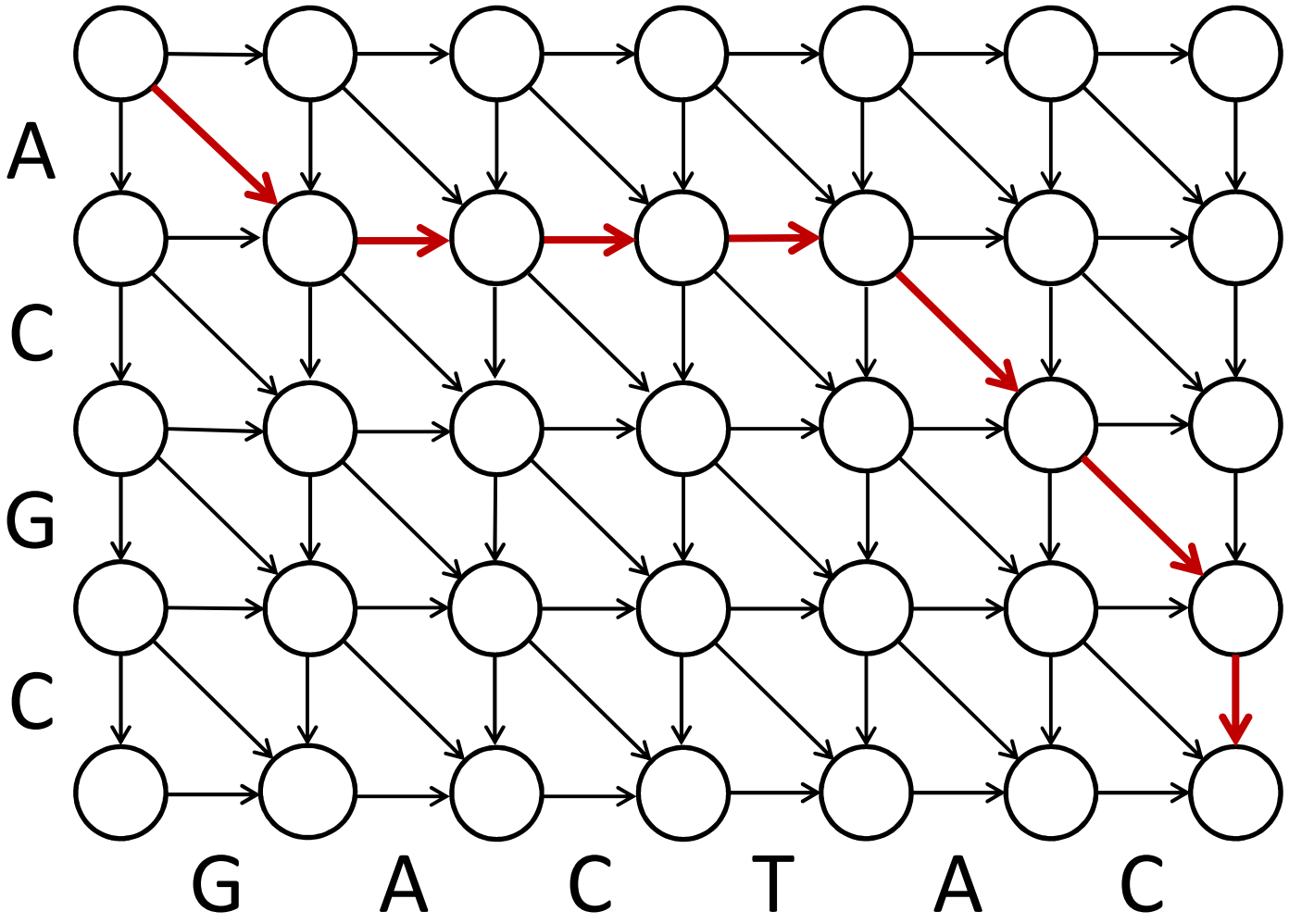
Problem: How can one find the optimal alignment?

Path graphs

A global alignment may be viewed as a path through a directed *path graph* which begins at the upper left corner and ends at the lower right. Diagonal steps correspond to substitutions, while horizontal or vertical steps correspond to indels. Scores are associated with each edge, and the score of an alignment is the sum of the scores of the edges it traverses. Each alignment corresponds to a unique path, and vice versa.

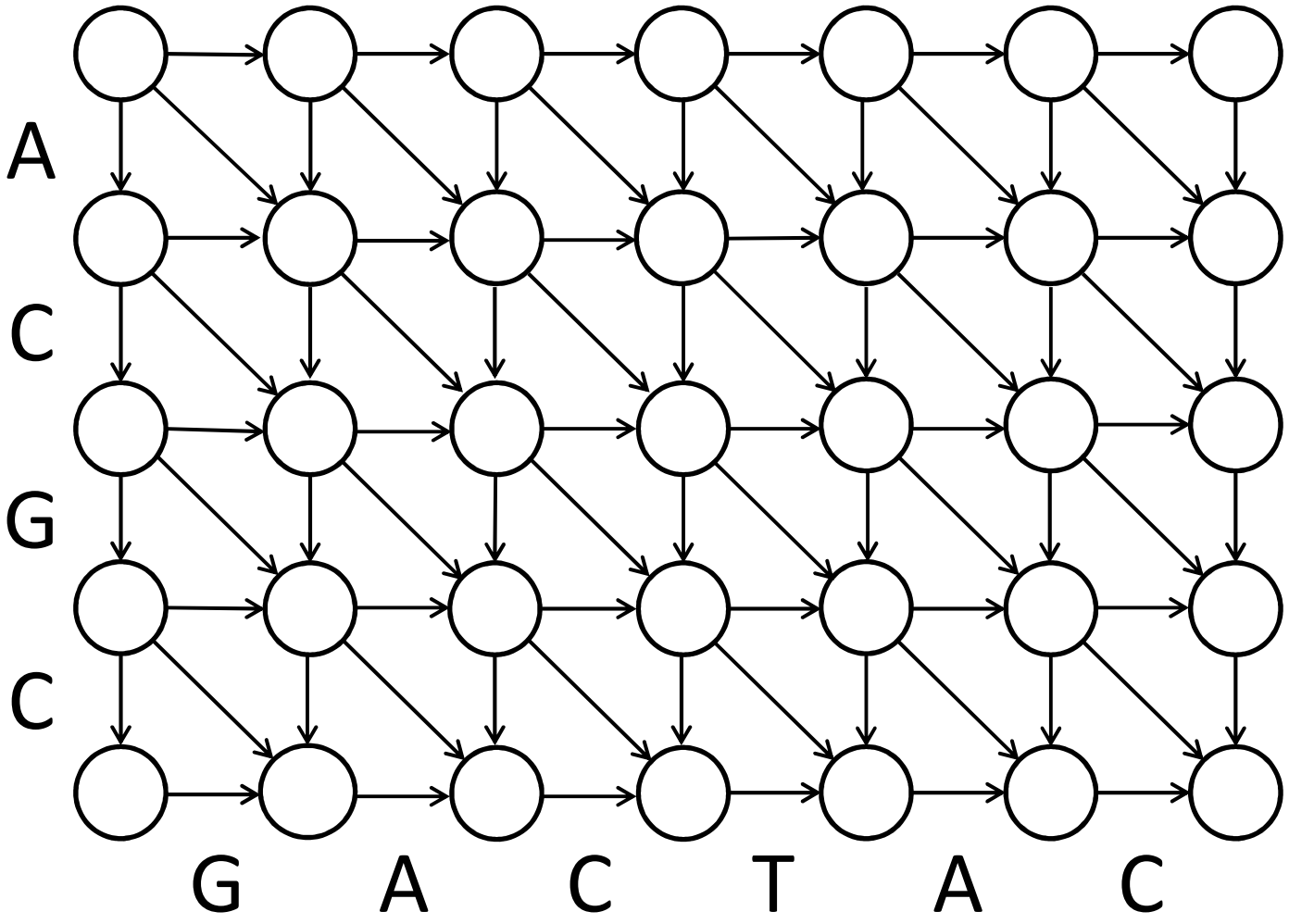


What alignment is this? What is its score?



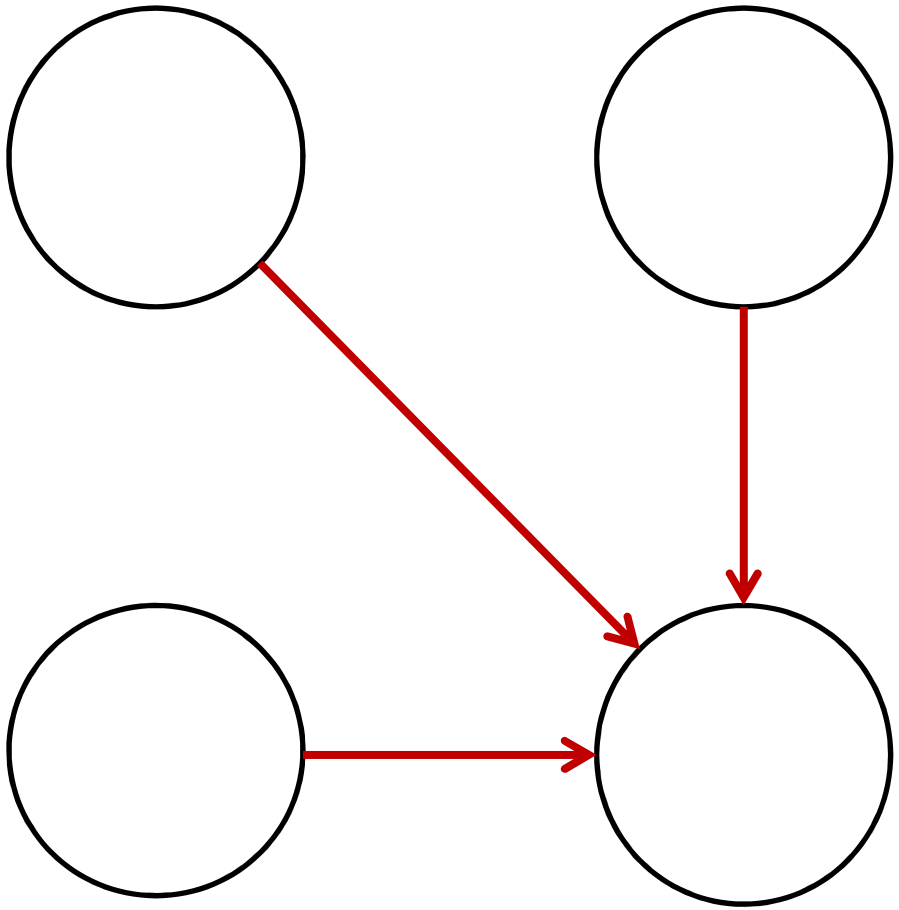
Scores: Match +1 Mismatch 0 Indel -1

How can one find the optimal alignment?

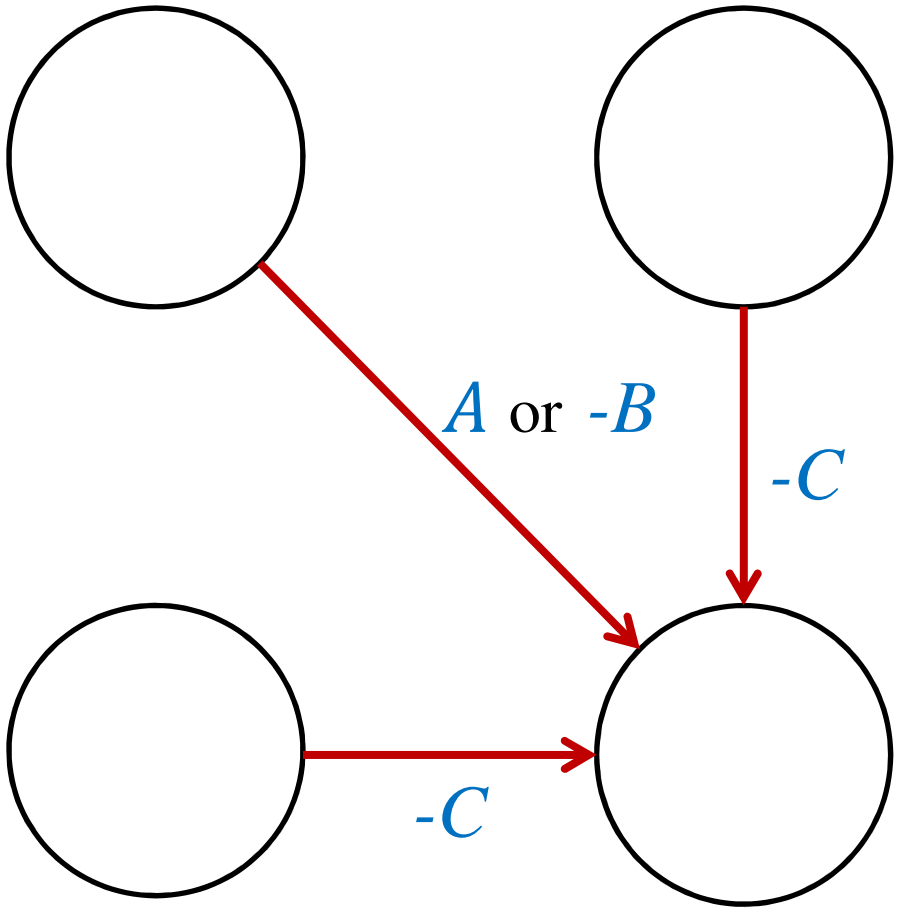


Scores: Match +1 Mismatch 0 Indel -1

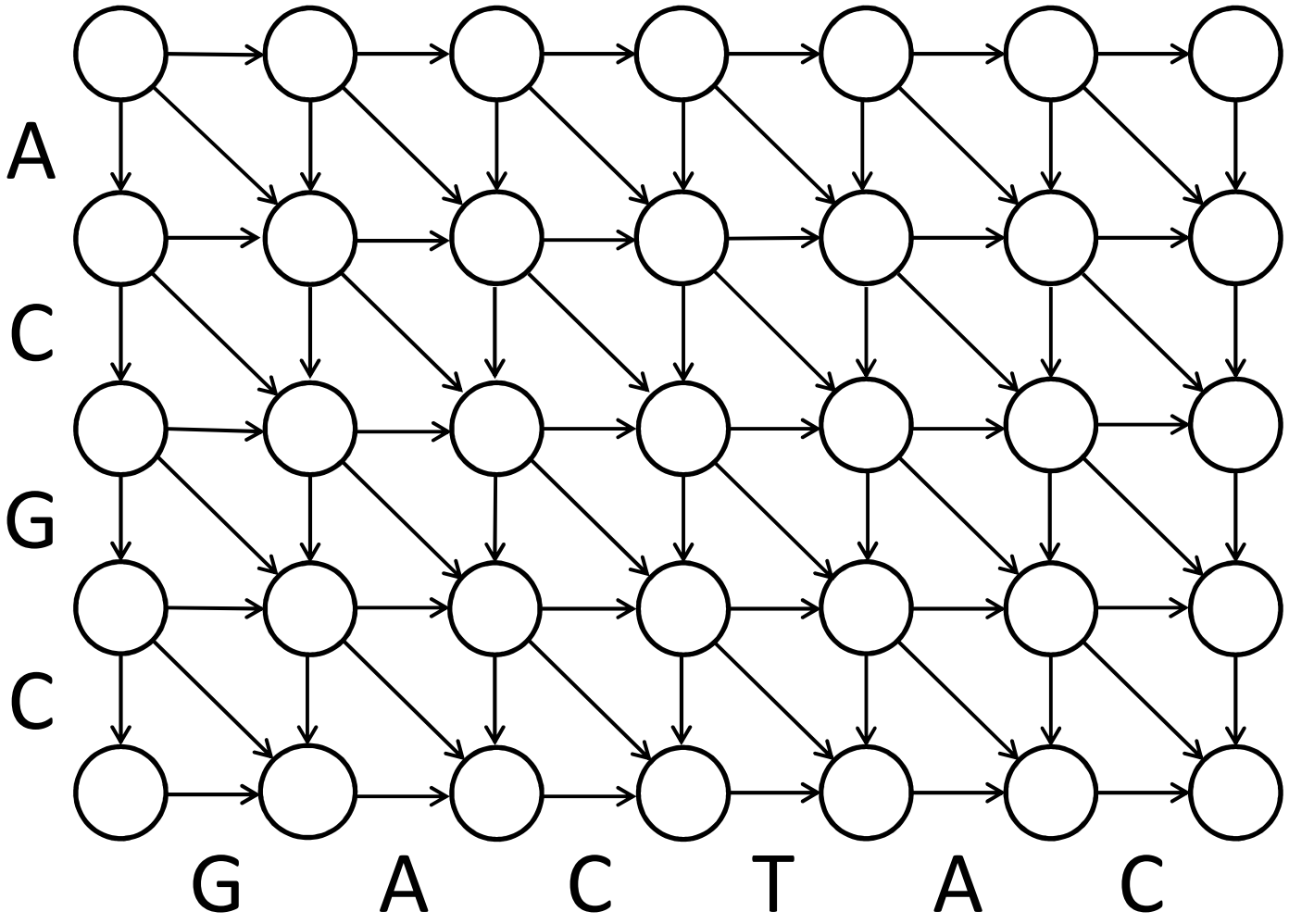
There are at most three ways to enter a node



Each edge has an associated score

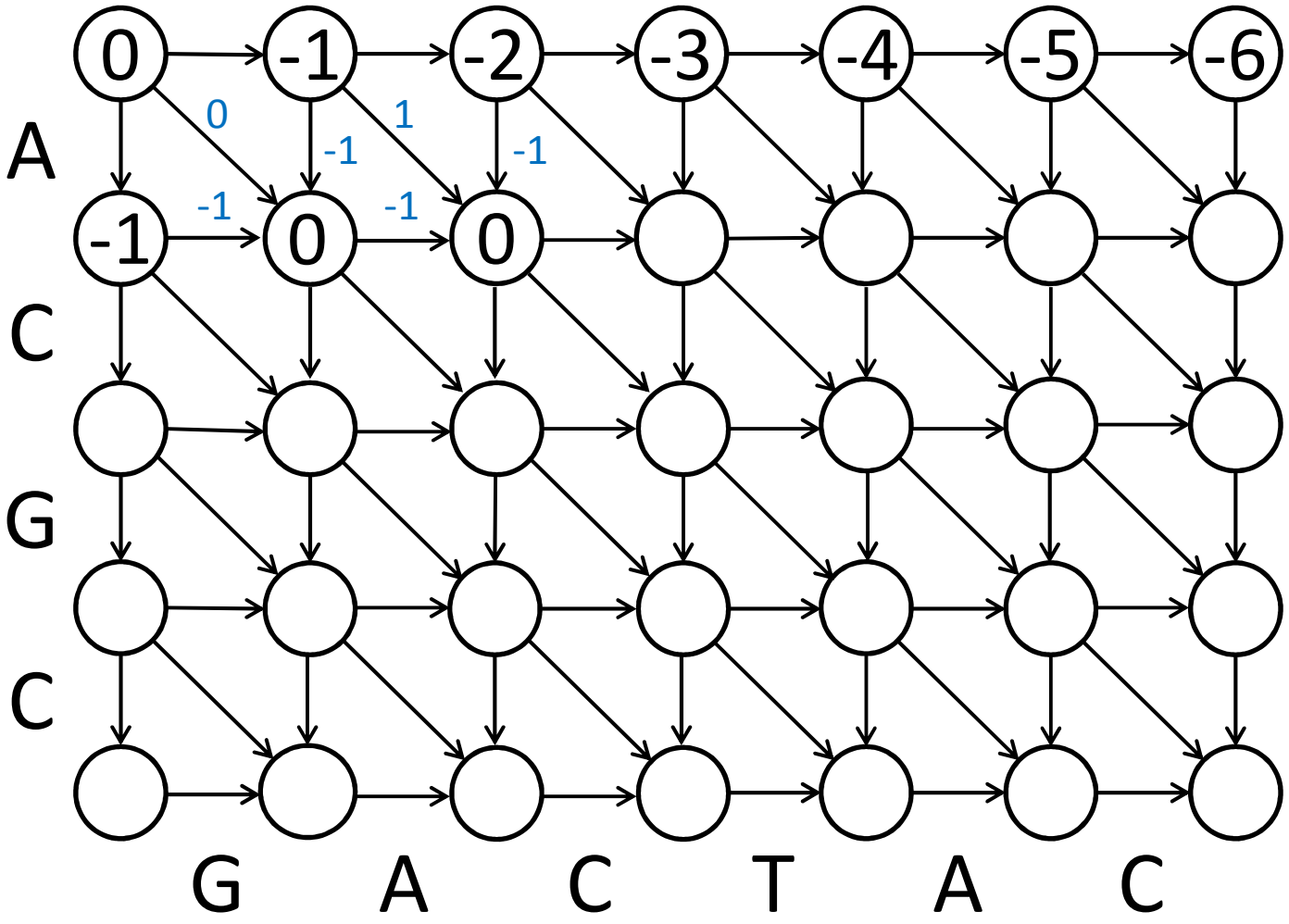


How can one find the optimal alignment?



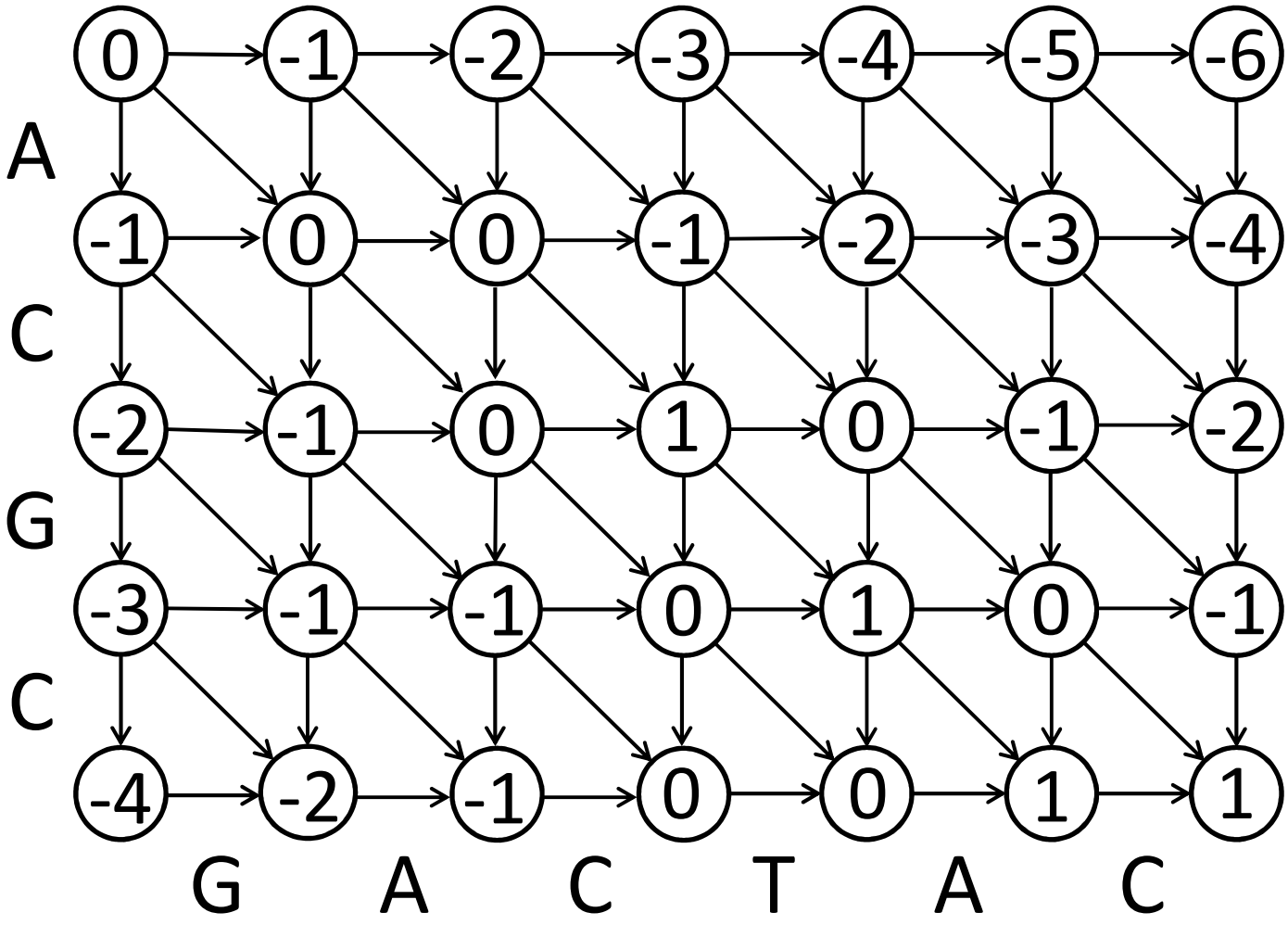
Scores: Match +1 Mismatch 0 Indel -1

Fill in the nodes…



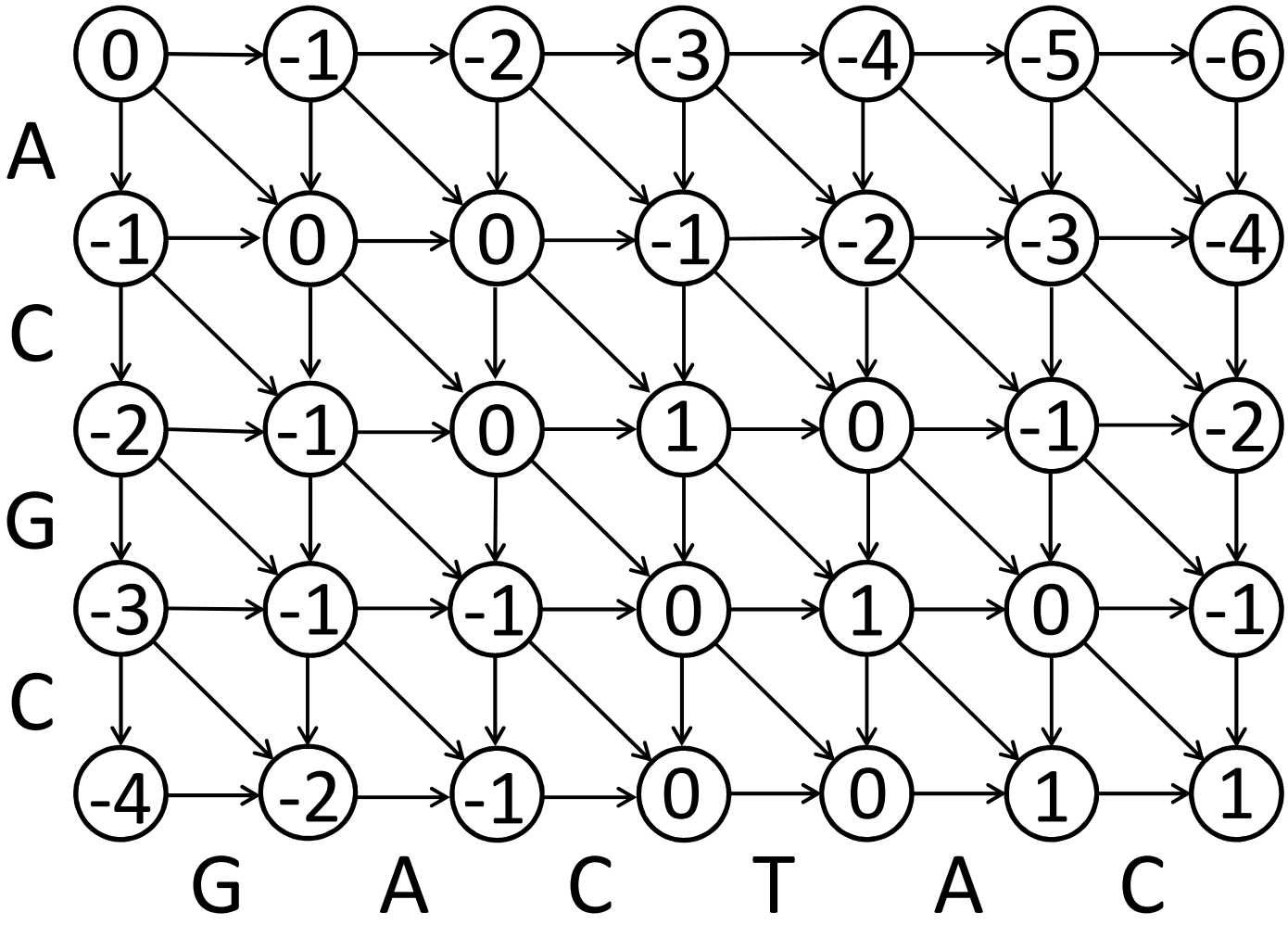
Scores: Match +1 Mismatch 0 Indel -1

The completed path graph



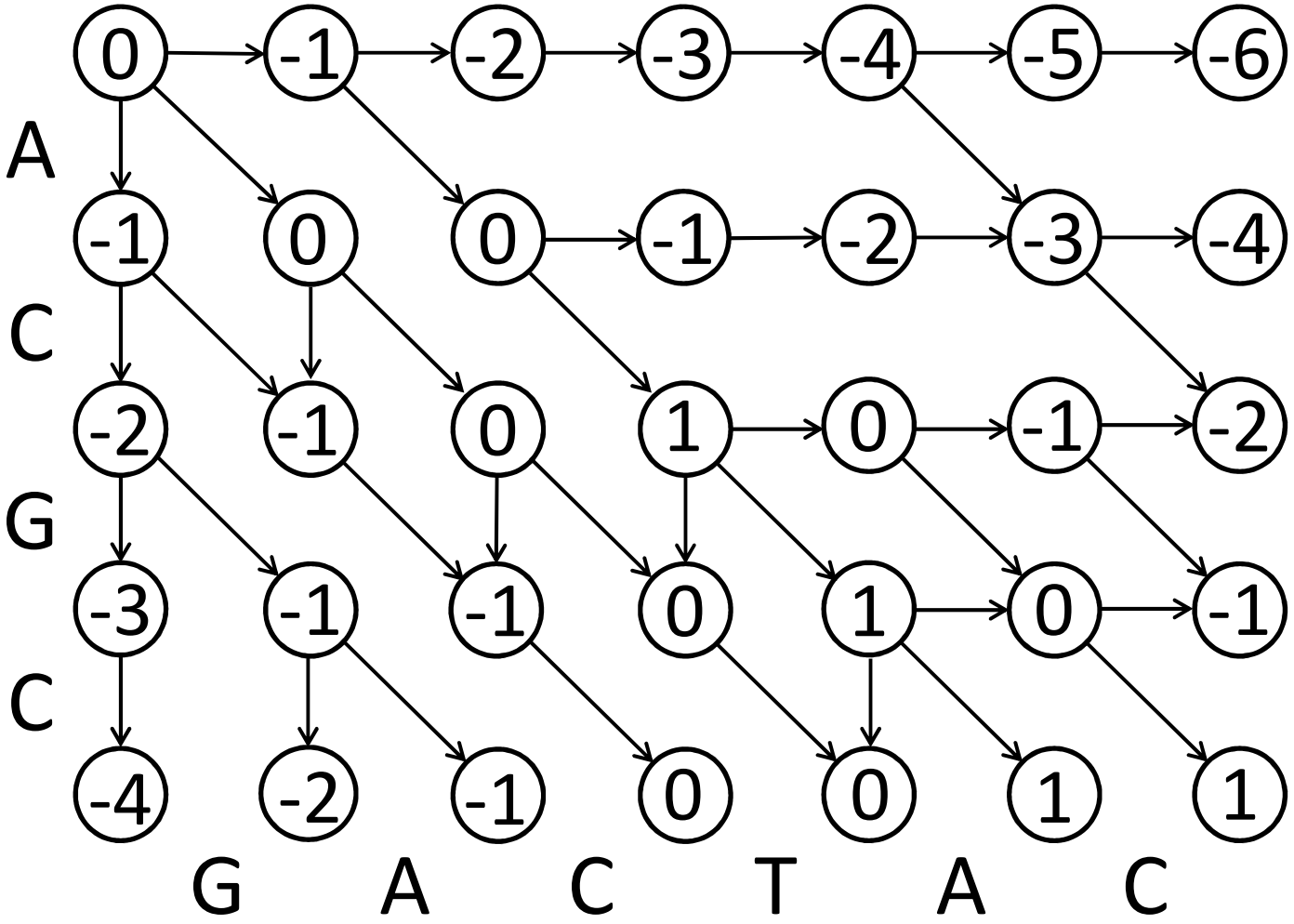
Scores: Match +1 Mismatch 0 Indel -1

But… What is the optimal alignment?

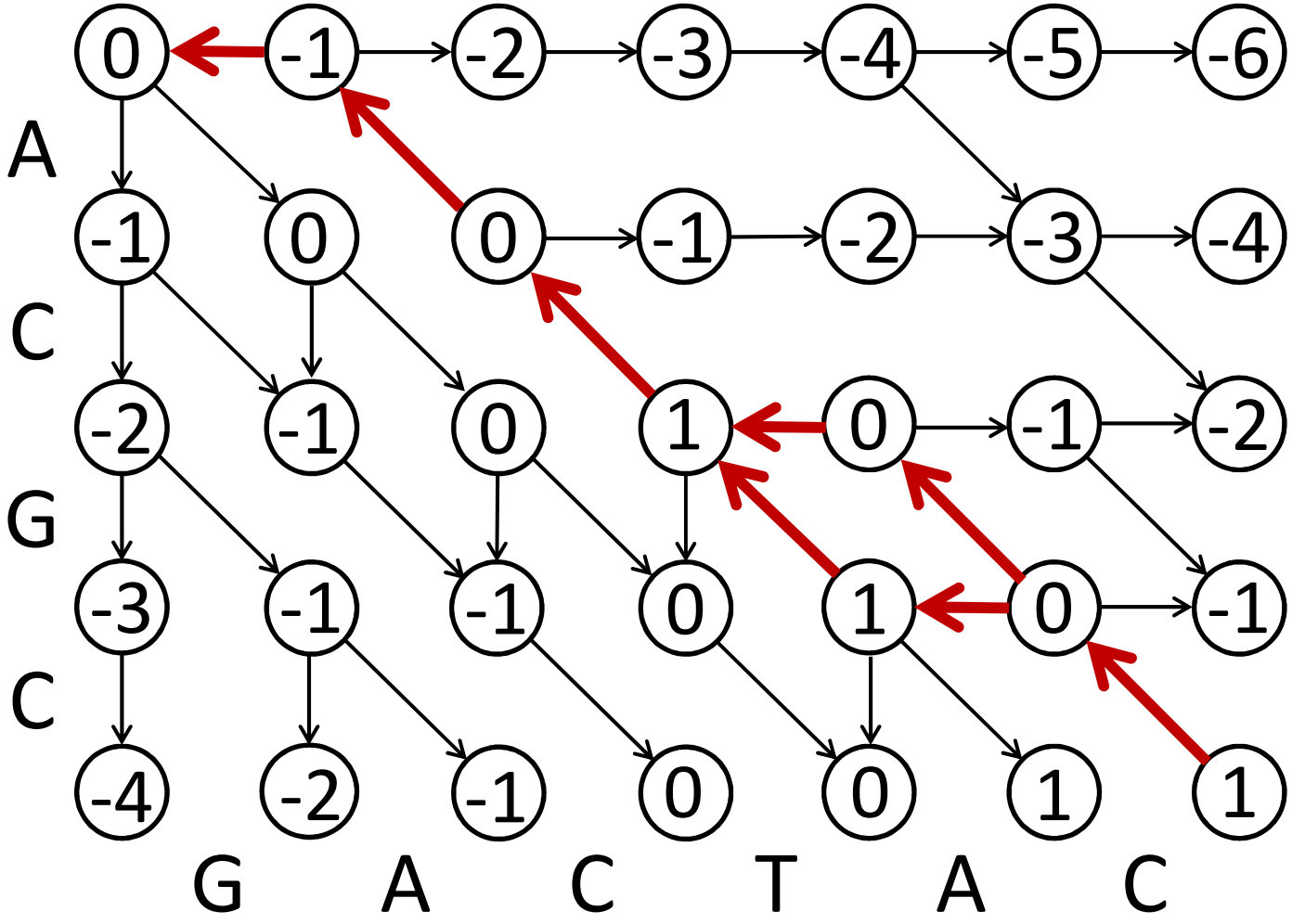


Scores: Match +1 Mismatch 0 Indel -1

Record the best path or paths into each node…



Follow *traceback* edges from the final node



Optimal alignments: **–ACG–C** / **GACTAC** and **–AC–GC** / **GACTAC**

Dynamic Programming and Global Alignment

Dynamic programming is a method by which a larger problem may be solved by first solving smaller, partial versions of the problem. We demonstrate here how it may be applied to global sequence alignment, where at first we are interested only in the similarity of two sequences, and not the alignment that yields this score.

Definitions:

$s(a, b)$	the substitution score for aligning letters a and b
g	the gap score for aligning any letter to a null
X_i	the partial sequence consisting of the first i letters of $X \equiv x_1 x_2 \ldots x_m$
Y_j	the partial sequence consisting of the first j letters of $Y \equiv y_1 y_2 \ldots y_n$
$SIM(i, j)$	the similarity of X_i and Y_j

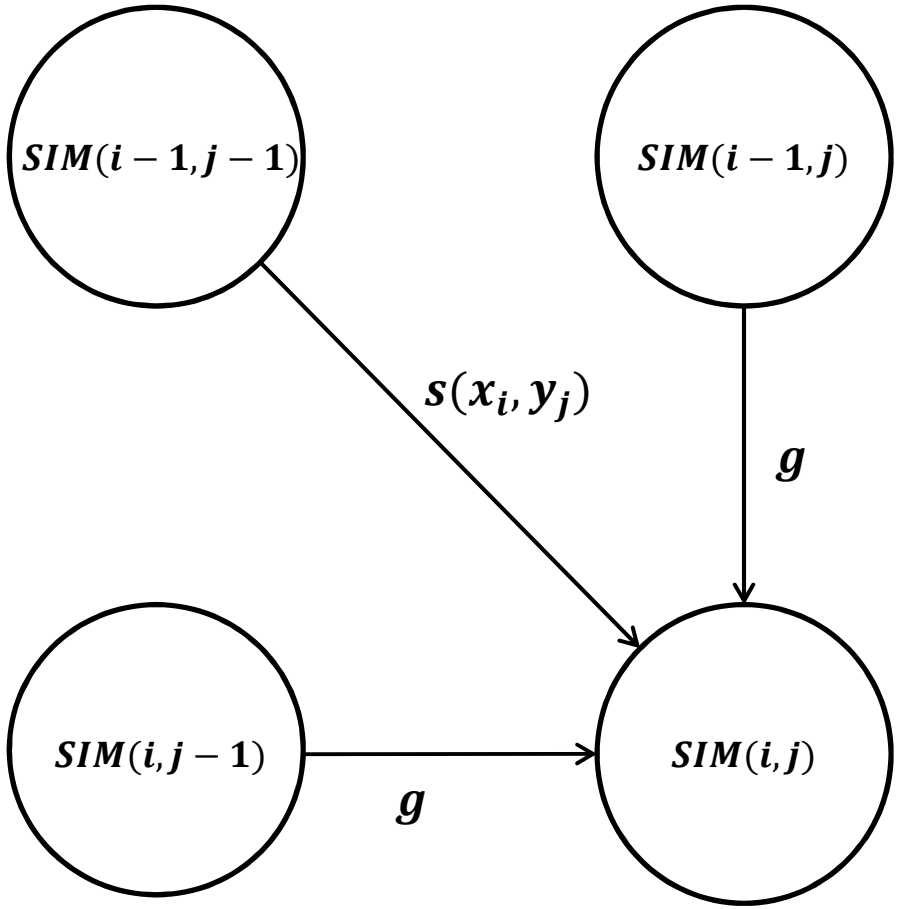
Consider the *last column* of an optimal alignment of X_i and Y_j. This column either aligns x_i to y_j, or x_i to a null, or y_j to a null. Because we do not allow "crossing", there are no other possibilities. This observation yields the following recurrence:

$$\boldsymbol{SIM(i,j) = max}\begin{cases} \boldsymbol{SIM(i-1, j-1) + s(x_i, y_j)} & x_i \text{ and } y_j \text{ aligned} \\ \boldsymbol{SIM(i-1,\ j) + g} & x_i \text{ aligned with a null} \\ \boldsymbol{SIM(i,\ j-1) + g} & y_j \text{ aligned with a null} \end{cases}$$

In brief, we can solve for $SIM(m, n)$ by solving smaller versions of the problem first.

Dynamic programming on path graphs

One may associate a partial similarity with each node of a path graph. If the values of $SIM(i-1, j-1)$, $SIM(i-1, j)$ and $SIM(i, j-1)$ are known, the value of $SIM(i, j)$ may be calculated.



Pseudocode for Finding Sequence Similarity

```
Similarity(X,Y):
   For i = 0,...,m:  SIM[i,0] = i*g
   For j = 1,...,n:   SIM[0,j] = j*g
   For i = 1,...,m:
      For j = 1,...,n:
         SIM[i,j] = max(
            SIM[i-1,j-1] + s(X[i],Y[j]),
            SIM[i-1,j]+g,
            SIM[i,j-1]+g
         )
      EndFor
   EndFor
Return SIM[m,n]
```

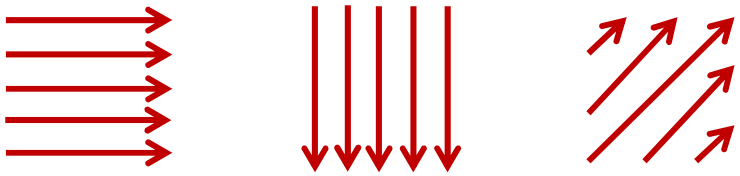
Exercise: Generalize the code to include traceback information, and produce one optimal alignment.

Note: This is generally known as the *Needleman-Wunsch algorithm*, after the first paper in the field of computational molecular biology to apply *dynamic programming* to the global alignment problem. However, the paper actually describes a somewhat different algorithm which is almost never used.

Needleman, S.B. & Wunsch, C.D. (1970) "A general method applicable to the search for similarities in the amino acid sequences of two proteins." *J. Mol. Biol.* **48**:443-453.

Observations and Generalizations

The nodes can be expanded in a variety of orders, so long as all nodes that "feed into" a given node are expanded before that node. Possible expansion orders are:



The time complexity of the algorithm is $O(mn)$.

If only the similarity is desired, the space complexity is $O[\min(m,n)]$; if an optimal alignment is sought, the space complexity is $O(mn)$, but as we shall see, this too can be reduced to $O[\min(m,n)]$.

It is possible to save time (but in general no more than a constant factor) by not expanding nodes that can not possibly participate in an optimal path.

Fickett, J.W. (1984) *Nucl. Acids Res*. **12**:175-180; Spouge, J.L. (1989) *SIAM J. Appl. Math*. **49**:1552-1566.

Global Alignment Scores

Multiplying all substitution and gap scores by a positive constant does not change the optimal alignment. Why?

Adding a constant k to all substitution scores, and $k/2$ to all gap scores, does not change the optimal alignment. Why?

A global alignment scoring system with the three nominal parameters of match score a, mismatch score b, and gap score g, in fact has a single free parameter. For example, assuming $a > g$, one can always construct an equivalent scoring system with $a = 1$ and $g = 0$. What is the scoring system of this form equivalent to $(a = 1,\ b = 0,\ g = -1)$?

Modifying global alignment scores so that $g = 0$ can speed up the inner loop of the dynamic programming algorithm.

What next?

Are there better *substitution scores* than match-mismatch scores?

```
elephant
|||: |||
eleg-ant
```

Are there better *gap scores* than scores for just individual indels?

```
disestablishment
|||        |  |||
dis-------s--ent

disestablishment
|||          :|||
dis---------sent
```

Does changing the definition of an alignment's score require a change to the algorithm? If so, does it slow down the algorithm, and by how much?

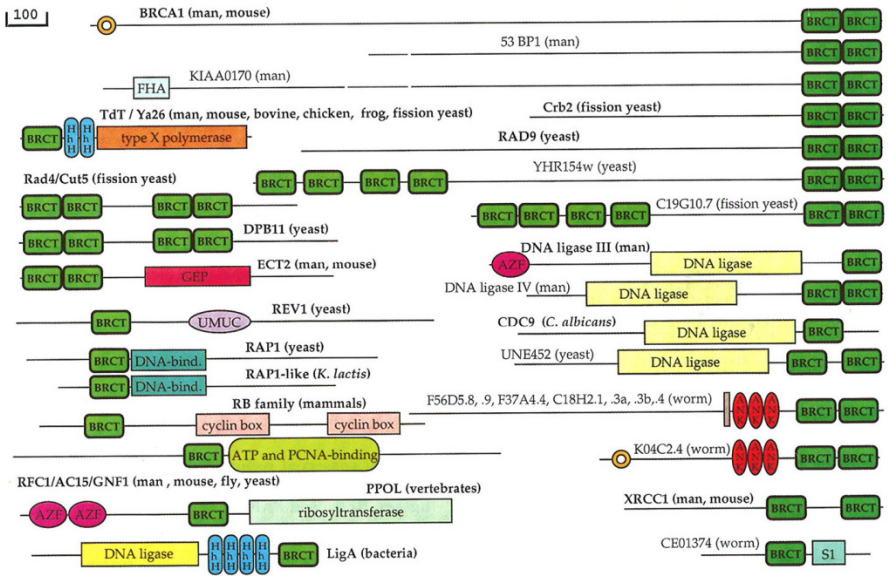
What next?

How does one define a *local alignment* and its score?

How can one modify the algorithm to deal with local alignments?

```
vermiform
::||:::::
formation
```

```
vermiform-----
    ||||
-----formation
```



What next?

Can one speed up the algorithm?

How high an alignment score can one expect to find by chance?

How can one align multiple sequences?

How can one account for *correlations* between positions?