# Sequence Alignment using *Align*

## Part IV: Alignment Algorithms

Richard M. Salter
Oberlin College


Supported by NSF CCLI DUE 0618252

# Semiglobal Alignment

- Recall from Part III that the Needleman-Wunsch algorithm is a **global algorithm** because the two sequences are aligned over their entire lengths.

- Since the algorithm does not distinguish between gaps that appear internally and those that appear at the beginning and/or end of the sequence, this algorithm is not useful for locating a smaller sequence within a larger one.

  - Example: Consider aligning `cgta` within `acatcgatcgtataccgac`

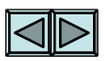
    Each of the following scores the same using the (0, -1, 1, 0)* scoring scheme:

```
acatcgatcgtataccgac   acatcgatcgtataccgac   acatcgatcgtataccgac
-c---g-t---a-------   ----cg----ta-------   --------cgta-------
```

  - Clearly the last of these is the most interesting alignment, since it finds the shorter sequence intact within the larger one. Unfortunately, the global algorithm is unable to distinguish among these alignments.

- The problem is easy to fix by making a few adjustments to our dynamic programming algorithm.

*Origination penalty: 0, gap penalty: -1, match score: 1, mismatch score: 0



# Needleman-Wunsch Semiglobal Alignment

- We wish to modify the original Needleman-Wunsch global alignment algorithm so that gaps at either end of a sequence (let's call them *terminal gaps*) do not incur a gap penalty.

- But how are terminal gaps distinguished in the program from internal gaps?

- First, note that gaps appear when movement in the matrix is either horizontal or vertical.

  - This is illustrated in the two figures below. Horizontal movement corresponds to gaps introduced into the query sequence, while vertical movement corresponds to gaps introduced into the target string.

Target acatgcatcgtaaccgac
Query -----c---gta-----

Reset

| | | a | c | a | t | g | c | a | t | c | g | t | a | a | c | c | g | a | c |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| c | 0.00 | | | | | | 1.00 | 0.00 | -1.00 | -2.00 | | | | | | | | | |
| g | 0.00 | | | | | | | | | | -1.00 | | | | | | | | |
| t | 0.00 | | | | | | | | | | | 0.00 | | | | | | | |
| a | 0.00 | | | | | | | | | | | | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |

Target --acatgcatcgtaaccgac
Query cgta----------------

Reset

| | | a | c | a | t | g | c | a | t | c | g | t | a | a | c | c | g | a | c |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| c | 0.00 | | | | | | | | | | | | | | | | | | |
| g | 0.00 | | | | | | | | | | | | | | | | | | |
| t | 0.00 | 0.00 | | | | | | | | | | | | | | | | | |
| a | 0.00 | | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

# Needleman-Wunsch Semiglobal Alignment (cont'd)

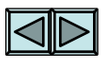
- It should be clear from the illustration on the previous page that terminal gaps correspond in the matrix to horizontal motion in the first or last row, and vertical motion in the first or last column.

- To create a semiglobal algorithm, it is therefore sufficient to remove the gap penalties from the first and last row, and first and last columns of our table.

- Gap penalties are easily removed from the first row and column by simply initializing the them to 0.

- To remove gap penalties from the last row and column, we modify the algorithm to allow free horizontal and vertical movement there.

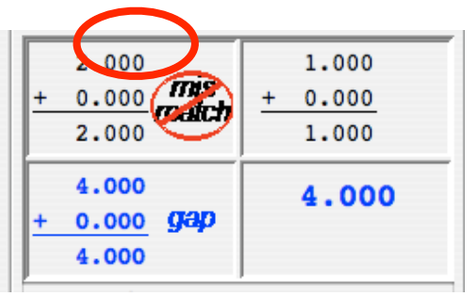
- The result is called the *Needleman-Wunsch Semiglobal Alignment.*



# Example

- Let us try aligning `cgta` within `acatcgatcgtataccgac` using Needleman-Wunsch semiglobal alignment.
  - Fire up *Align* and click *Needleman/Wunsch Semiglobal Alignment.*
  - Check your configuration. Under *Manual Scoring* select the *DNA* alphabet and *Simple Scoring.* Set your scoring scheme to (0, -1, 1, 0).
  - Select **File | Load Sequences ...** from the menu and enter the two sequences as Query and Target, respectively. (You can simply cut and paste them from above).
  - Click *Start,* then *Solve.* Note that the first row and first column of the table are now filled with 0's. Similarly, note that no penalty is applied in the last seven horizontal steps.
    - Click on one of these boxes to see the details.

| | | a | c | a | t | c | g | a | t | c | g | t | a | t | a | c | c | g | a | c |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| c | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 1.00 | 0.00 | 0.00 | 1.00 |
| g | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 2.00 | 1.00 | 0.00 | 0.00 | 2.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 2.00 | 1.00 | 1.00 |
| t | 0.00 | 0.00 | 0.00 | 0.00 | 2.00 | 1.00 | 1.00 | 2.00 | 2.00 | 1.00 | 1.00 | 3.00 | 2.00 | 1.00 | 0.00 | 0.00 | 0.00 | 1.00 | 2.00 | 1.00 |
| a | 0.00 | 1.00 | 1.00 | 1.00 | 1.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 4.00 | 4.00 | 4.00 | 4.00 | 4.00 | 4.00 | 4.00 | 4.00 |




- The semiglobal algorithm selected the preferred alignment, giving it a score of 4.

# Exercise 4.1

- To see why the semiglobal alignment algorithm selected the preferred alignment, let us now score using the semiglobal algorithm the other alignments that achieved the same optimal score when using the global algorithm.

  a)
  ```
  acatcgatcgtataccgac
  -c---g-t---a-------
  ```

  b)
  ```
  acatcgatcgtataccgac
  ----cg----ta-------
  ```

- Load the sequences (if necessary):
  ```
  acatcgatcgtataccgac
  cgta
  ```

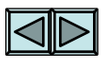
  - make sure you are using the (0, -1, 1, 0) scoring scheme.

- Click *Manual.*

- Using left and right mouse buttons, shift the Query string so that it aligns as in (a). Then click *Evaluate*. What score does it get? Can you see why?

- Similarly, click *Restart*, align the Query string as in (b), and then click *Evaluate.* What is the score? Why?

Exercise 4.1 Solution



# Exercise 4.2

- For each of the following sequence pairs, using the scoring schemes shown:

  i. Load the sequences shown in (i).

  ii. Using semiglobal alignment in Manual mode, shift the sequences (using left and right mouse buttons) until they appear as shown in the Trial Alignment (ii), then score the Trial Alignment;

  iii. Use Auto mode to find the optimal alignment for the pair;

  iv. Compare results; find the locations in the table where they differ, if any.
  Recall that you can flip between Manual and Auto mode to compare results

a) i) Sequences:
```
agatagaaactgatatata
agaaaacagagt
```
Scoring: (0, -1, 1, 0)
(0, -1, 1, -1)

ii) Trial Alignment:
```
agatagaaactgatatata
ag---aaaacagagt----
```

b) i) Sequences:
```
MVAIWRAILVGTVIAML
AILMVGTAVWIML
```
Scoring: (0, -1, 1, -1)
(-12, -1, Blosum 45)*
(-10, -1, Pam 30)**

ii) Trial Alignment:
```
MVAIWRAILVGTVIAML-
--AI---LMVGTAVWIML
```

Before trying part (b) you must click **Edit | Configuration ...** and select the Protein alphabet.

* e.g., origination penalty: 0; gap penalty: -1; match/mismatch: Blosum 45 matrix
** Recall that when loading Pam matrices you need to select Other from the alphabet choices.

Exercise 4.2 Solution

# Local Alignment

- The semiglobal algorithm works well when the query string can be found largely intact within the target string.

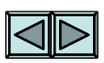
- In some situations, we are interested in matching subsequences of the query string to counterparts in the target string. Semiglobal alignment does not work in this case since internal gaps are penalized.

- We turn instead to **local alignment**. Consider matching `aacctatagct` and `gcgatata` with semiglobal alignment, using the (0, -1, 1, -1) scoring scheme. *Align* gives the following result:

```
aac-ctatagct
-gcgatata---
```

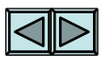
- Note that this alignment only succeeds in matching the subsequences `tata`. We need an algorithm that will identify this match while ignoring the mismatches and gaps around it.

- A modification to the basic dynamic programming algorithm achieves this. It is called the *Smith-Waterman Algorithm.*.



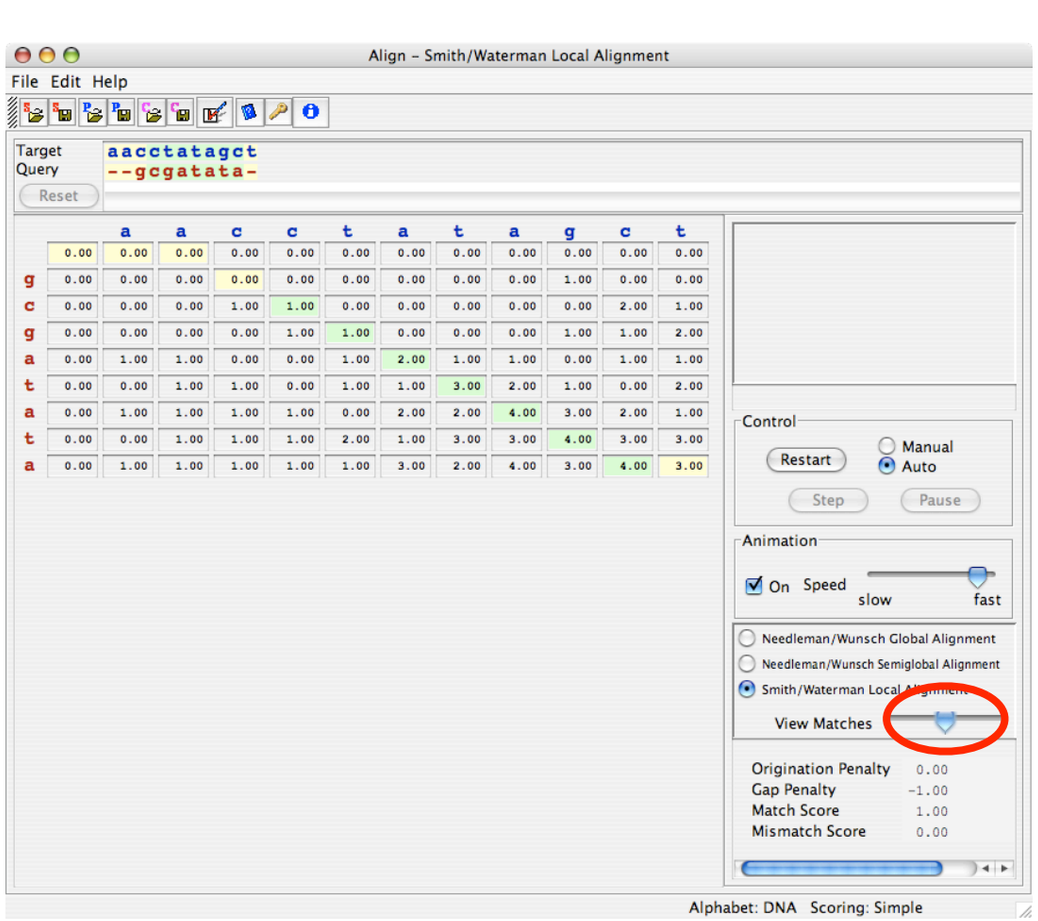
# The Smith-Waterman Algorithm

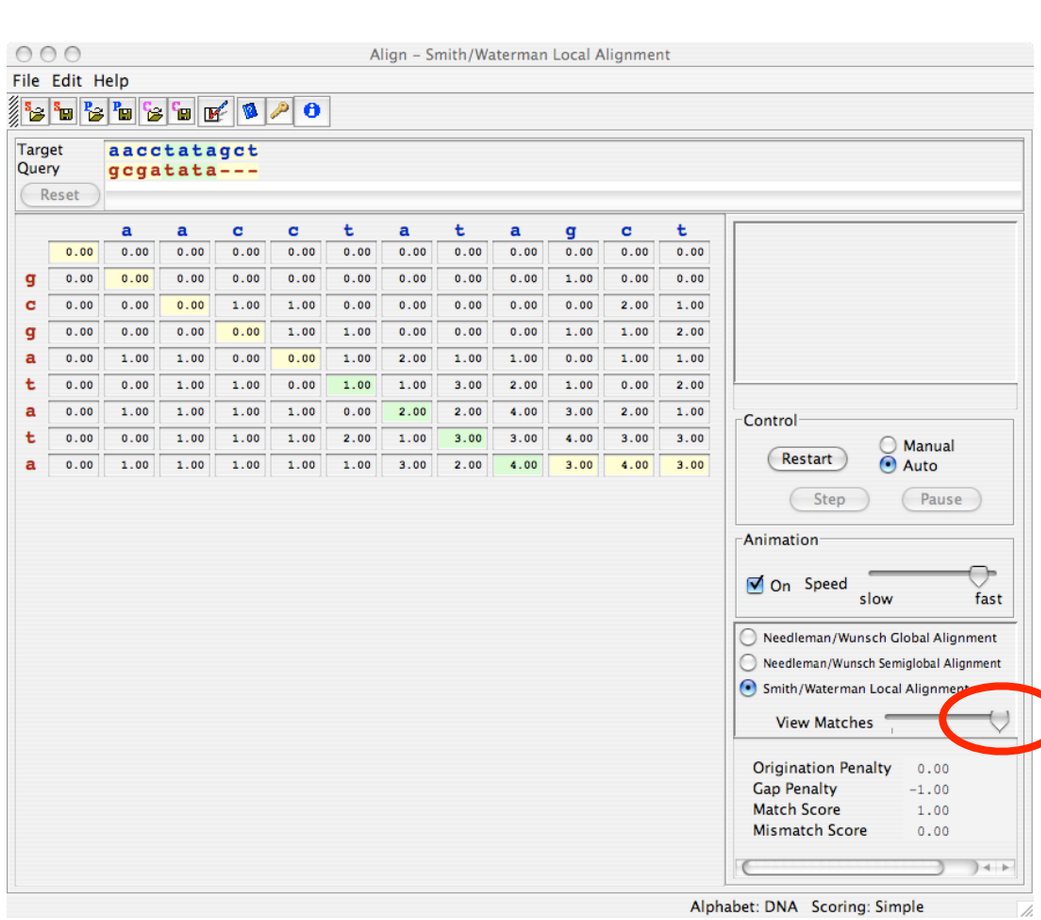
- In the Smith-Waterman algorithm, we proceed by filling in the the table as with our earlier algorithms, but with one additional option:
  - If the value that would be entered in a box would be negative, enter 0 instead.
- When the table is complete, we find the maximal alignment score wherever it occurs in the table, and work backwards to 0. Each such sequence yields an optimal local alignment.
- *Align* can be used with the Smith-Waterman algorithm by selecting that algorithm.
  - When Smith-Waterman is selected, the *View Matches* slider appears. The slider is used to locate all of the local alignments that achieve the optimal score.



# Example

- Again consider matching `aacctatagct` and `gcgatata`, using the (0, -1, 1, 0) scoring scheme.
  - Using **File | Load Sequences ...** enter the sequences shown above.
  - Select *Smith/Waterman Local Alignment* as the algorithm, click *Auto*, then *Solve.* The filled table is shown below in the left-hand figure.
- When the matrix is completely filled, the *View Matches* slider is used to locate all local matches that achieve the optimal score.
  - Click on the hash marks below the slider to highlight each local alignment in turn. The local alignment will be highlighted in green in the both matrix and sequence display.
  - The two local alignments in our example achieving the optimal score (4.0) are shown below.



# Another Example

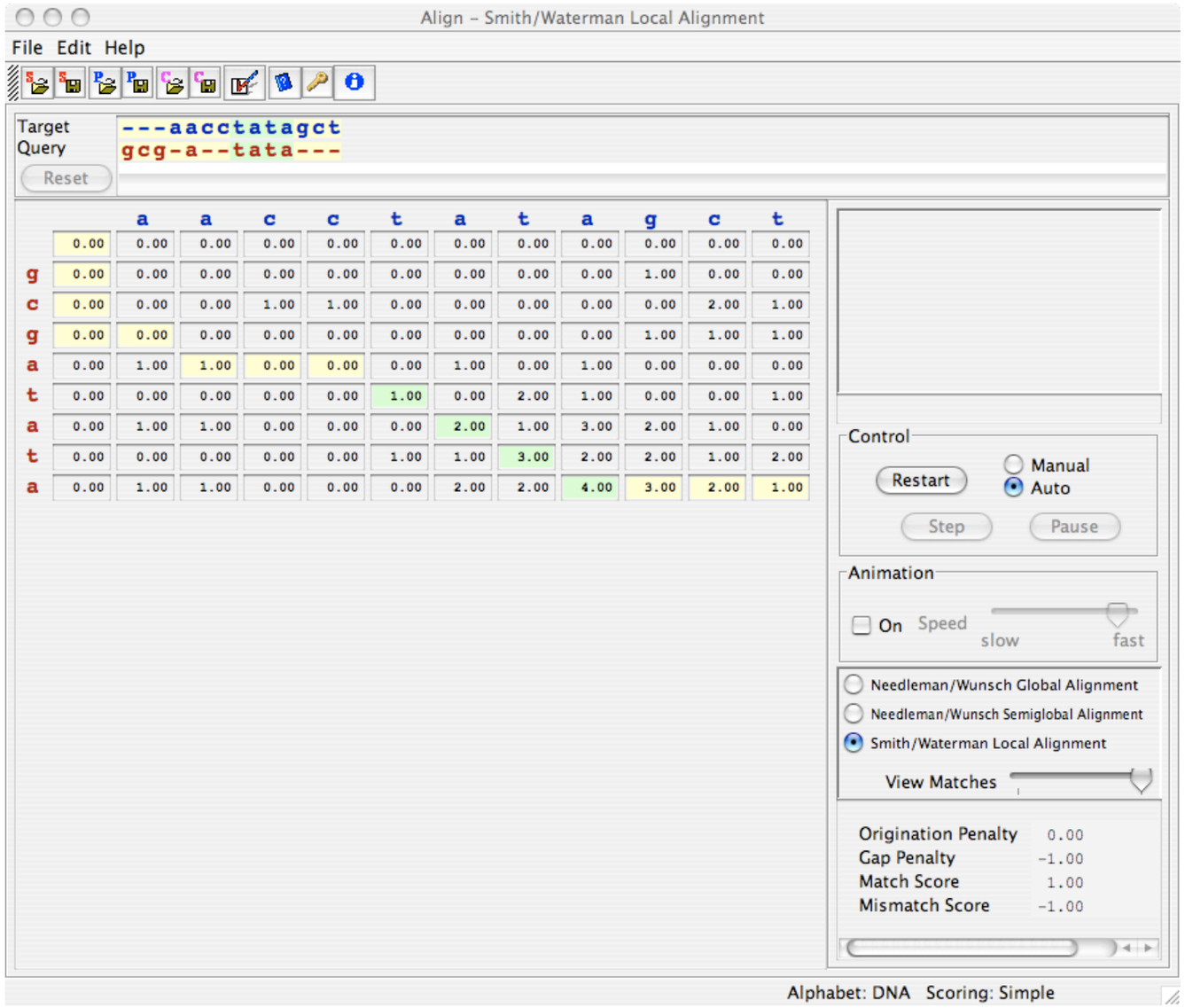
- Let's now try a local match between `aacctatagct` and `gcgatata` this time using the (0, -1, 1, -1) scoring scheme.

- As the figures on the right demonstrate, there is only one optimal local alignment.

- Choice of scoring scheme can greatly affect the selection of optimal alignment. (See Exercise 4.3.)

  - The unique optimal local alignment obtained here was one of the two produced in the previous example.

    Do you think we improved the outcome by switching scoring schemes?



# Exercise 4.3

Find all local matches between the DNA sequences

```
cgtcatatctataagat
actatatatctcaagt
```

using the following scoring schemes:

a) (0, -1, 1, 0)

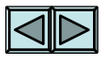
b) (0, -1, 1, -1)

c) (0, -2, 1, -1)

d) (-2, -1, 1, -1)

Analyze the results

Exercise 4.3 Solution



# Exercise 4.4

- Find all local matches between the protein sequences

```
AILMWGVVAMWILVMVGTAVVIAIML
AILMVGTAVWIML
```

using the following scoring schemes:

a) (-12, -1, Blosum 45)
(-12, -5, Blosum 45)

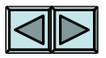
b) (-12, -1, Blosum 62)
(-12, -5, Blosum 62)

c) (-10, -1, Pam 30)
(-10, -5, Pam 30)

d) (-10, -1, Pam 250)
(-10, -5, Pam 250)
(-10, -10, Pam 250)

- Analyze the result

Exercise 4.4 Solution



# Conclusion

- These four laboratory modules and the *Align* tool have introduced some important ideas about aligning sequences:
  - What makes sequence alignment difficult.
  - How dynamic programming simplifies the computational aspects of the problem.
  - How the basic global dynamic programming algorithm works.
  - The importance of proper scoring schemes.
  - The use of scoring matrices, in particular for protein sequence alignment (e.g. BLOSUM and PAM matrices.)
  - Semiglobal and local variations of the basic global dynamic programming algorithm.
- There are excellent online resources available to continue your study of sequence alignment. Among the topics you might explore are:
  - Proper selection of origination and gap penalties.
  - Alignment of 3 or more sequences.
  - Using online sequence alignment tools.

Good Luck!

# Solution to Exercise 4.1

- a) `acatcgatcgtataccgac`
  `-c---g-t---a-------`

  - The internal gaps between `c`, `g`, `t` and `a` show gap penalties in the circled boxes below.

| | | a | c | a | t | c | g | a | t | c | g | t | a | t | a | c | c | g | a | c |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| c | 0.00 | | 1.00 | 0.00 | -1.00 | -2.00 | | | | | | | | | | | | | | |
| g | 0.00 | | | | | | -1.00 | -2.00 | | | | | | | | | | | | |
| t | 0.00 | | | | | | | | -1.00 | -2.00 | -3.00 | -4.00 | | | | | | | | |
| a | 0.00 | | | | | | | | | | | | -3.00 | -3.00 | -3.00 | -3.00 | -3.00 | -3.00 | -3.00 | -3.00 |

- b) `acatcgatcgtataccgac`
  `----cg----ta-------`

  - The internal gaps between g and t show gap penalties in the circled boxes below.

| | | a | c | a | t | c | g | a | t | c | g | t | a | t | a | c | c | g | a | c |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| c | 0.00 | | | | | 1.00 | | | | | | | | | | | | | | |
| g | 0.00 | | | | | | 2.00 | 1.00 | 0.00 | -1.00 | -2.00 | | | | | | | | | |
| t | 0.00 | | | | | | | | | | | -1.00 | | | | | | | | |
| a | 0.00 | | | | | | | | | | | | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

- Note that no such gap penalties are incurred in the preferred alignment.

Return to Exercise 4.1

# Solution to Exercise 4.2

a) (0, -1, 1, 0)

i. trial

Target agatagaaactgatatata
Query ag---aaaacagagt----

Reset

| | | a | g | a | t | a | g | a | a | a | c | t | g | a | t | a | t | a | t | a |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| a | 0.00 | 1.00 | | | | | | | | | | | | | | | | | | |
| g | 0.00 | | 2.00 | 1.00 | 0.00 | -1.00 | | | | | | | | | | | | | | |
| a | 0.00 | | | | | | -1.00 | | | | | | | | | | | | | |
| a | 0.00 | | | | | | | 0.00 | | | | | | | | | | | | |
| a | 0.00 | | | | | | | | 1.00 | | | | | | | | | | | |
| a | 0.00 | | | | | | | | | 2.00 | | | | | | | | | | |
| c | 0.00 | | | | | | | | | | 3.00 | | | | | | | | | |
| a | 0.00 | | | | | | | | | | | 3.00 | | | | | | | | |
| g | 0.00 | | | | | | | | | | | | 4.00 | | | | | | | |
| a | 0.00 | | | | | | | | | | | | | 5.00 | | | | | | |
| g | 0.00 | | | | | | | | | | | | | | 5.00 | | | | | |
| t | 0.00 | | | | | | | | | | | | | | | 5.00 | 5.00 | 5.00 | 5.00 | 5.00 |

ii. optimal

Target agatag-aaactgatatata
Query ----agaaaacaga-gt---

Reset

| | | a | g | a | t | a | g | a | a | a | c | t | g | a | t | a | t | a | t | a |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| a | 0.00 | 1.00 | 0.00 | 1.00 | 0.00 | 1.00 | 0.00 | 1.00 | 1.00 | 1.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 1.00 | 0.00 | 1.00 | 0.00 | 1.00 |
| g | 0.00 | 0.00 | 2.00 | 1.00 | 1.00 | 0.00 | 2.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.00 | 1.00 | 0.00 | 1.00 | 0.00 | 1.00 | 0.00 | 1.00 | 1.00 |
| a | 0.00 | 1.00 | 1.00 | 3.00 | 2.00 | 2.00 | 1.00 | 3.00 | 2.00 | 2.00 | 1.00 | 1.00 | 0.00 | 2.00 | 1.00 | 2.00 | 1.00 | 2.00 | 1.00 | 2.00 |
| a | 0.00 | 1.00 | 1.00 | 2.00 | 3.00 | 3.00 | 2.00 | 2.00 | 4.00 | 3.00 | 2.00 | 1.00 | 1.00 | 1.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 |
| a | 0.00 | 1.00 | 1.00 | 2.00 | 2.00 | 4.00 | 3.00 | 3.00 | 3.00 | 5.00 | 4.00 | 3.00 | 2.00 | 2.00 | 1.00 | 3.00 | 2.00 | 3.00 | 2.00 | 3.00 |
| a | 0.00 | 1.00 | 1.00 | 2.00 | 2.00 | 3.00 | 4.00 | 4.00 | 4.00 | 4.00 | 5.00 | 4.00 | 3.00 | 3.00 | 2.00 | 2.00 | 3.00 | 3.00 | 3.00 | 3.00 |
| c | 0.00 | 0.00 | 1.00 | 1.00 | 2.00 | 2.00 | 3.00 | 4.00 | 4.00 | 4.00 | 5.00 | 5.00 | 4.00 | 3.00 | 3.00 | 2.00 | 2.00 | 3.00 | 3.00 | 3.00 |
| a | 0.00 | 1.00 | 0.00 | 2.00 | 1.00 | 3.00 | 2.00 | 4.00 | 5.00 | 5.00 | 4.00 | 5.00 | 5.00 | 5.00 | 4.00 | 4.00 | 3.00 | 3.00 | 3.00 | 4.00 |
| g | 0.00 | 0.00 | 2.00 | 1.00 | 2.00 | 2.00 | 4.00 | 3.00 | 4.00 | 5.00 | 5.00 | 4.00 | 6.00 | 5.00 | 5.00 | 4.00 | 4.00 | 3.00 | 3.00 | 4.00 |
| a | 0.00 | 1.00 | 1.00 | 3.00 | 2.00 | 3.00 | 3.00 | 5.00 | 4.00 | 5.00 | 5.00 | 5.00 | 5.00 | 7.00 | 6.00 | 6.00 | 5.00 | 5.00 | 4.00 | 4.00 |
| g | 0.00 | 0.00 | 2.00 | 2.00 | 3.00 | 2.00 | 4.00 | 4.00 | 5.00 | 4.00 | 5.00 | 5.00 | 6.00 | 6.00 | 7.00 | 6.00 | 6.00 | 5.00 | 5.00 | 4.00 |
| t | 0.00 | 0.00 | 1.00 | 2.00 | 3.00 | 3.00 | 3.00 | 4.00 | 4.00 | 5.00 | 5.00 | 6.00 | 6.00 | 6.00 | 7.00 | 7.00 | 7.00 | 7.00 | 7.00 | 7.00 |

Return to Exercise 4.2

# Solution to Exercise 4.2 (cont'd)

a) (0, -1, 1, -1)

i. trial

Target agatagaaactgatatata
Query ag---aaaacagagt----

Reset

| | | a | g | a | t | a | g | a | a | a | c | t | g | a | t | a | t | a | t | a |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| a | 0.00 | 1.00 | | | | | | | | | | | | | | | | | | |
| g | 0.00 | | 2.00 | 1.00 | 0.00 | -1.00 | | | | | | | | | | | | | | |
| a | 0.00 | | | | | | -2.00 | | | | | | | | | | | | | |
| a | 0.00 | | | | | | | -1.00 | | | | | | | | | | | | |
| a | 0.00 | | | | | | | | 0.00 | | | | | | | | | | | |
| a | 0.00 | | | | | | | | | 1.00 | | | | | | | | | | |
| c | 0.00 | | | | | | | | | | 2.00 | | | | | | | | | |
| a | 0.00 | | | | | | | | | | | 1.00 | | | | | | | | |
| g | 0.00 | | | | | | | | | | | | 2.00 | | | | | | | |
| a | 0.00 | | | | | | | | | | | | | 3.00 | | | | | | |
| g | 0.00 | | | | | | | | | | | | | | 2.00 | | | | | |
| t | 0.00 | | | | | | | | | | | | | | | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |

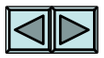
ii. optimal

Target agatag-aaactga-tatata
Query ----agaaaacagagt-----

Reset

| | | a | g | a | t | a | g | a | a | a | c | t | g | a | t | a | t | a | t | a |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| a | 0.00 | 1.00 | 0.00 | 1.00 | 0.00 | 1.00 | 0.00 | 1.00 | 1.00 | 1.00 | 0.00 | -1.00 | -1.00 | 1.00 | 0.00 | 1.00 | 0.00 | 1.00 | 0.00 | 1.00 |
| g | 0.00 | 0.00 | 2.00 | 1.00 | 0.00 | 0.00 | 2.00 | 1.00 | 0.00 | 0.00 | 0.00 | -1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 |
| a | 0.00 | 1.00 | 1.00 | 3.00 | 2.00 | 1.00 | 1.00 | 3.00 | 2.00 | 1.00 | 0.00 | -1.00 | -1.00 | 1.00 | 0.00 | 1.00 | 0.00 | 1.00 | 0.00 | 1.00 |
| a | 0.00 | 1.00 | 0.00 | 2.00 | 2.00 | 3.00 | 2.00 | 2.00 | 4.00 | 3.00 | 2.00 | 1.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 1.00 | 0.00 | 1.00 |
| a | 0.00 | 1.00 | 0.00 | 1.00 | 1.00 | 3.00 | 2.00 | 3.00 | 3.00 | 5.00 | 4.00 | 3.00 | 2.00 | 1.00 | 0.00 | 1.00 | 0.00 | 1.00 | 0.00 | 1.00 |
| a | 0.00 | 1.00 | 0.00 | 1.00 | 0.00 | 2.00 | 2.00 | 3.00 | 4.00 | 4.00 | 4.00 | 3.00 | 2.00 | 3.00 | 2.00 | 1.00 | 0.00 | 1.00 | 0.00 | 1.00 |
| c | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 1.00 | 2.00 | 3.00 | 3.00 | 5.00 | 4.00 | 3.00 | 2.00 | 2.00 | 1.00 | 0.00 | 0.00 | 0.00 | 1.00 |
| a | 0.00 | 1.00 | 0.00 | 1.00 | 0.00 | 1.00 | 0.00 | 2.00 | 3.00 | 4.00 | 4.00 | 4.00 | 3.00 | 4.00 | 3.00 | 3.00 | 2.00 | 1.00 | 0.00 | 1.00 |
| g | 0.00 | 0.00 | 2.00 | 1.00 | 0.00 | 0.00 | 2.00 | 1.00 | 2.00 | 3.00 | 3.00 | 3.00 | 5.00 | 4.00 | 3.00 | 2.00 | 2.00 | 1.00 | 0.00 | 1.00 |
| a | 0.00 | 1.00 | 1.00 | 3.00 | 2.00 | 1.00 | 1.00 | 3.00 | 2.00 | 3.00 | 2.00 | 2.00 | 4.00 | 6.00 | 5.00 | 4.00 | 3.00 | 3.00 | 2.00 | 1.00 |
| g | 0.00 | 0.00 | 2.00 | 2.00 | 2.00 | 1.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 1.00 | 3.00 | 5.00 | 5.00 | 4.00 | 3.00 | 2.00 | 2.00 | 1.00 |
| t | 0.00 | 0.00 | 1.00 | 1.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 4.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 |

Return to Exercise 4.2



# Solution to Exercise 4.2 (cont'd)

b) (0, -1, 1, 0)

i. trial

Target MVAIWRAILVGTVIAML-
Query --AI---LMVGTAVWIML
Reset

| | | M | V | A | I | W | R | A | I | L | V | G | T | V | I | A | M | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A | 0.00 | | | 1.00 | | | | | | | | | | | | | | |
| I | 0.00 | | | | 2.00 | 1.00 | 0.00 | -1.00 | | | | | | | | | | |
| L | 0.00 | | | | | | | | -1.00 | | | | | | | | | |
| M | 0.00 | | | | | | | | | -1.00 | | | | | | | | |
| V | 0.00 | | | | | | | | | | 0.00 | | | | | | | |
| G | 0.00 | | | | | | | | | | | 1.00 | | | | | | |
| T | 0.00 | | | | | | | | | | | | 2.00 | | | | | |
| A | 0.00 | | | | | | | | | | | | | 2.00 | | | | |
| V | 0.00 | | | | | | | | | | | | | | 2.00 | | | |
| W | 0.00 | | | | | | | | | | | | | | | 2.00 | | |
| I | 0.00 | | | | | | | | | | | | | | | | 2.00 | |
| M | 0.00 | | | | | | | | | | | | | | | | | 2.00 |
| L | 0.00 | | | | | | | | | | | | | | | | | 2.00 |

ii. optimal

Target MVAIWRAIL-VGT-VIAML
Query ------AILMVGTAVWIML
Reset

| | | M | V | A | I | W | R | A | I | L | V | G | T | V | I | A | M | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 |
| I | 0.00 | 0.00 | 0.00 | 0.00 | 2.00 | 1.00 | 0.00 | 0.00 | 2.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 1.00 | 0.00 |
| L | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 2.00 | 1.00 | 0.00 | 1.00 | 3.00 | 2.00 | 1.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 2.00 |
| M | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 1.00 | 2.00 | 1.00 | 0.00 | 2.00 | 3.00 | 2.00 | 1.00 | 0.00 | 0.00 | 0.00 | 2.00 | 2.00 |
| V | 0.00 | 0.00 | 2.00 | 1.00 | 0.00 | 0.00 | 1.00 | 2.00 | 1.00 | 1.00 | 3.00 | 3.00 | 2.00 | 2.00 | 1.00 | 0.00 | 1.00 | 2.00 |
| G | 0.00 | 0.00 | 1.00 | 2.00 | 1.00 | 0.00 | 0.00 | 1.00 | 2.00 | 1.00 | 2.00 | 4.00 | 3.00 | 2.00 | 2.00 | 1.00 | 0.00 | 2.00 |
| T | 0.00 | 0.00 | 0.00 | 1.00 | 2.00 | 1.00 | 0.00 | 0.00 | 1.00 | 2.00 | 1.00 | 3.00 | 5.00 | 4.00 | 3.00 | 2.00 | 1.00 | 2.00 |
| A | 0.00 | 0.00 | 0.00 | 1.00 | 1.00 | 2.00 | 1.00 | 1.00 | 0.00 | 1.00 | 2.00 | 2.00 | 4.00 | 5.00 | 4.00 | 4.00 | 3.00 | 2.00 |
| V | 0.00 | 0.00 | 1.00 | 0.00 | 1.00 | 1.00 | 2.00 | 1.00 | 1.00 | 0.00 | 2.00 | 2.00 | 3.00 | 5.00 | 5.00 | 4.00 | 4.00 | 3.00 |
| W | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 2.00 | 1.00 | 2.00 | 1.00 | 1.00 | 1.00 | 2.00 | 2.00 | 4.00 | 5.00 | 5.00 | 4.00 | 4.00 |
| I | 0.00 | 0.00 | 0.00 | 0.00 | 2.00 | 1.00 | 2.00 | 1.00 | 3.00 | 2.00 | 1.00 | 1.00 | 2.00 | 3.00 | 5.00 | 5.00 | 5.00 | 4.00 |
| M | 0.00 | 1.00 | 0.00 | 0.00 | 1.00 | 2.00 | 1.00 | 2.00 | 2.00 | 3.00 | 2.00 | 1.00 | 1.00 | 2.00 | 4.00 | 5.00 | 6.00 | 5.00 |
| L | 0.00 | 0.00 | 1.00 | 1.00 | 1.00 | 1.00 | 2.00 | 2.00 | 2.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 4.00 | 5.00 | 7.00 |

Return to Exercise 4.2

# Solution to Exercise 4.2 (cont'd)

b) (-12, -1, Blosum 45)

i. trial

Target MVAIWRAILVGTVIAML-
Query --AI---LMVGTAVWIML

Reset

| | | M | V | A | I | W | R | A | I | L | V | G | T | V | I | A | M | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A | 0.00 | | | 5.00 | | | | | | | | | | | | | | |
| I | 0.00 | | | | 10.00 | -3.00 | -4.00 | -5.00 | | | | | | | | | | |
| L | 0.00 | | | | | | | | -3.00 | | | | | | | | | |
| M | 0.00 | | | | | | | | | -1.00 | | | | | | | | |
| V | 0.00 | | | | | | | | | | 4.00 | | | | | | | |
| G | 0.00 | | | | | | | | | | | 11.00 | | | | | | |
| T | 0.00 | | | | | | | | | | | | 16.00 | | | | | |
| A | 0.00 | | | | | | | | | | | | | 16.00 | | | | |
| V | 0.00 | | | | | | | | | | | | | | 19.00 | | | |
| W | 0.00 | | | | | | | | | | | | | | | 17.00 | | |
| I | 0.00 | | | | | | | | | | | | | | | | 19.00 | |
| M | 0.00 | | | | | | | | | | | | | | | | | 21.00 |
| L | 0.00 | | | | | | | | | | | | | | | | | 21.00 |

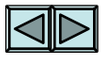
ii. optimal

Target MVAIWR-AILVGTVIAML-
Query ------AILMVGTAVWIML

Reset

| | | M | V | A | I | W | R | A | I | L | V | G | T | V | I | A | M | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A | 0.00 | -1.00 | 0.00 | 5.00 | -1.00 | -1.00 | -1.00 | 5.00 | -1.00 | -1.00 | 0.00 | 0.00 | 0.00 | 0.00 | -1.00 | 5.00 | -1.00 | 0.00 |
| I | 0.00 | 2.00 | 2.00 | -1.00 | 10.00 | -2.00 | -2.00 | -2.00 | 10.00 | 1.00 | 2.00 | -4.00 | -1.00 | 3.00 | 5.00 | -2.00 | 7.00 | 1.00 |
| L | 0.00 | 2.00 | 3.00 | 1.00 | 1.00 | 8.00 | -3.00 | -3.00 | 0.00 | 15.00 | 2.00 | -1.00 | -5.00 | 0.00 | 5.00 | 4.00 | 0.00 | 12.00 |
| M | 0.00 | 6.00 | 3.00 | 2.00 | 3.00 | -1.00 | 7.00 | -4.00 | -1.00 | 2.00 | 16.00 | 3.00 | 2.00 | 1.00 | 2.00 | 4.00 | 10.00 | 12.00 |
| V | 0.00 | 2.00 | 11.00 | 3.00 | 5.00 | 0.00 | -3.00 | 7.00 | -1.00 | 0.00 | 7.00 | 13.00 | 3.00 | 7.00 | 4.00 | 2.00 | 6.00 | 12.00 |
| G | 0.00 | -1.00 | -1.00 | 11.00 | -1.00 | 3.00 | -2.00 | -3.00 | 3.00 | -4.00 | -3.00 | 14.00 | 11.00 | 0.00 | 3.00 | 4.00 | 0.00 | 12.00 |
| T | 0.00 | -1.00 | -1.00 | -1.00 | 10.00 | -3.00 | 2.00 | -2.00 | -4.00 | 2.00 | -4.00 | 1.00 | 19.00 | 11.00 | -1.00 | 3.00 | 3.00 | 12.00 |
| A | 0.00 | -1.00 | -1.00 | 4.00 | -2.00 | 8.00 | -5.00 | 7.00 | -3.00 | -5.00 | 2.00 | 0.00 | 6.00 | 19.00 | 10.00 | 4.00 | 2.00 | 12.00 |
| V | 0.00 | 2.00 | 4.00 | -1.00 | 7.00 | -5.00 | 6.00 | -5.00 | 10.00 | -2.00 | 0.00 | -1.00 | 5.00 | 11.00 | 22.00 | 10.00 | 6.00 | 12.00 |
| W | 0.00 | -1.00 | -1.00 | 2.00 | -3.00 | 22.00 | 9.00 | 8.00 | 7.00 | 8.00 | -5.00 | -2.00 | 4.00 | 3.00 | 9.00 | 20.00 | 8.00 | 12.00 |
| I | 0.00 | 2.00 | 2.00 | -2.00 | 7.00 | 9.00 | 19.00 | 8.00 | 13.00 | 9.00 | 11.00 | -2.00 | 3.00 | 7.00 | 8.00 | 8.00 | 22.00 | 12.00 |
| M | 0.00 | 6.00 | 3.00 | 1.00 | 0.00 | 8.00 | 8.00 | 18.00 | 10.00 | 15.00 | 10.00 | 9.00 | 2.00 | 4.00 | 9.00 | 7.00 | 14.00 | 24.00 |
| L | 0.00 | 2.00 | 7.00 | 7.00 | 7.00 | 7.00 | 7.00 | 7.00 | 20.00 | 20.00 | 20.00 | 20.00 | 20.00 | 20.00 | 20.00 | 20.00 | 20.00 | 24.00 |

Return to Exercise 4.2



# Solution to Exercise 4.2 (cont'd)

b) (-10, -1, Pam 30)

i. trial

Target MVAIWRAILVGTVIAML-
Query --AI---LMVGTAVWIML

Reset

| | | M | V | A | I | W | R | A | I | L | V | G | T | V | I | A | M | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A | 0.00 | | | 6.00 | | | | | | | | | | | | | | |
| I | 0.00 | | | | 14.00 | 3.00 | 2.00 | 1.00 | | | | | | | | | | |
| L | 0.00 | | | | | | | | 0.00 | | | | | | | | | |
| M | 0.00 | | | | | | | | | 1.00 | | | | | | | | |
| V | 0.00 | | | | | | | | | | 8.00 | | | | | | | |
| G | 0.00 | | | | | | | | | | | 14.00 | | | | | | |
| T | 0.00 | | | | | | | | | | | | 21.00 | | | | | |
| A | 0.00 | | | | | | | | | | | | | 19.00 | | | | |
| V | 0.00 | | | | | | | | | | | | | | 21.00 | | | |
| W | 0.00 | | | | | | | | | | | | | | | 8.00 | | |
| I | 0.00 | | | | | | | | | | | | | | | | 7.00 | |
| M | 0.00 | | | | | | | | | | | | | | | | | 8.00 |
| L | 0.00 | | | | | | | | | | | | | | | | | 8.00 |

ii. optimal

Target MVAIWRAIL-VGTVIA--ML
Query ------AILMVGTAV-WIML

Reset

| | | M | V | A | I | W | R | A | I | L | V | G | T | V | I | A | M | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A | 0.00 | -1.00 | -1.00 | 6.00 | -1.00 | -1.00 | -1.00 | 6.00 | -1.00 | -1.00 | -1.00 | -1.00 | -1.00 | -1.00 | -1.00 | 6.00 | -1.00 | 0.00 |
| I | 0.00 | -1.00 | 1.00 | -5.00 | 14.00 | 3.00 | 2.00 | 1.00 | 14.00 | 3.00 | 2.00 | 1.00 | 0.00 | 1.00 | 7.00 | -4.00 | 5.00 | 0.00 |
| L | 0.00 | 1.00 | -3.00 | -5.00 | 3.00 | 8.00 | 1.00 | 0.00 | 3.00 | 21.00 | 10.00 | 9.00 | 8.00 | 7.00 | 6.00 | 5.00 | 4.00 | 12.00 |
| M | 0.00 | 11.00 | 0.00 | -8.00 | 2.00 | 1.00 | 4.00 | -1.00 | 2.00 | 10.00 | 20.00 | 9.00 | 8.00 | 7.00 | 6.00 | 4.00 | 16.00 | 12.00 |
| V | 0.00 | 0.00 | 18.00 | 7.00 | 6.00 | 5.00 | 4.00 | 3.00 | 2.00 | 9.00 | 17.00 | 15.00 | 7.00 | 15.00 | 9.00 | 4.00 | 5.00 | 14.00 |
| G | 0.00 | -1.00 | 7.00 | 16.00 | 5.00 | 4.00 | 3.00 | 2.00 | 1.00 | 8.00 | 7.00 | 23.00 | 12.00 | 11.00 | 10.00 | 9.00 | 8.00 | 14.00 |
| T | 0.00 | -1.00 | 6.00 | 6.00 | 14.00 | 3.00 | 2.00 | 2.00 | 0.00 | 7.00 | 6.00 | 12.00 | 30.00 | 19.00 | 18.00 | 17.00 | 16.00 | 15.00 |
| A | 0.00 | -1.00 | 5.00 | 12.00 | 3.00 | 2.00 | 1.00 | 8.00 | -3.00 | 6.00 | 5.00 | 11.00 | 19.00 | 28.00 | 17.00 | 24.00 | 15.00 | 15.00 |
| V | 0.00 | -1.00 | 6.00 | 3.00 | 14.00 | 3.00 | 2.00 | 1.00 | 10.00 | 5.00 | 13.00 | 10.00 | 18.00 | 26.00 | 30.00 | 19.00 | 23.00 | 15.00 |
| W | 0.00 | -1.00 | -2.00 | -3.00 | 3.00 | 27.00 | 16.00 | 15.00 | 14.00 | 13.00 | 12.00 | 11.00 | 17.00 | 16.00 | 19.00 | 18.00 | 17.00 | 17.00 |
| I | 0.00 | -1.00 | 1.00 | -4.00 | 5.00 | 16.00 | 22.00 | 14.00 | 23.00 | 13.00 | 15.00 | 10.00 | 16.00 | 19.00 | 24.00 | 17.00 | 17.00 | 17.00 |
| M | 0.00 | 11.00 | 0.00 | -1.00 | -2.00 | 15.00 | 14.00 | 17.00 | 13.00 | 24.00 | 13.00 | 12.00 | 15.00 | 15.00 | 18.00 | 19.00 | 28.00 | 18.00 |
| L | 0.00 | 1.00 | 9.00 | 9.00 | 9.00 | 14.00 | 14.00 | 14.00 | 16.00 | 20.00 | 22.00 | 22.00 | 22.00 | 22.00 | 22.00 | 22.00 | 22.00 | 35.00 |

Return to Exercise 4.2

# Solution to Exercise 4.3

a) (0, -1, 1, 0)
2 solutions, optimal score of 3.0.

```
-cgtcatatctataagat
actatatatct-caag-t
```

```
-cgtcatatctataagat
actatatatct-caagt-
```

b) (0, -1, 1, -1)
1 solution, optimal score of 7.0.

```
----cgtcatatctataagat
acta--t-atatct-caag-t
```

c) (0, -2, 1, -1)
1 solution, optimal score of 3.0

```
----cgtcatatctataagat
acta--t-atatct-caagt-
```

d) (-2, -1, 1, -1)

```
----cgtcatatctataagat
acta--t-atatctcaagt--
```

In (a), where gaps and mismatches have equal value, the algorithm is not sufficiently selective, producing 2 long solutions containing mismatches and gaps.

In (b), the mismatch penalty is raised. Gaps are introduced to avoid mismatches in a shorter local alignment.

In (c), the gap penalty is raised, resulting in an even shorter local alignment, but one containing a gap.

In (d), an origination penalty is introduced, reducing the tolerance for internal gaps. This produces the tightest local alignment.

Return to Exercise 4.3

# Solution to Exercise 4.4

a) (-12, -1, Blosum 45)
(-12, -5, Blosum 45)

```
AILMWGVVAMWILVMVGTAVVIAIML
AILMVG-TAVWIML------------
```

```
AILMWGVVAMWILVMVGTAVVIAIML
--------A---ILMVGTAVWIML--
```

b) (-12, -1, Blosum 62)
(-12, -5, Blosum 62)

```
AILMWGVVAMWILVMVGTAVVIAIML
--------A---ILMVGTAVWIML--
```

```
AILMWGVVAMWILVMVGTAVVIAIML
--------A---ILMVGTAVWIML--
```

c) (-10, -1, Pam 30)
(-10, -5, Pam 30)

```
AILMWGVVAMWILVMVGTAV-VIAIML
--------A--IL-MVGTAVW---IML
```

```
AILMWGVVAMWILVMVGTAVVIAIML
--------A--IL-MVGTAVWI--ML
```

d) (-10, -1, Pam 250)
(-10, -5, Pam 250)
(-10, -10, Pam 250)

```
AILMWGVVAMWILVMVGTAVVIAIML
AILMVG-TAVWIML------------
```

```
AILMWGVVAMWILVMVGTAVVIAIML
AILMVG-TAVWIML------------
```

```
AILMWGVVAMWILVMVGTAVVIAIML
A-ILMVGTAVWIML------------
```

- Larger gap penalty often results in narrower local match (see (d) in particular).
  Proper choice of origination and gap penalties is crucial (see (c) in particular).
  Different matrices may single out different local match regions.

Return to Exercise 4.4