Bioinformatics

CS300

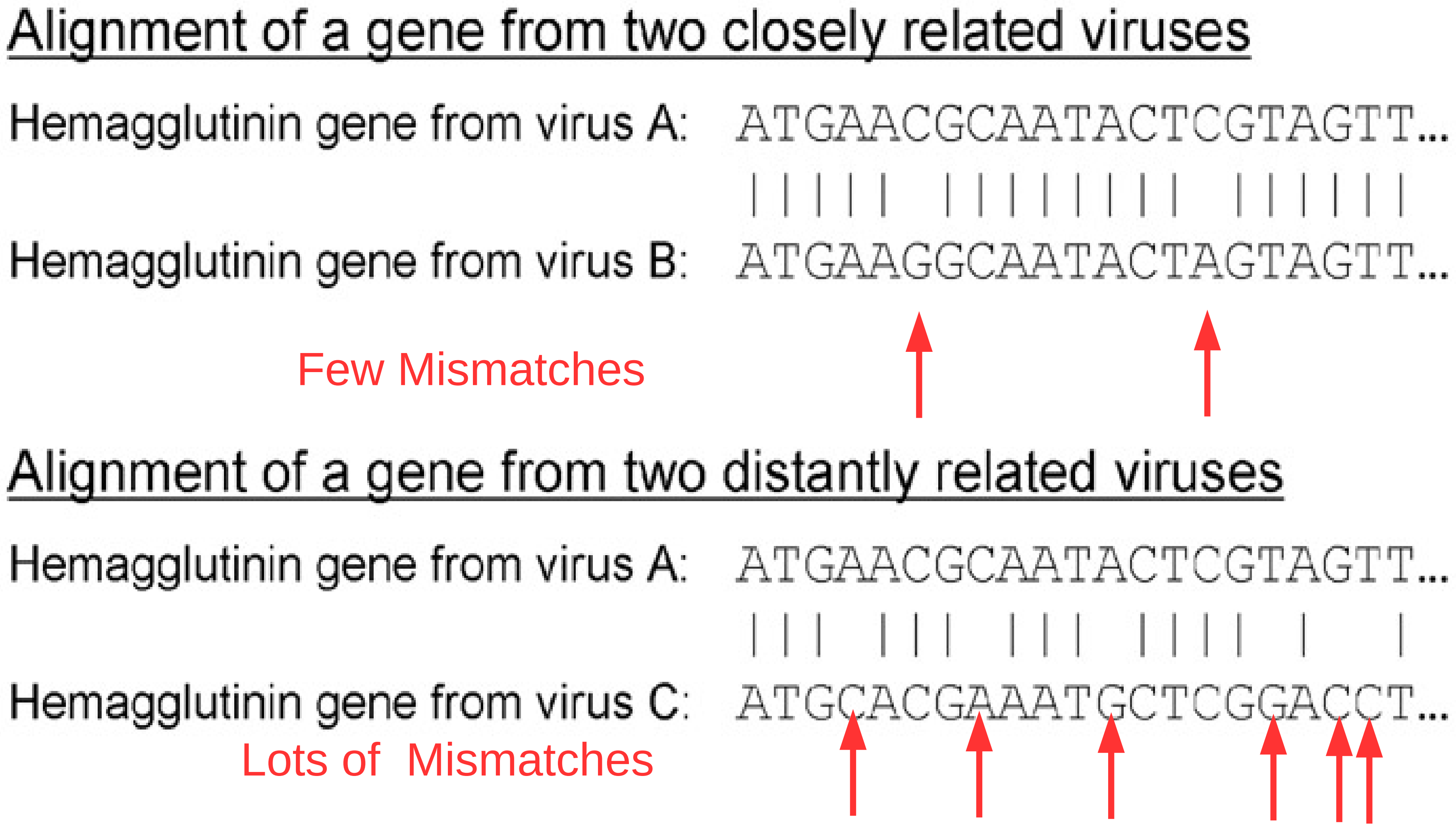
Chap 3

Sequence Alignment:
Investigating an Influenza Outbreak

Fall 2017

Oliver Bonham-Carter

Pairwise Alignment
Similarity and Relatedness



Alignment of a gene from two closely related viruses

```
Hemagglutinin gene from virus A:   ATGAACGCAATACTCGTAGTT...
                                   ||||| |||||||| ||||||
Hemagglutinin gene from virus B:   ATGAAGGCAATACTAGTAGTT...
```

Few Mismatches

Alignment of a gene from two distantly related viruses

```
Hemagglutinin gene from virus A:   ATGAACGCAATACTCGTAGTT...
                                   ||| ||| ||| |||| |  |
Hemagglutinin gene from virus C:   ATGCACGAAATGCTCGGACCT...
```

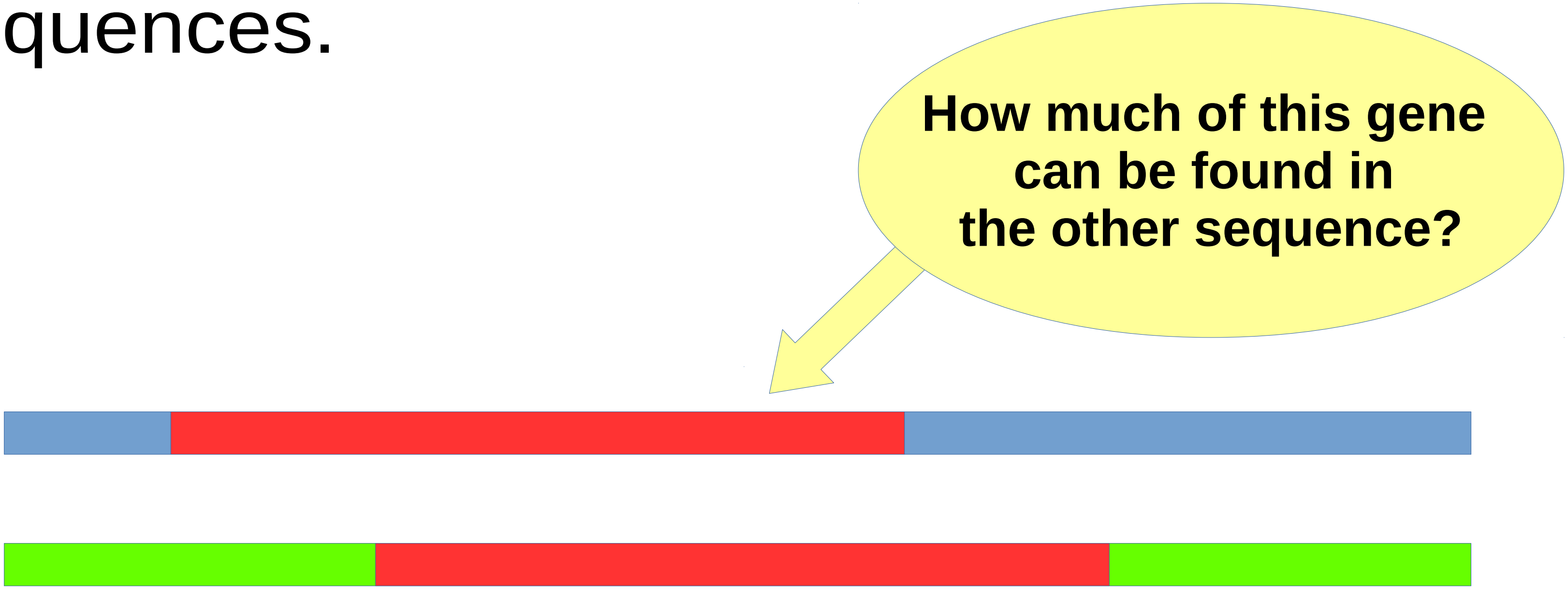
Lots of Mismatches

Concept Questions: Discuss With Your Group

- How is similarity between genes related to the biological concept of descent from a common ancestor?
- The influenza virus mutates so rapidly that you would likely be able to identify at least a couple of mutations over the length of the complete virus genome even if you sequenced two viruses from two different patients within the same influenza outbreak. What do you think might be some considerations in deciding whether two viruses with different genome sequences actually represent two different strains?
- RNA viruses (retro viruses) are prone to genetic mutations during replication. After a mutation, parts of their genetics have been completely changed. Is it still possible to study their genetics (over long periods of time) given their code will quickly change? Why or why not?

What is Sequence Alignment?

- Sequence alignment is a way of arranging the sequence of genetic material (DNA, RNA or protein) to identify regions of similarity that may be a consequence of functional, structural or evolutional relationships between the sequences.

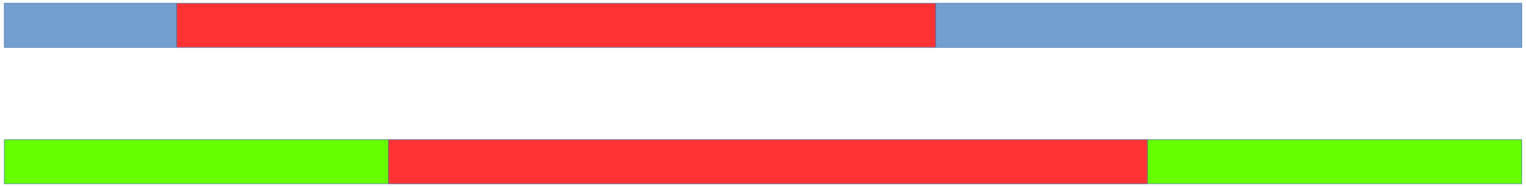


How much of this gene can be found in the other sequence?

What is Global Sequence Alignment?

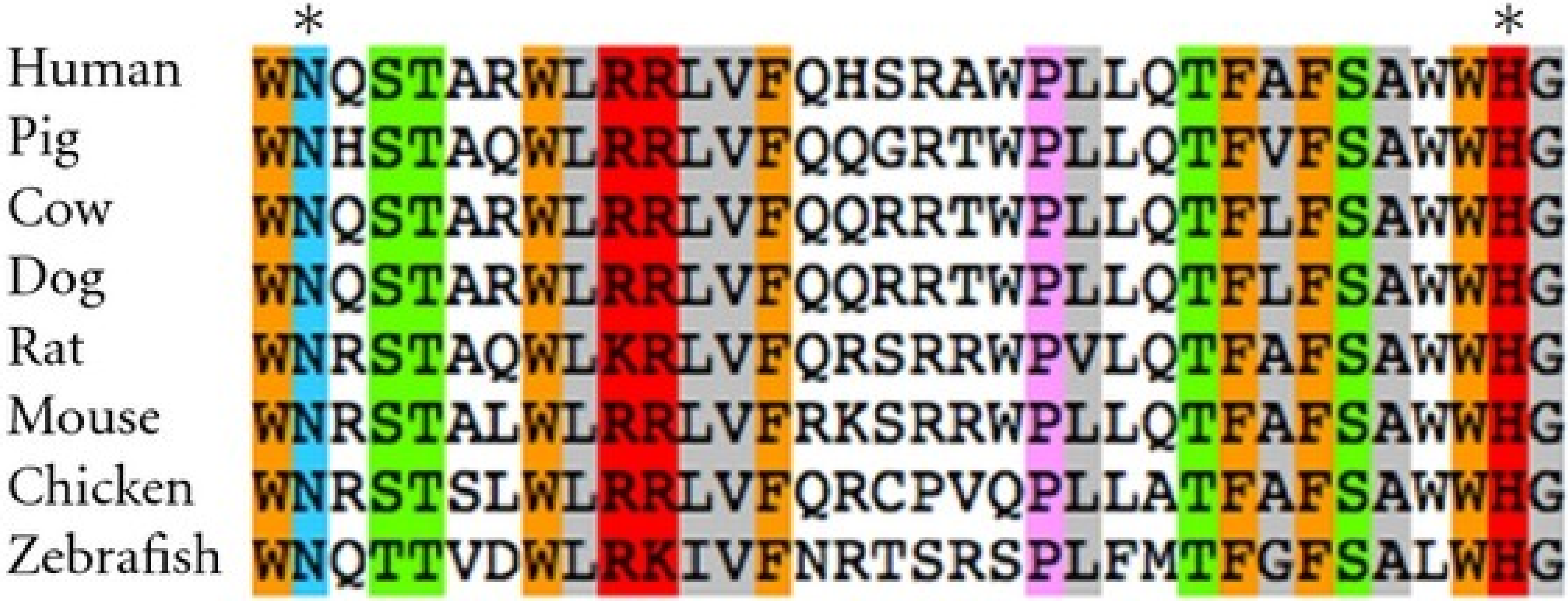
- We search for matches, matches and gaps between two sequences to determine their relatedness.
- (*) indicate matches or similar nucleotides along sequence
- Here, the sequences may share some common ancestor

```
ACGTACT    ACGTAC-T    ACGTACT----
ACTACGT    AC-TACGT    ----ACTACGT
**    *    ** *** *        ***
```



Ex:Sequence Alignment of Some Organisms

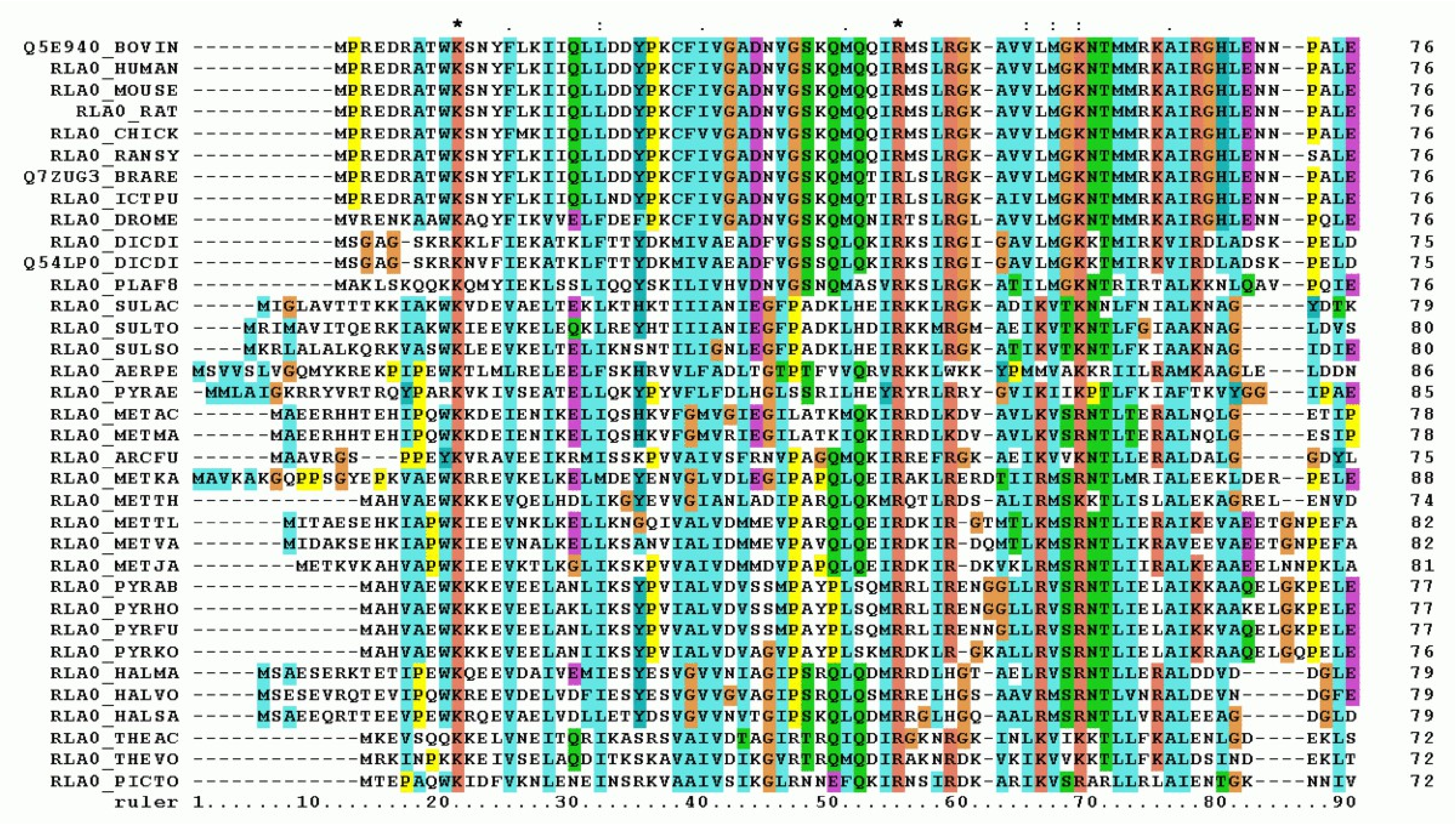
- We compare protein samples from several different organisms.



```
            *                                *
Human       WNQSTARWLRRLVFQHSRAWPLLQTFAFSAWWHG
Pig         WNHSTAQWLRRLVFQQGRTWPLLQTFVFSAWWHG
Cow         WNQSTARWLRRLVFQQRRTWPLLQTFLFSAWWHG
Dog         WNQSTARWLRRLVFQQRRTWPLLQTFLFSAWWHG
Rat         WNRSTAQWLKRLVFQRSRRWPVLQTFAFSAWWHG
Mouse       WNRSTALWLRRLVFRKSRRWPLLQTFAFSAWWHG
Chicken     WNRSTSLWLRRLVFQRCPVQPLLATFAFSAWWHG
Zebrafish   WNQTTVDWLRKIVFNRTSRSPLFMTFGFSALWHG
```

Ex: Sequence Alignment of Many Organisms

- The common code have same nucleotides but there are still breaks in these common regions.



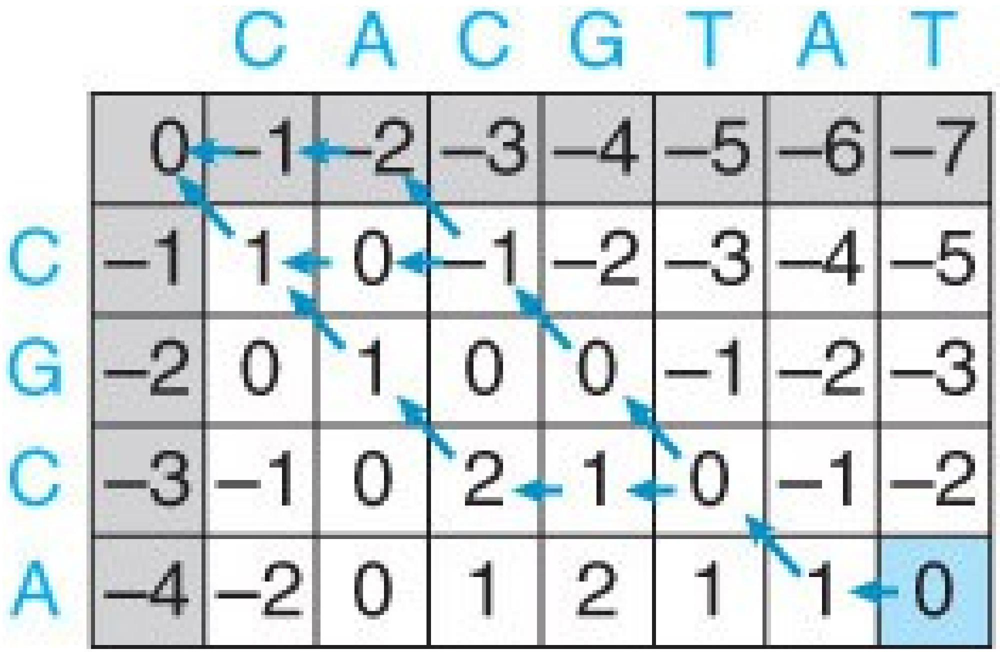
```
                                 *   .      :       .          .   *         : : :      .
Q5E940_BOVIN -----------MPREDRATWKSNYFLKIIQLLDDYPKCFIVGADNVGSKQMQQIRMSLRGK-AVVLMGKNTMMRKAIRGHLENN--PALE    76
  RLA0_HUMAN -----------MPREDRATWKSNYFLKIIQLLDDYPKCFIVGADNVGSKQMQQIRMSLRGK-AVVLMGKNTMMRKAIRGHLENN--PALE    76
  RLA0_MOUSE -----------MPREDRATWKSNYFLKIIQLLDDYPKCFIVGADNVGSKQMQQIRMSLRGK-AVVLMGKNTMMRKAIRGHLENN--PALE    76
    RLA0_RAT -----------MPREDRATWKSNYFLKIIQLLDDYPKCFIVGADNVGSKQMQQIRMSLRGK-AVVLMGKNTMMRKAIRGHLENN--PALE    76
  RLA0_CHICK -----------MPREDRATWKSNYFMKIIQLLDDYPKCFVVGADNVGSKQMQQIRMSLRGK-AVVLMGKNTMMRKAIRGHLENN--PALE    76
  RLA0_RANSY -----------MPREDRATWKSNYFLKIIQLLDDYPKCFIVGADNVGSKQMQQIRMSLRGK-AVVLMGKNTMMRKAIRGHLENN--SALE    76
Q7ZUG3_BRARE -----------MPREDRATWKSNYFLKIIQLLDDYPKCFIVGADNVGSKQMQTIRLSLRGK-AVVLMGKNTMMRKAIRGHLENN--PALE    76
  RLA0_ICTPU -----------MPREDRATWKSNYFLKIIQLLNDYPKCFIVGADNVGSKQMQTIRLSLRGK-AIVLMGKNTMMRKAIRGHLENN--PALE    76
  RLA0_DROME -----------MVRENKAAWKAQYFIKVVELFDEFPKCFIVGADNVGSKQMQNIRTSLRGL-AVVLMGKNTMMRKAIRGHLENN--PQLE    76
  RLA0_DICDI -----------MSGAG-SKRKKLFIEKATKLFTTYDKMIVAEADFVGSSQLQKIRKSIRGI-GAVLMGKKTMIRKVIRDLADSK--PELD    75
Q54LP0_DICDI -----------MSGAG-SKRKNVFIEKATKLFTTYDKMIVAEADFVGSSQLQKIRKSIRGI-GAVLMGKKTMIRKVIRDLADSK--PELD    75
  RLA0_PLAF8 -----------MAKLSKQQKKQMYIEKLSSLIQQYSKILIVHVDNVGSNQMASVRKSLRGK-ATILMGKNTRIRTALKKNLQAV--PQIE    76
  RLA0_SULAC -----MIGLAVTTTKKIAKWKVDEVAELTEKLKTHKTIIIANIEGFPADKLHEIRKKLRGK-ADIKVTKNNLFNIALKNAG-----YDTK    79
  RLA0_SULTO ----MRIMAVITQERKIAKWKIEEVKELEQKLREYHTIIIANIEGFPADKLHDIRKKMRGM-AEIKVTKNTLFGIAAKNAG-----LDVS    80
  RLA0_SULSO ----MKRLALALKQRKVASWKLEEVKELTELIKNSNTILIGNLEGFPADKLHEIRKKLRGK-ATIKVTKNTLFKIAAKNAG-----IDIE    80
  RLA0_AERPE MSVVSLVGQMYKREKPIPEWKTLMLRELEELFSKHRVVLFADLTGTPTFVVQRVRKKLWKK-YPMMVAKKRIILRAMKAAGLE---LDDN    86
  RLA0_PYRAE -MMLAIGKRRYVRTRQYPARKVKIVSEATELLQKYPYVFLFDLHGLSSRILHEYRYRLRRY-GVIKIIKPTLFKIAFTKVYGG---IPAE    85
  RLA0_METAC ------MAEERHHTEHIPQWKKDEIENIKELIQSHKVFGMVGIEGILATKMQKIRRDLKDV-AVLKVSRNTLTERALNQLG-----ETIP    78
  RLA0_METMA ------MAEERHHTEHIPQWKKDEIENIKELIQSHKVFGMVRIEGILATKIQKIRRDLKDV-AVLKVSRNTLTERALNQLG-----ESIP    78
  RLA0_ARCFU ------MAAVRGS---PPEYKVRAVEEIKRMISSKPVVAIVSFRNVPAGQMQKIRREFRGK-AEIKVVKNTLLERALDALG-----GDYL    75
  RLA0_METKA MAVKAKGQPPSGYEPKVAEWKRREVKELKELMDEYENVGLVDLEGIPAPQLQEIRAKLRERDTIIRMSRNTLMRIALEEKLDER--PELE    88
  RLA0_METTH -------------MAHVAEWKKKEVQELHDLIKGYEVVGIANLADIPARQLQKMRQTLRDS-ALIRMSKKTLISLALEKAGREL--ENVD    74
  RLA0_METTL -------MITAESEHKIAPWKIEEVNKLKELLKNGQIVALVDMMEVPARQLQEIRDKIR-GTMTLKMSRNTLIERAIKEVAEETGNPEFA    82
  RLA0_METVA -------MIDAKSEHKIAPWKIEEVNALKELLKSANVIALIDMMEVPAVQLQEIRDKIR-DQMTLKMSRNTLIKRAVEEVAEETGNPEFA    82
  RLA0_METJA --------METKVKAHVAPWKIEEVKTLKGLIKSKPVVAIVDMMDVPAPQLQEIRDKIR-DKVKLRMSRNTLIIRALKEAAEELNNPKLA    81
  RLA0_PYRAB -------------MAHVAEWKKKEVEELANLIKSYPVIALVDVSSMPAYPLSQMRRLIRENGGLLRVSRNTLIELAIKKAAQELGKPELE    77
  RLA0_PYRHO -------------MAHVAEWKKKEVEELAKLIKSYPVIALVDVSSMPAYPLSQMRRLIRENGGLLRVSRNTLIELAIKKAAKELGKPELE    77
  RLA0_PYRFU -------------MAHVAEWKKKEVEELANLIKSYPVVALVDVSSMPAYPLSQMRRLIRENNGLLRVSRNTLIELAIKKVAQELGKPELE    77
  RLA0_PYRKO -------------MAHVAEWKKKEVEELANIIKSYPVIALVDVAGVPAYPLSKMRDKLR-GKALLRVSRNTLIELAIKRAAQELGQPELE    76
  RLA0_HALMA -----MSAESERKTETIPEWKQEEVDAIVEMIESYESVGVVNIAGIPSRQLQDMRRDLHGT-AELRVSRNTLLERALDDVD-----DGLE    79
  RLA0_HALVO -----MSESEVRQTEVIPQWKREEVDELVDFIESYESVGVVGVAGIPSRQLQSMRRELHGS-AAVRMSRNTLVNRALDEVN-----DGFE    79
  RLA0_HALSA -----MSAEEQRTTEEVPEWKRQEVAELVDLLETYDSVGVVNVTGIPSKQLQDMRRGLHGQ-AALRMSRNTLLVRALEEAG-----DGLD    79
  RLA0_THEAC -------------MKEVSQQKKELVNEITQRIKASRSVAIVDTAGIRTRQIQDIRGKNRGK-INLKVIKKTLLFKALENLGD----EKLS    72
  RLA0_THEVO -------------MRKINPKKKEIVSELAQDITKSKAVAIVDIKGVRTRQMQDIRAKNRDK-VKIKVVKKTLLFKALDSIND----EKLT    72
  RLA0_PICTO -------------MTEPAQWKIDFVKNLENEINSRKVAAIVSIKGLRNNEFQKIRNSIRDK-ARIKVSRARLLRLAIENTGK----NNIV    72
       ruler 1.......10........20........30........40........50........60........70........80........90
```

Needleman-Wunsch Algorithm Background

- Global Alignment: Used to determine which parts of a sequence (inside the sequence) are shared (common) with another sequence.
- Developed by Saul B. Needleman and Christian D. Wunsch in 1970.
- Dynamic programming to find optimal solution for matching the characters of the two sequences.

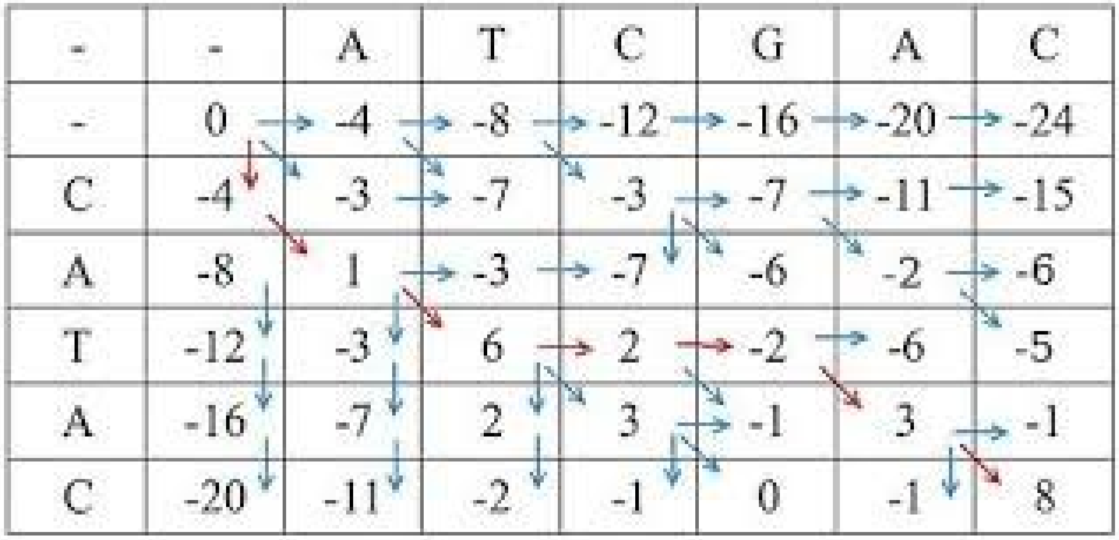
Global Pairwise Alignment

- Steps to begin
  - Initialization of the matrix
  - Calculation of the scores given for a character by character comparison.
  - Filling in a system of arrows in a trace-back matrix to uncover a path back to the start in the score matrix.
  - Deducing the alignment by following the arrows in the trace-back matrix.



		C	A	C	G	T	A	T
	0	–1	–2	–3	–4	–5	–6	–7
C	–1	1	0	–1	–2	–3	–4	–5
G	–2	0	1	0	0	–1	–2	–3
C	–3	–1	0	2	1	0	–1	–2
A	–4	–2	0	1	2	1	1	0

Needleman-Wunsch Algorithm

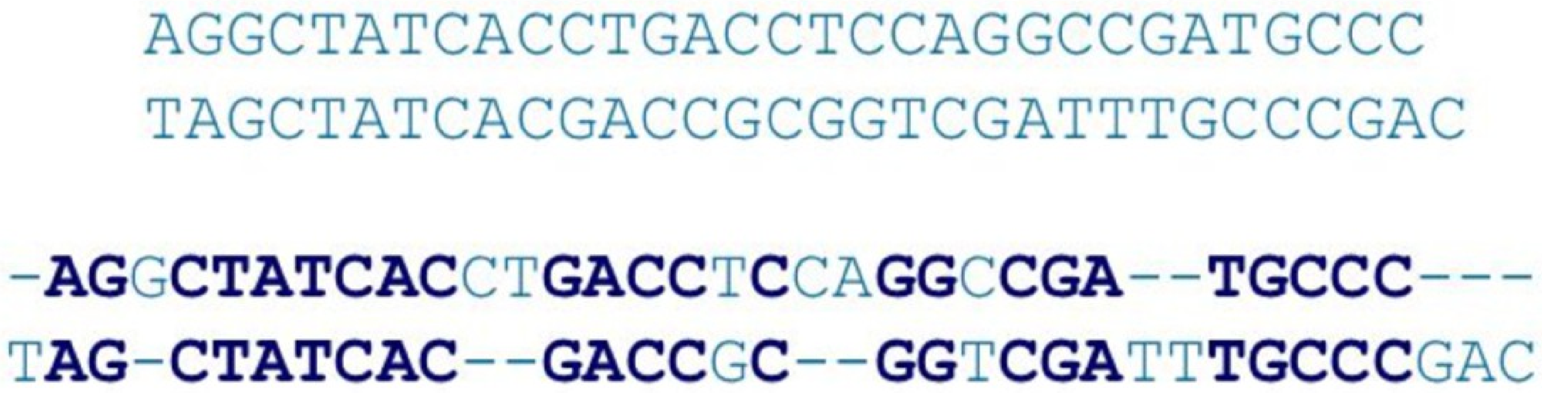


-	-	A	T	C	G	A	C
-	0	-4	-8	-12	-16	-20	-24
C	-4	-3	-7	-3	-7	-11	-15
A	-8	1	-3	-7	-6	-2	-6
T	-12	-3	6	2	-2	-6	-5
A	-16	-7	2	3	-1	3	-1
C	-20	-11	-2	-1	0	-1	8

- Create N x M matrix and place each sequence along one axis
- Place score 0 at the up-left corner
- Fill in 1st row & column with gap penalty multiples
- Fill in the matrix with max value of 3 possible moves:
  - Vertical move: Score + gap penalty
  - Horizontal move: Score + gap penalty
  - Diagonal move: Score + match/mismatch score
- To reconstruct the optimal alignment, trace back where the max at each step came from, stop when hit the origin.

Terms

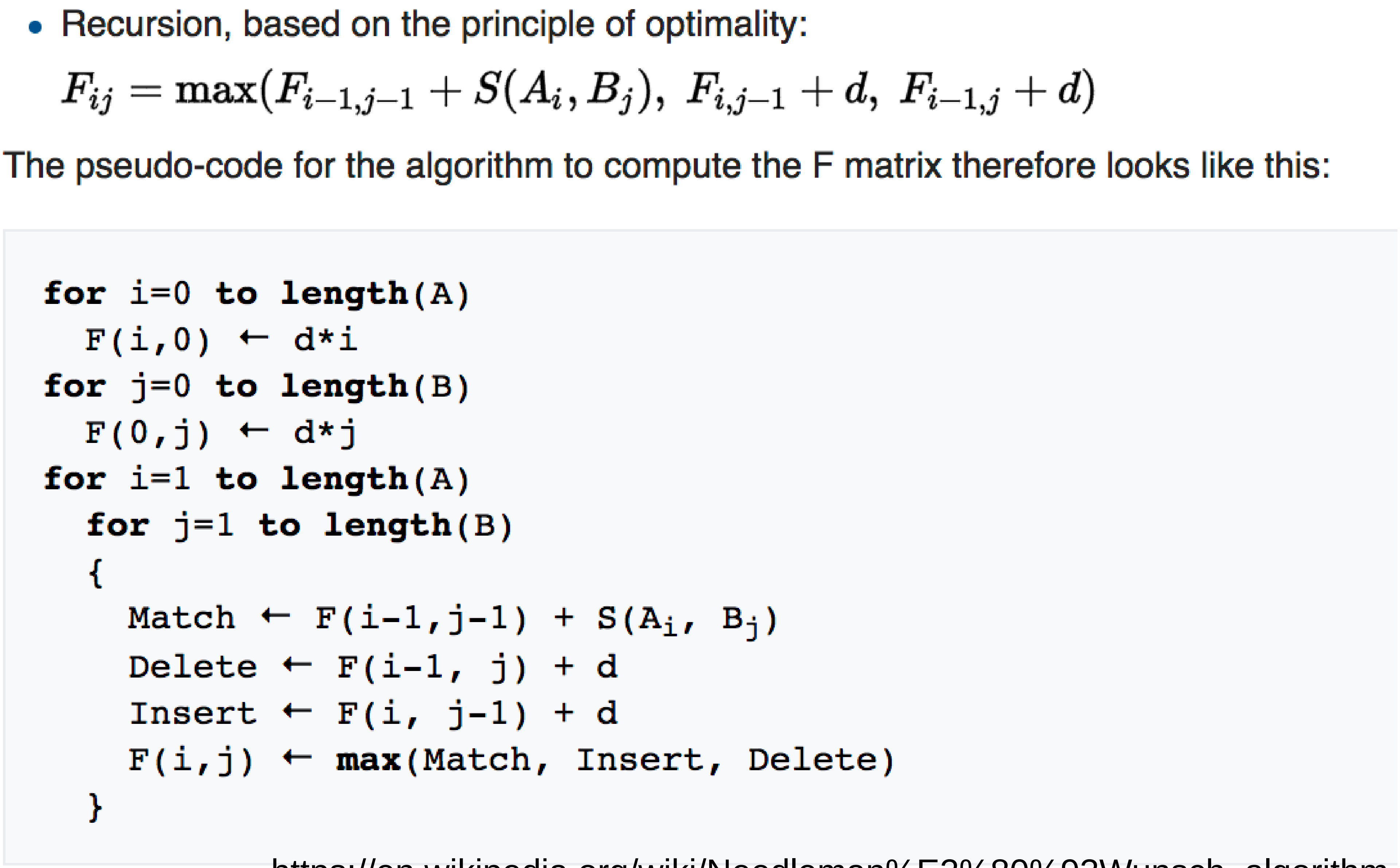
- Alignment is divided up unto sub problems
- Solutions are scored; the best solutions for char by char comparison are kept in the overall solution.
- **Match** – bases of each sequence at position ARE same
- **Mismatch** – bases of each sequence at position are NOT same
- **Gap** – bases are not the same, some insert or deletion may have occurred.



```
AGGCTATCACCTGACCTCCAGGCCGATGCCC
TAGCTATCACGACCGCGGTCGATTTGCCCGAC

-AGGCTATCACCTGACCTCCAGGCCGA--TGCCC---
TAG-CTATCAC--GACCGC--GGTCGATTTGCCCGAC
```

For each Element: Three Calculations



- Recursion, based on the principle of optimality:

$$F_{ij} = \max(F_{i-1,j-1} + S(A_i, B_j),\ F_{i,j-1} + d,\ F_{i-1,j} + d)$$

The pseudo-code for the algorithm to compute the F matrix therefore looks like this:

```
for i=0 to length(A)
  F(i,0) ← d*i
for j=0 to length(B)
  F(0,j) ← d*j
for i=1 to length(A)
  for j=1 to length(B)
  {
    Match ← F(i-1,j-1) + S($A_i$, $B_j$)
    Delete ← F(i-1, j) + d
    Insert ← F(i, j-1) + d
    F(i,j) ← max(Match, Insert, Delete)
  }
```

https://en.wikipedia.org/wiki/Needleman%E2%80%93Wunsch_algorithm

Let’s Calculate!

	_	A	T	C	G
_					
T					
C					
A					

- Gap: -1 ← ↑

- Match: 1
- Mismatch: 0

↖

Add The Outer Values

	_	A	T	C	G
_	0	-1	-2	-3	-4
T	-1				
C	-2				
A	-3				

- Gap: -1 ← ↑

- Match: 1
- Mismatch: 0

↖

Mismatch: A != T

	_	A	T	C	G
_	0	-1	-2	-3	-4
T	-1				
C	-2				
A	-3				

- Gap: -1 ← ↑

- Match: 1
- Mismatch: 0

↖

Upperbox: -1 – 1
= -2

Sidebox: -1 – 1
= -2

Diag: 0 – 0
= 0

Mismatch: A != T

	_	**A**	T	C	G
_	0	-1	-2	-3	-4
T	-1	**0**			
C	-2				
A	-3				

- Gap: -1

- Match: 1
- Mismatch: 0

Upperbox: -1 – 1
= -2

Sidebox: -1 – 1
= -2

Diag: 0 – 0
= 0

Max value

Match: T = T

	_	A	T	C	G
_	0	-1	-2	-3	-4
T	-1	0	**0**		
C	-2				
A	-3				

- Gap: -1

- Match: 1
- Mismatch: 0

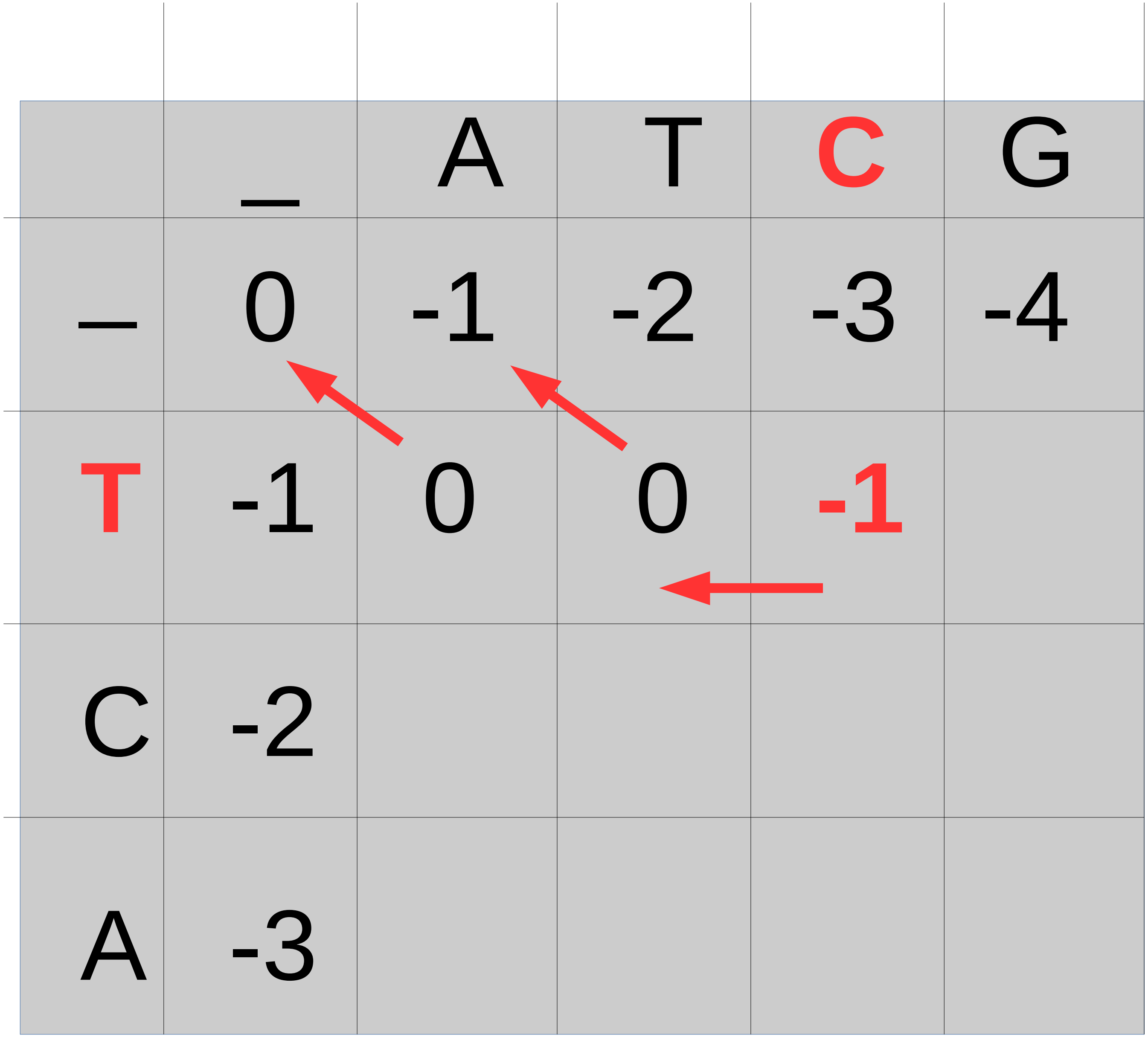
Upperbox: -2 - 1
= -3

Sidebox: 0 – 1
= -1

Diag: -1 + 1
= 0

Max value

Mismatch: T != C



	_	A	T	C	G
_	0	-1	-2	-3	-4
T	-1	0	0	-1	
C	-2				
A	-3				

- Gap: -1

- Match: 1
- Mismatch: 0

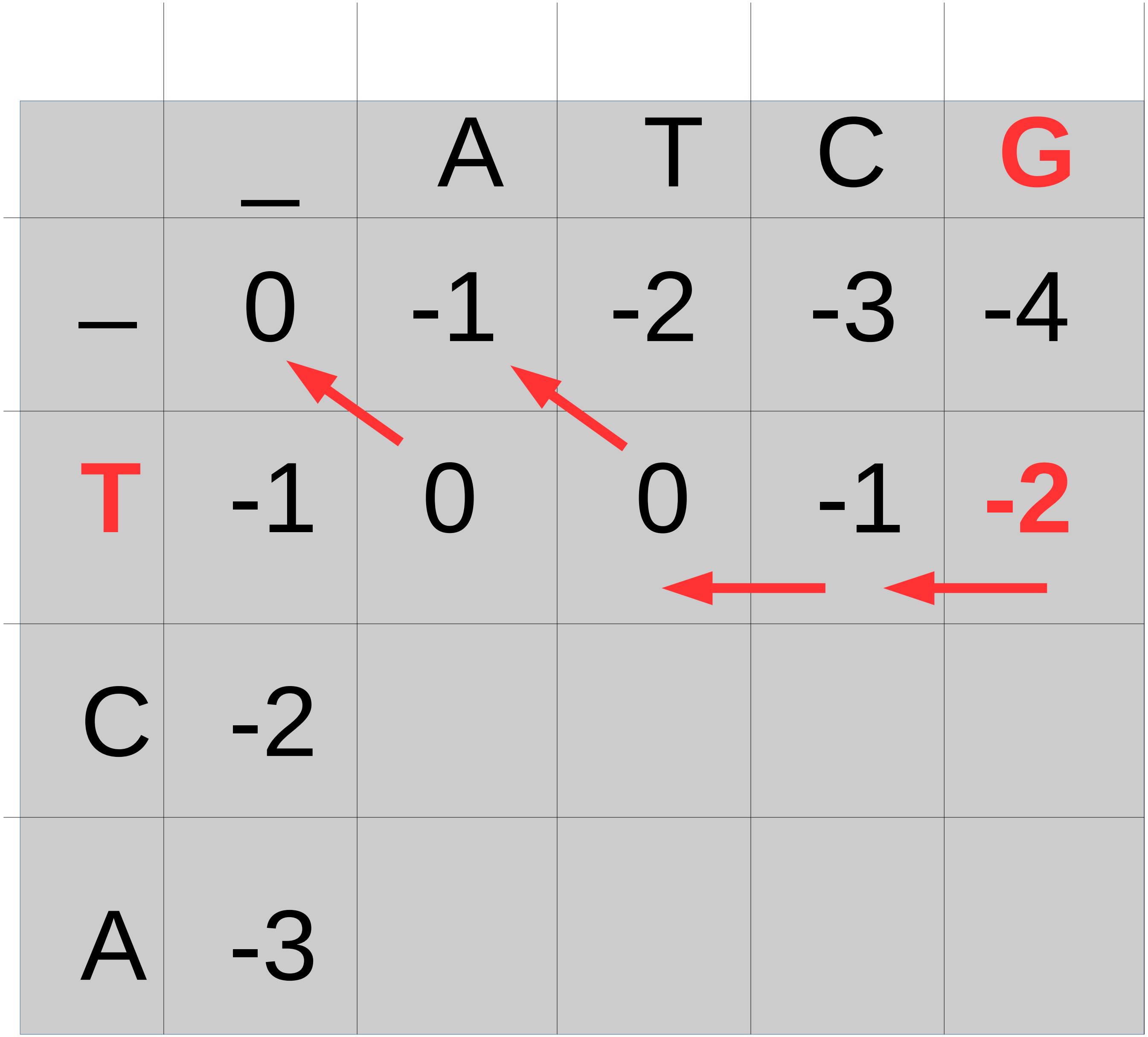
Upperbox: -3 – 1
= -4

Sidebox: 0 – 1
= -1

Max value

Diag: -2 + 0
= -2

Mismatch: T != G



	_	A	T	C	G
_	0	-1	-2	-3	-4
T	-1	0	0	-1	-2
C	-2				
A	-3				

- Gap: -1

- Match: 1
- Mismatch: 0

Upperbox: -4 – 1 = -5

Sidebox: -1 - 1 = -2 **Max value**

Diag: -3 - 0 = -3

Filling in The Rest of the Values

	_	A	T	C	G
_	0	-1	-2	-3	-4
T	-1	0	0	-1	-2
C	-2	-1	0	1	0
A	-3	-1	-1	0	1

- Gap: -1 ←↑

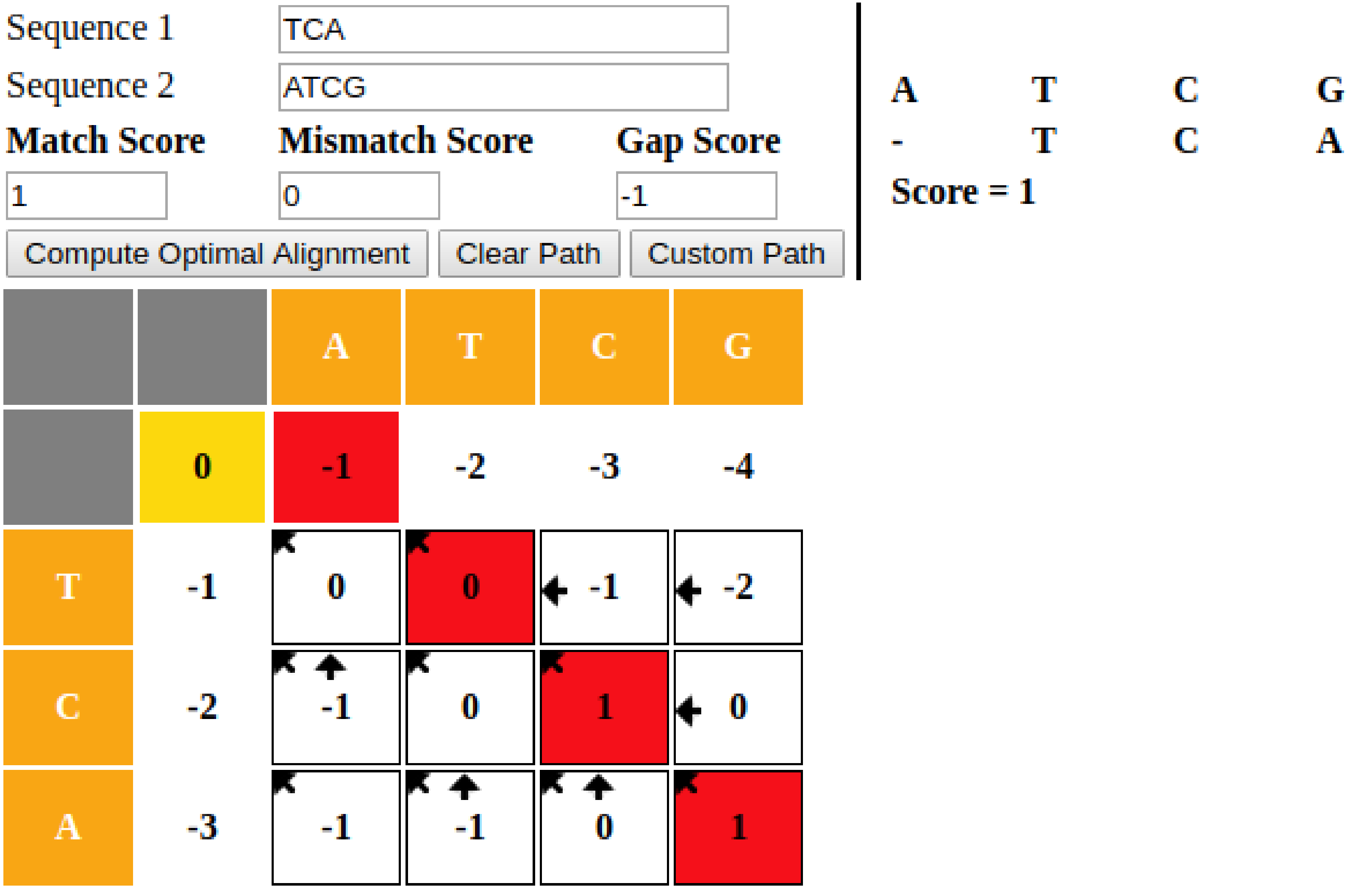
- Match: 1
- Mismatch: 0

↖

Alignment:

A T C G
_ T C A

Follow the Arrows Back To Find the Sequence Alignment



Sequence 1: TCA

Sequence 2: ATCG

Match Score: 1

Mismatch Score: 0

Gap Score: -1

		A	T	C	G
	0	-1	-2	-3	-4
T	-1	↖ 0	↖ 0	← -1	← -2
C	-2	↖ ↑ -1	↖ 0	↖ 1	← 0
A	-3	↖ -1	↖ ↑ -1	↖ ↑ 0	↖ 1

A T C G

- T C A

Score = 1

http://experiments.mostafa.io/public/needleman-wunsch/

To Get the Alignment

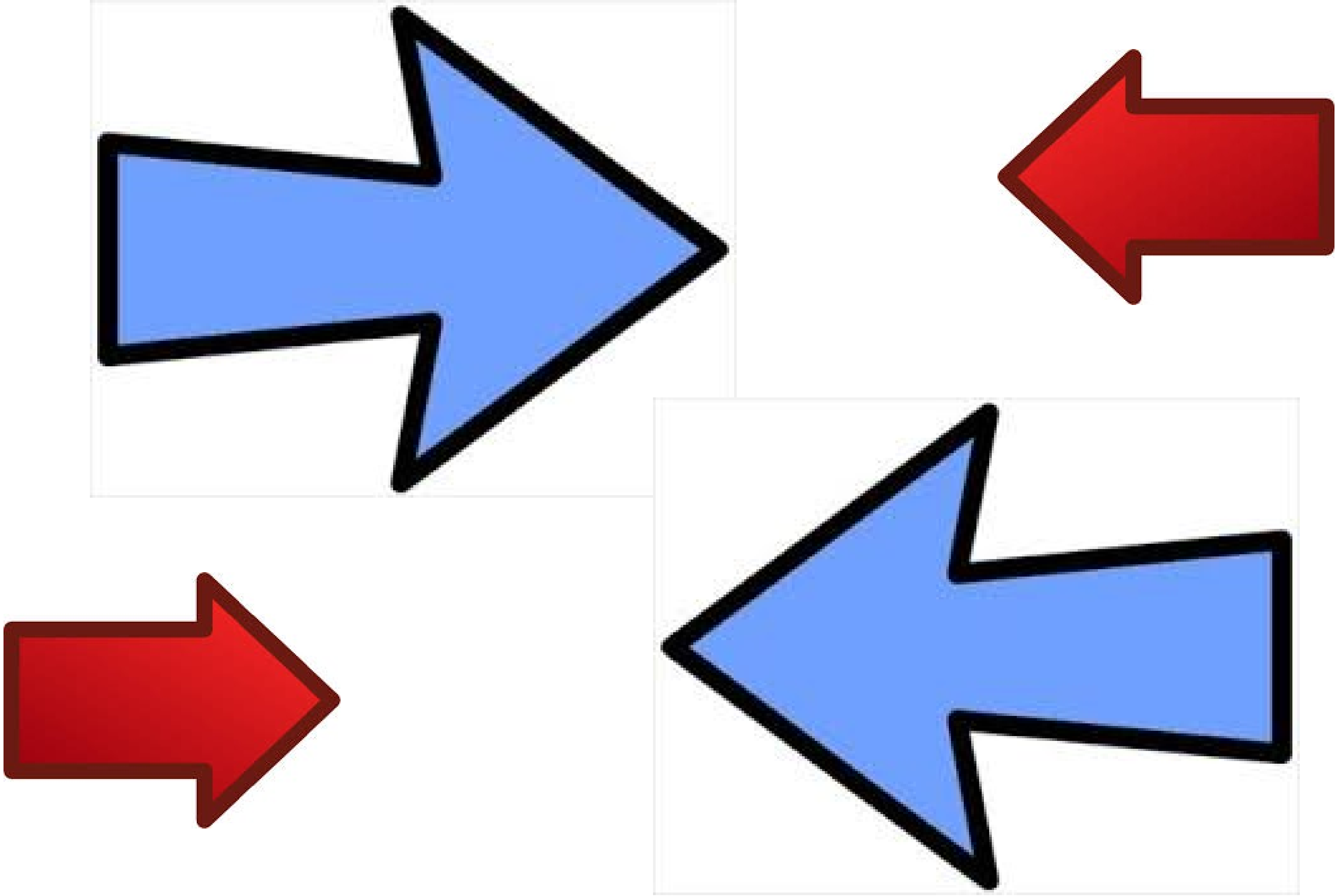
- With each calculation, we placed an arrow to show how the score was calculated and to give us the actual alignment.

How do we read this output?

Alignment:

```
A T C G
_ T C A
```

Up Ahead:
More Examples of Arrows



Example Alignment score = 0

Let:

Match = +1

Mismatch = 0

Gap = -1

		C	A	C	G	T	A	T
	0	-1	-2	-3	-4	-5	-6	-7
C	-1	1	0	-1	-2	-3	-4	-5
G	-2	0	1	0	0	-1	-2	-3
C	-3	-1	0	2	1	0	-1	-2
A	-4	-2	0	1	2	1	1	0

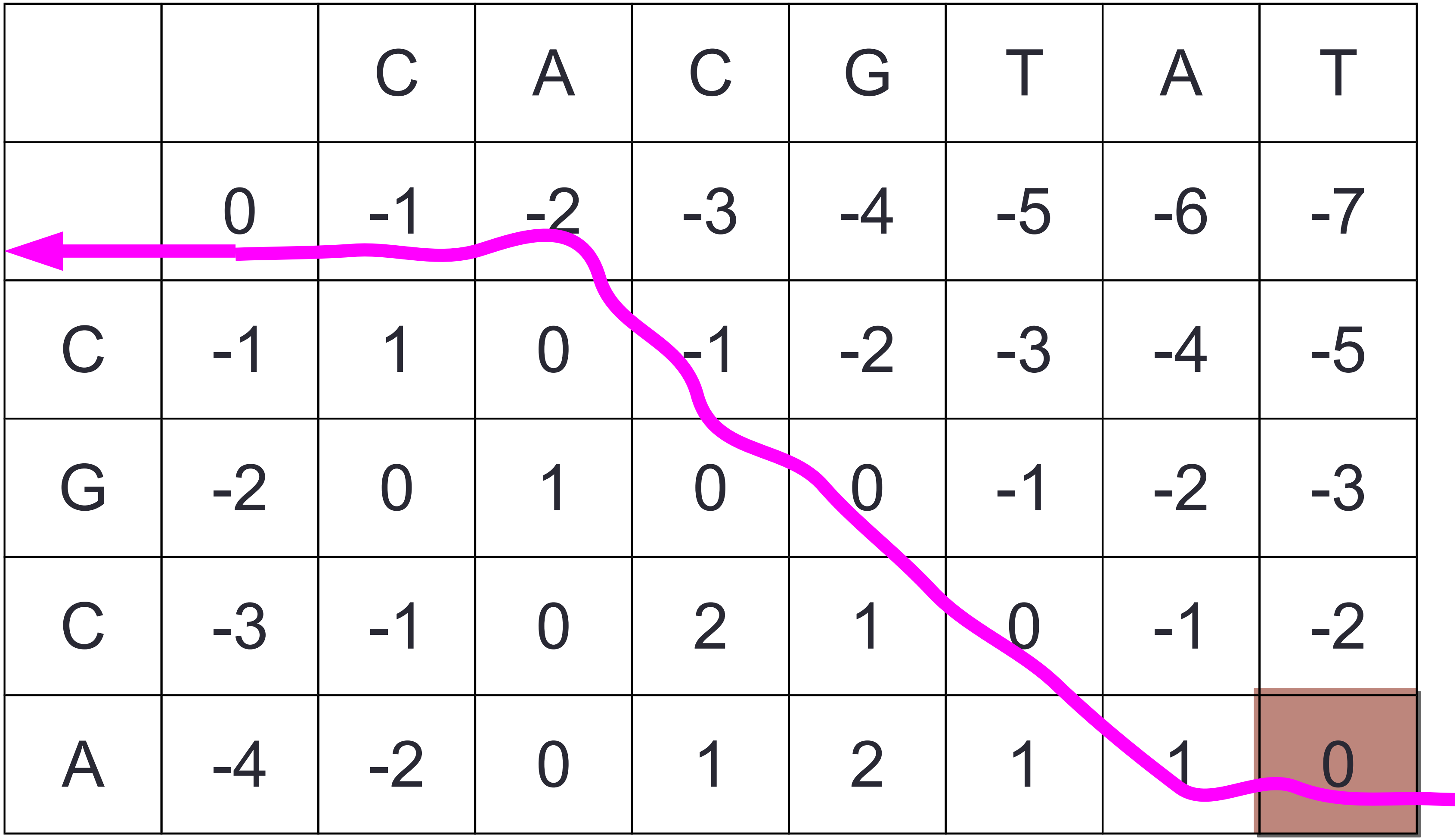
Example Alignment score = 0

Let:

Match = +1

Mismatch = 0

Gap = -1



		C	A	C	G	T	A	T
	0	-1	-2	-3	-4	-5	-6	-7
C	-1	1	0	-1	-2	-3	-4	-5
G	-2	0	1	0	0	-1	-2	-3
C	-3	-1	0	2	1	0	-1	-2
A	-4	-2	0	1	2	1	1	0

```
CACGTAT
--CGCA-
```

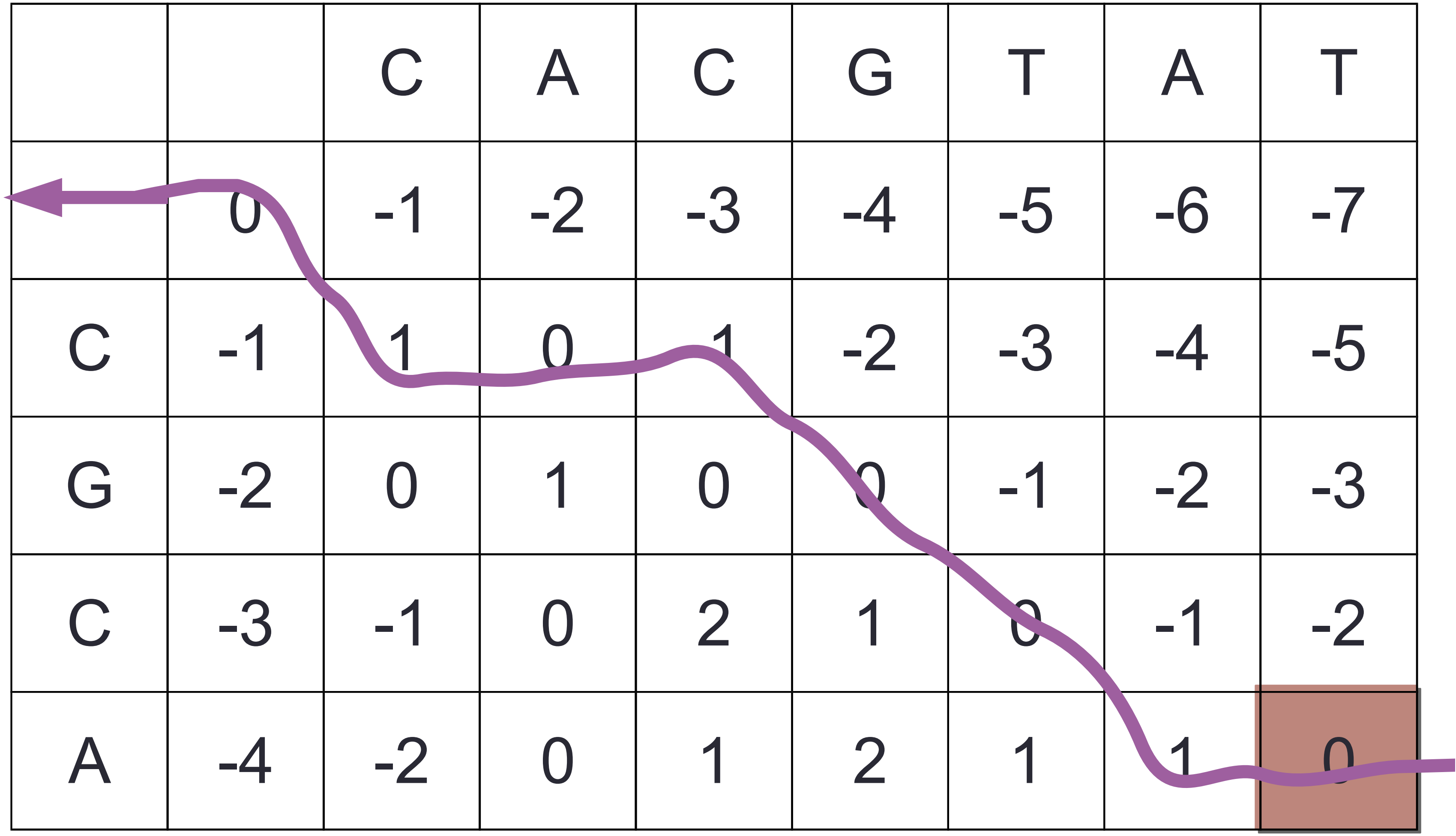
Example Alignment score = 0

Let:

Match = +1

Mismatch = 0

Gap = -1



		C	A	C	G	T	A	T
	0	-1	-2	-3	-4	-5	-6	-7
C	-1	1	0	1	-2	-3	-4	-5
G	-2	0	1	0	0	-1	-2	-3
C	-3	-1	0	2	1	0	-1	-2
A	-4	-2	0	1	2	1	1	0

```
CACGTAT
C--GCA-
```

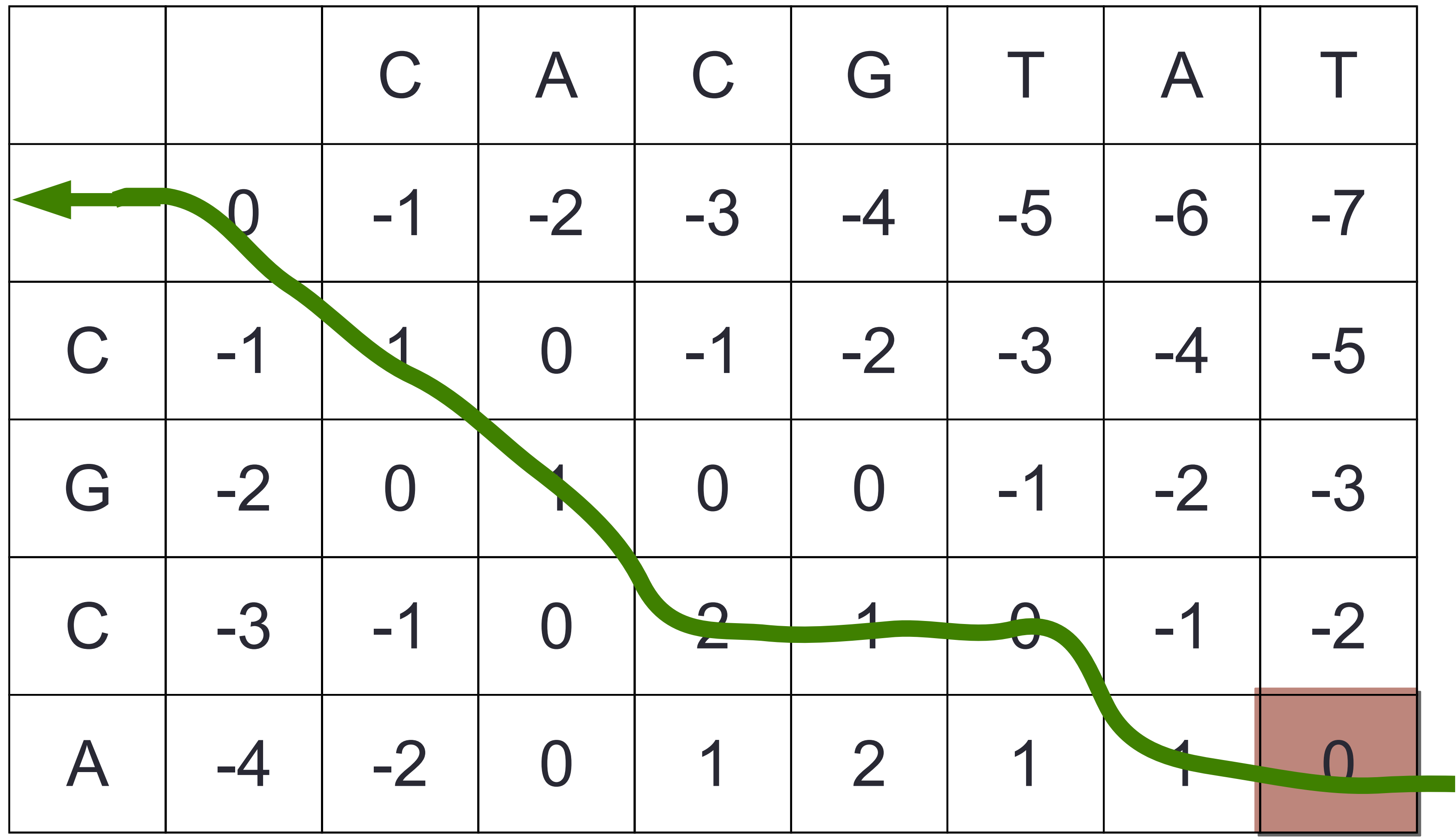
Example Alignment score = 0

Let:

Match = +1

Mismatch = 0

Gap = -1



		C	A	C	G	T	A	T
	0	-1	-2	-3	-4	-5	-6	-7
C	-1	1	0	-1	-2	-3	-4	-5
G	-2	0	1	0	0	-1	-2	-3
C	-3	-1	0	2	1	0	-1	-2
A	-4	-2	0	1	2	1	1	0

CACGTAT

CGC--A-

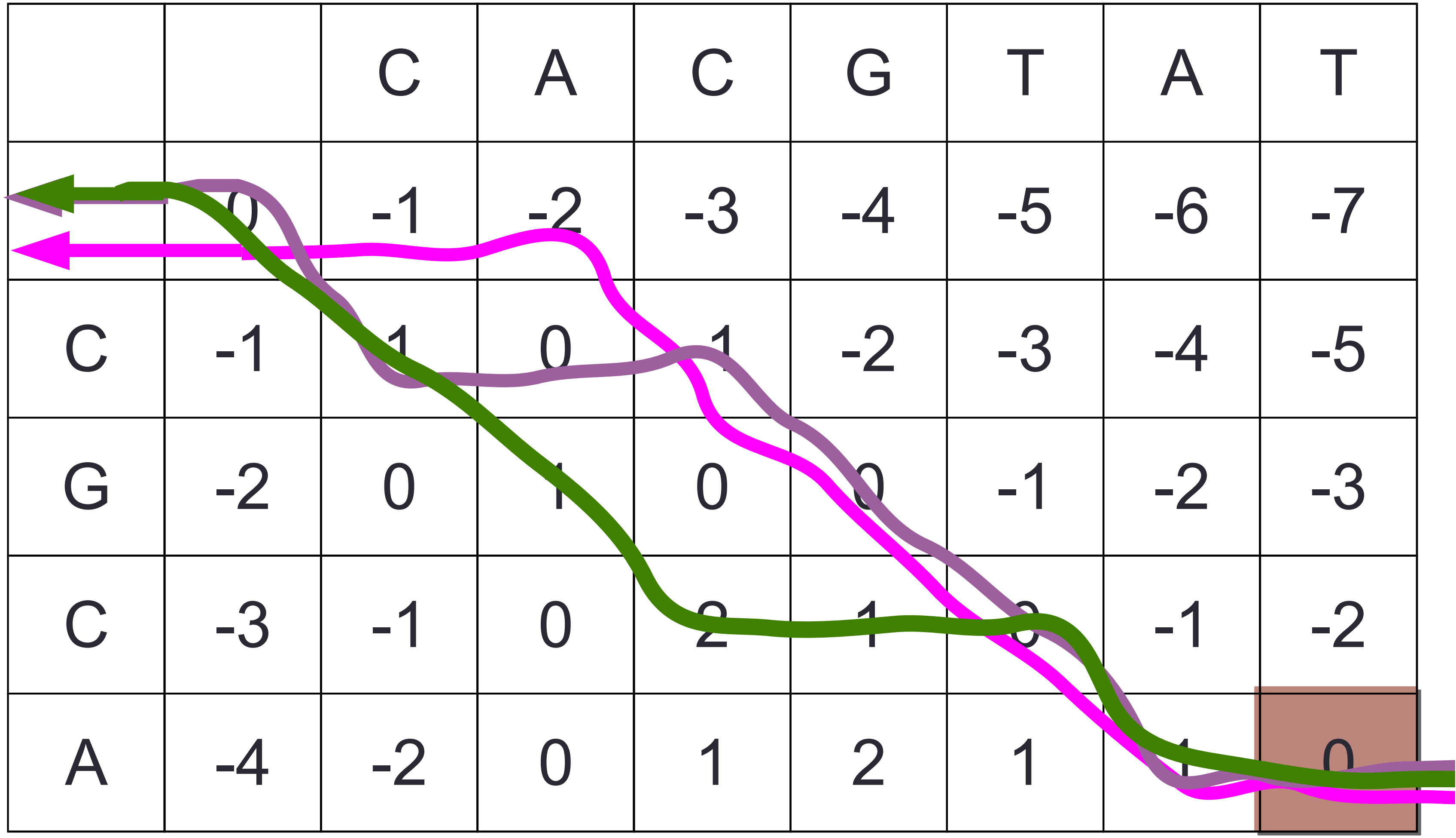
Example Alignment score = 0

Let:

Match = +1

Mismatch = 0

Gap = -1



		C	A	C	G	T	A	T
	0	-1	-2	-3	-4	-5	-6	-7
C	-1	1	0	1	-2	-3	-4	-5
G	-2	0	1	0	0	-1	-2	-3
C	-3	-1	0	2	1	0	-1	-2
A	-4	-2	0	1	2	1	1	0

```
CACGTAT
--CGCA-
```

```
CACGTAT
C--GCA-
```

```
CACGTAT
CGC--A-
```