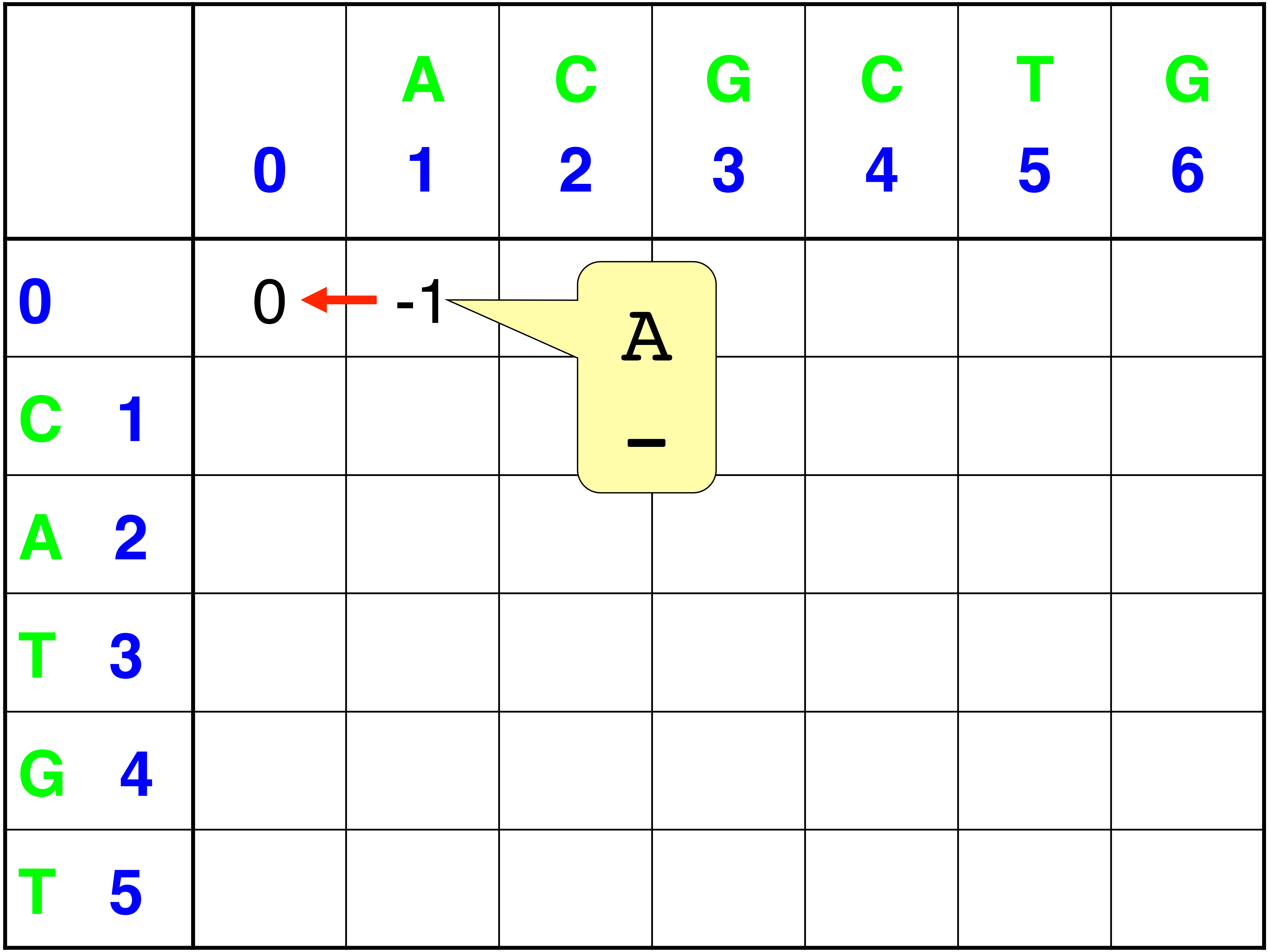
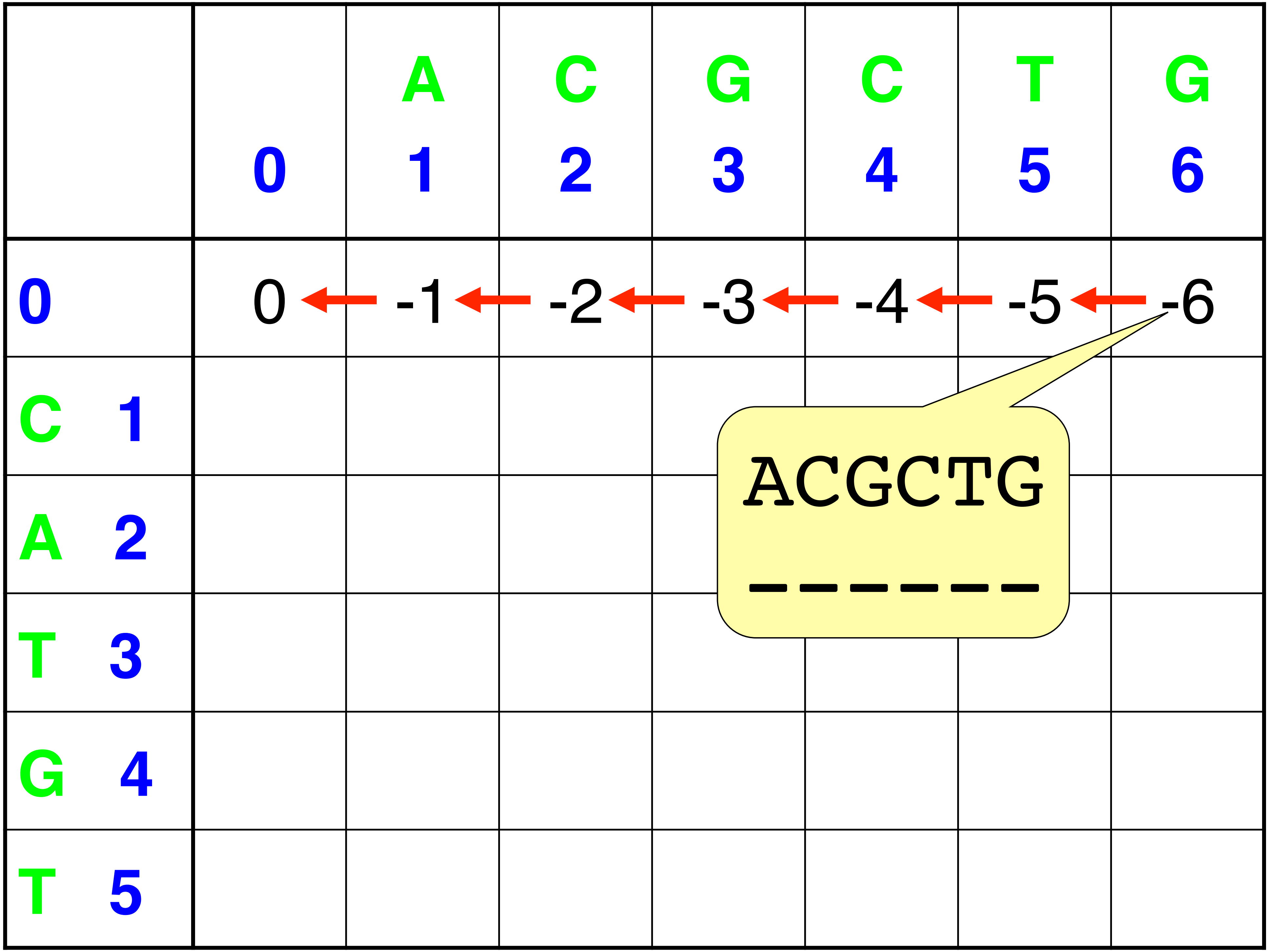
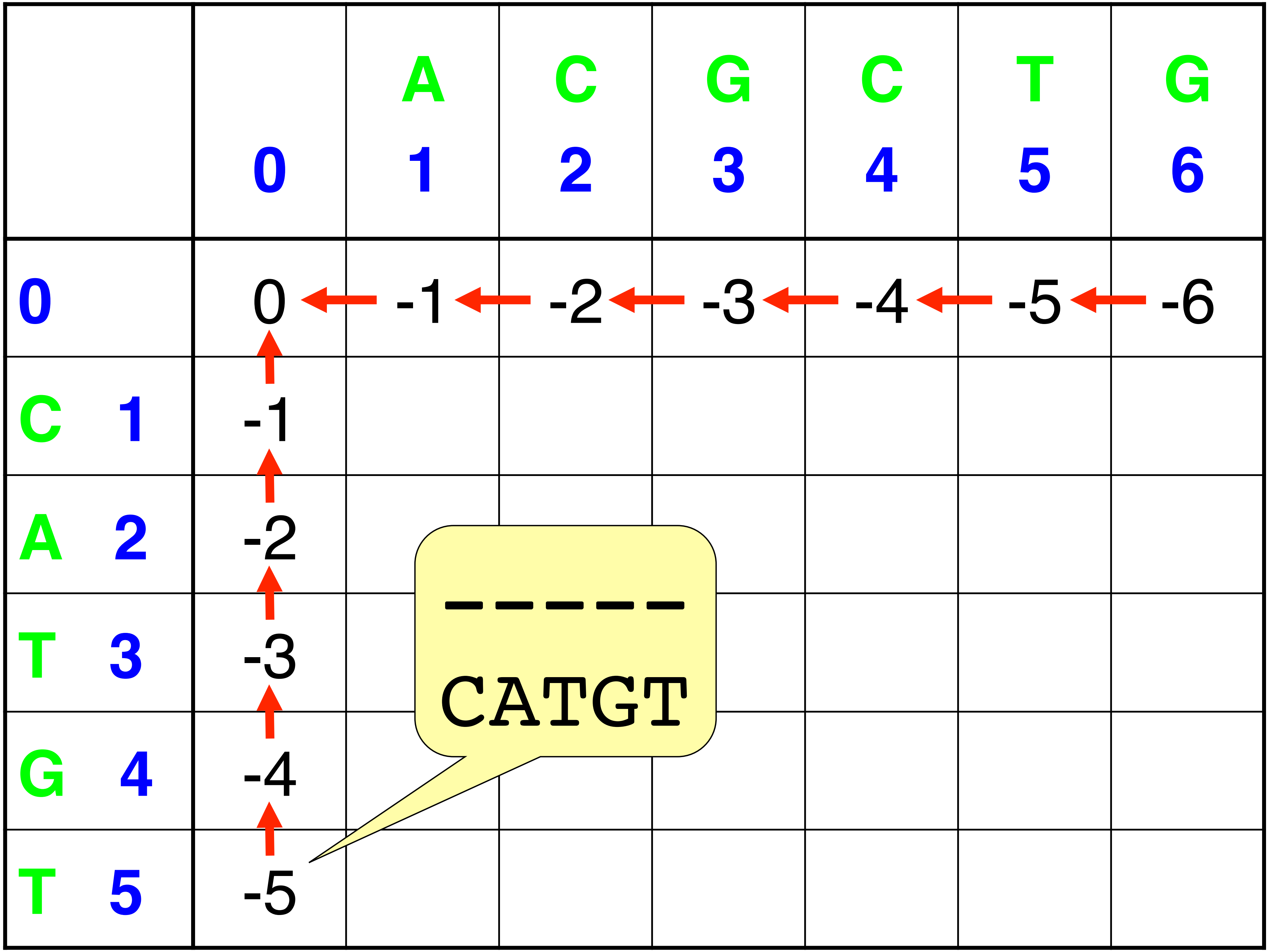
		A	C	G	C	T	G
	0	1	2	3	4	5	6
0	0						
C 1							
A 2							
T 3							
G 4							
T 5							



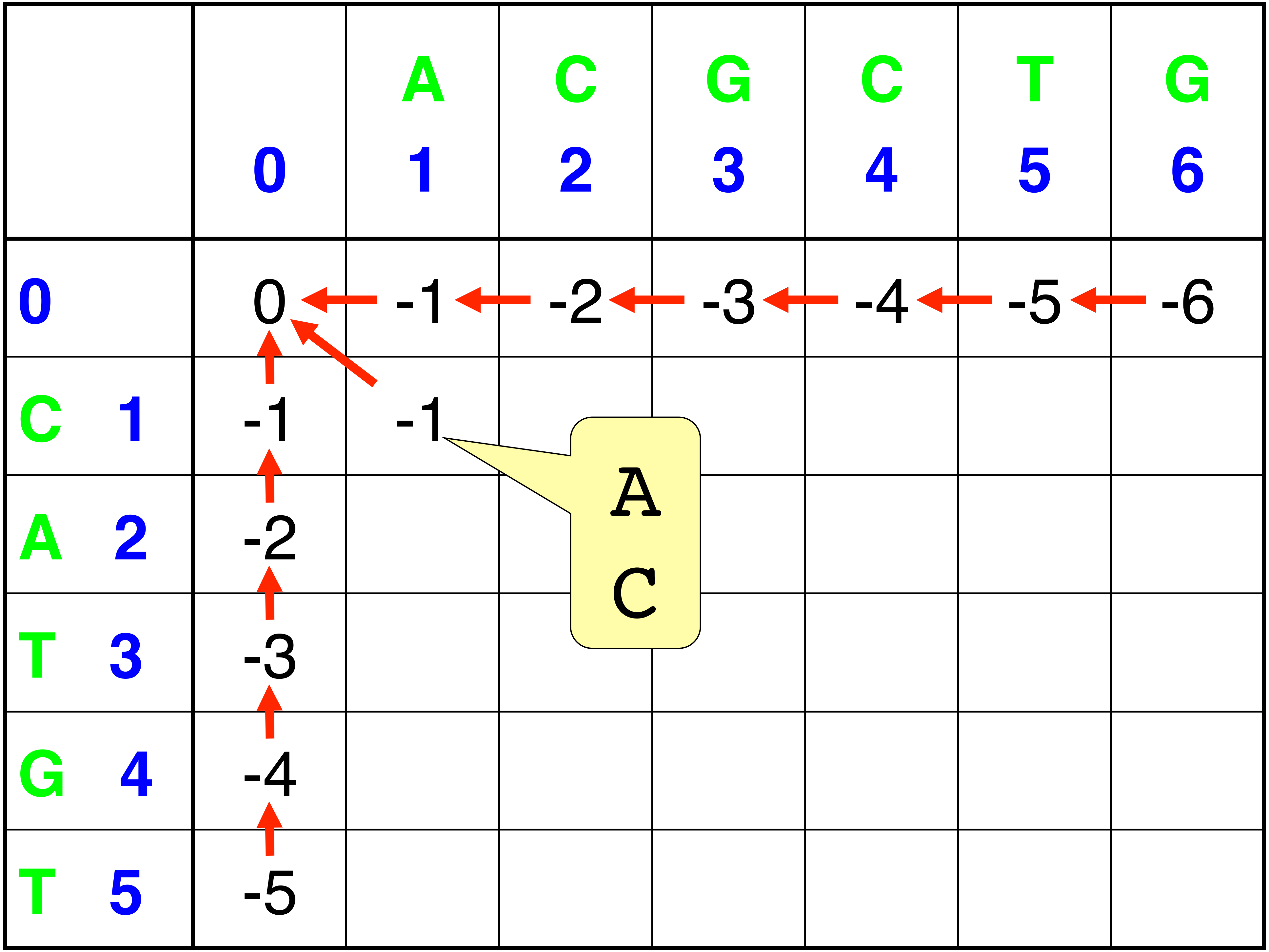
	0	A 1	C 2	G 3	C 4	T 5	G 6
0	0	← -1					
C 1							
A 2							
T 3							
G 4							
T 5							



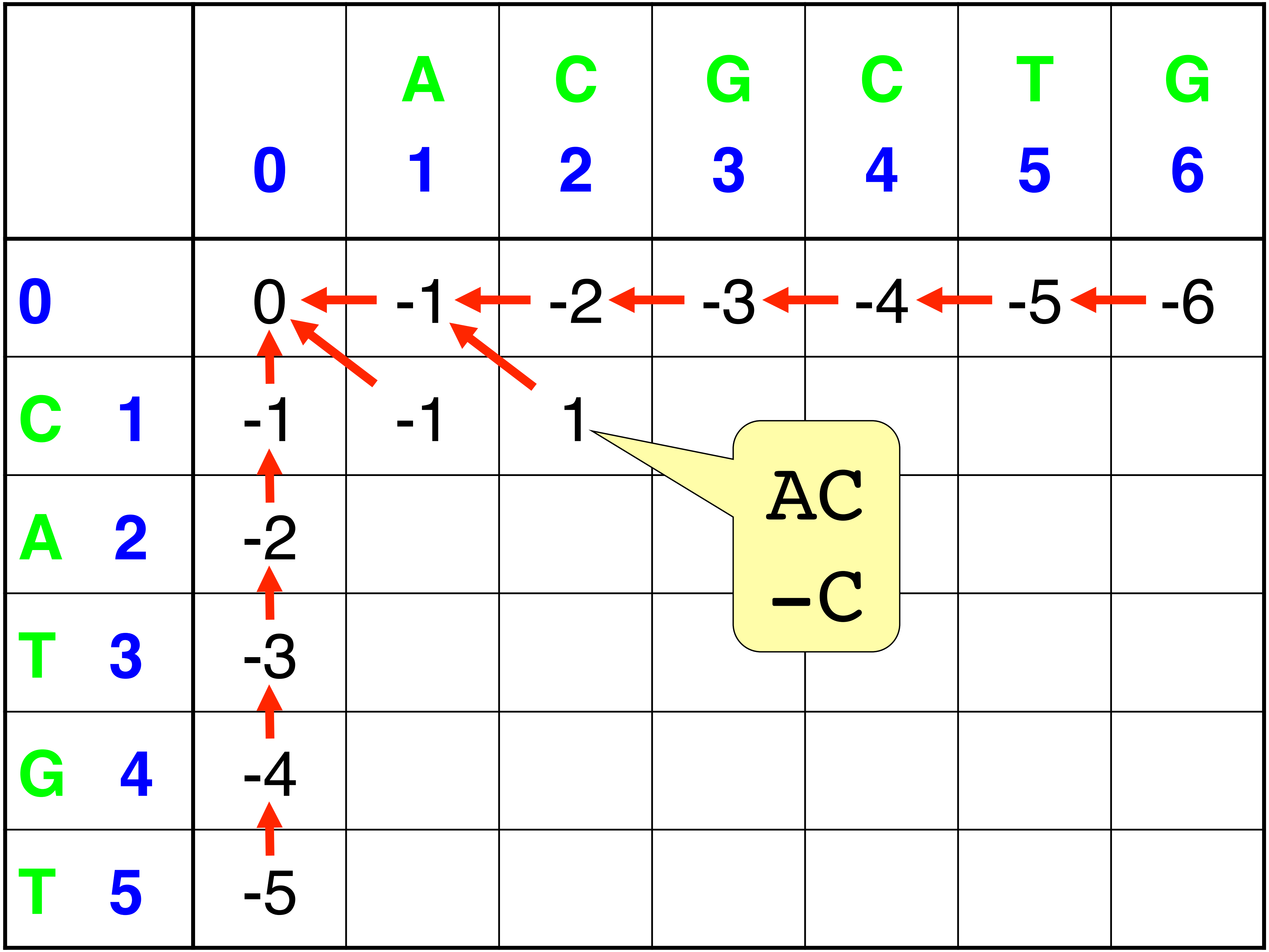


		A	C	G	C	T	G
	0	1	2	3	4	5	6
0	0	-1	-2	-3	-4	-5	-6
C 1	-1						
A 2	-2						
T 3	-3						
G 4	-4						
T 5	-5						

```
-----
CATGT
```

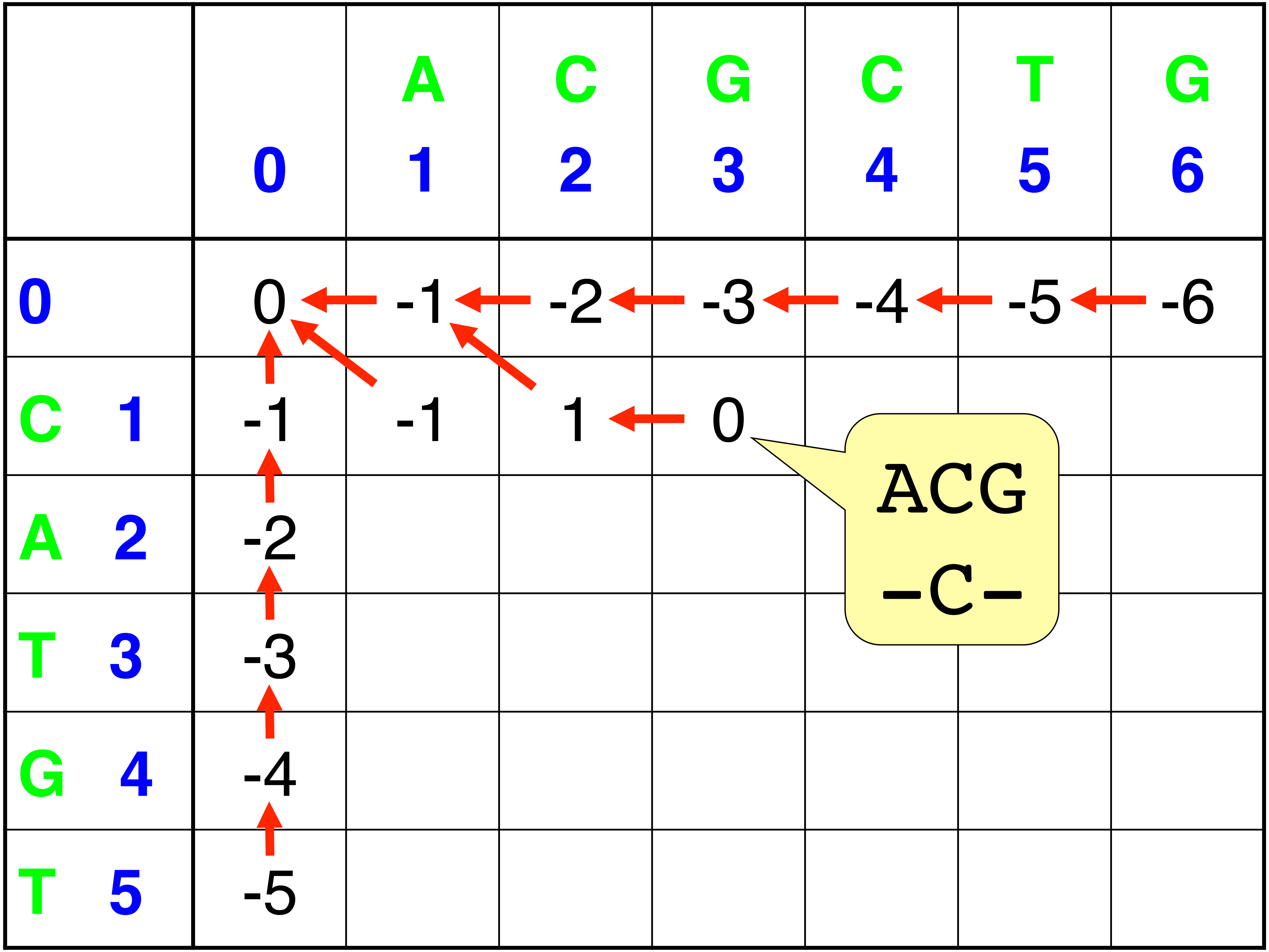


	0	A 1	C 2	G 3	C 4	T 5	G 6
0	0	-1	-2	-3	-4	-5	-6
C 1	-1	-1					
A 2	-2						
T 3	-3						
G 4	-4						
T 5	-5						



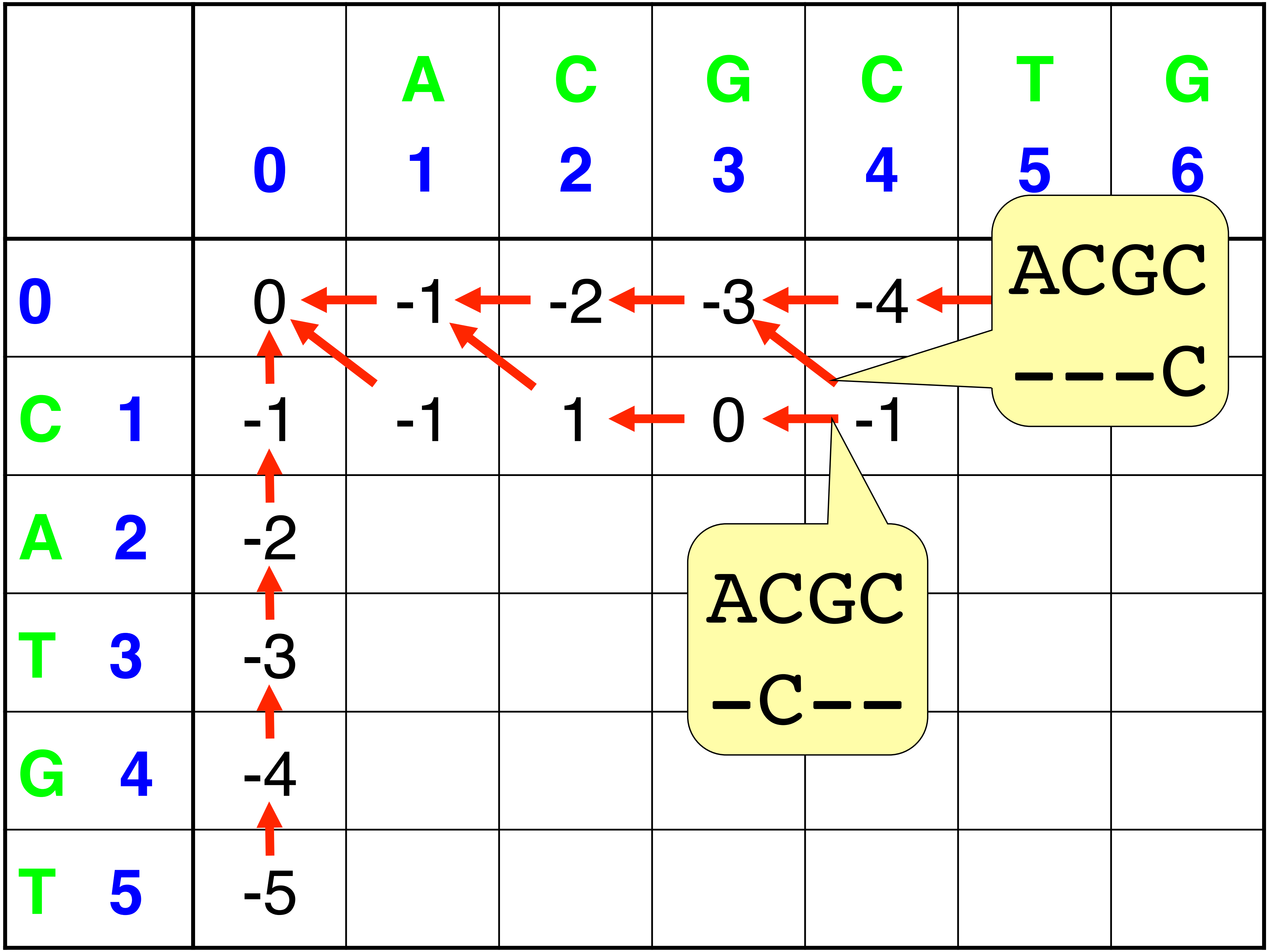
		A	C	G	C	T	G
	0	1	2	3	4	5	6
0	0	-1	-2	-3	-4	-5	-6
C 1	-1	-1	1				
A 2	-2						
T 3	-3						
G 4	-4						
T 5	-5						

AC
–C



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

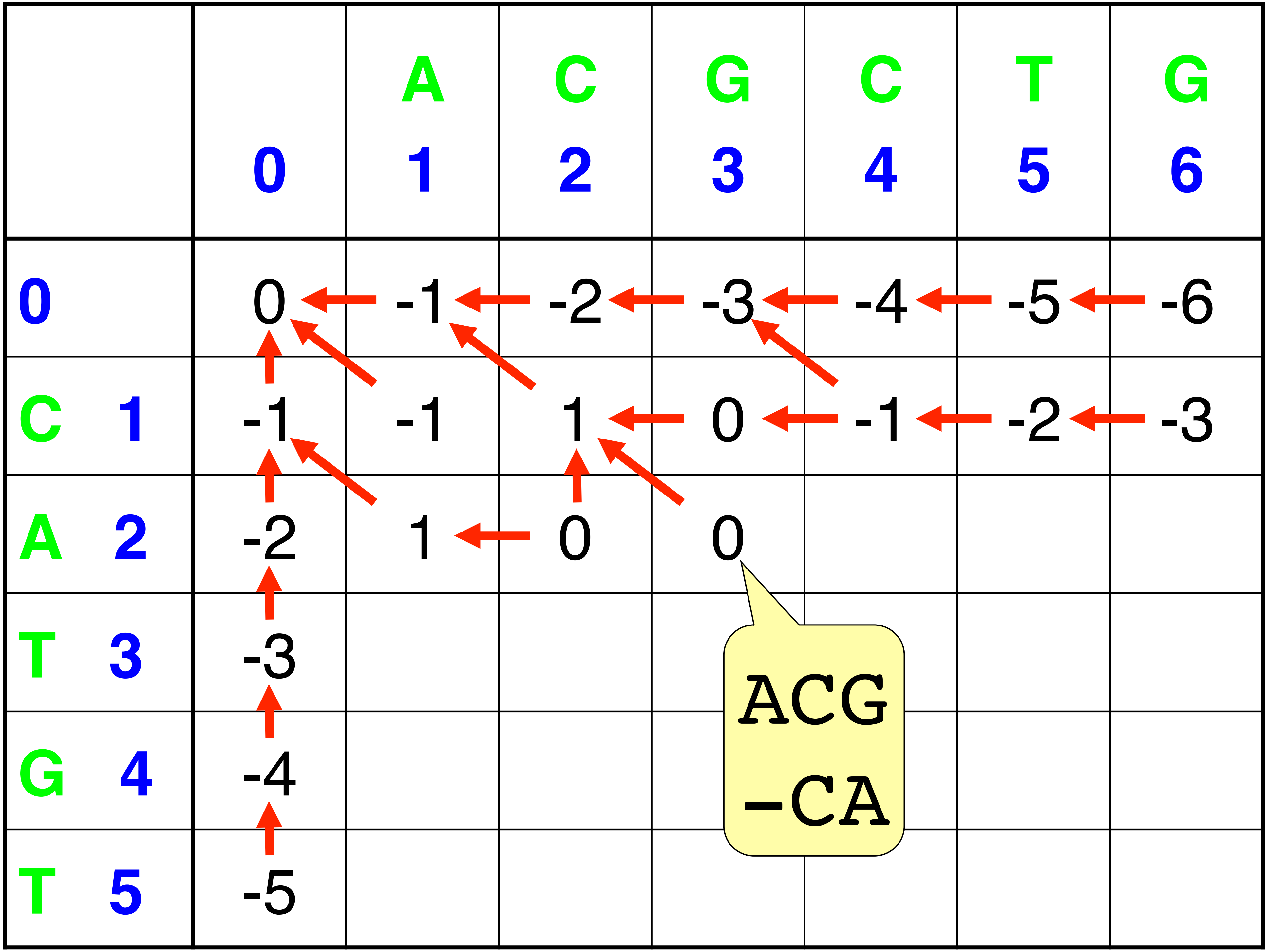
```
ACG
-C-
```



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

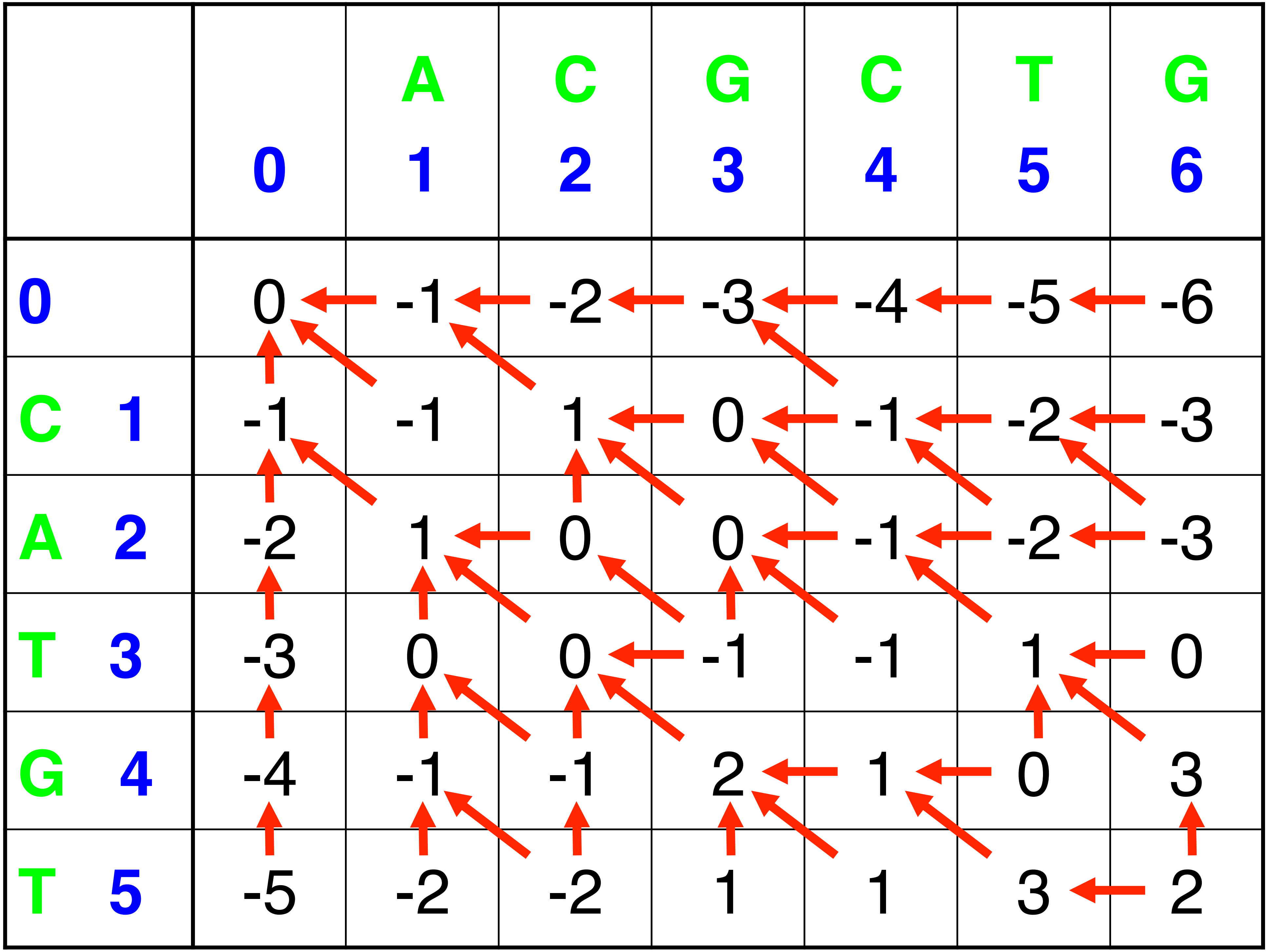
```
ACGC
---C
```

```
ACGC
-C--
```

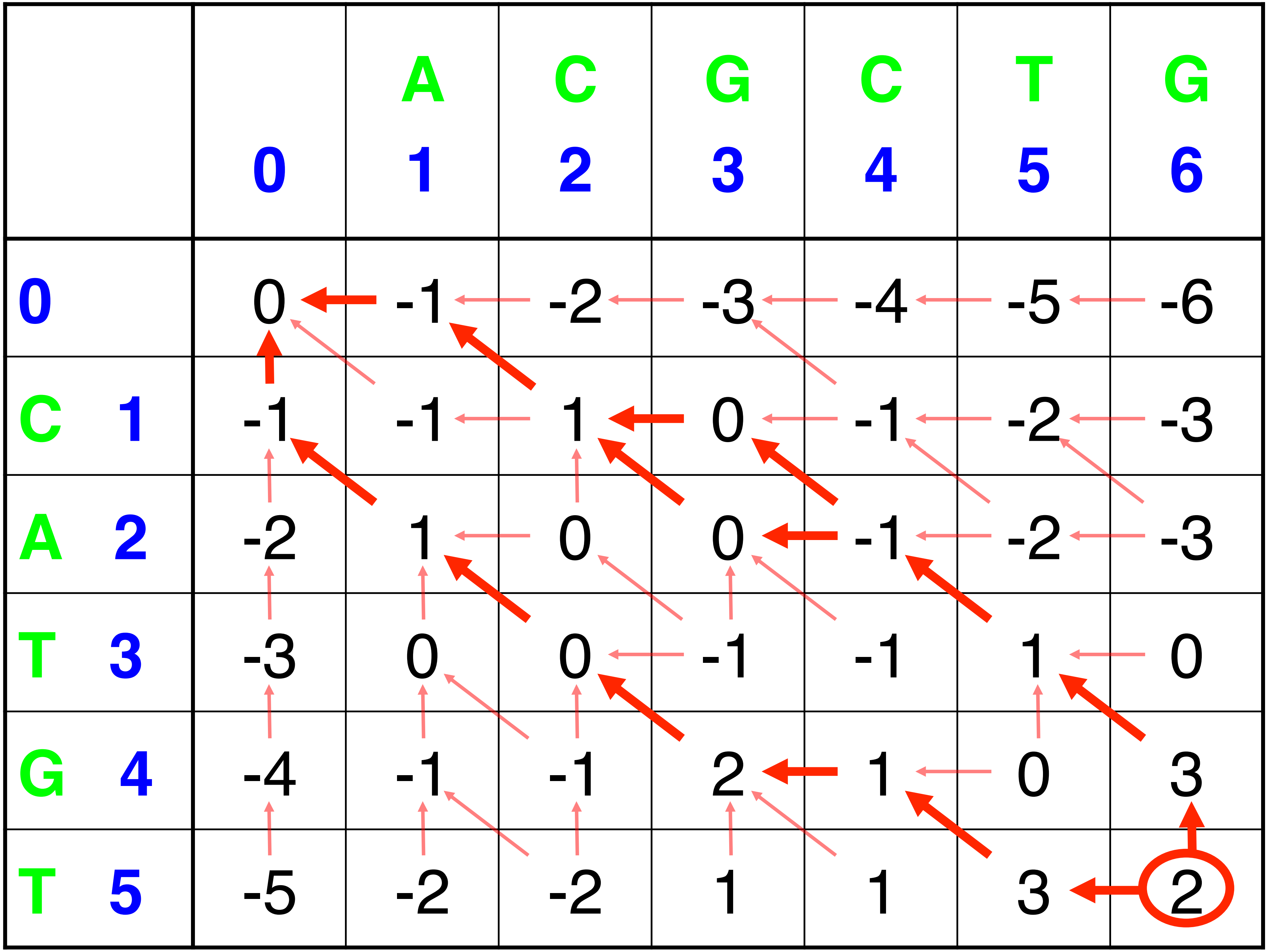


	0	A 1	C 2	G 3	C 4	T 5	G 6
0	0	-1	-2	-3	-4	-5	-6
C 1	-1	-1	1	0	-1	-2	-3
A 2	-2	1	0	0			
T 3	-3						
G 4	-4						
T 5	-5						

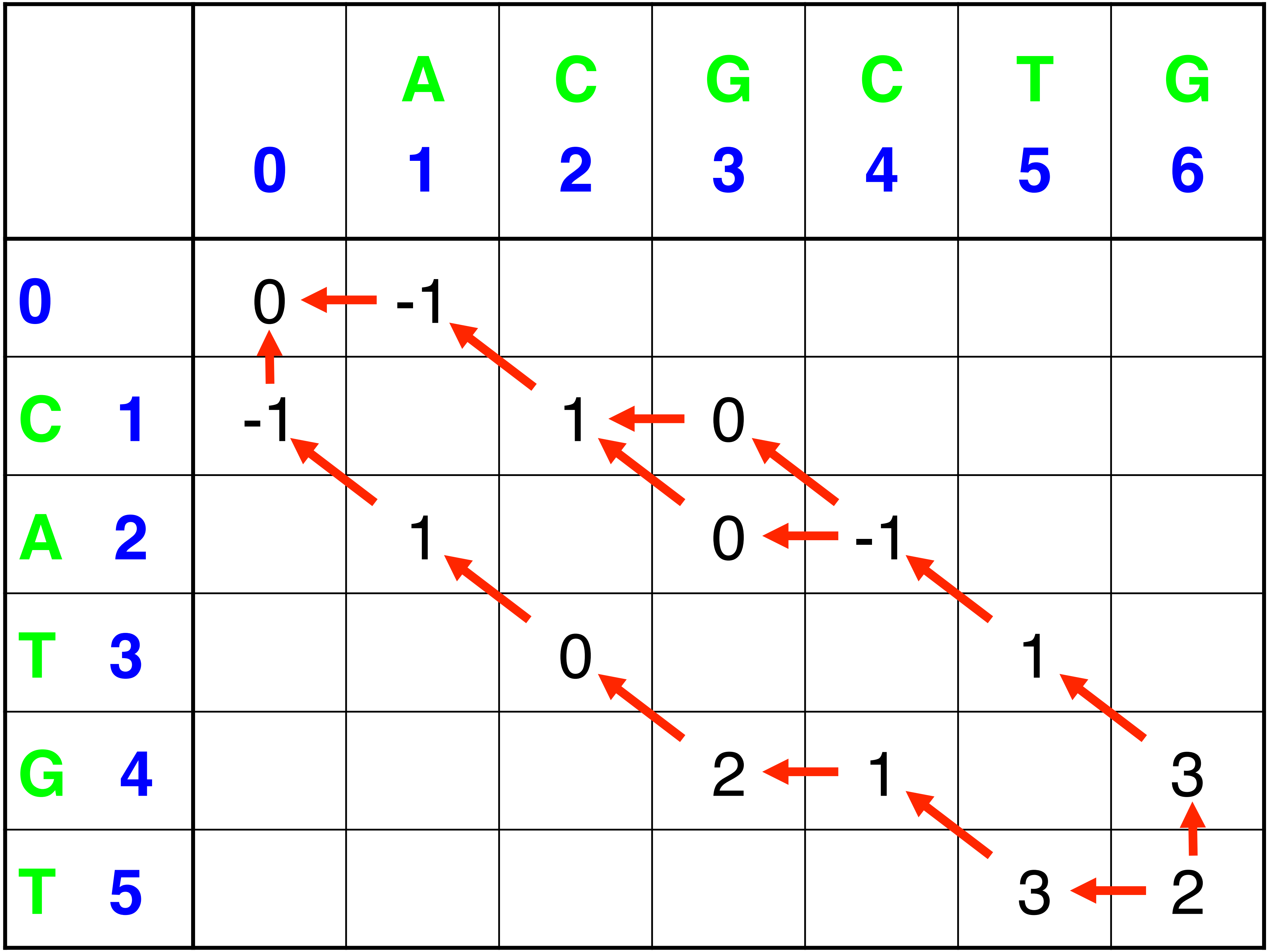
```
ACG
–CA
```



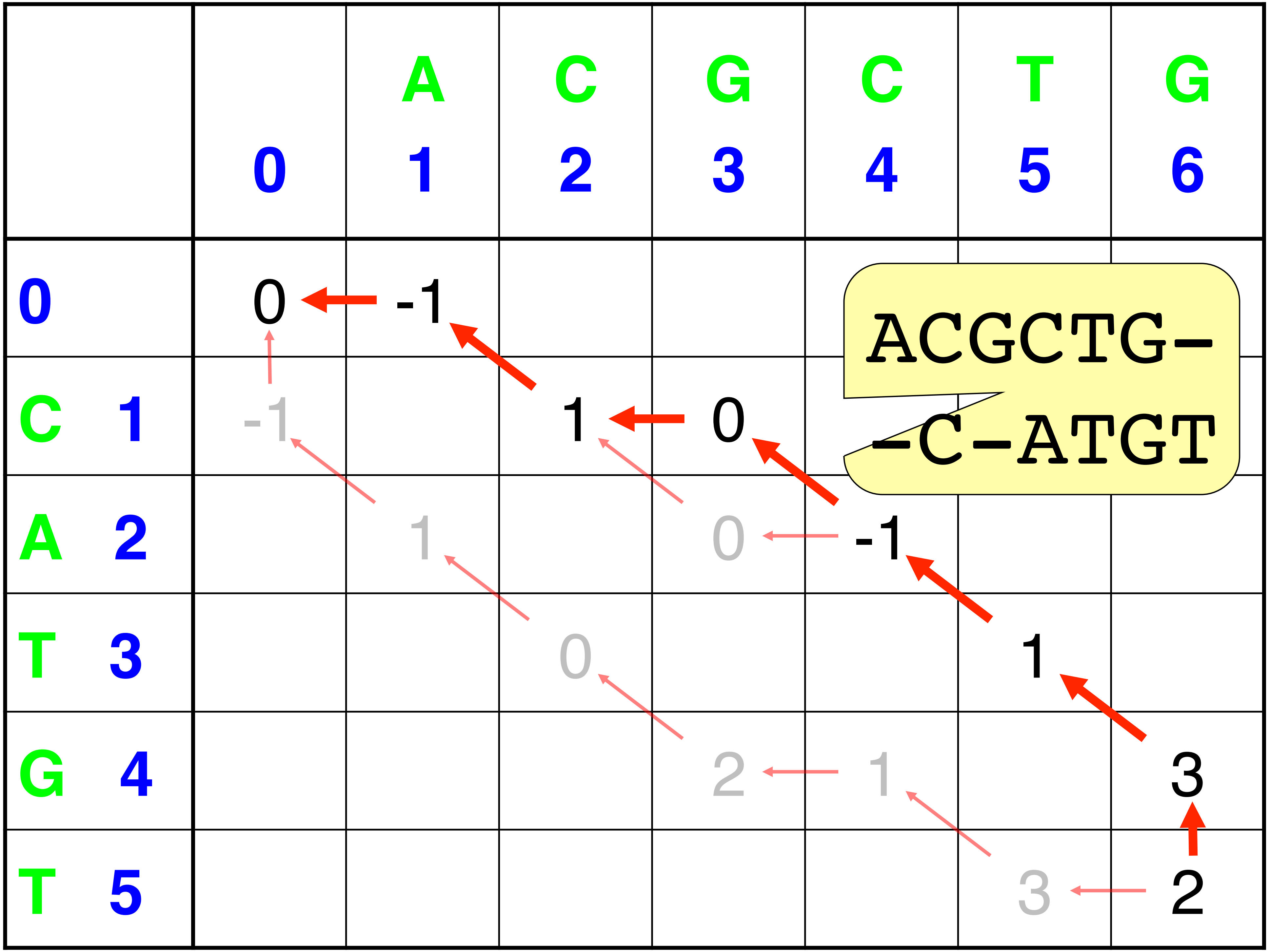
			A	C	G	C	T	G
		0	1	2	3	4	5	6
	0	0	-1	-2	-3	-4	-5	-6
C	1	-1	-1	1	0	-1	-2	-3
A	2	-2	1	0	0	-1	-2	-3
T	3	-3	0	0	-1	-1	1	0
G	4	-4	-1	-1	2	1	0	3
T	5	-5	-2	-2	1	1	3	2



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

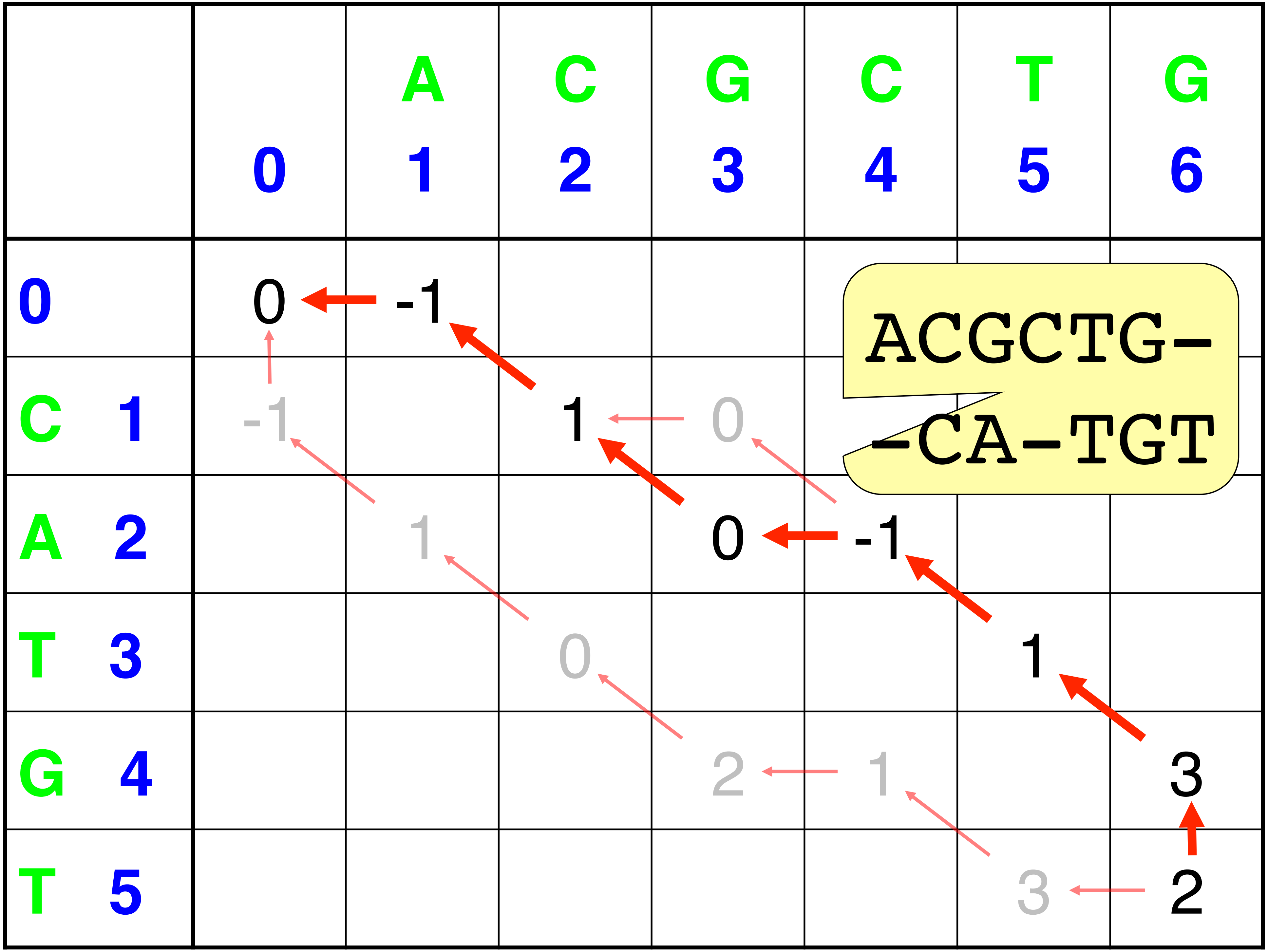


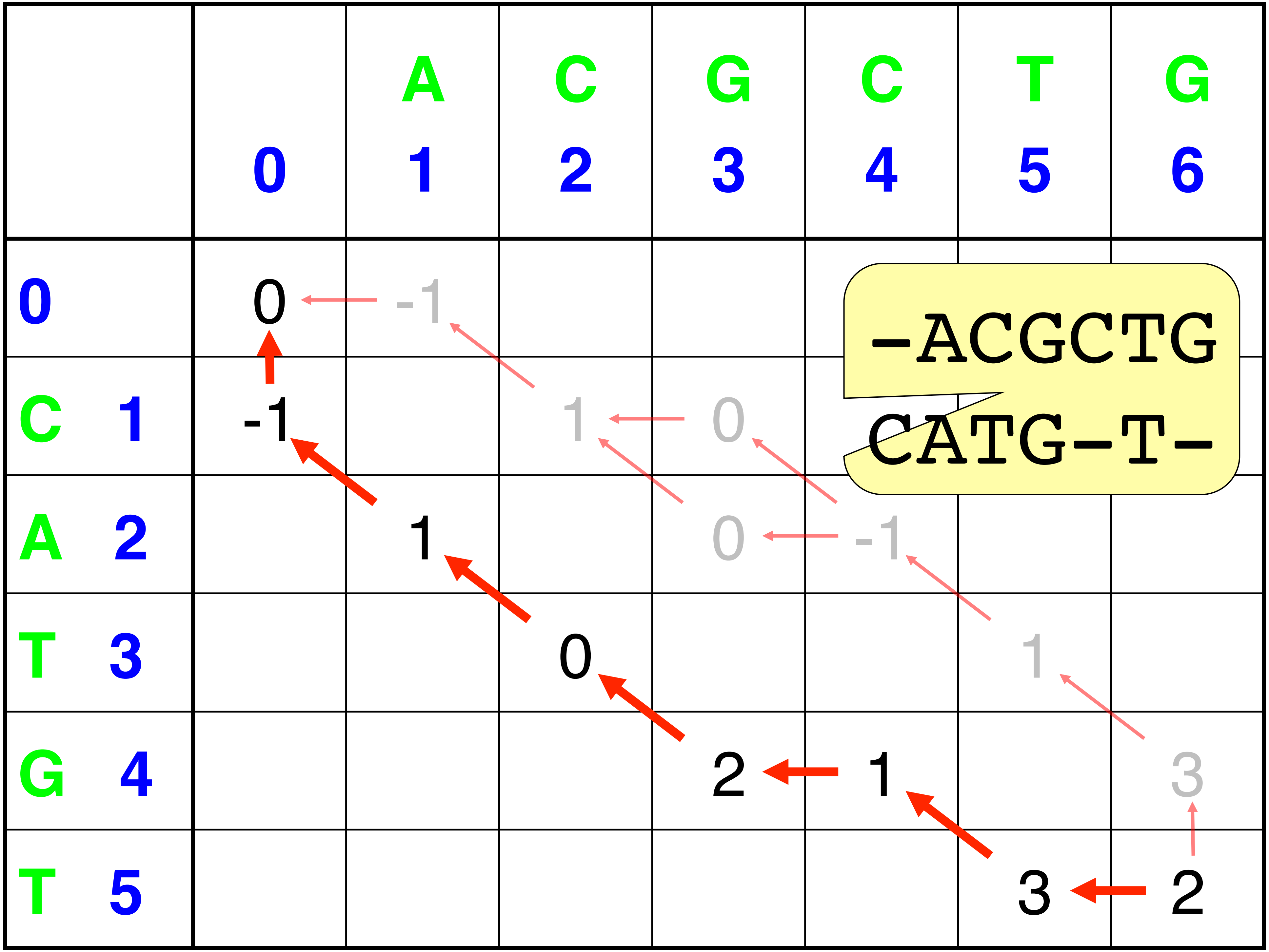
	0	A 1	C 2	G 3	C 4	T 5	G 6
0	0	-1					
C 1	-1		1	0			
A 2		1		0	-1		
T 3			0			1	
G 4				2	1		3
T 5						3	2



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

ACGCTG-
-C-ATGT





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

```
-ACGCTG
CATG-T-
```

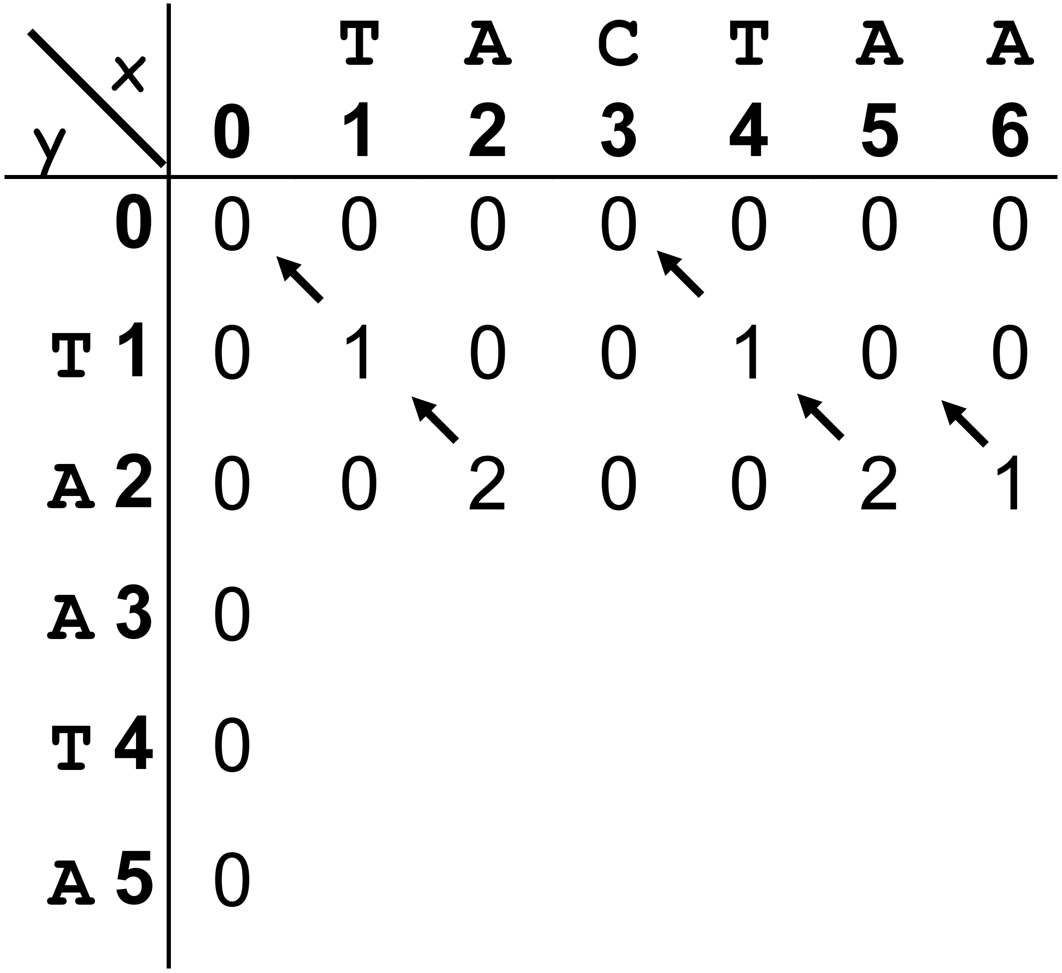
Local Alignment Example

y = TAATA
x = TACTAA

x y	**0**	**A** **1**	**T** **2**	**C** **3**	**T** **4**	**A** **5**	**A** **6**
0	0	0	0	0	0	0	0
T 1	0						
A 2	0						
A 3	0						
T 4	0						
A 5	0						

Local Alignment Example

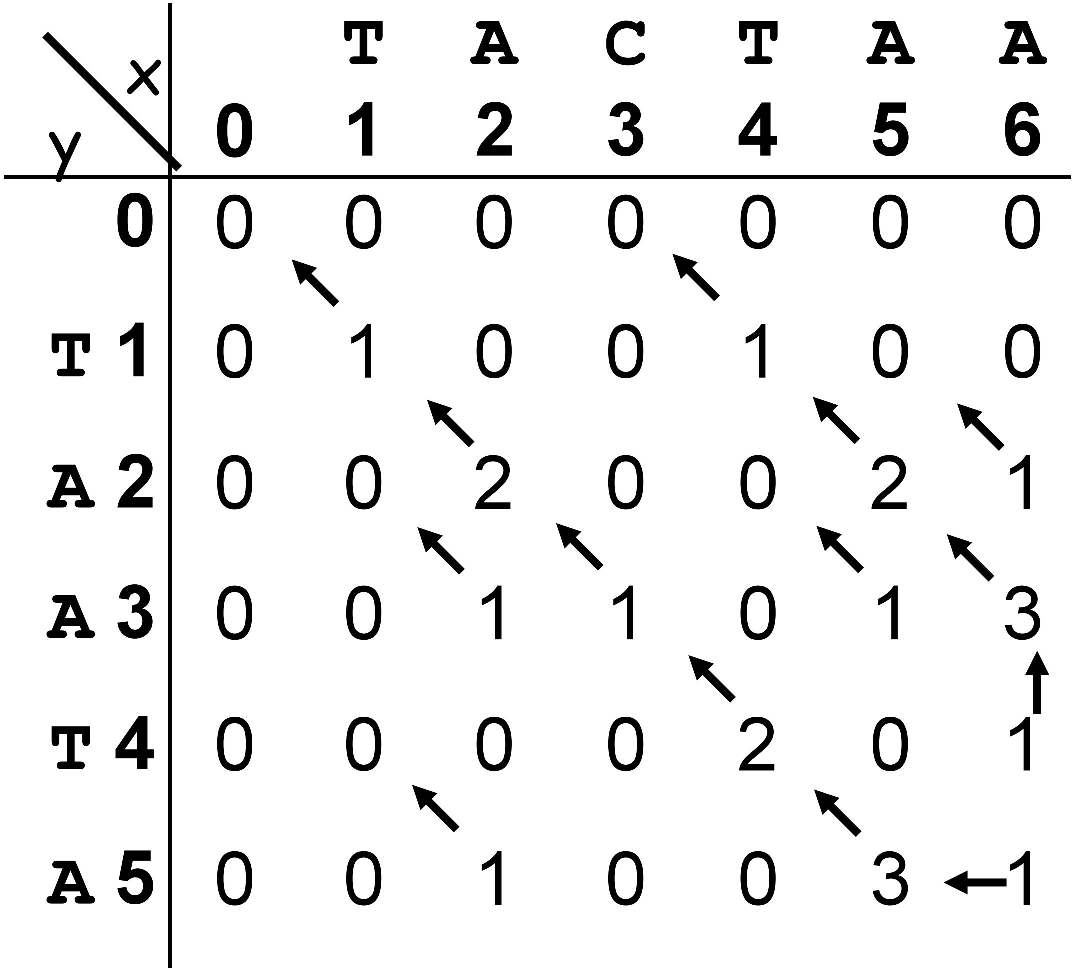
y = TAATA
x = TACTAA



x y	0	T 1	A 2	C 3	T 4	A 5	A 6
0	0	0	0	0	0	0	0
T 1	0	1	0	0	1	0	0
A 2	0	0	2	0	0	2	1
A 3	0						
T 4	0						
A 5	0						

Local Alignment Example

y = TAATA
x = TACTAA

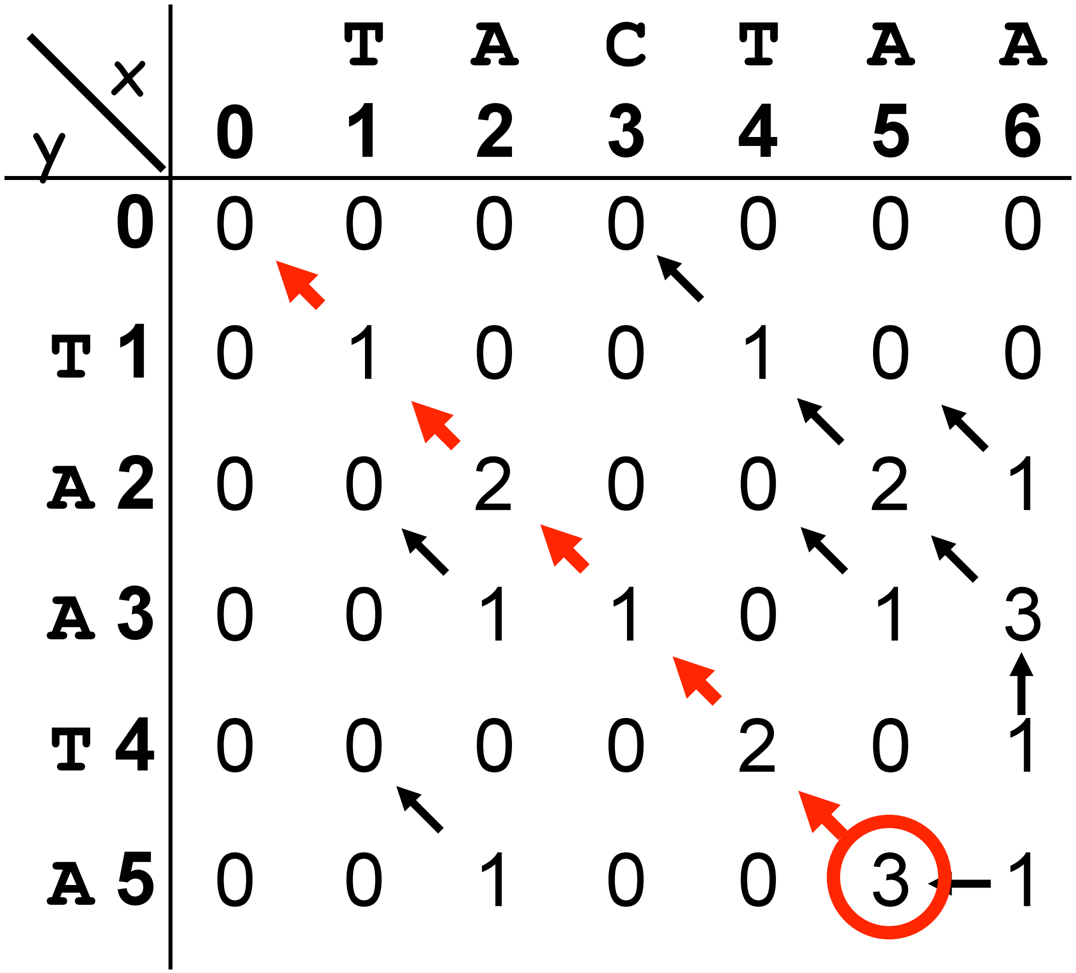


y \ x	0	T 1	A 2	C 3	T 4	A 5	A 6
0	0	0	0	0	0	0	0
T 1	0	1	0	0	1	0	0
A 2	0	0	2	0	0	2	1
A 3	0	0	1	1	0	1	3
T 4	0	0	0	0	2	0	1
A 5	0	0	1	0	0	3	1

Local Alignment Example

y = TAATA

x = TACTAA

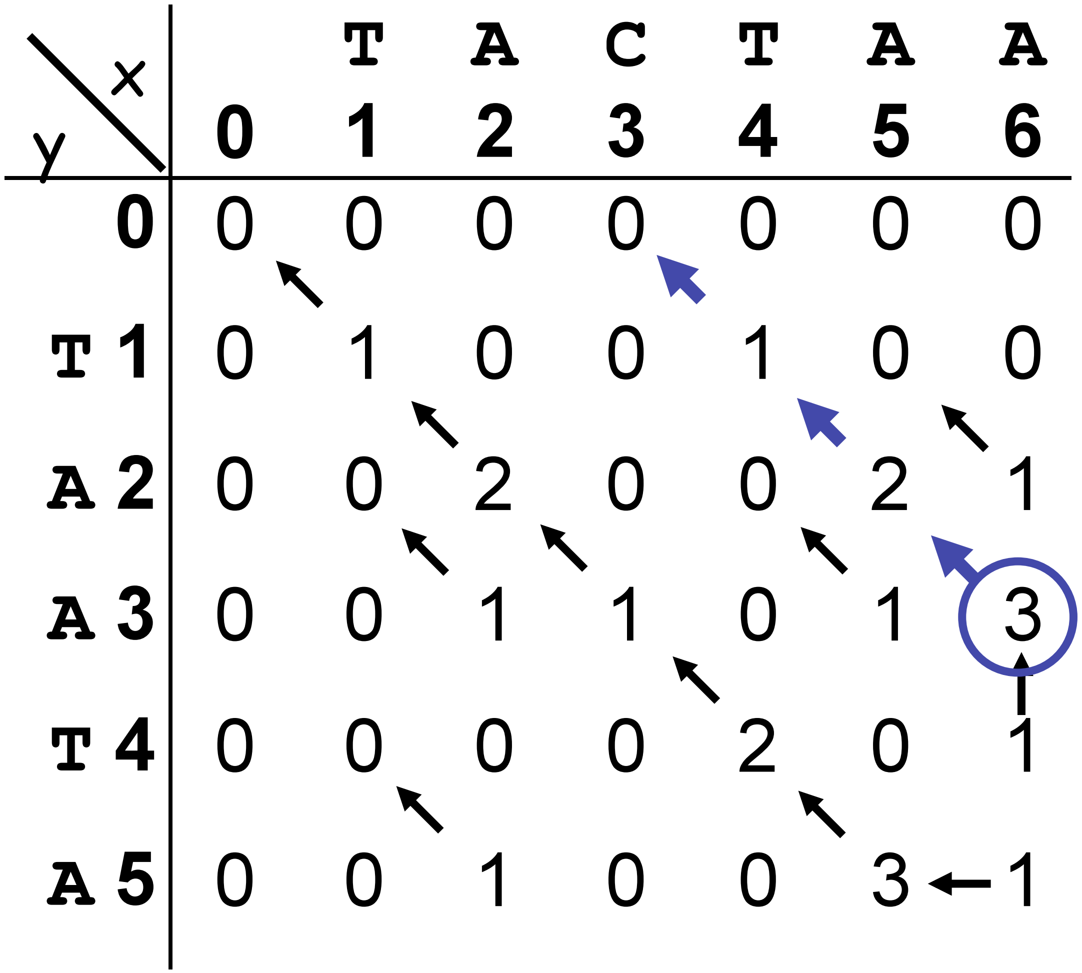


y \ x	0	T 1	A 2	C 3	T 4	A 5	A 6
0	0	0	0	0	0	0	0
T 1	0	1	0	0	1	0	0
A 2	0	0	2	0	0	2	1
A 3	0	0	1	1	0	1	3
T 4	0	0	0	0	2	0	1
A 5	0	0	1	0	0	3	1

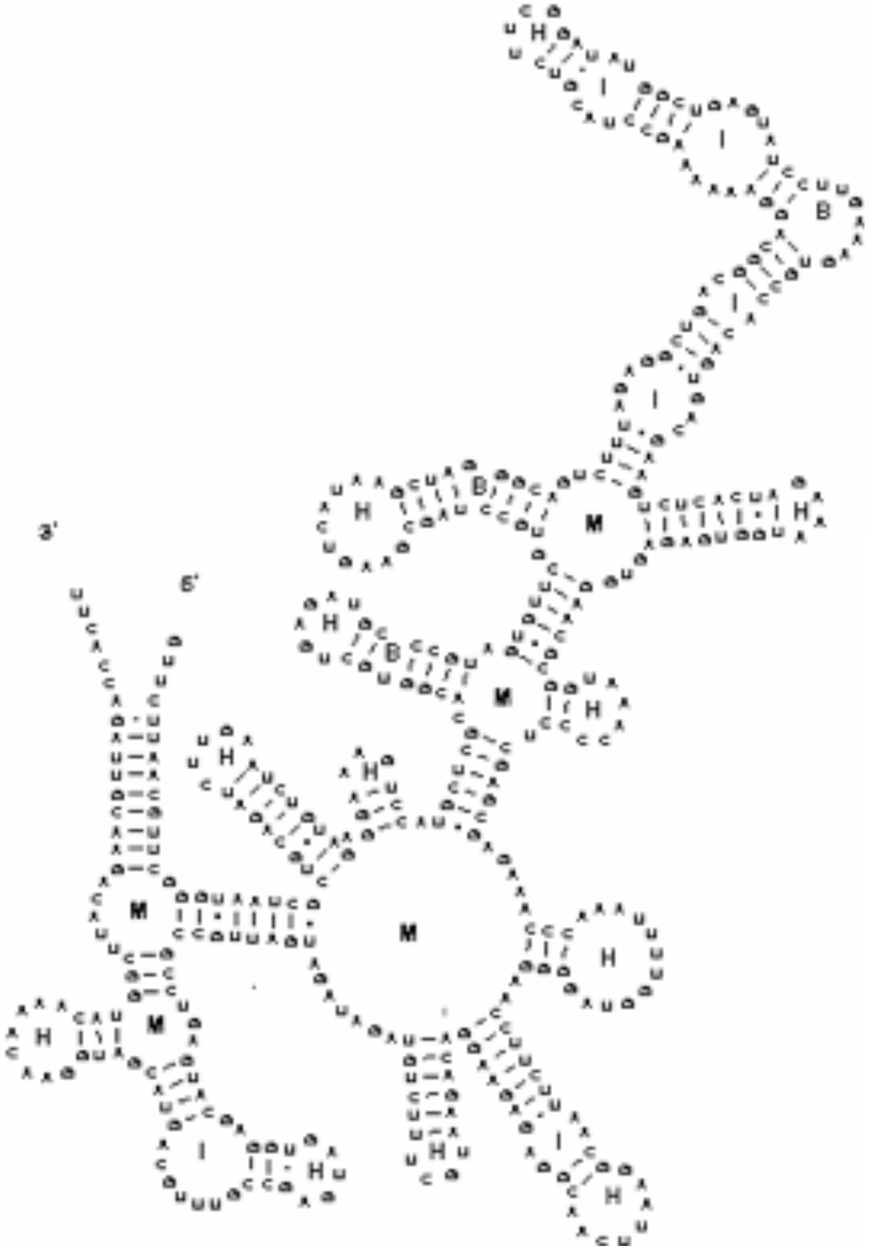
Local Alignment Example

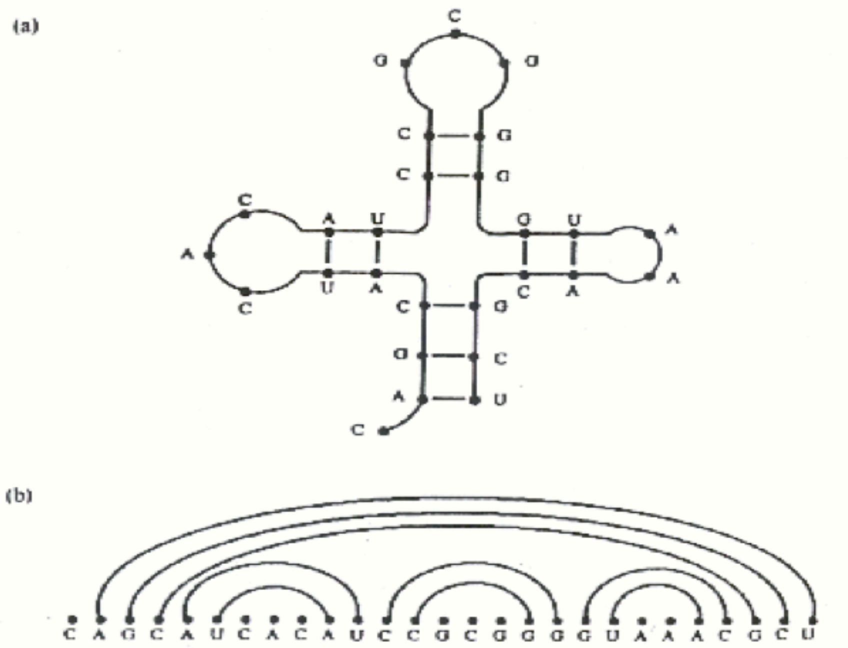
y = TAATA

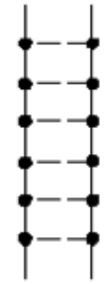
x = TACTAA

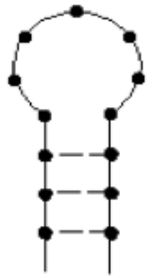


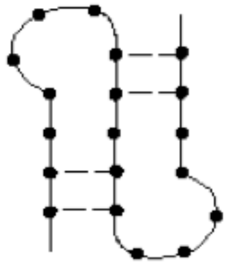
y \ x	0	T 1	A 2	C 3	T 4	A 5	A 6
0	0	0	0	0	0	0	0
T 1	0	1	0	0	1	0	0
A 2	0	0	2	0	0	2	1
A 3	0	0	1	1	0	1	3
T 4	0	0	0	0	2	0	1
A 5	0	0	1	0	0	3	1

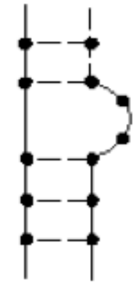


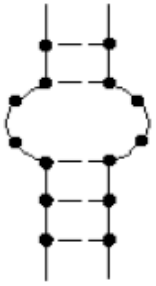


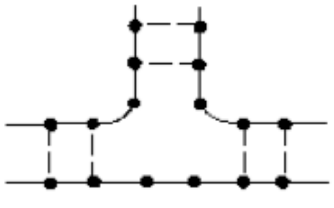

Stem


Hairpin loop

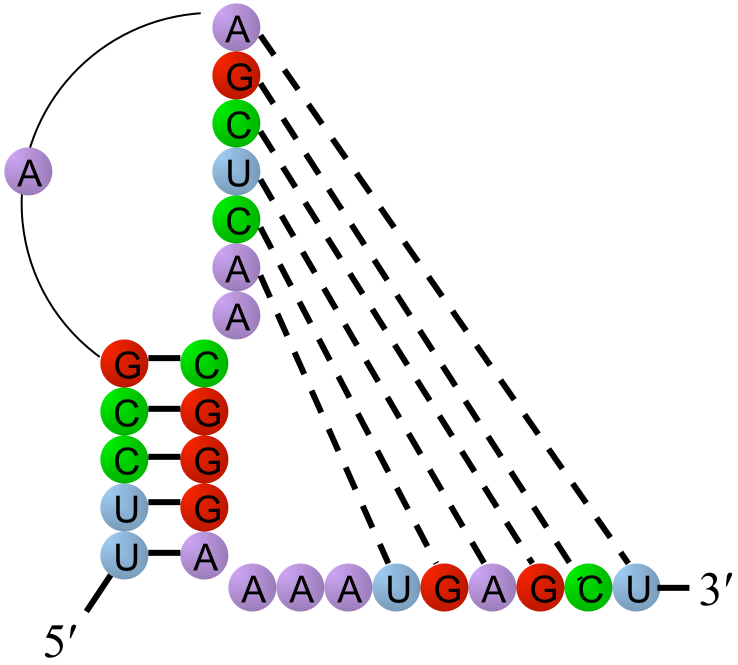

Pseudo knot

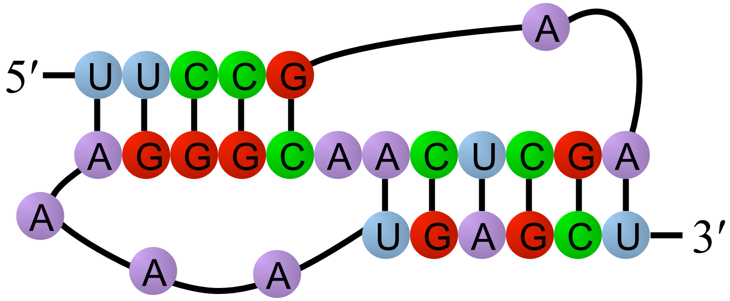

Bulge loop


Internal loop


Branch loop

Pseudoknots

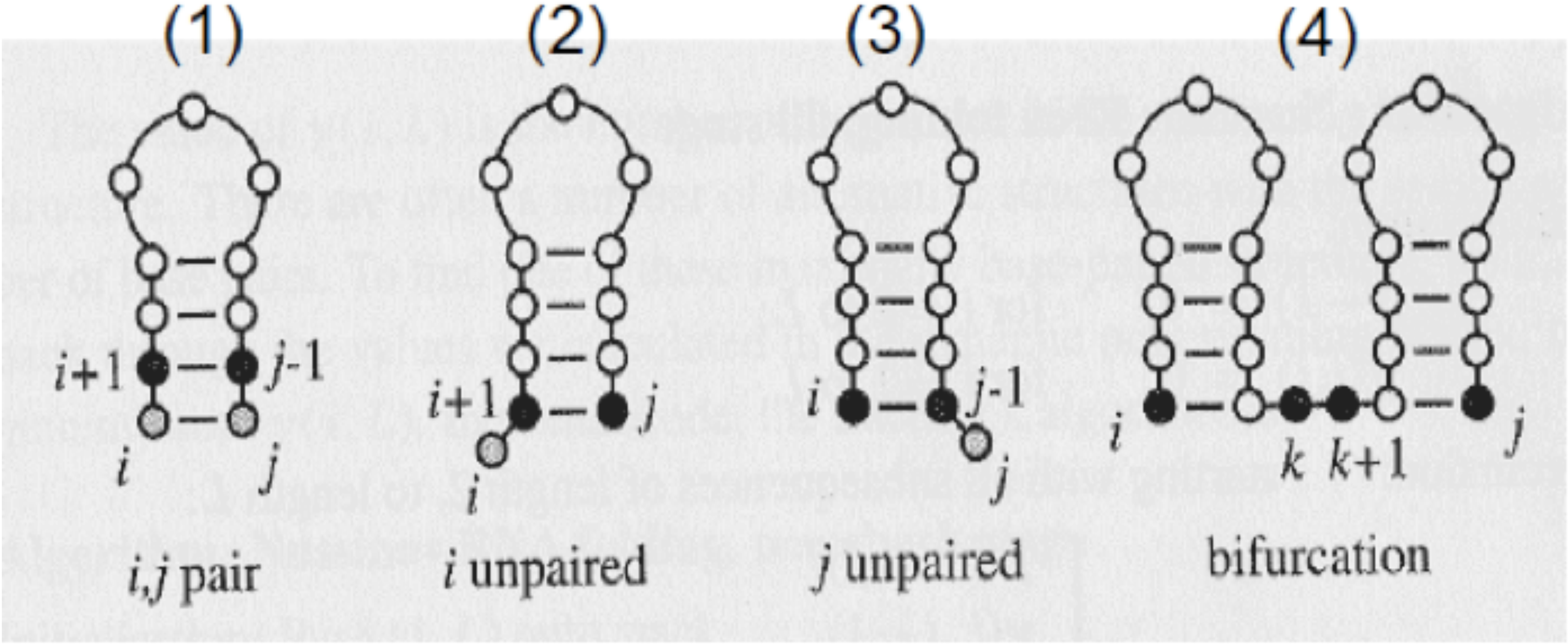




- bases pairs between a loop and positions outside the enclosing stem
- two stems can stack coaxially and mimic a contiguous A-form helix

Image Source: Durbin et al. (2002) “Biological Sequence Analysis”

Nussinov algorithm



Objective: To find the secondary structure with the maximal number of base pairs under the pseudo-knot exclusion constraint.
Principle: Recursive procedure (dynamic programming algorithm). Scoring function: sum of base-pair scores, no penalties for loops Optimal score computed from the optimal scores of subsequences.
Filling-stage. Scores for subsequences are recursively computed from and recorded in a quadratic table.
Trace-back: Reconstruction of filling steps indicates optimal structure
Time-complexity: O(N3)

The Nussinov Folding Algorithm

Example: GGGAAAUCC

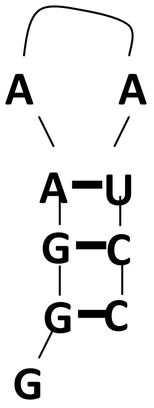
$\gamma(i,j)$ is the maximum number of base pairs in segment $[i,j]$

$$\boxed{\textit{Initialisation}\ \gamma(\mathrm{i},\mathrm{i}-1)=0\ \&\ \gamma(\mathrm{i},\mathrm{i})=0}$$

Starting with all subsequences of length 2, to length L:

$$\gamma(\mathrm{i},\mathrm{j}) = \max\begin{cases}\gamma(\mathrm{i}+1,\mathrm{j})\\ \gamma(\mathrm{i},\mathrm{j}-1)\\ \gamma(\mathrm{i}+1,\mathrm{j}-1)+\delta(\mathrm{i},\mathrm{j})\\ \max_{\mathrm{i}<\mathrm{k}<\mathrm{j}}[\gamma(\mathrm{i},\mathrm{k})+\gamma(\mathrm{k}+1,\mathrm{j})]\end{cases}$$

Where $\delta(i,j) = 1$ if x_i and x_j are a complementary base pair, and $\delta(i,j) = 0$, otherwise.



j →

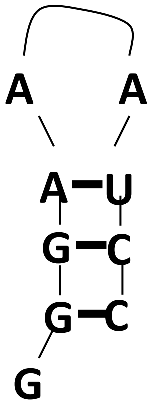
i ↓

	G	G	G	A	A	A	U	C	C
G	0								
G	0	0							
G		0	0						
A			0	0					
A				0	0				
A					0	0			
U						0	0		
C							0	0	
								0	0

j →

i ↓

	G	G	G	A	A	A	U	C	C
G	0	**0**							
G	0	0	**0**						
G		0	0	**0**					
A			0	0	**0**				
A				0	0	**0**			
A					0	0	**1**		
U						0	0	**0**	
C							0	0	**0**
C								0	0



$$\gamma(\mathrm{i},\mathrm{j}) = \max\begin{cases} \gamma(\mathrm{i}+1,\mathrm{j}) \\ \gamma(\mathrm{i},\mathrm{j}-1) \\ \gamma(\mathrm{i}+1,\mathrm{j}-1)+\delta(\mathrm{i},\mathrm{j}) \\ \max_{\mathrm{i}<\mathrm{k}<\mathrm{j}}[\gamma(\mathrm{i},\mathrm{k})+\gamma(\mathrm{k}+1,\mathrm{j})] \end{cases}$$

j →

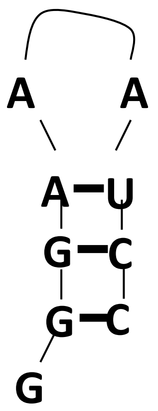
i ↓

	G	G	G	A	A	A	U	C	C
G	0	0	**0**						
G	0	0	0	**0**					
G		0	0	0	**0**				
A			0	0	0	**0**			
A				0	0	0	**1**		
A					0	0	1	**1**	
U						0	0	0	**0**
C							0	0	0
								0	0

After scores for subsequences of length 2 (left) and 3 (right)

Nussinov Folding Algorithm
After scores for subsequences of length 4

$$\gamma(\mathrm{i},\mathrm{j}) = \max\begin{cases} \gamma(\mathrm{i}+1,\mathrm{j}) \\ \gamma(\mathrm{i},\mathrm{j}-1) \\ \gamma(\mathrm{i}+1,\mathrm{j}-1)+\delta(\mathrm{i},\mathrm{j}) \\ \max_{\mathrm{i}<\mathrm{k}<\mathrm{j}}[\gamma(\mathrm{i},\mathrm{k})+\gamma(\mathrm{k}+1,\mathrm{j})] \end{cases}$$



j →

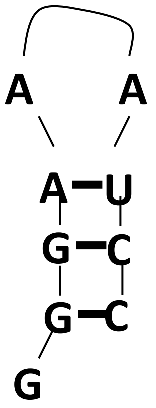
i ↓

	G	G	G	A	A	A	U	C	C
G	0	0	0	**0**					
G	0	0	0	0	**0**				
G		0	0	0	0	**0**			
A			0	0	0	0	**1**		
A				0	0	**0**	**1**	**1**	
A					0	0	1	1	**1**
U						0	0	0	0
C							0	0	0
								0	0

Two optimal substructures for same subsequence

Nussinov Folding Algorithm
After scores for subsequences of length 5

$$\gamma(\mathrm{i},\mathrm{j}) = \max\begin{cases} \gamma(\mathrm{i}+1,\mathrm{j}) \\ \gamma(\mathrm{i},\mathrm{j}-1) \\ \gamma(\mathrm{i}+1,\mathrm{j}-1)+\delta(\mathrm{i},\mathrm{j}) \\ \max_{\mathrm{i}<\mathrm{k}<\mathrm{j}}[\gamma(\mathrm{i},\mathrm{k})+\gamma(\mathrm{k}+1,\mathrm{j})] \end{cases}$$



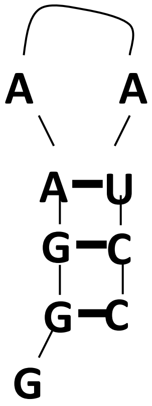
j →

i ↓

	G	G	G	A	A	A	U	C	C
G	0	0	0	0	**0**				
G	0	0	0	0	0	**0**			
G		0	0	0	0	0	**1**		
A			0	0	0	0	1	**1**	
A				0	0	0	1	1	**1**
A					0	0	1	1	1
U						0	0	0	0
C							0	0	0
C								0	0

Nussinov Folding Algorithm
After scores for subsequences of length 6

$$\gamma(\mathrm{i},\mathrm{j}) = \max\begin{cases}\gamma(\mathrm{i}+1,\mathrm{j})\\ \gamma(\mathrm{i},\mathrm{j}-1)\\ \gamma(\mathrm{i}+1,\mathrm{j}-1)+\delta(\mathrm{i},\mathrm{j})\\ \max_{\mathrm{i}<\mathrm{k}<\mathrm{j}}[\gamma(\mathrm{i},\mathrm{k})+\gamma(\mathrm{k}+1,\mathrm{j})]\end{cases}$$



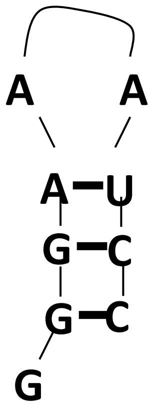
j →

i ↓

	G	G	G	A	A	A	U	C	C
G	0	0	0	0	0	**0**			
G	0	0	0	0	0	0	**1**		
G		0	0	0	0	0	1	**2**	
A			0	0	0	0	1	1	**1**
A				0	0	0	1	1	1
A					0	0	1	1	1
U						0	0	0	0
C							0	0	0
								0	0

Nussinov Folding Algorithm
After scores for subsequences of length 7

$$\gamma(\mathrm{i},\mathrm{j}) = \max\begin{cases}\gamma(\mathrm{i}+1,\mathrm{j})\\ \gamma(\mathrm{i},\mathrm{j}-1)\\ \gamma(\mathrm{i}+1,\mathrm{j}-1)+\delta(\mathrm{i},\mathrm{j})\\ \max_{\mathrm{i}<\mathrm{k}<\mathrm{j}}[\gamma(\mathrm{i},\mathrm{k})+\gamma(\mathrm{k}+1,\mathrm{j})]\end{cases}$$



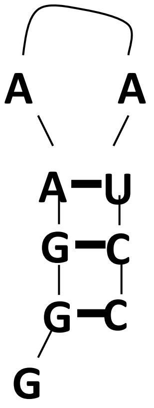
j →

i ↓

	G	G	G	A	A	A	U	C	C
G	0	0	0	0	0	0	**1**		
G	0	0	0	0	0	0	1	**2**	
G		0	0	0	0	0	1	2	**2**
A			0	0	0	0	1	1	1
A				0	0	0	1	1	1
A					0	0	1	1	1
U						0	0	0	0
C							0	0	0
								0	0

Nussinov Folding Algorithm
After scores for subsequences of length 8

$$\gamma(\mathrm{i},\mathrm{j}) = \max\begin{cases} \gamma(\mathrm{i}+1,\mathrm{j}) \\ \gamma(\mathrm{i},\mathrm{j}-1) \\ \gamma(\mathrm{i}+1,\mathrm{j}-1)+\delta(\mathrm{i},\mathrm{j}) \\ \max_{\mathrm{i}<\mathrm{k}<\mathrm{j}}[\gamma(\mathrm{i},\mathrm{k})+\gamma(\mathrm{k}+1,\mathrm{j})] \end{cases}$$



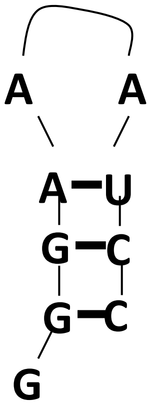
j →

i ↓

	G	G	G	A	A	A	U	C	C
G	0	0	0	0	0	0	1	**2**	
G	0	0	0	0	0	0	1	2	**3**
G		0	0	0	0	0	1	2	2
A			0	0	0	0	1	1	1
A				0	0	0	1	1	1
A					0	0	1	1	1
U						0	0	0	0
C							0	0	0
								0	0

Nussinov Folding Algorithm
After scores for subsequences of length 9

$$\gamma(\mathrm{i},\mathrm{j}) = \max\begin{cases}\gamma(\mathrm{i}+1,\mathrm{j})\\ \gamma(\mathrm{i},\mathrm{j}-1)\\ \gamma(\mathrm{i}+1,\mathrm{j}-1)+\delta(\mathrm{i},\mathrm{j})\\ \max_{\mathrm{i}<\mathrm{k}<\mathrm{j}}[\gamma(\mathrm{i},\mathrm{k})+\gamma(\mathrm{k}+1,\mathrm{j})]\end{cases}$$

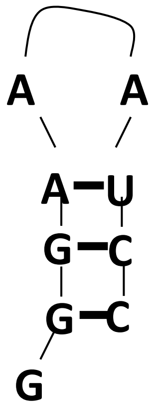


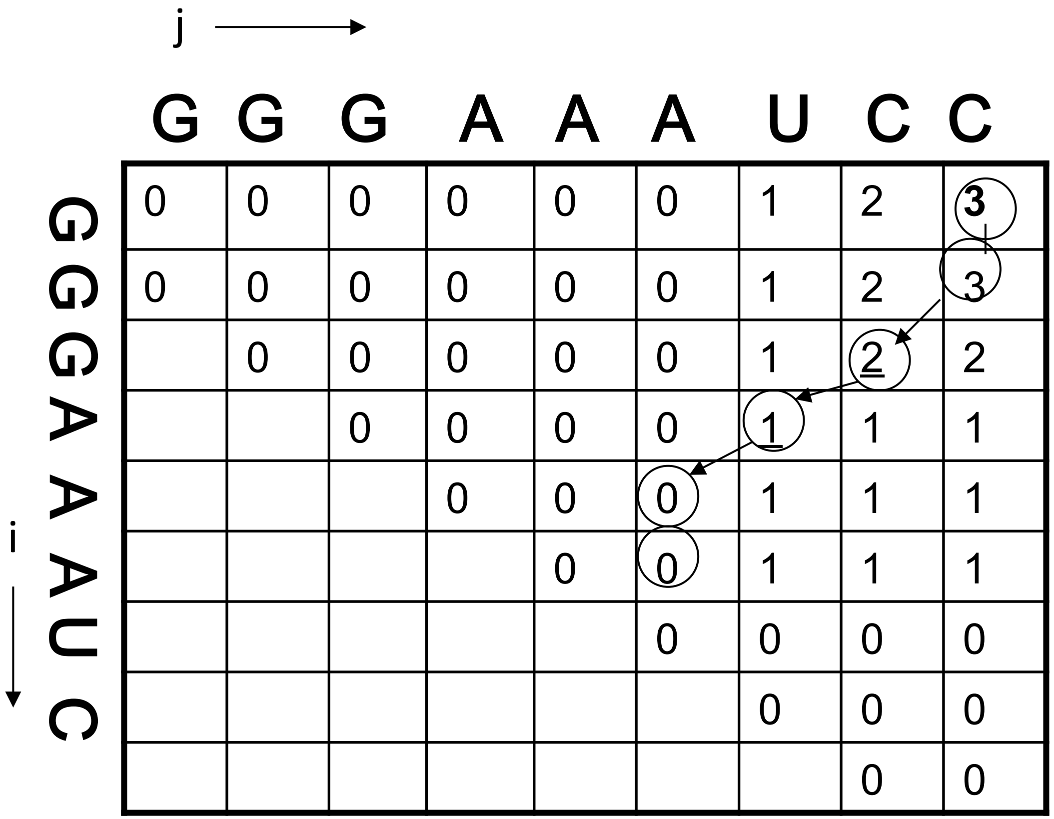
j ⟶

i ↓

	G	G	G	A	A	A	U	C	C
G	0	0	0	0	0	0	1	2	**3**
G	0	0	0	0	0	0	1	2	3
G		0	0	0	0	0	1	$\underline{2}$	2
A			0	0	0	0	$\underline{1}$	1	1
A				0	0	$\underline{0}$	1	1	1
A					0	0	1	1	1
U						0	0	0	0
C							0	0	0
C								0	0

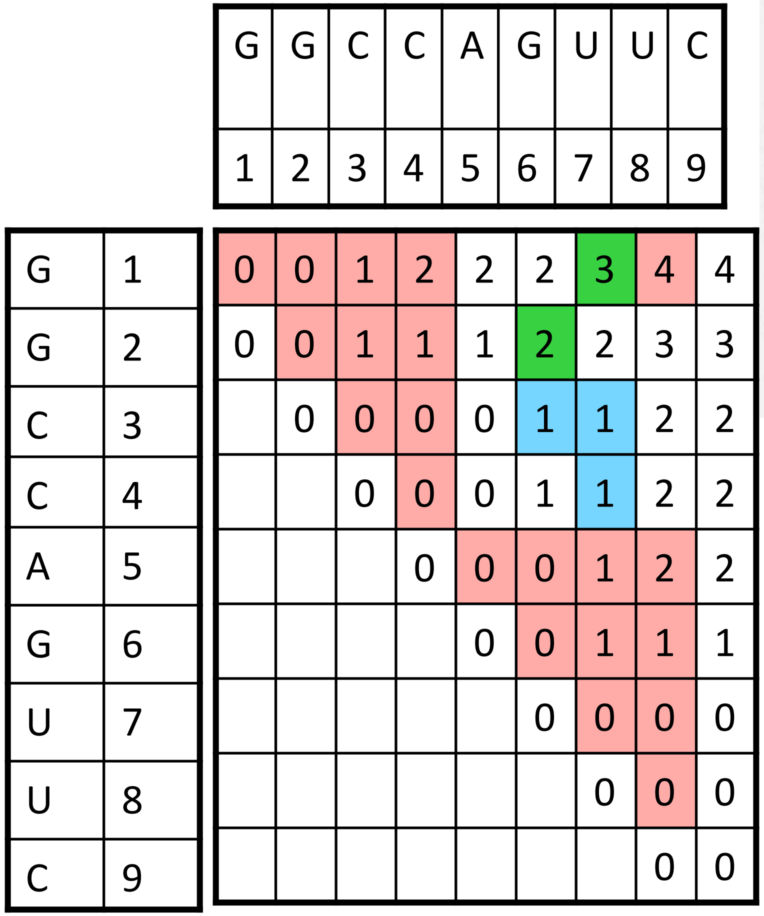
Nussinov Folding Algorithm
Traceback



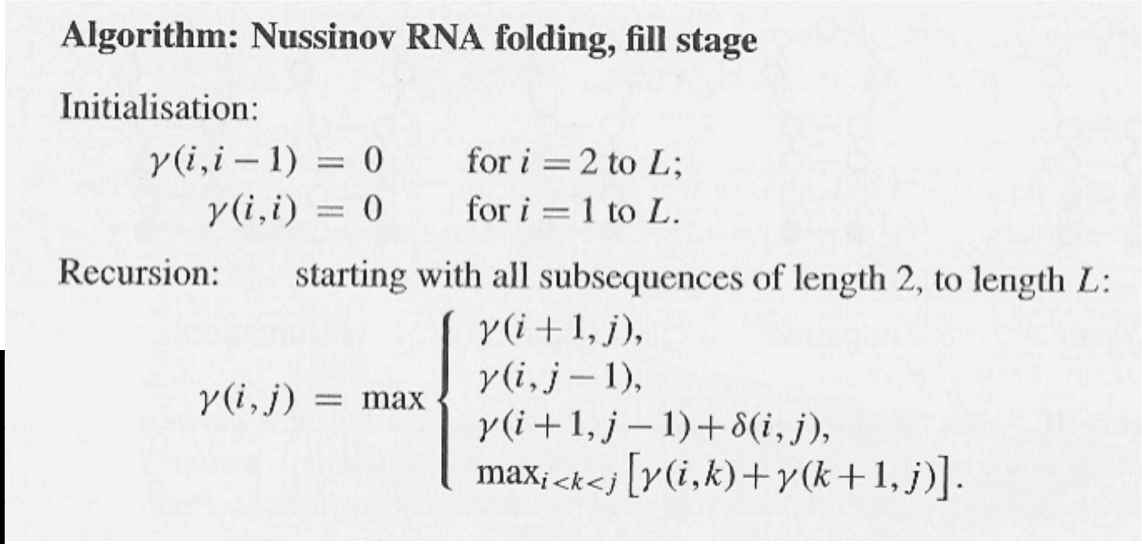


	G	G	G	A	A	A	U	C	C
G	0	0	0	0	0	0	1	2	3
G	0	0	0	0	0	0	1	2	3
G		0	0	0	0	0	1	2	2
A			0	0	0	0	1	1	1
A				0	0	0	1	1	1
A					0	0	1	1	1
U						0	0	0	0
C							0	0	0
C								0	0

Nussinov algorithm: fill-stage



		G	G	C	C	A	G	U	U	C
		1	2	3	4	5	6	7	8	9
G	1	0	0	1	2	2	2	3	4	4
G	2	0	0	1	1	1	2	2	3	3
C	3		0	0	0	0	1	1	2	2
C	4			0	0	0	1	1	2	2
A	5				0	0	0	1	2	2
G	6					0	0	1	1	1
U	7						0	0	0	0
U	8							0	0	0
C	9								0	0



Algorithm: Nussinov RNA folding, fill stage

Initialisation:

$$\gamma(i,i-1) = 0 \qquad \text{for } i = 2 \text{ to } L;$$
$$\gamma(i,i) = 0 \qquad \text{for } i = 1 \text{ to } L.$$

Recursion: starting with all subsequences of length 2, to length L:

$$\gamma(i,j) = \max \begin{cases} \gamma(i+1,j), \\ \gamma(i,j-1), \\ \gamma(i+1,j-1)+\delta(i,j), \\ \max_{i<k<j}\left[\gamma(i,k)+\gamma(k+1,j)\right]. \end{cases}$$

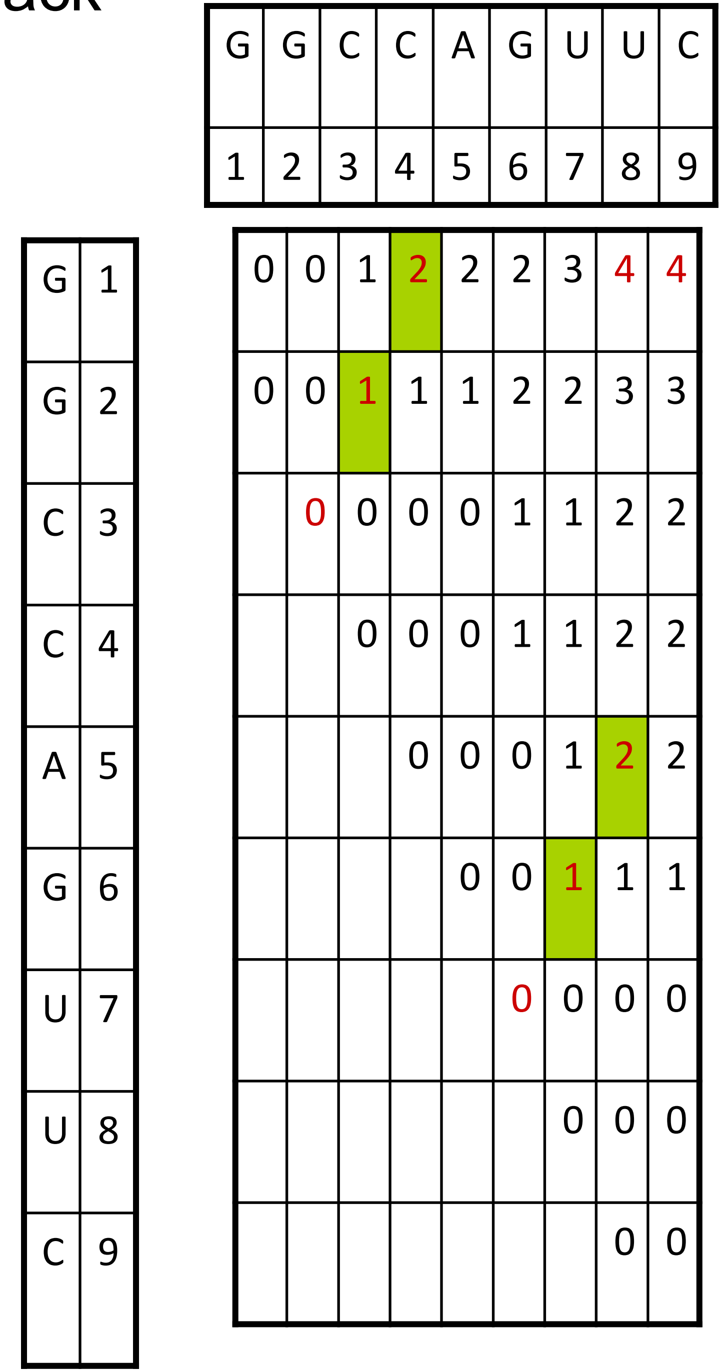
Scoring system:
$\delta(i,j) = 1$ for all RNA Watson-Crick base-pairs including G-U else $\delta(i,j) = 0$.

Blue: addition of unpaired base 3 or 7

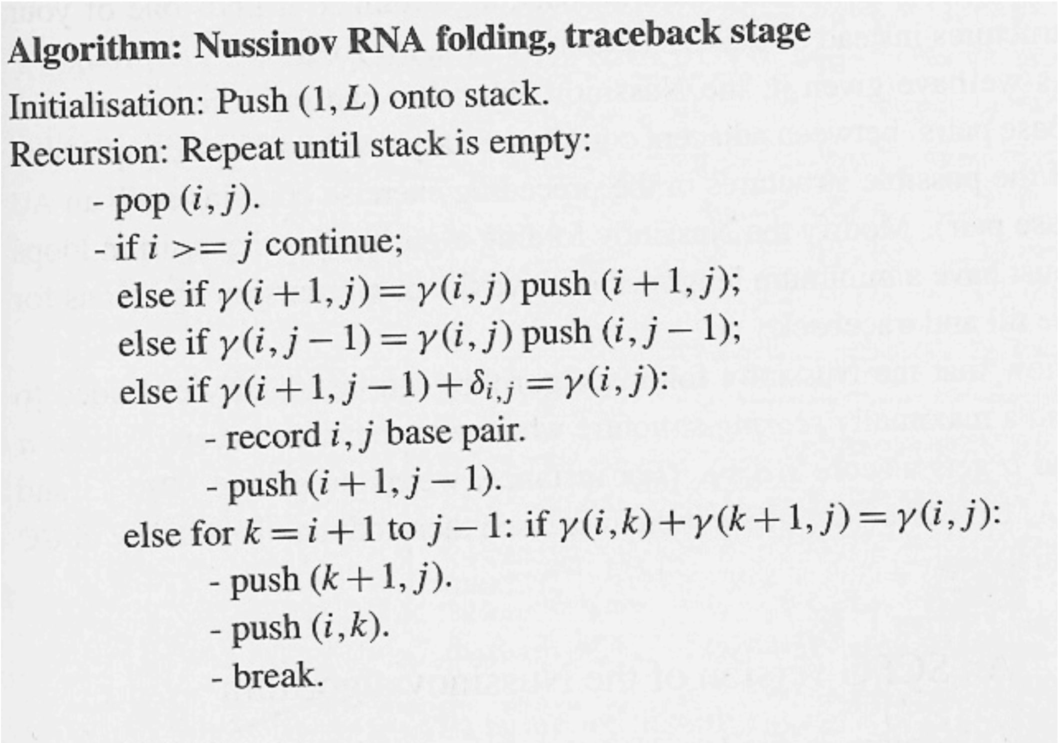
Green: addition of paired bases 1,7

Pink: joining of substructures 1..4 and 5..8

Nussinov algorithm: trace-back



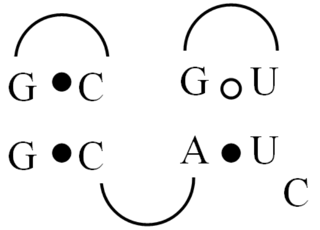
	G 1	G 2	C 3	C 4	A 5	G 6	U 7	U 8	C 9
G 1	0	0	1	2	2	2	3	4	4
G 2	0	0	1	1	1	2	2	3	3
C 3		0	0	0	0	1	1	2	2
C 4			0	0	0	1	1	2	2
A 5				0	0	0	1	2	2
G 6					0	0	1	1	1
U 7						0	0	0	0
U 8							0	0	0
C 9								0	0



Algorithm: Nussinov RNA folding, traceback stage

```
Initialisation: Push $(1, L)$ onto stack.
Recursion: Repeat until stack is empty:
    - pop $(i, j)$.
    - if $i >= j$ continue;
      else if $\gamma(i+1, j) = \gamma(i, j)$ push $(i+1, j)$;
      else if $\gamma(i, j-1) = \gamma(i, j)$ push $(i, j-1)$;
      else if $\gamma(i+1, j-1) + \delta_{i,j} = \gamma(i, j)$:
            - record $i, j$ base pair.
            - push $(i+1, j-1)$.
      else for $k = i+1$ to $j-1$: if $\gamma(i,k) + \gamma(k+1, j) = \gamma(i, j)$:
            - push $(k+1, j)$.
            - push $(i, k)$.
            - break.
```

```
current  record  stack
                 1,9
1,9              1,8
1,8              1,4  5,8
1,4      1,4     2,3  5,8
2,3      2,3     3,2  5,8
3,2              5,8
5,8      5,8     6,7
6,7      6,7     7,6
7,6
```



- souce code http://www.ibluemojo.com/school/RNA/RNAFolding.java

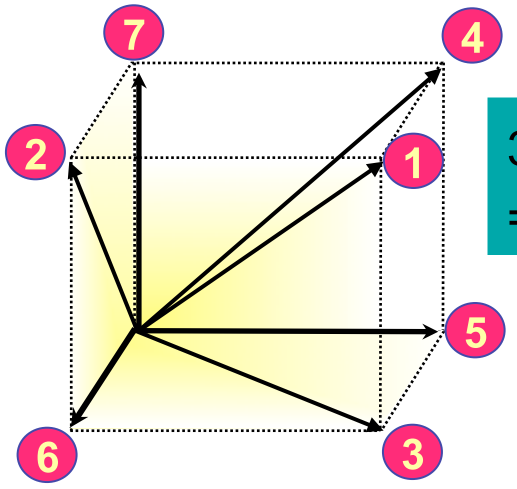
- Blast source code http://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=BlastDocs&DOC_TYPE=DeveloperInfo

- BioJava – www.biojava.org
- BioPerl – www.bioperl.org
- BioPython – www.biopython.org
- BioCorba – www.biocorba.org
- C++ www.ncbi.nlm.nih.gov/IEB/ToolBox/CPP_DOC/

Three-dimensional Alignment Path Matrix

For two sequences, there are three ways to extend an alignment.

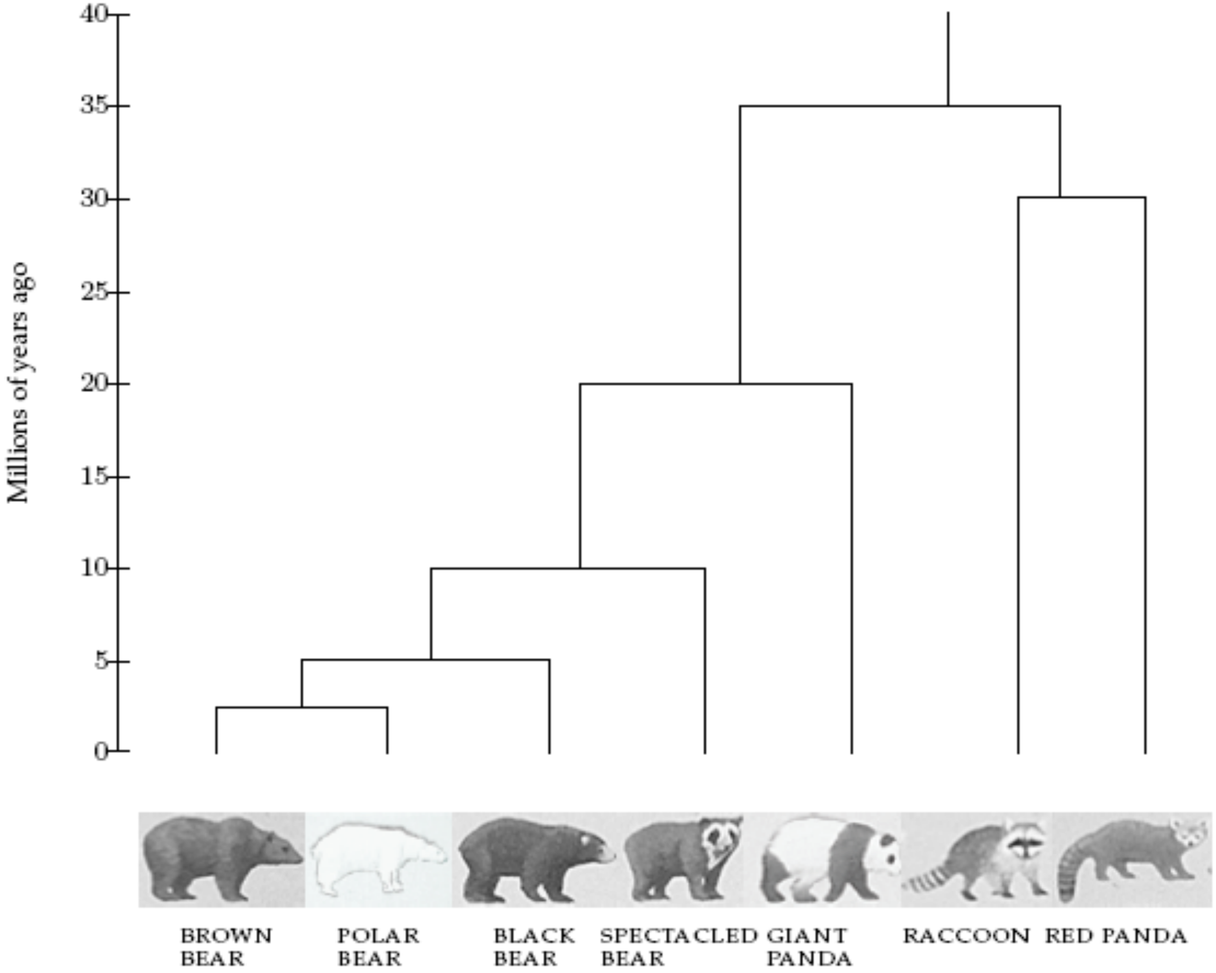
For N sequences, there are 2^N-1 ways to extend an alignment.

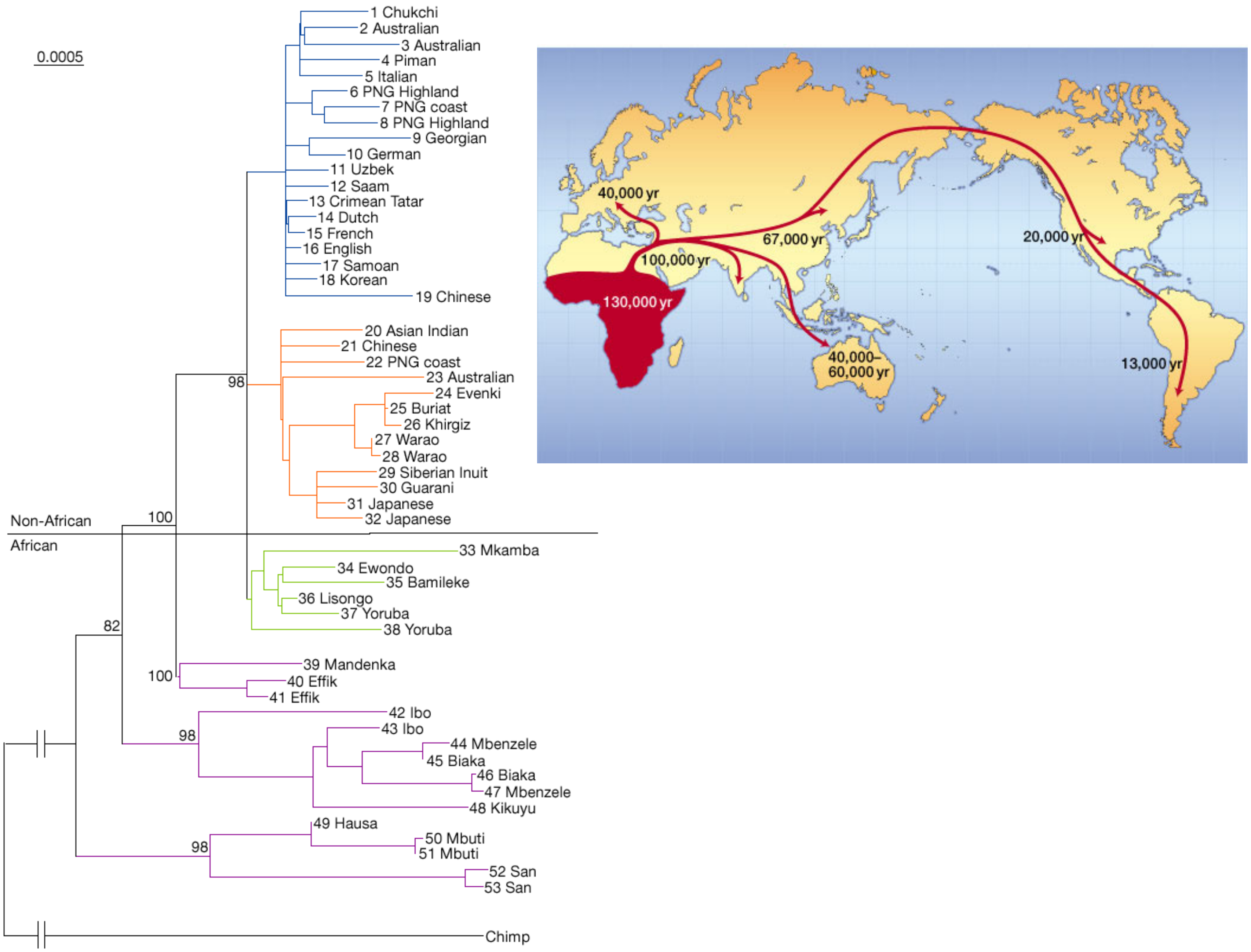


3 sequences
=> 7 ways

1	2	3	4	5	6	7
ABC	**AB–**	**ABC**	**ABC**	**ABC**	**AB–**	**AB–**
ABC	**ABC**	**AB–**	**ABC**	**AB–**	**AB–**	**ABC**
ABC	**ABC**	**ABC**	**AB–**	**AB–**	**ABC**	**AB–**

Evolutionary Tree of Bears and Raccoons




0.0005
1 Chukchi
2 Australian
3 Australian
4 Piman
5 Italian
6 PNG Highland
7 PNG coast
8 PNG Highland
9 Georgian
10 German
11 Uzbek
12 Saam
13 Crimean Tatar
14 Dutch
15 French
16 English
17 Samoan
18 Korean
19 Chinese
20 Asian Indian
21 Chinese
22 PNG coast
23 Australian
24 Evenki
25 Buriat
26 Khirgiz
27 Warao
28 Warao
29 Siberian Inuit
30 Guarani
31 Japanese
32 Japanese
98
100
Non-African
African
33 Mkamba
34 Ewondo
35 Bamileke
36 Lisongo
37 Yoruba
38 Yoruba
82
39 Mandenka
100
40 Effik
41 Effik
42 Ibo
98
43 Ibo
44 Mbenzele
45 Biaka
46 Biaka
47 Mbenzele
48 Kikuyu
49 Hausa
98
50 Mbuti
51 Mbuti
52 San
53 San
Chimp
40,000 yr
67,000 yr
100,000 yr
20,000 yr
130,000 yr
40,000–60,000 yr
13,000 yr

H3 HA

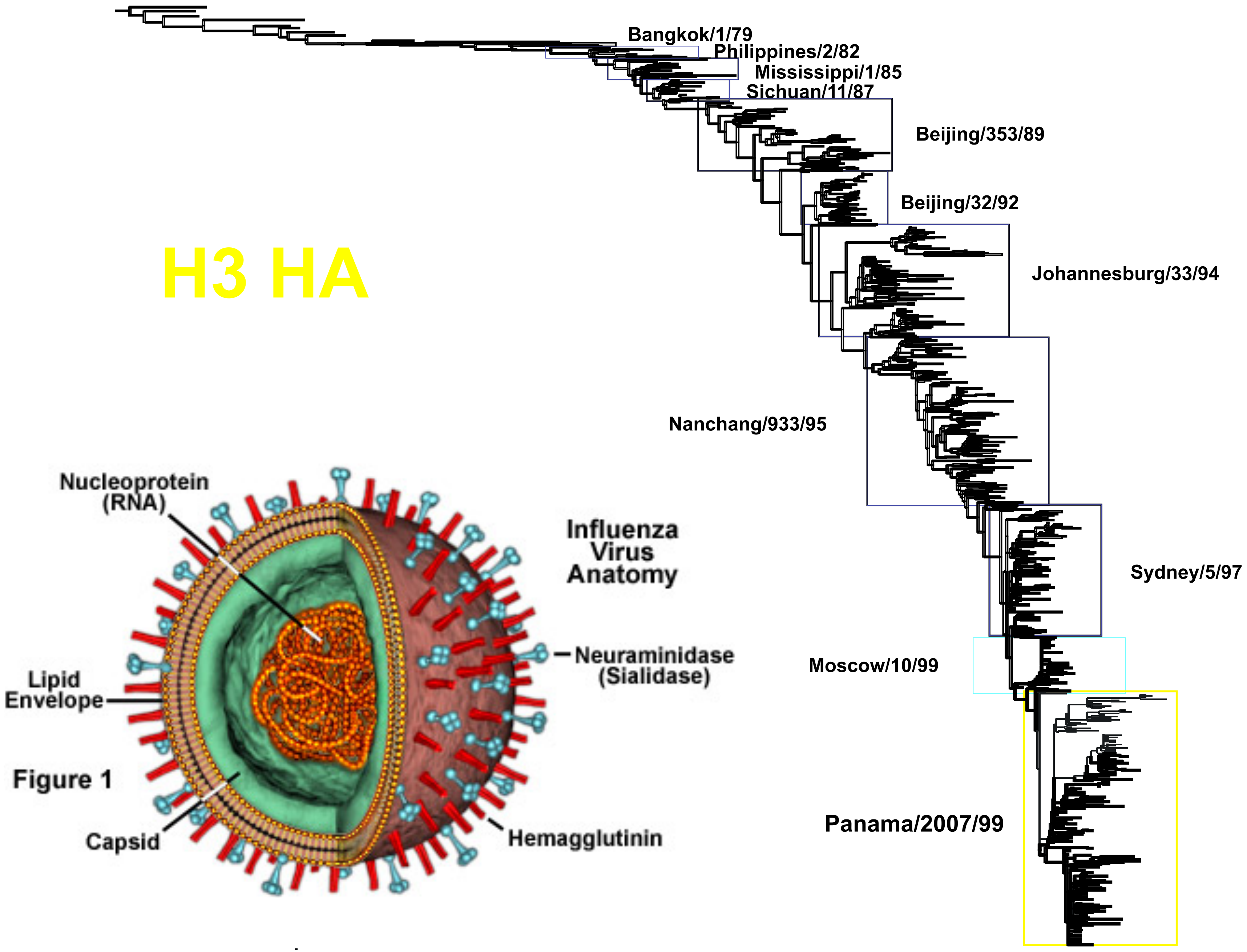
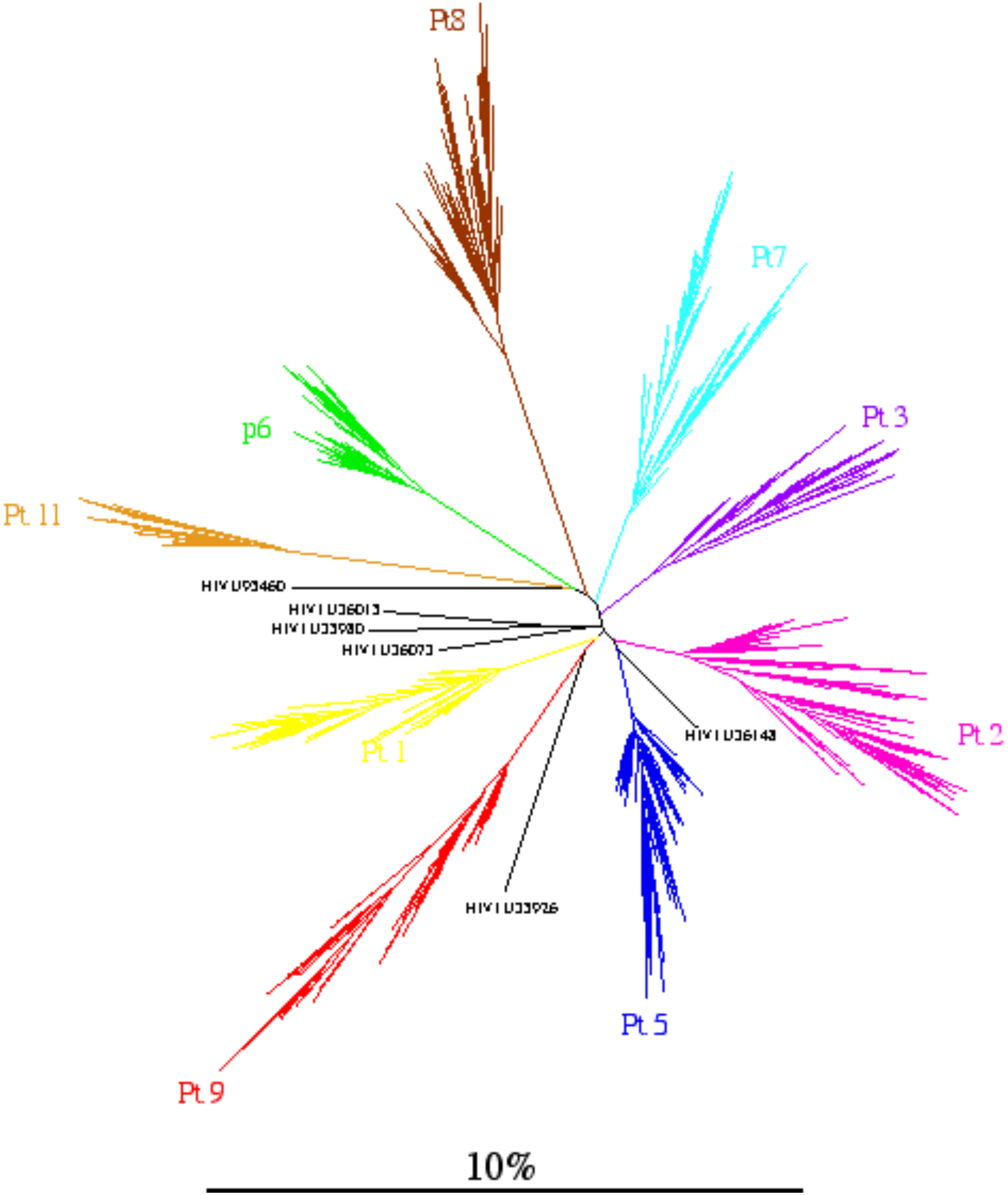


Figure 1


Pt8
Pt7
Pt 3
p6
Pt 11
HIVU93460
HIV1U36013
HIV1U33980
HIV1U36073
Pt 2
HIV1U36143
Pt 1
HIV1U33926
Pt 5
Pt 9
10%