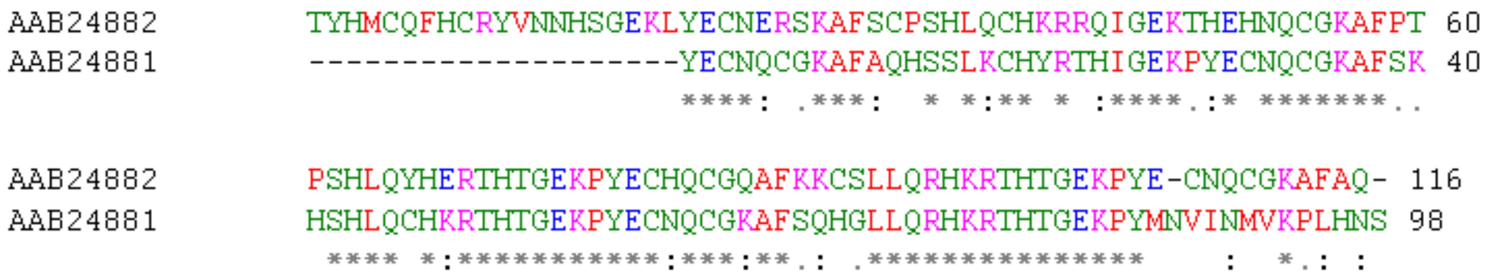
Sequence alignments and scoring matrices

Arne Elofsson

To read: http://perso.fundp.ac.be/~lambertc/DEA-bioinfo/CLambert_curr_gen_2003.pdf

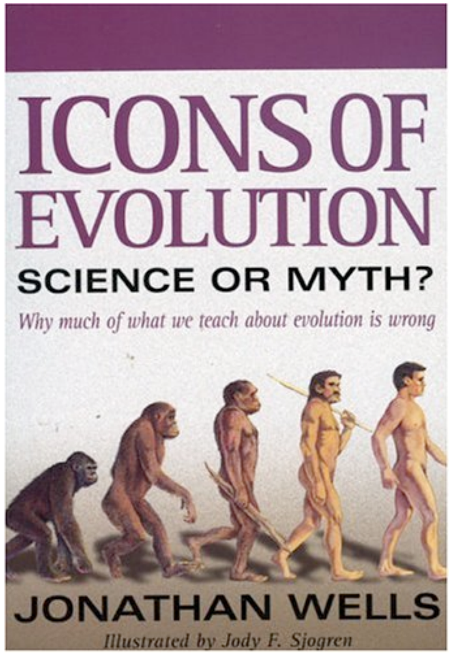
To read: Wikipidea about Sequence Alignment

Why alignments ?



```
AAB24882        TYHMCQFHCRYVNNHSGEKLYECNERSKAFSCPSHLQCHKRRQIGEKTHEHNQCGKAFPT 60
AAB24881        --------------------YECNQCGKAFAQHSSLKCHYRTHIGEKPYECNQCGKAFSK 40
                                    ****: .***:  * *:** * :****.:* *******..

AAB24882        PSHLQYHERTHTGEKPYECHQCGQAFKKCSLLQRHKRTHTGEKPYE-CNQCGKAFAQ- 116
AAB24881        HSHLQCHKRTHTGEKPYECNQCGKAFSQHGLLQRHKRTHTGEKPYMNVINMVKPLHNS 98
                 **** *:***********:***:**.: .***************    :  *.: :
```

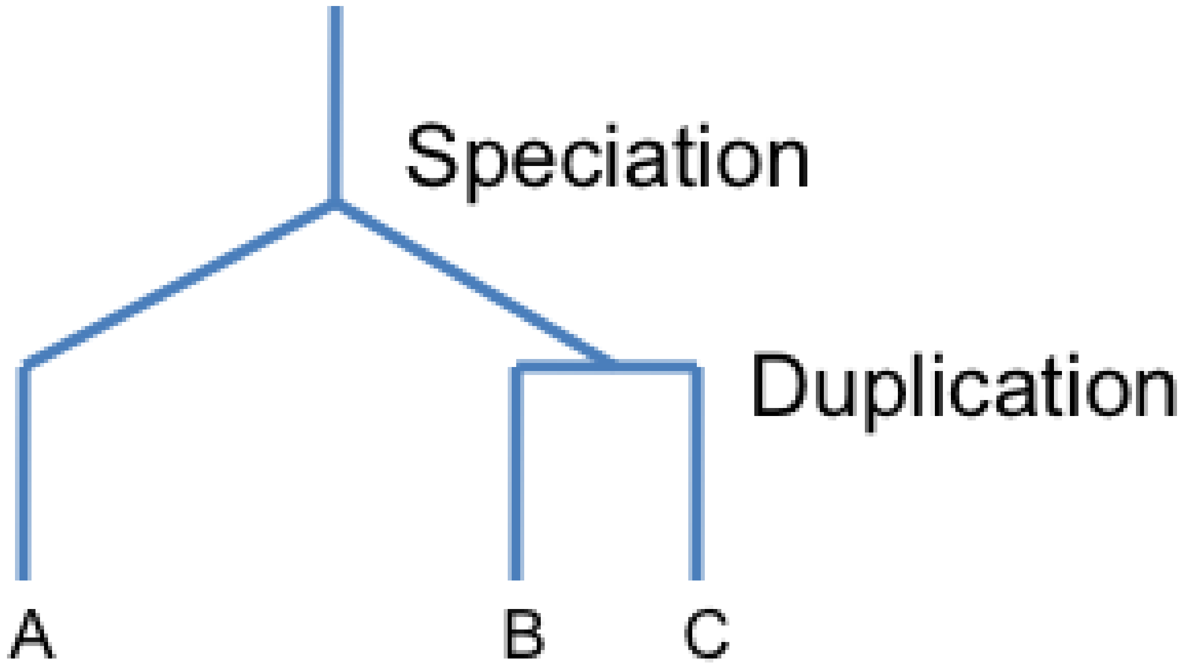
- Detect homology
- Study evolution
- Predict functions
- Model 3D-structure


ICONS OF EVOLUTION
SCIENCE OR MYTH?
Why much of what we teach about evolution is wrong
JONATHAN WELLS
Illustrated by Jody F. Sjogren

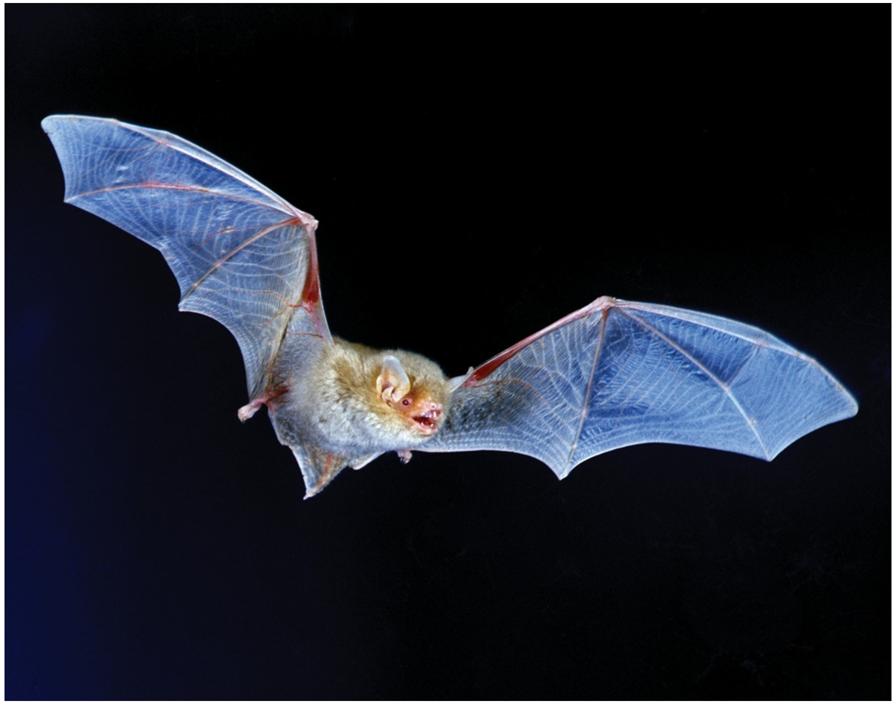
Sequence similarity

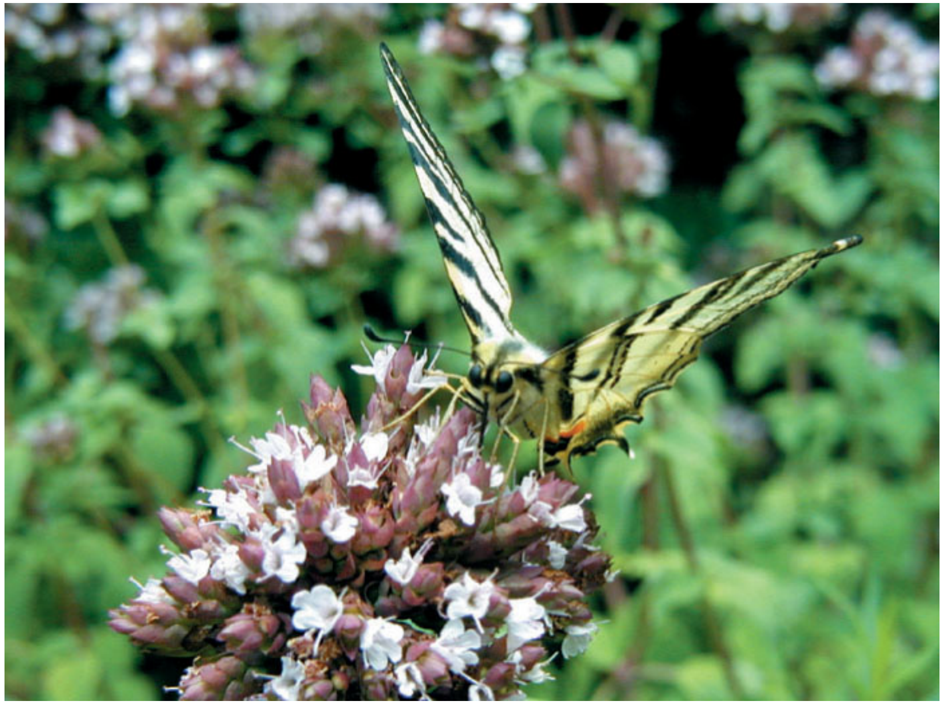
- Homologs have a common ancestor
- Gene duplication or speciation
- High sequence similarity indicates homology
- Homologs have similar 3D-structure

Homology



Convergent evolution

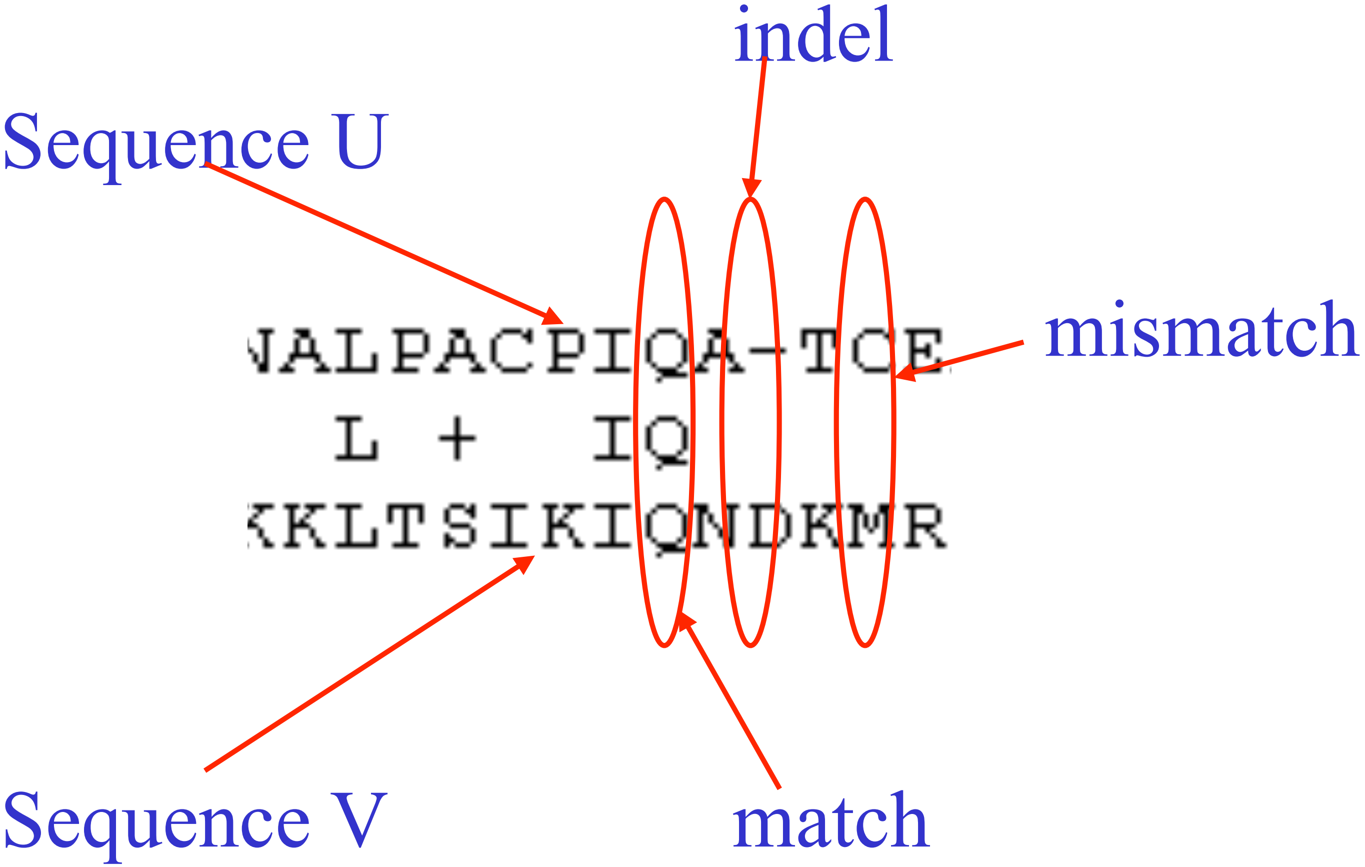




What is an alignment

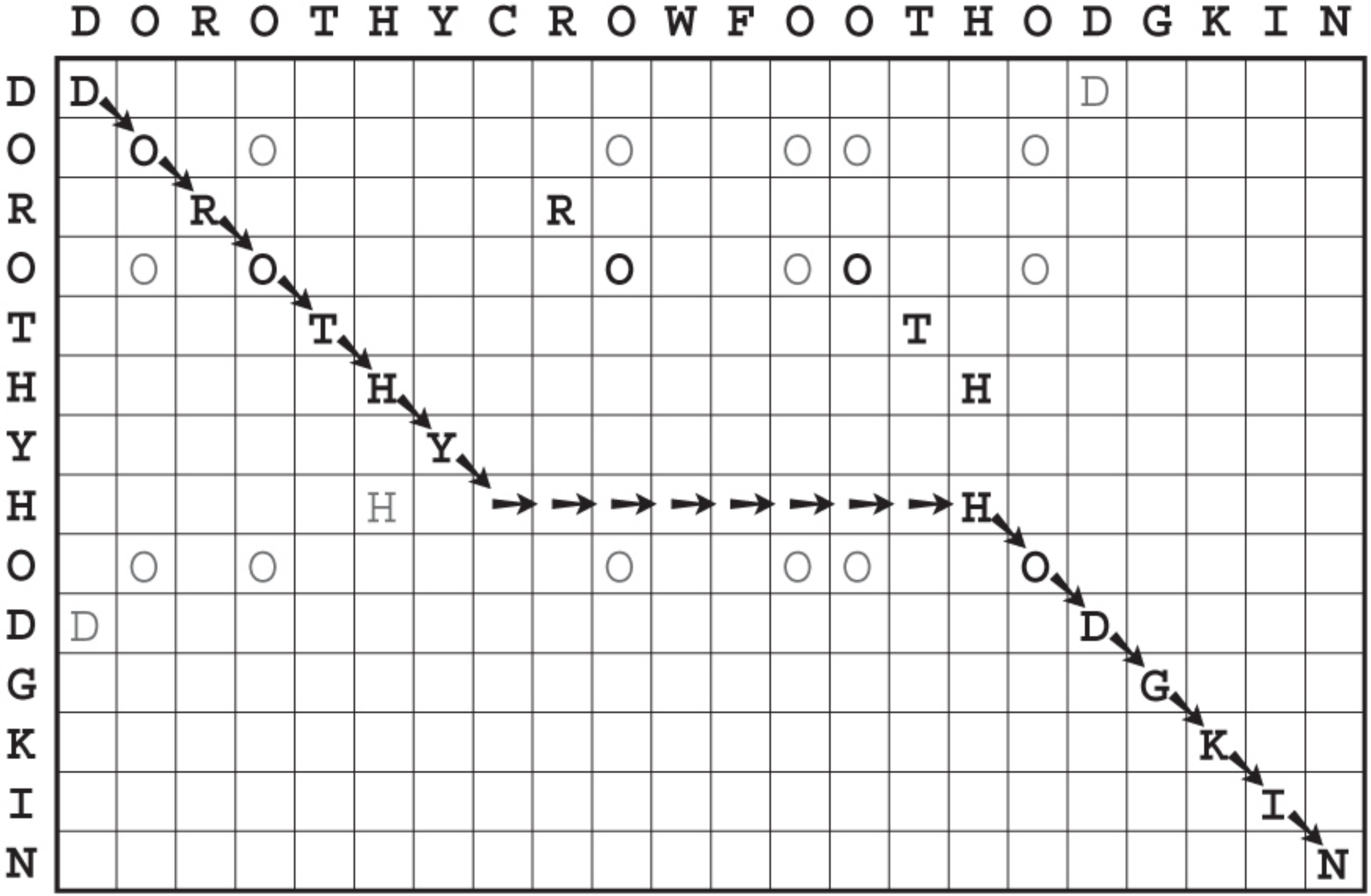
```
THISSEQUENCE
||  ||||||||      10/12 Identical
THATSEQUENCE

THATSEQUENCE
||      |  |       4/12 Identical
THISISASEQUENCE

THISISA-SEQUENCE
||    | ||||||||   11/12 Identical
TH----ATSEQUENCE
```

What can an alignment say ?



```
NALPACPIQA-TCE
  L +  IQ
KKLTSIKIQNDKMR
```

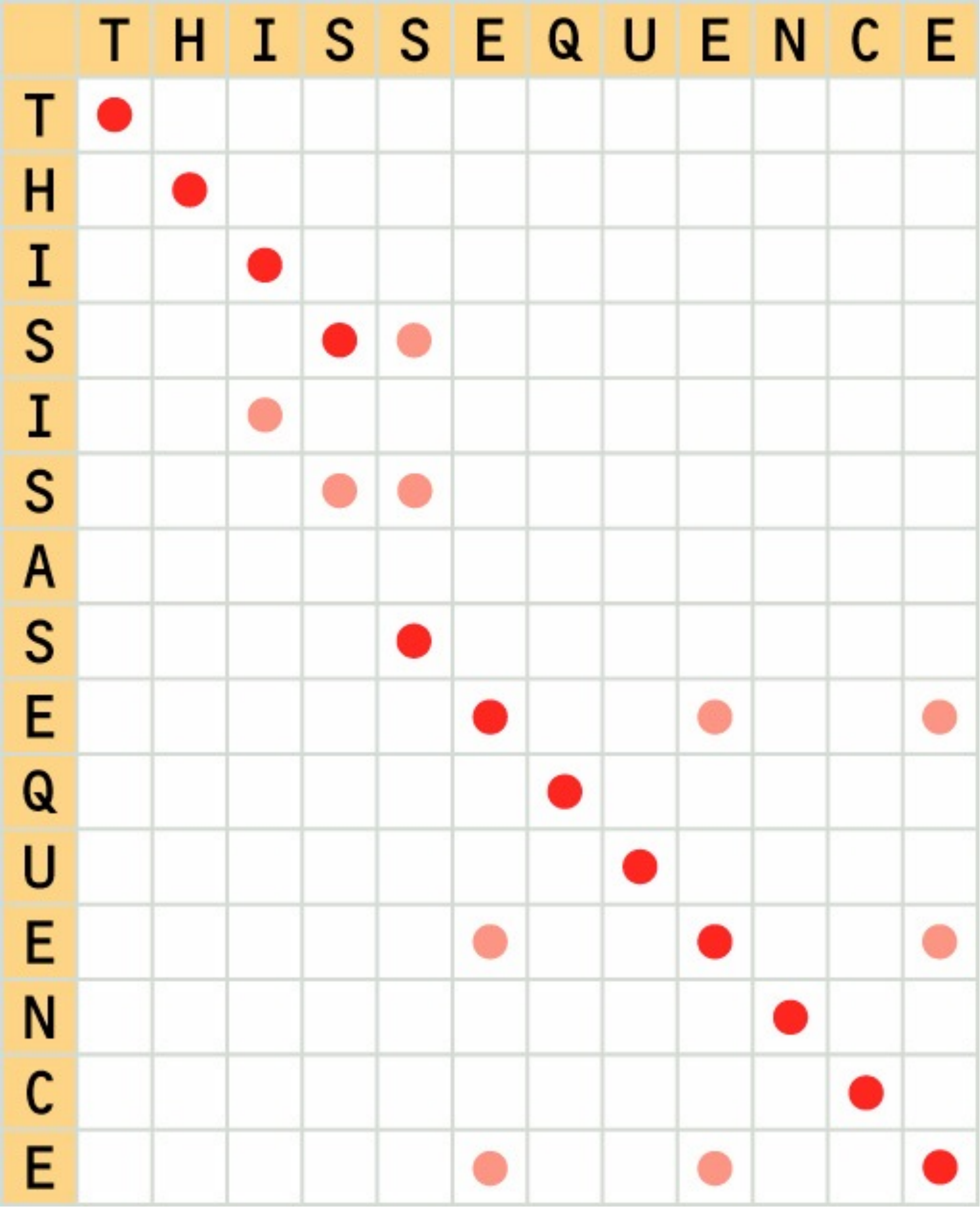
An alignment matrix



	D	O	R	O	T	H	Y	C	R	O	W	F	O	O	T	H	O	D	G	K	I	N
D	D																	D				
O		O		O						O			O	O			O					
R			R						R													
O		O		O						O			O	O			O					
T					T										T							
H						H										H						
Y							Y															
H						H		→	→	→	→	→	→	→	→	H						
O		O		O						O			O	O			O					
D	D																	D				
G																			G			
K																				K		
I																					I	
N																						N

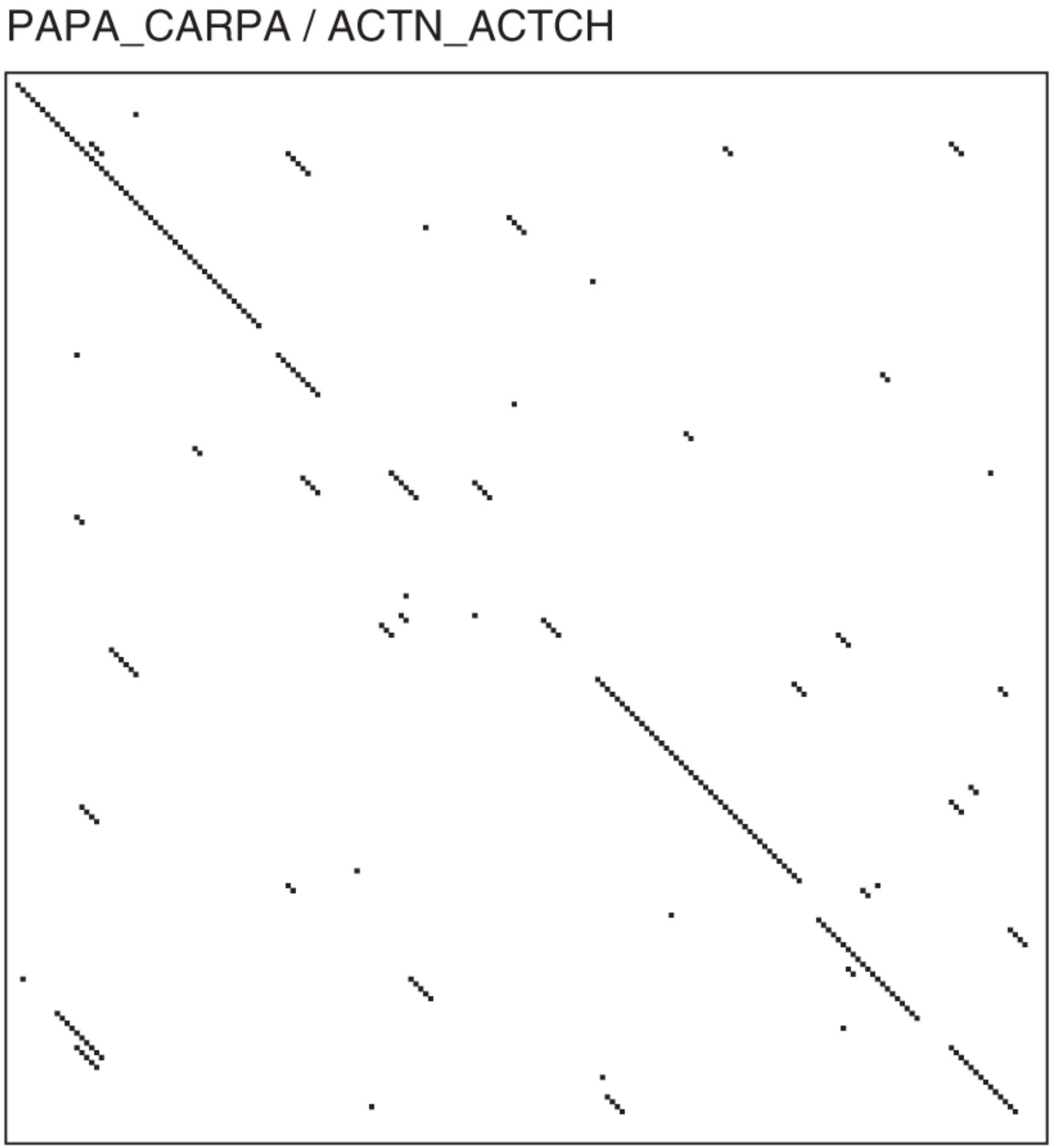
Dotplots

Dotplots

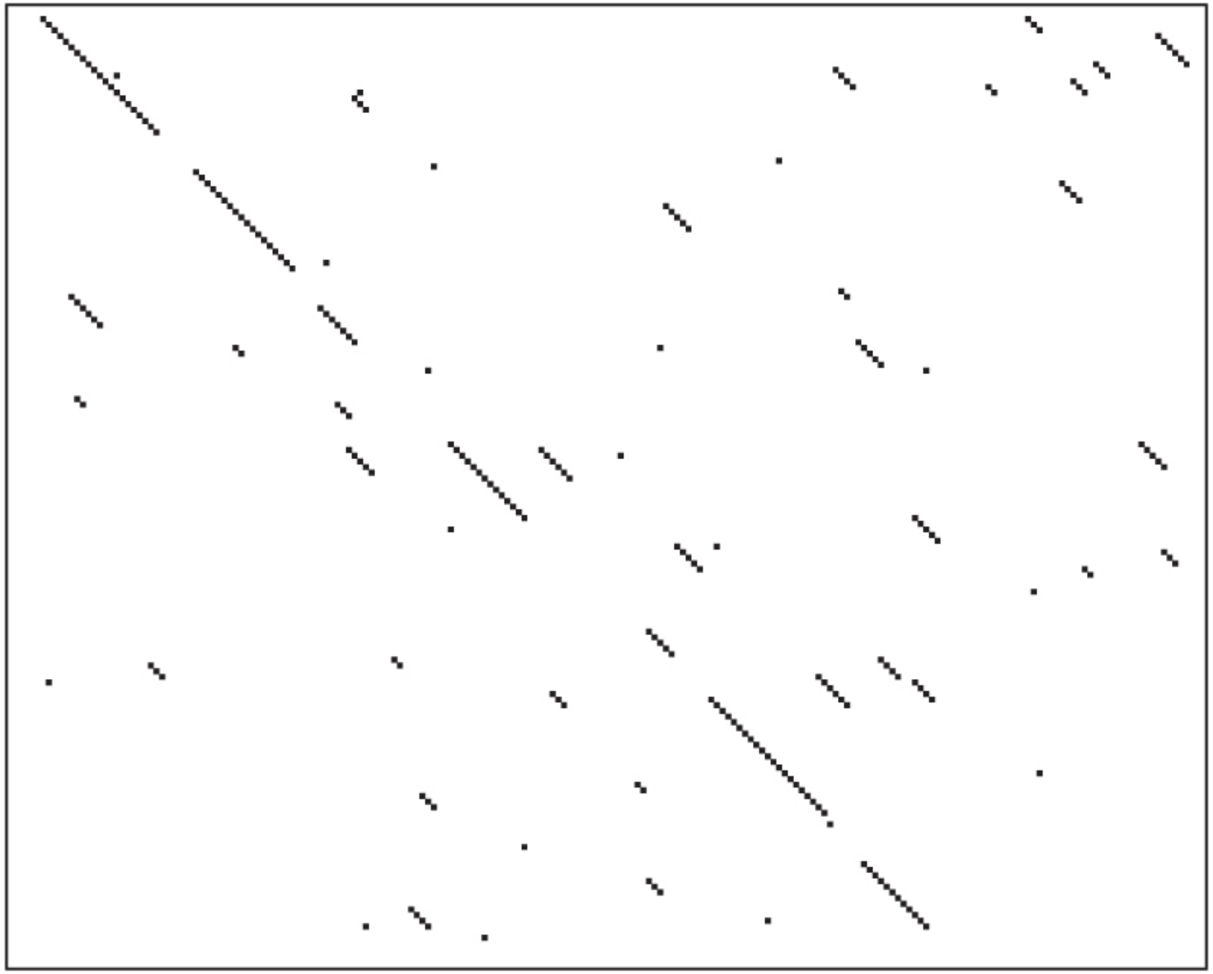


	T	H	I	S	S	E	Q	U	E	N	C	E
T	●											
H		●										
I			●									
S				●	○							
I			○									
S				○	○							
A												
S					●							
E						●			○			○
Q							●					
U								●				
E						○			●			○
N										●		
C											●	
E						○			○			●

Dotplots

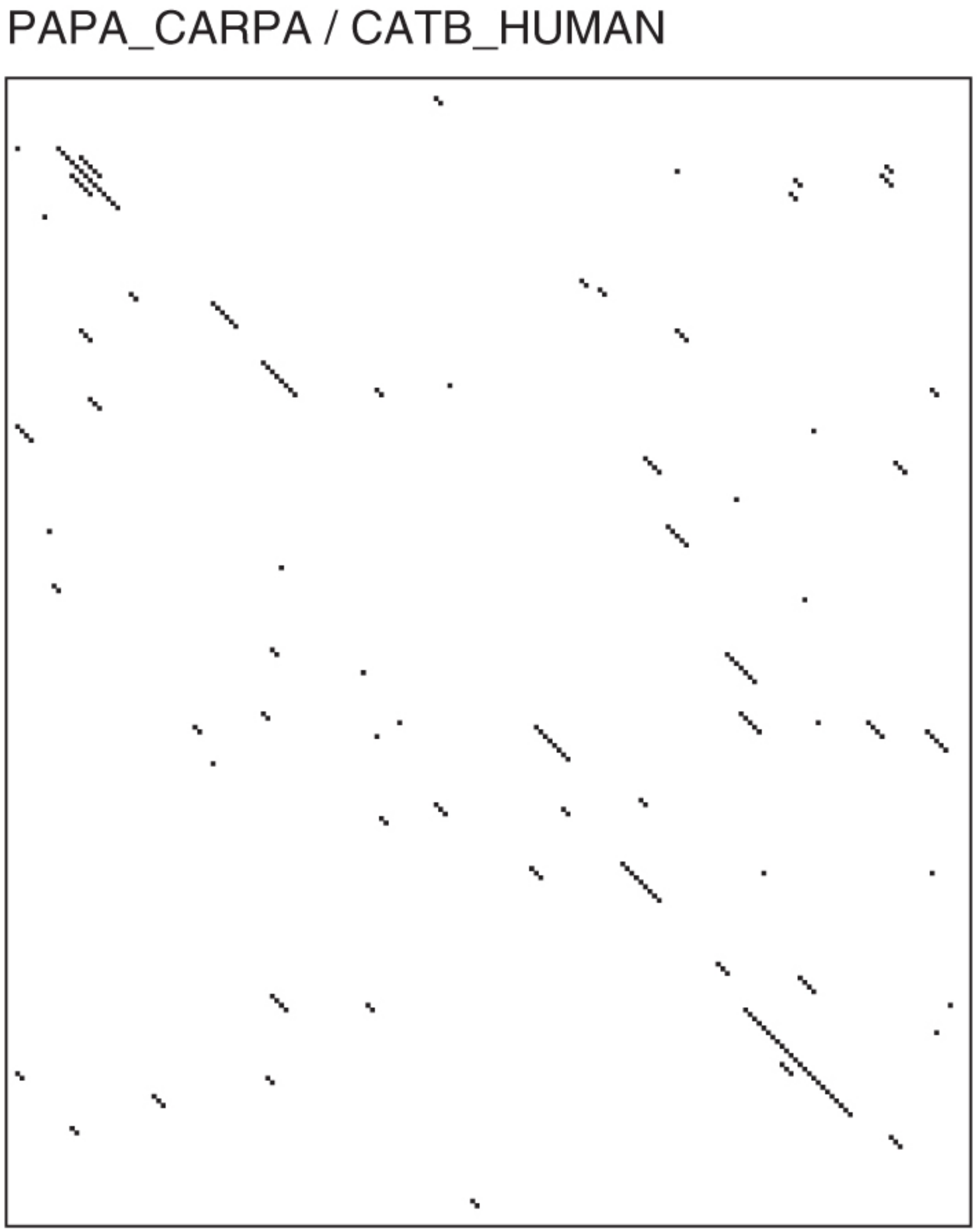


Dotplots

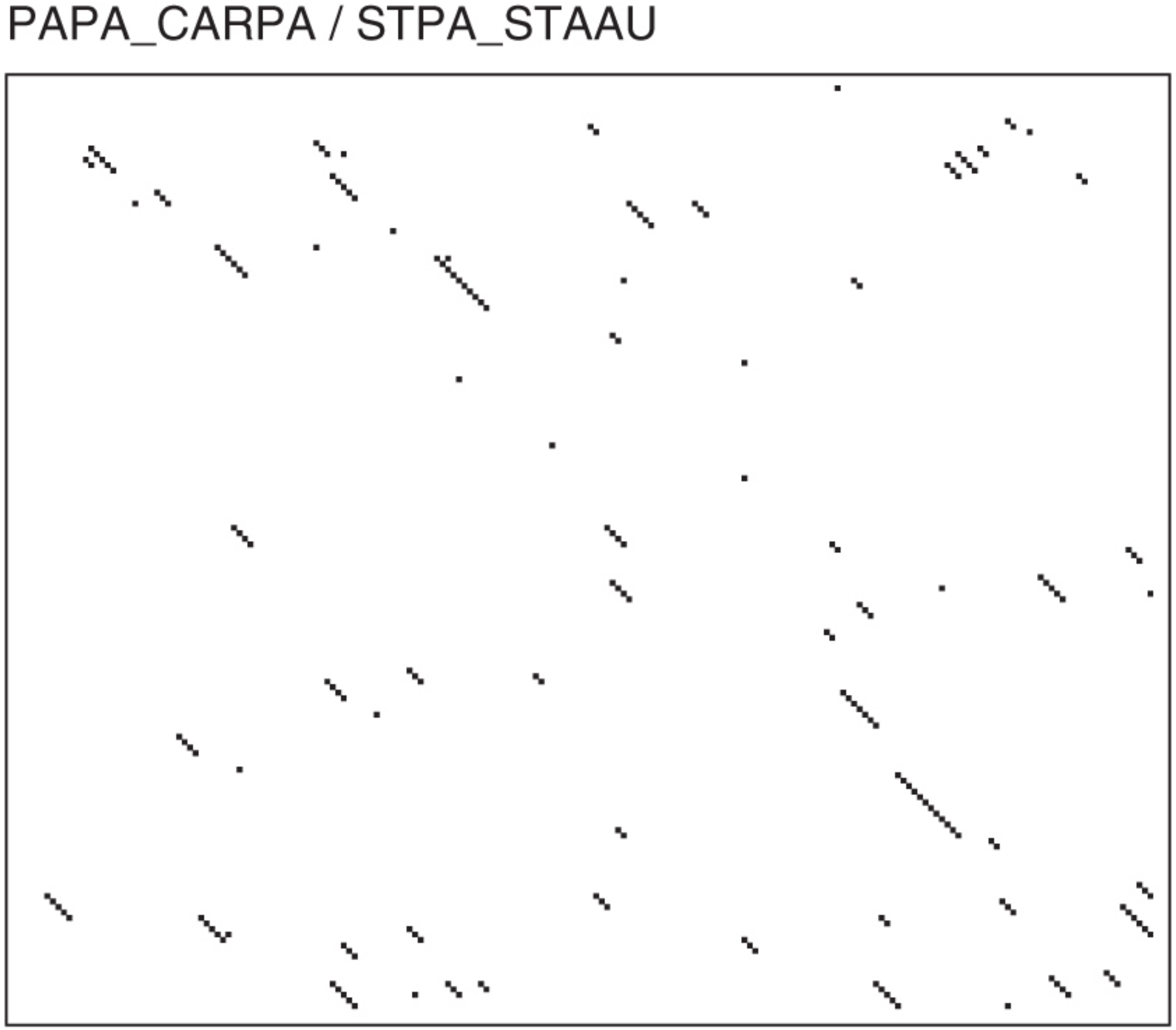


PAPA_CARPA / CATL_HUMAN

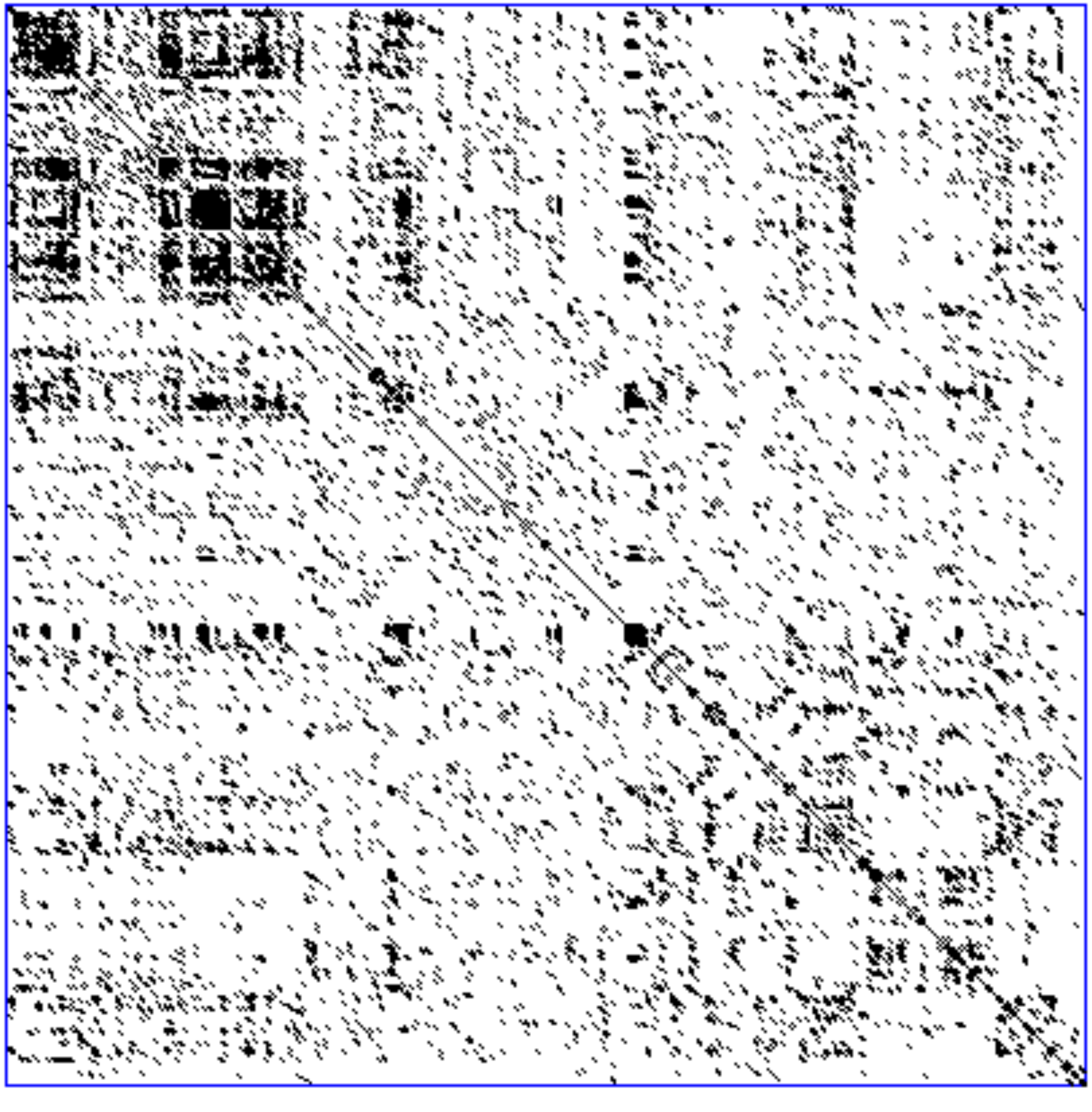
Dotplots



Dotplots

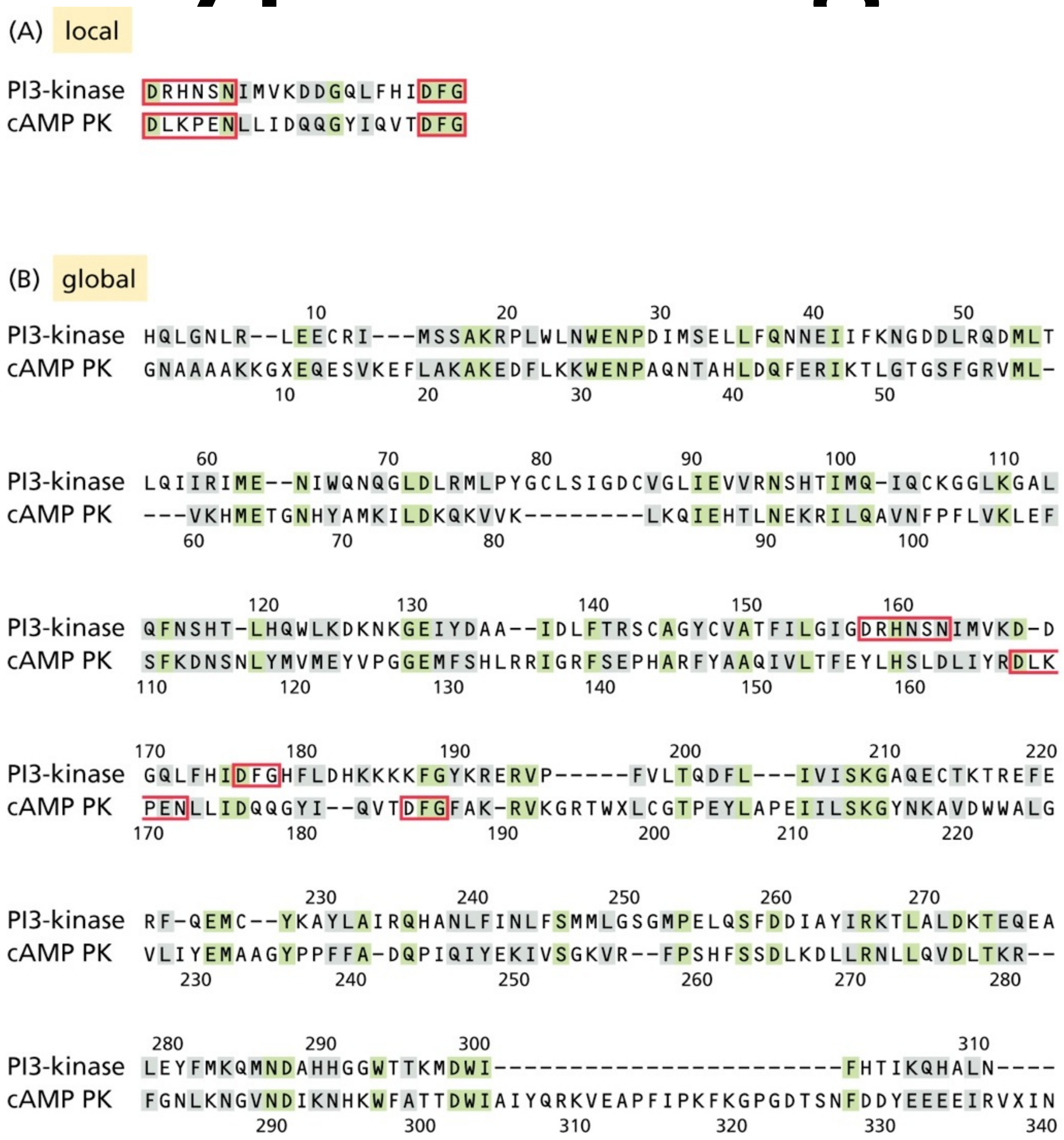


Dotplots



Types of alignment

Types of alignment



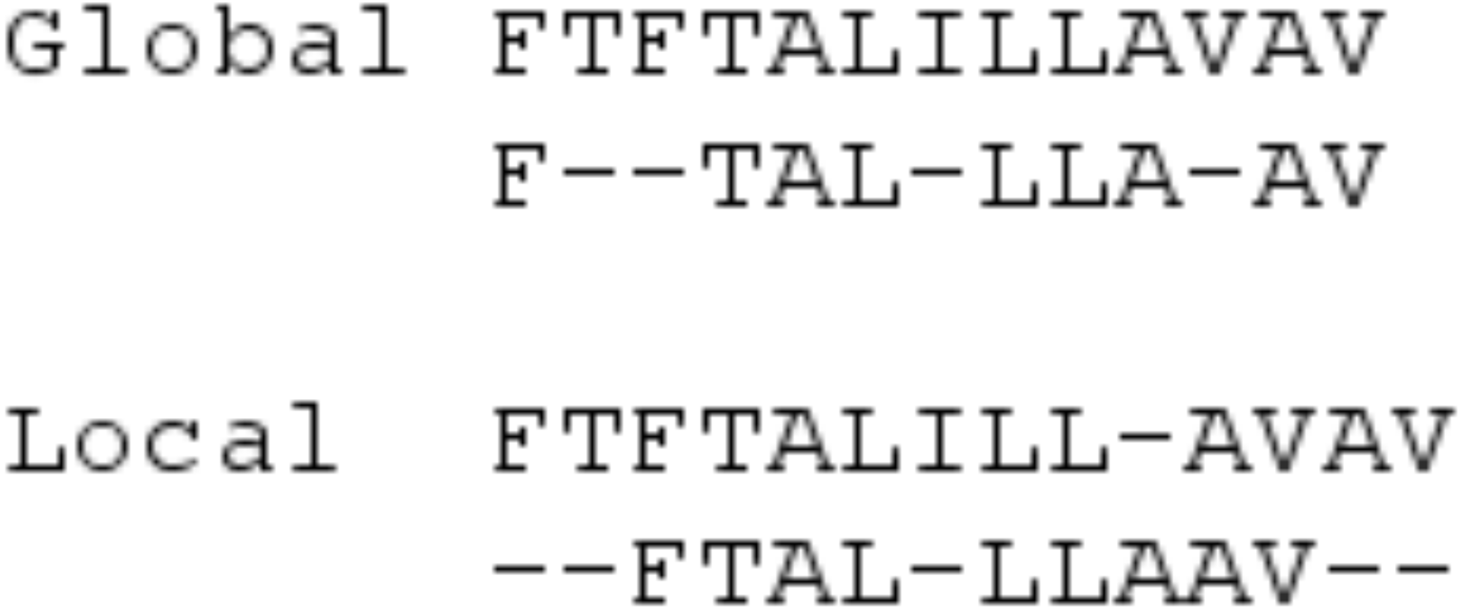
(A) local

```
PI3-kinase  DRHNSNIMVKDDGQLFHIDFG
cAMP PK     DLKPENLLIDQQGYIQVTDFG
```

(B) global

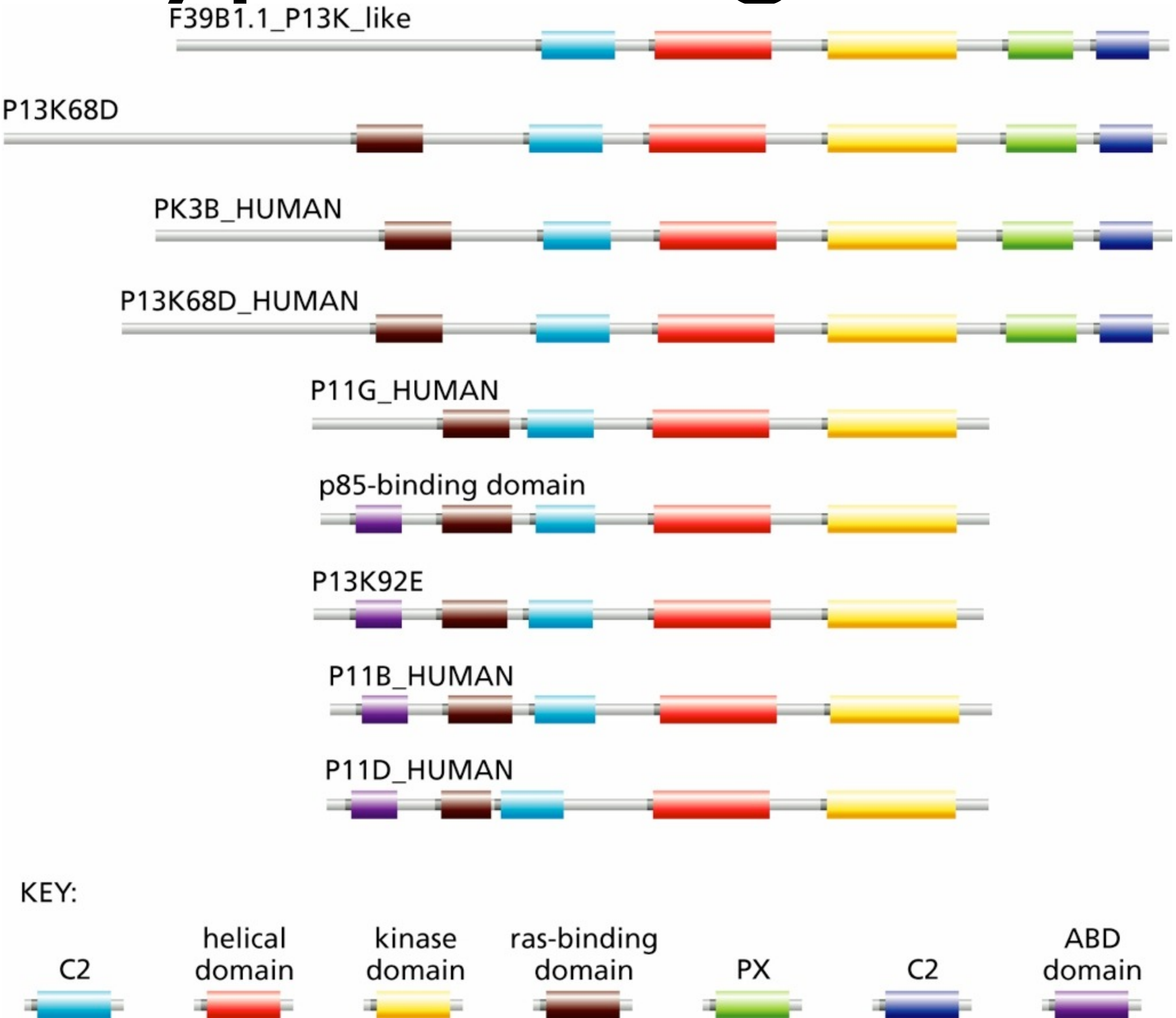
```
                      10           20        30        40        50
PI3-kinase  HQLGNLR--LEECRI---MSSAKRPLWLNWENPDIMSELLFQNNEIIFKNGDDLRQDMLT
cAMP PK     GNAAAAKKGXEQESVKEFLAKAKEDFLKKWENPAQNTAHLDQFERIKTLGTGSFGRVML-
                    10        20        30        40        50

               60          70        80        90       100        110
PI3-kinase  LQIIRIME--NIWQNQGLDLRMLPYGCLSIGDCVGLIEVVRNSHTIMQ-IQCKGGLKGAL
cAMP PK     ---VKHMETGNHYAMKILDKQKVVK--------LKQIEHTLNEKRILQAVNFPFLVKLEF
              60        70        80                90       100

                  120       130         140       150       160
PI3-kinase  QFNSHT-LHQWLKDKNKGEIYDAA--IDLFTRSCAGYCVATFILGIGDRHNSNIMVKD-D
cAMP PK     SFKDNSNLYMVMEYVPGGEMFSHLRRIGRFSEPHARFYAAQIVLTFEYLHSLDLIYRDLK
            110     120       130       140        150       160

            170      180       190            200          210       220
PI3-kinase  GQLFHIDFGHFLDHKKKKFGYKRERVP-----FVLTQDFL---IVISKGAQECTKTREFE
cAMP PK     PENLLIDQQGYI--QVTDFGFAK-RVKGRTWXLCGTPEYLAPEIILSKGYNKAVDWWALG
            170      180          190       200      210        220

                      230       240       250       260       270
PI3-kinase  RF-QEMC--YKAYLAIRQHANLFINLFSMMLGSGMPELQSFDDIAYIRKTLALDKTEQEA
cAMP PK     VLIYEMAAGYPPFFA-DQPIQIYEKIVSGKVR--FPSHFSSDLKDLLRNLLQVDLTKR--
              230       240        250         260       270       280

            280       290       300                              310
PI3-kinase  LEYFMKQMNDAHHGGWTTKMDWI-----------------------FHTIKQHALN----
cAMP PK     FGNLKNGVNDIKNHKWFATTDWIAIYQRKVEAPFIPKFKGPGDTSNFDDYEEEEIRVXIN
                   290       300       310       320       330       340
```

Types of alignment

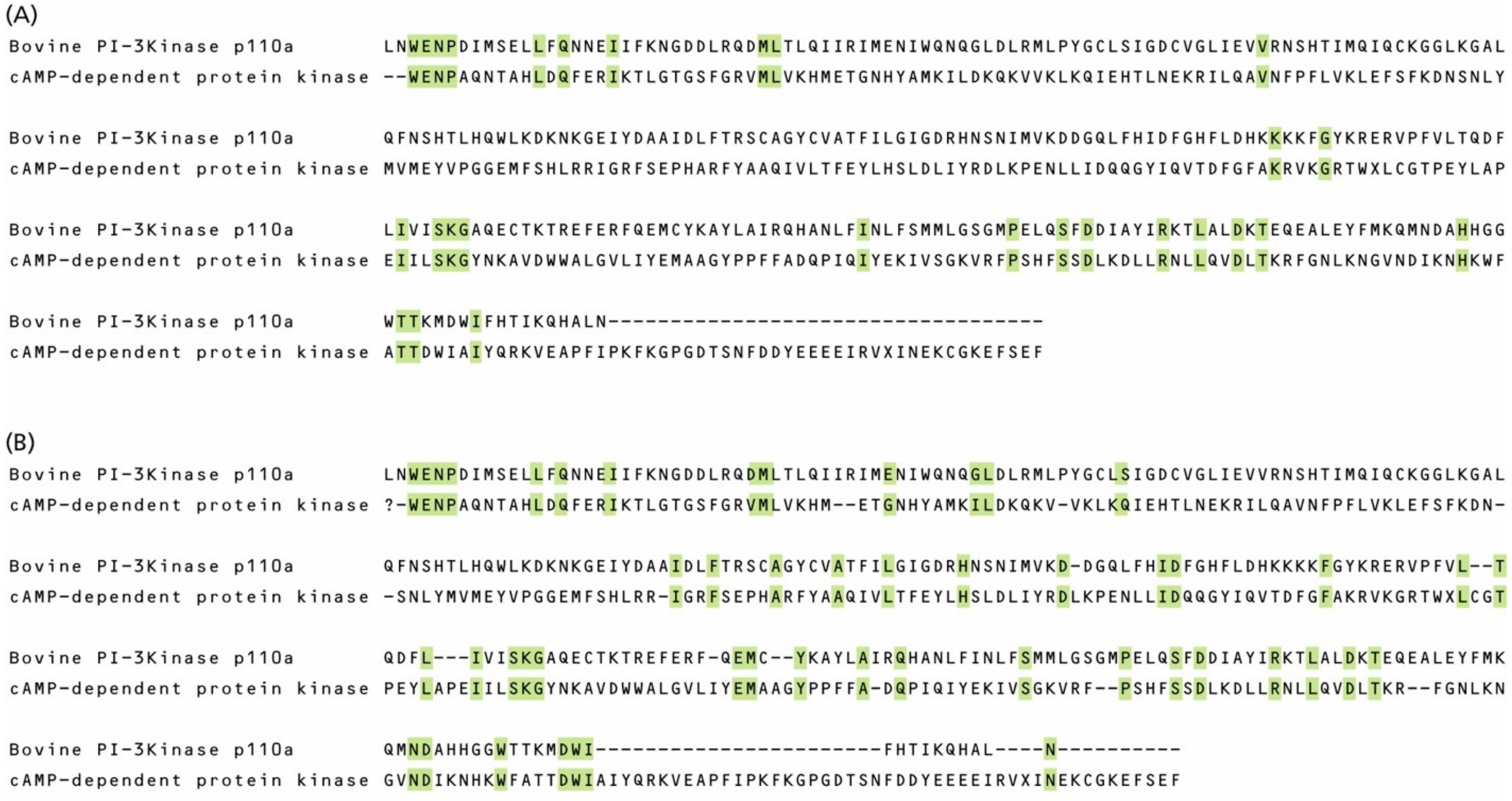


```
Global FTFTALILLAVAV
       F--TAL-LLA-AV

Local  FTFTALILL-AVAV
       --FTAL-LLAAV--
```

Types of alignment



Inserting gaps



(A)

```
Bovine PI-3Kinase p110a        LNWENPDIMSELLFQNNEIIFKNGDDLRQDMLTLQIIRIMENIWQNQGLDLRMLPYGCLSIGDCVGLIEVVRNSHTIMQIQCKGGLKGAL
cAMP-dependent protein kinase  --WENPAQNTAHLDQFERIKTLGTGSFGRVMLVKHMETGNHYAMKILDKQKVVKLKQIEHTLNEKRILQAVNFPFLVKLEFSFKDNSNLY

Bovine PI-3Kinase p110a        QFNSHTLHQWLKDKNKGEIYDAAIDLFTRSCAGYCVATFILGIGDRHNSNIMVKDDGQLFHIDFGHFLDHKKKKFGYKRERVPFVLTQDF
cAMP-dependent protein kinase  MVMEYVPGGEMFSHLRRIGRFSEPHARFYAAQIVLTFEYLHSLDLIYRDLKPENLLIDQQGYIQVTDFGFAKRVKGRTWXLCGTPEYLAP

Bovine PI-3Kinase p110a        LIVISKGAQECTKTREFERFQEMCYKAYLAIRQHANLFINLFSMMLGSGMPELQSFDDIAYIRKTLALDKTEQEALEYFMKQMNDAHHGG
cAMP-dependent protein kinase  EIILSKGYNKAVDWWALGVLIYEMAAGYPPFFADQPIQIYEKIVSGKVRFPSHFSSDLKDLLRNLLQVDLTKRFGNLKNGVNDIKNHKWF

Bovine PI-3Kinase p110a        WTTKMDWIFHTIKQHALN-----------------------------------
cAMP-dependent protein kinase  ATTDWIAIYQRKVEAPFIPKFKGPGDTSNFDDYEEEEIRVXINEKCGKEFSEF
```

(B)

```
Bovine PI-3Kinase p110a        LNWENPDIMSELLFQNNEIIFKNGDDLRQDMLTLQIIRIMENIWQNQGLDLRMLPYGCLSIGDCVGLIEVVRNSHTIMQIQCKGGLKGAL
cAMP-dependent protein kinase  ?-WENPAQNTAHLDQFERIKTLGTGSFGRVMLVKHM--ETGNHYAMKILDKQKV-VKLKQIEHTLNEKRILQAVNFPFLVKLEFSFKDN-

Bovine PI-3Kinase p110a        QFNSHTLHQWLKDKNKGEIYDAAIDLFTRSCAGYCVATFILGIGDRHNSNIMVKD-DGQLFHIDFGHFLDHKKKKFGYKRERVPFVL--T
cAMP-dependent protein kinase  -SNLYMVMEYVPGGEMFSHLRR-IGRFSEPHARFYAAQIVLTFEYLHSLDLIYRDLKPENLLIDQQGYIQVTDFGFAKRVKGRTWXLCGT

Bovine PI-3Kinase p110a        QDFL---IVISKGAQECTKTREFERF-QEMC--YKAYLAIRQHANLFINLFSMMLGSGMPELQSFDDIAYIRKTLALDKTEQEALEYFMK
cAMP-dependent protein kinase  PEYLAPEIILSKGYNKAVDWWALGVLIYEMAAGYPPFFA-DQPIQIYEKIVSGKVRF--PSHFSSDLKDLLRNLLQVDLTKR--FGNLKN

Bovine PI-3Kinase p110a        QMNDAHHGGWTTKMDWI-----------------------FHTIKQHAL----N----------
cAMP-dependent protein kinase  GVNDIKNHKWFATTDWIAIYQRKVEAPFIPKFKGPGDTSNFDDYEEEEIRVXINEKCGKEFSEF
```

What is an optimal alignment ?

```
T H I S S E Q U E N C E
| |     | | | | | | | |       10/12 Identical
T H A T S E Q U E N C E

T H A T S E Q U E N C E
| |             |     |       4/12 Identical
T H I S I S A S E Q U E N C E

T H I S I S A - S E Q U E N C E
| |         |   | | | | | | | |    11/12 Identical
T H - - - - A T S E Q U E N C E
```

Different scoring

```
T H I S S E Q U E N C E
5 8-1 1 4 5 6 0 5 6 9 5         Score = 52
T H A T S E Q U E N C E
```

```
T H A T S E Q U E N C E
5 8-1-1-2 0-1 0 5 0 0 5         Score = 18
T H I S I S A S E Q U E N C E
```

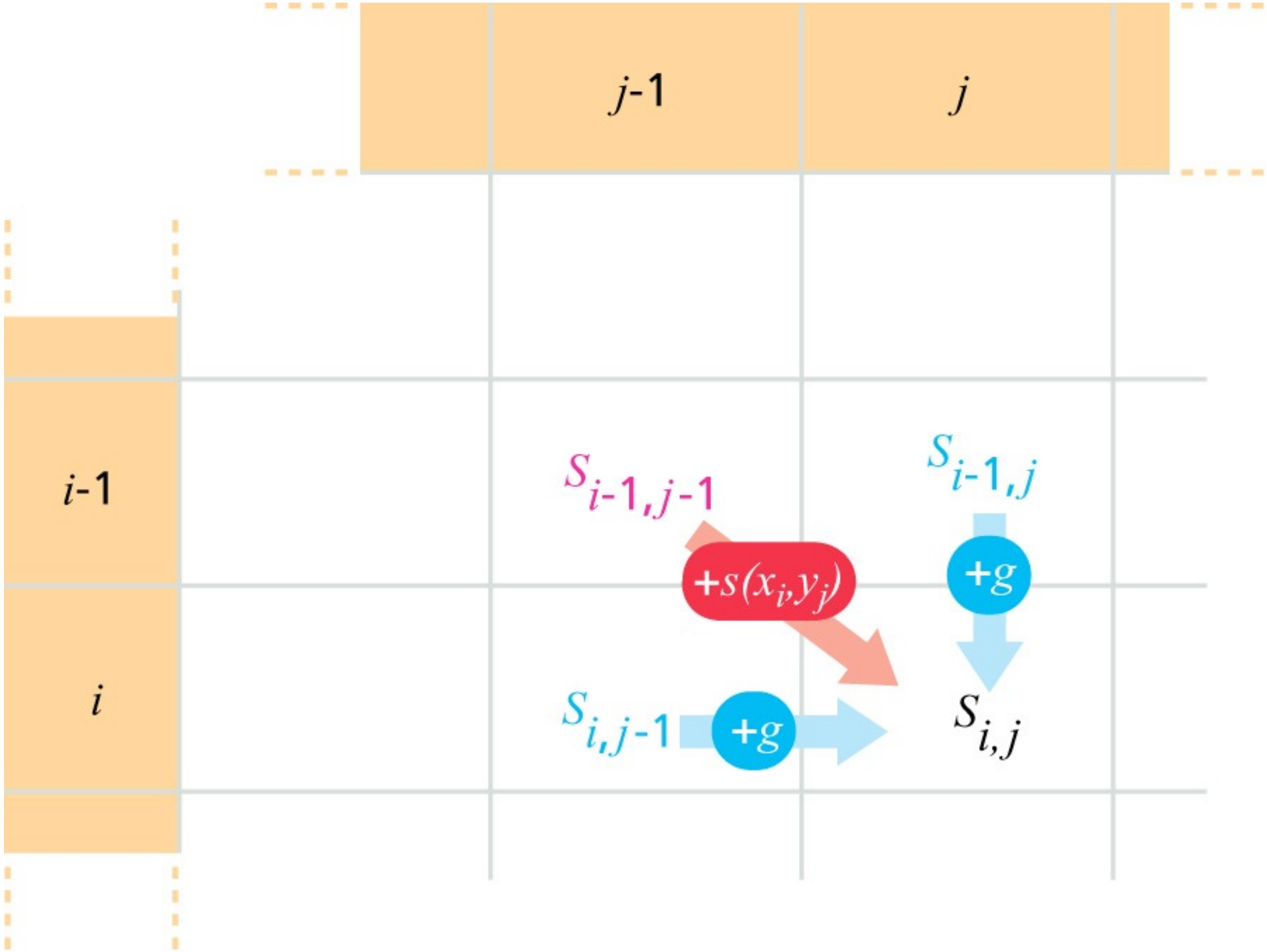
```
T H I S I S A - S E Q U E N C E
5 8 0 0 0 0 4 0 4 5 6 0 5 6 9 5         Score = 56
T H - - - - A T S E Q U E N C E
```

With Gap cost

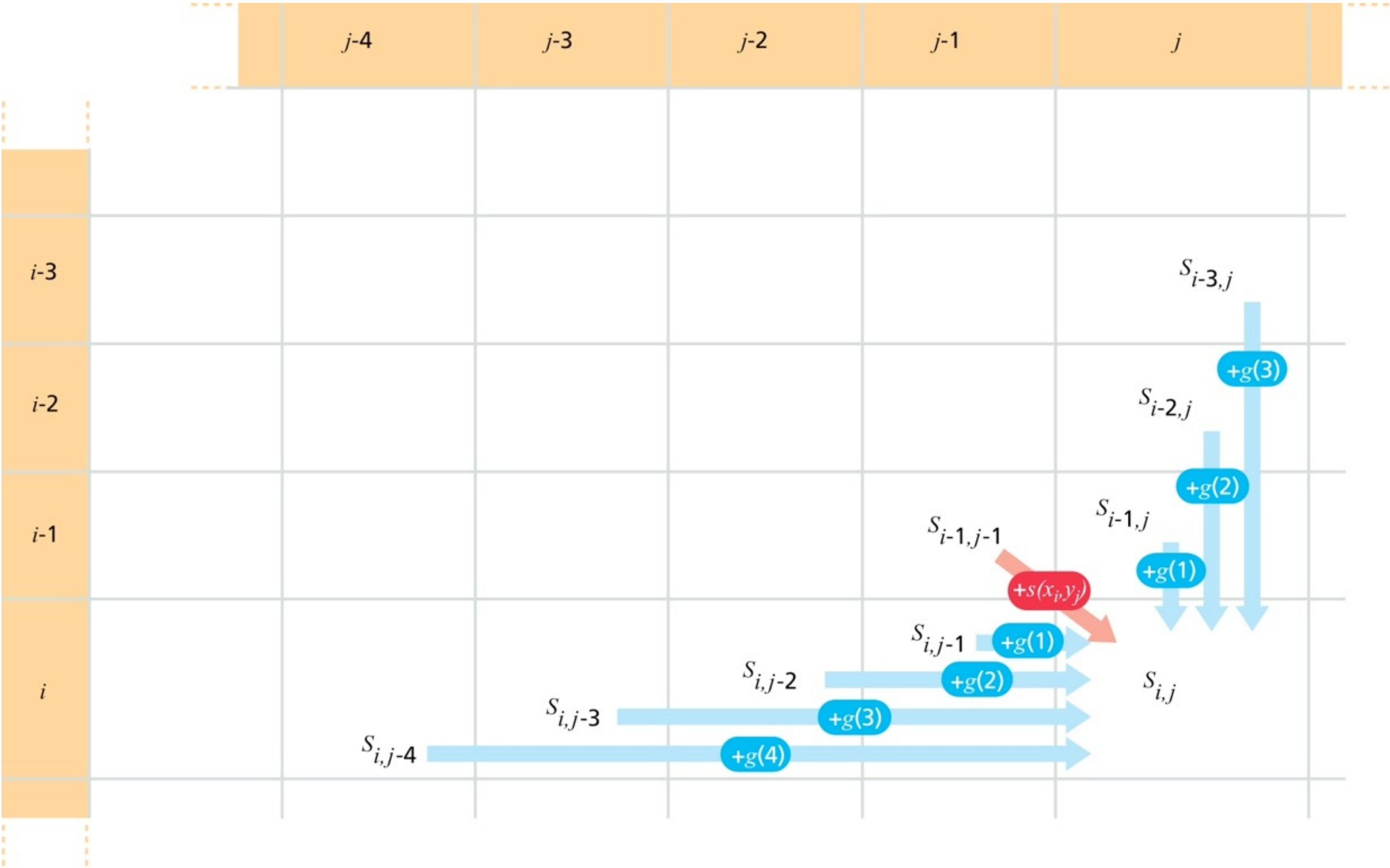
```
T H I S S E Q U E N C E
5 8-1 1 4 5 6 0 5 6 9 5      Score = 52
T H A T S E Q U E N C E

T H A T S E Q U E N C E
5 8-1-1-2 0-1 0 5 0 0 5      Score = 18
T H I S I S A S E Q U E N C E

T H I S I S A - S E Q U E N C E
5 8-1-1-1-1 4-1 4 5 6 0 5 6 9 5      Score = 51
T H - - - - A T S E Q U E N C E
```

Dynamic programming

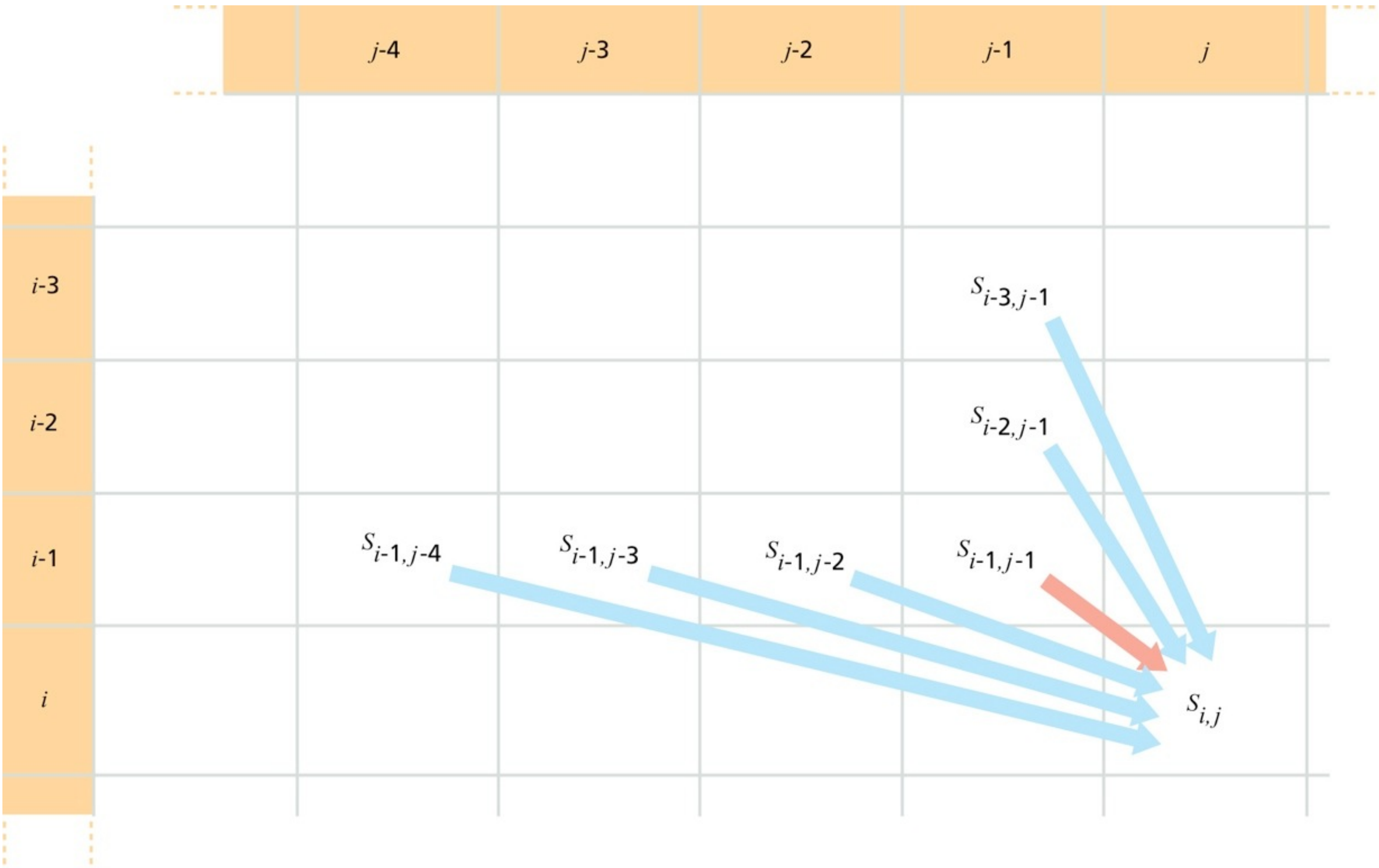
Dynamic programming



Dynamic programming

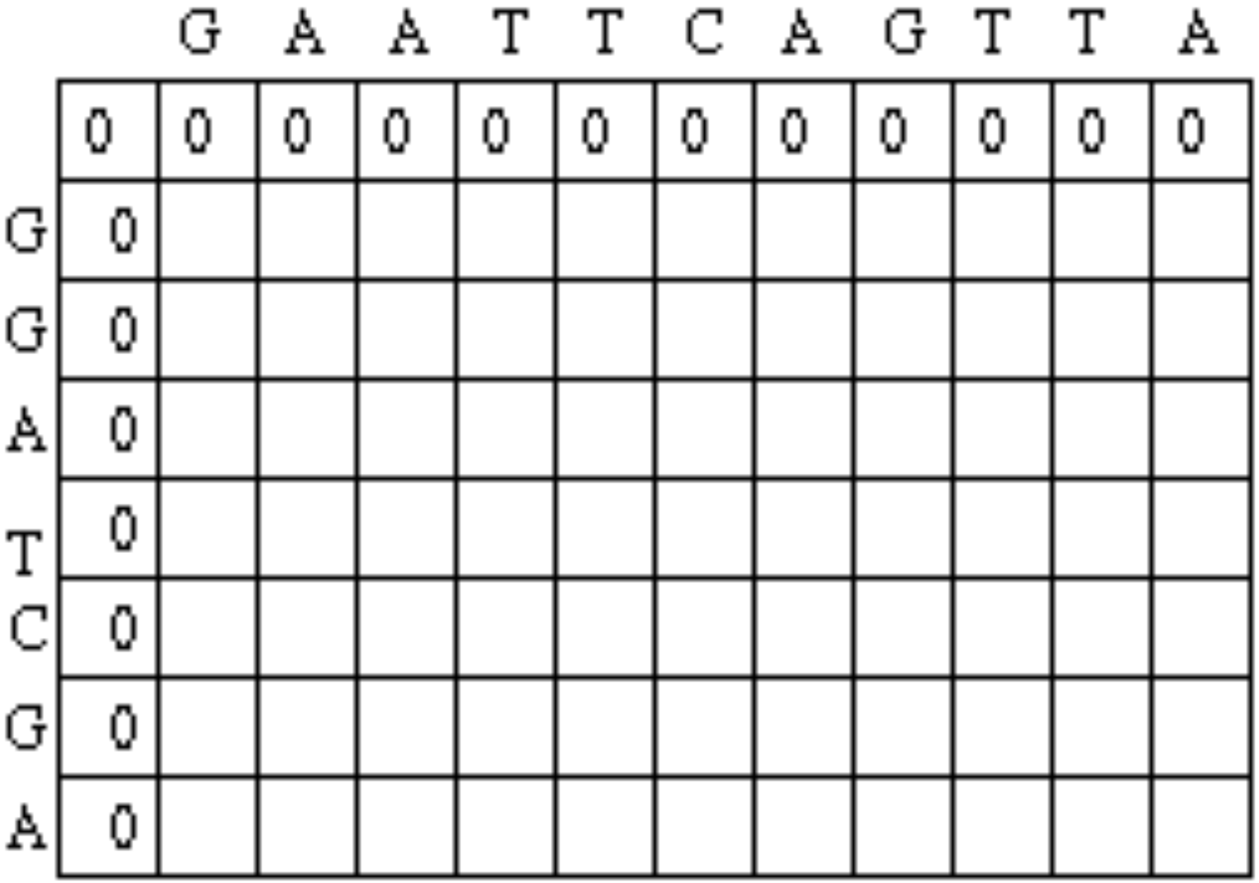


Dynamic programming



j-4
j-3
j-2
j-1
j
i-3
i-2
i-1
i
$s_{i\text{-}3,j\text{-}1}$
$s_{i\text{-}2,j\text{-}1}$
$s_{i\text{-}1,j\text{-}4}$
$s_{i\text{-}1,j\text{-}3}$
$s_{i\text{-}1,j\text{-}2}$
$s_{i\text{-}1,j\text{-}1}$
$s_{i,j}$

Initialisation step: Create Matrix with M + 1 columns and N + 1 rows. First row and column filled with 0.



		G	A	A	T	T	C	A	G	T	T	A
	0	0	0	0	0	0	0	0	0	0	0	0
G	0											
G	0											
A	0											
T	0											
C	0											
G	0											
A	0											

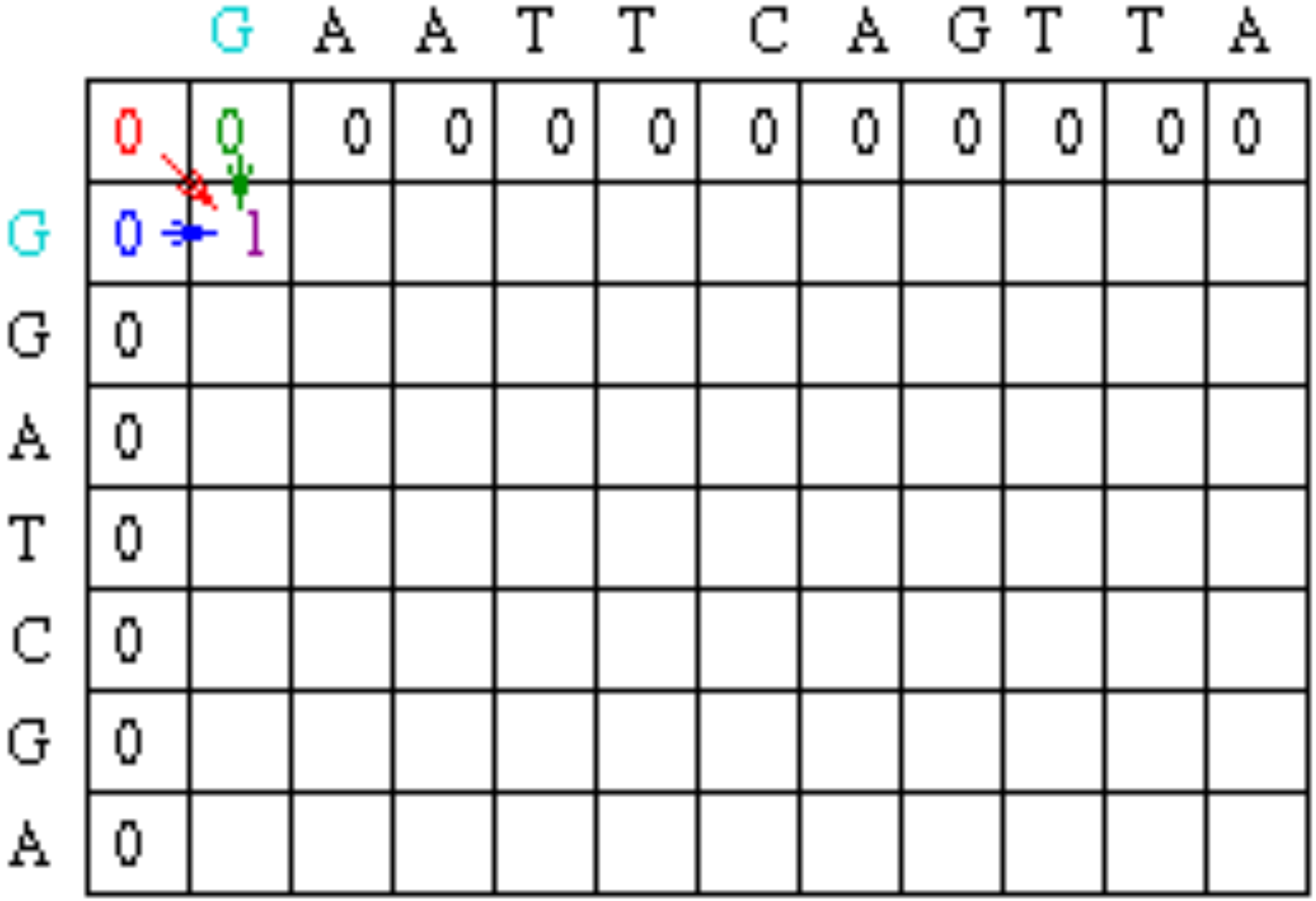
Matrix fill step: Each position $M_{i,j}$ is defined to be the MAXIMUM score at position i,j

$M_{i,j}$ = MAXIMUM [

$M_{i-1,\ j-1}$ + $s_{i,,j}$ (match or mismatch in the diagonal)

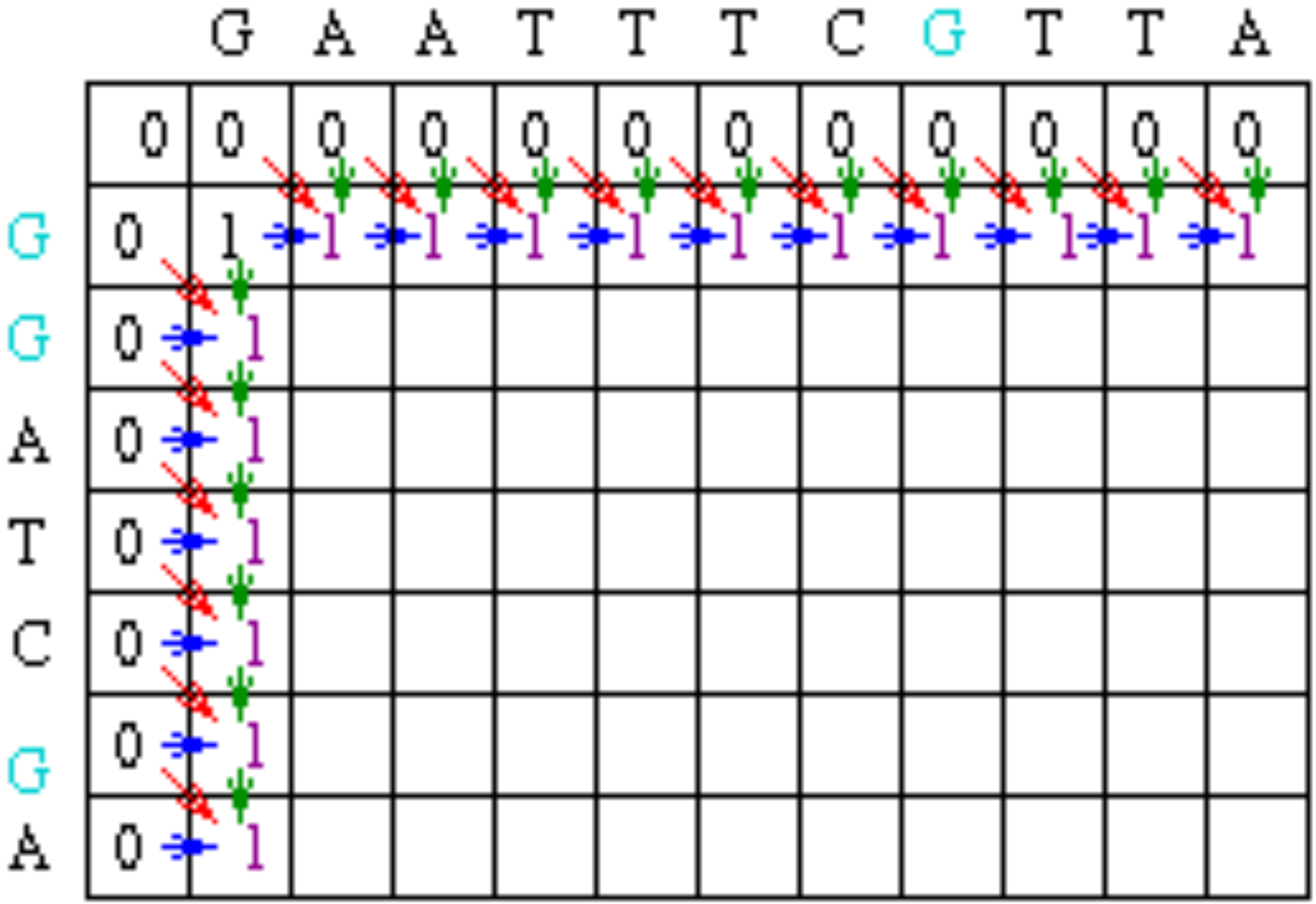
$M_{i,\ j-1}$ + w (gap in sequence #1)

$M_{i-1,\ j}$ + w (gap in sequence #2)]



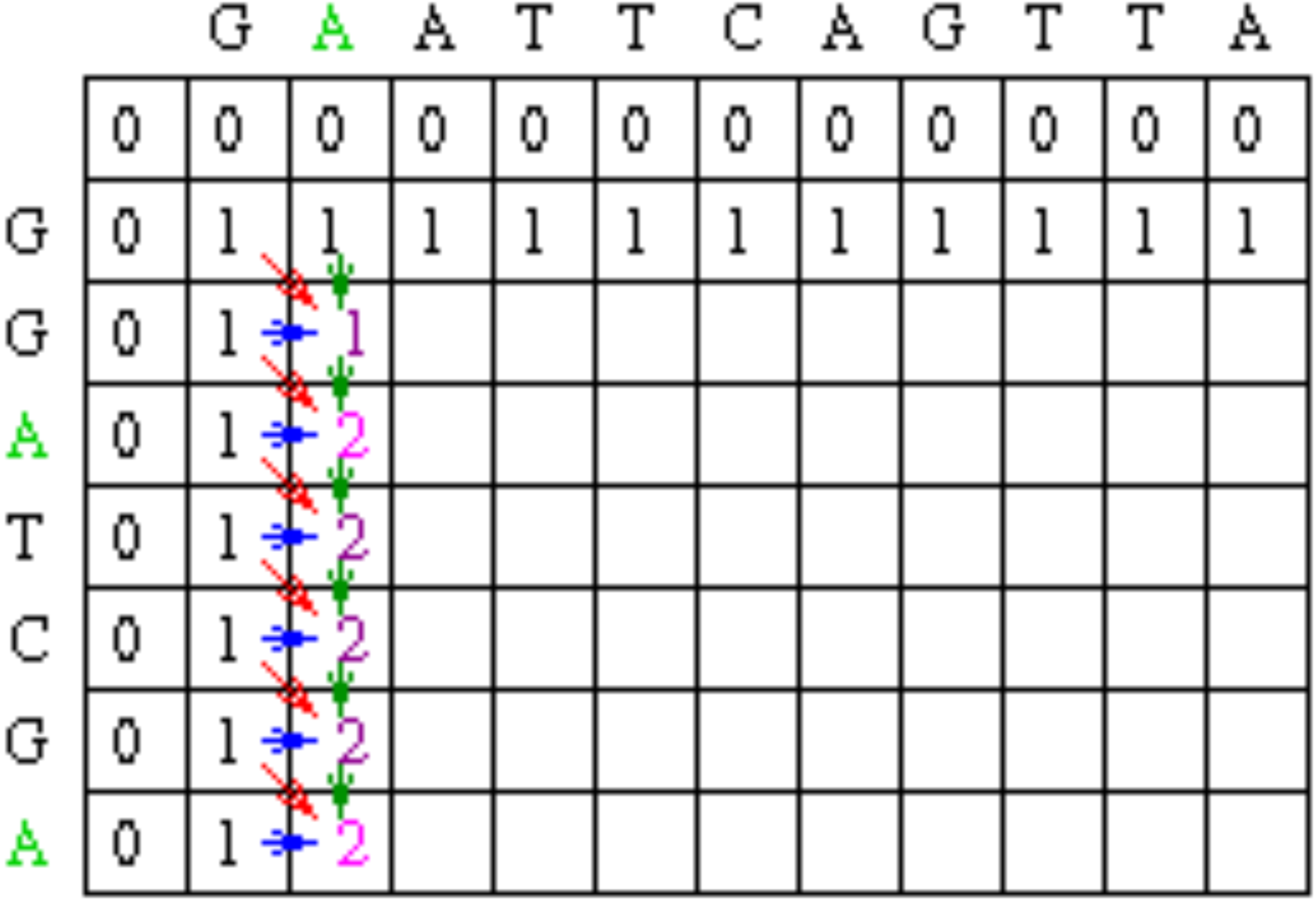
		G	A	A	T	T	C	A	G	T	T	A
	0	0	0	0	0	0	0	0	0	0	0	0
G	0	1										
G	0											
A	0											
T	0											
C	0											
G	0											
A	0											

Fill in rest of row 1 and column 1



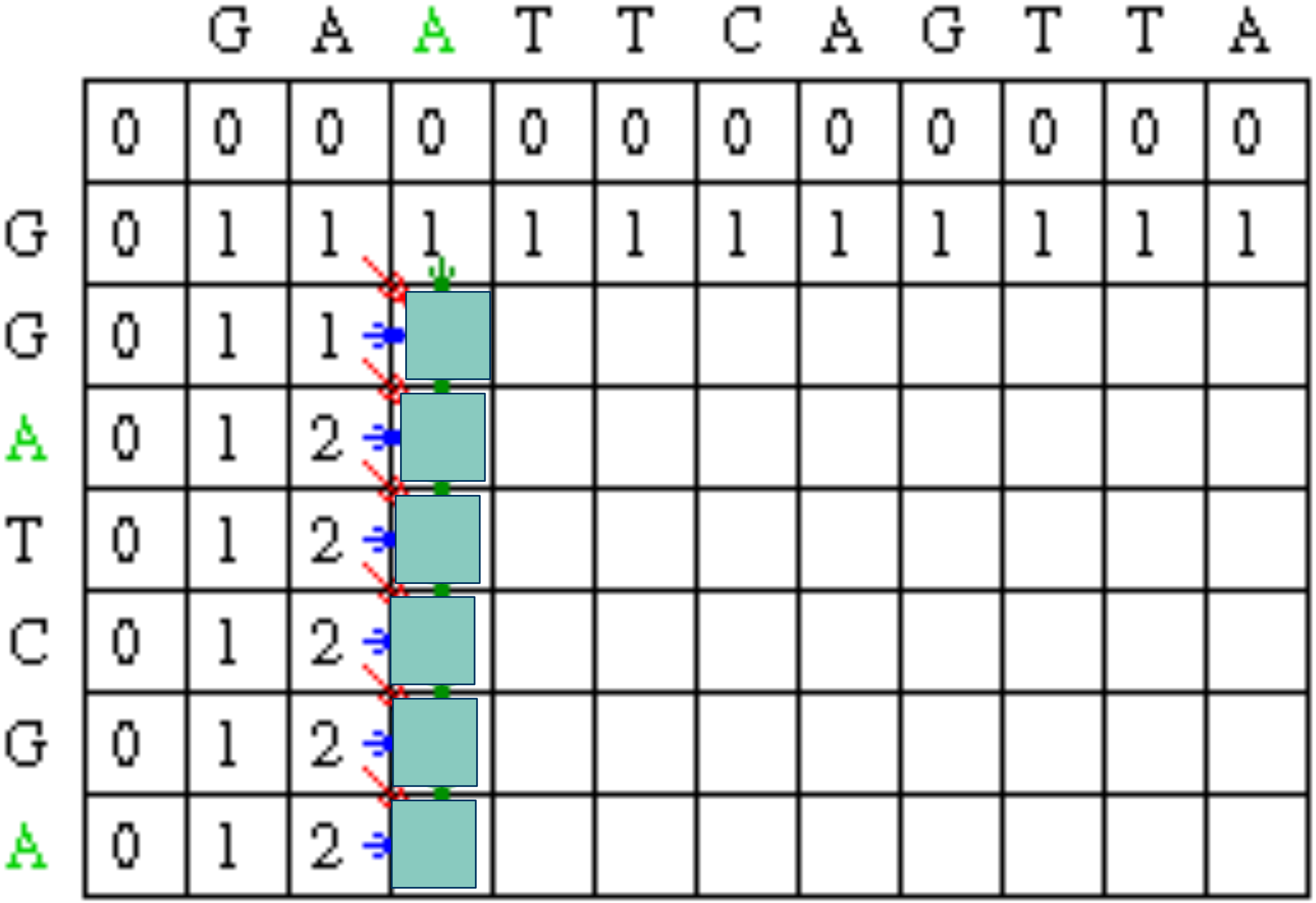
		G	A	A	T	T	T	C	G	T	T	A
	0	0	0	0	0	0	0	0	0	0	0	0
G	0	1	1	1	1	1	1	1	1	1	1	1
G	0	1										
A	0	1										
T	0	1										
C	0	1										
G	0	1										
A	0	1										

Fill in column 2



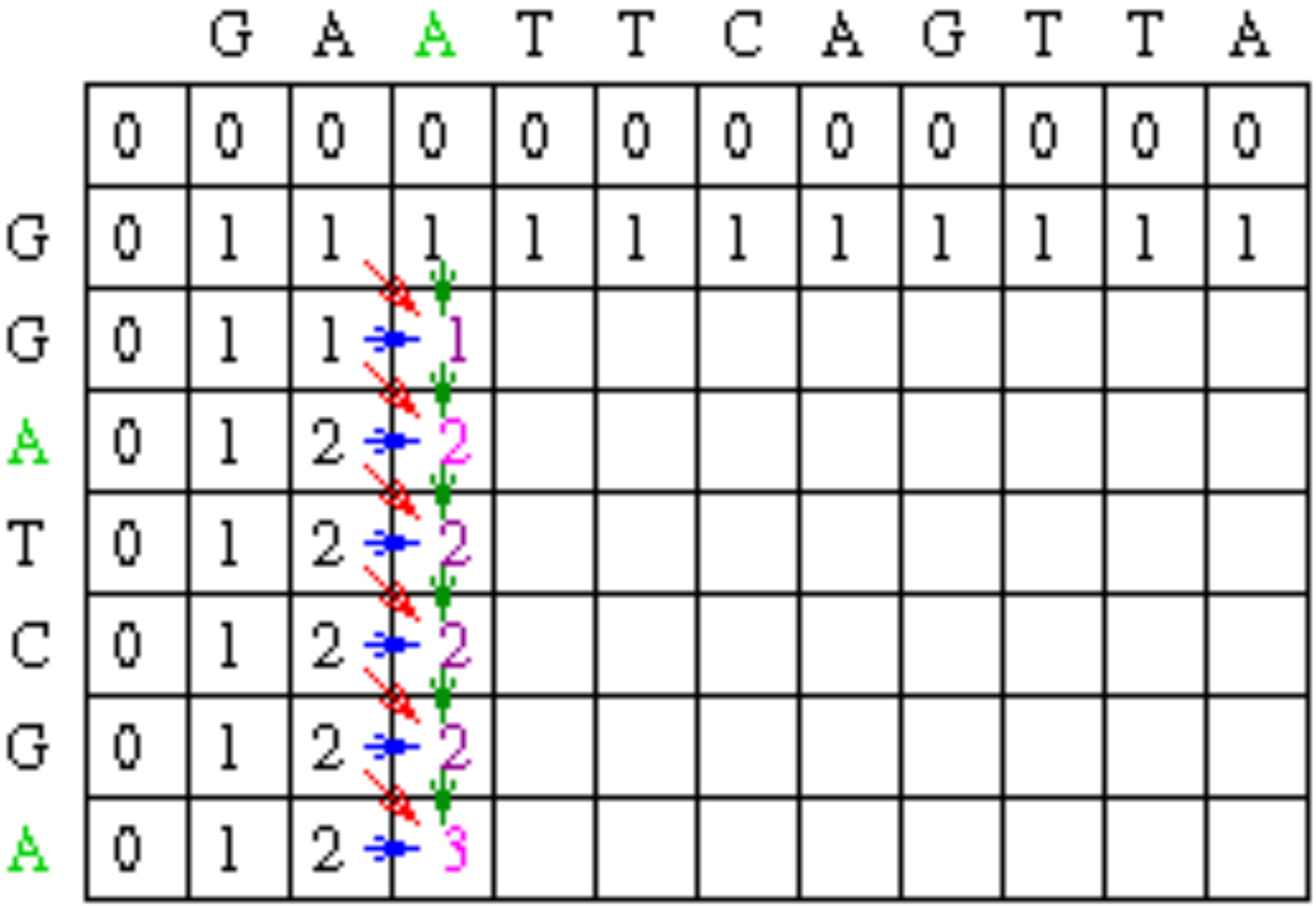
		G	A	A	T	T	C	A	G	T	T	A
	0	0	0	0	0	0	0	0	0	0	0	0
G	0	1	1	1	1	1	1	1	1	1	1	1
G	0	1	1									
A	0	1	2									
T	0	1	2									
C	0	1	2									
G	0	1	2									
A	0	1	2									

Fill in column 3



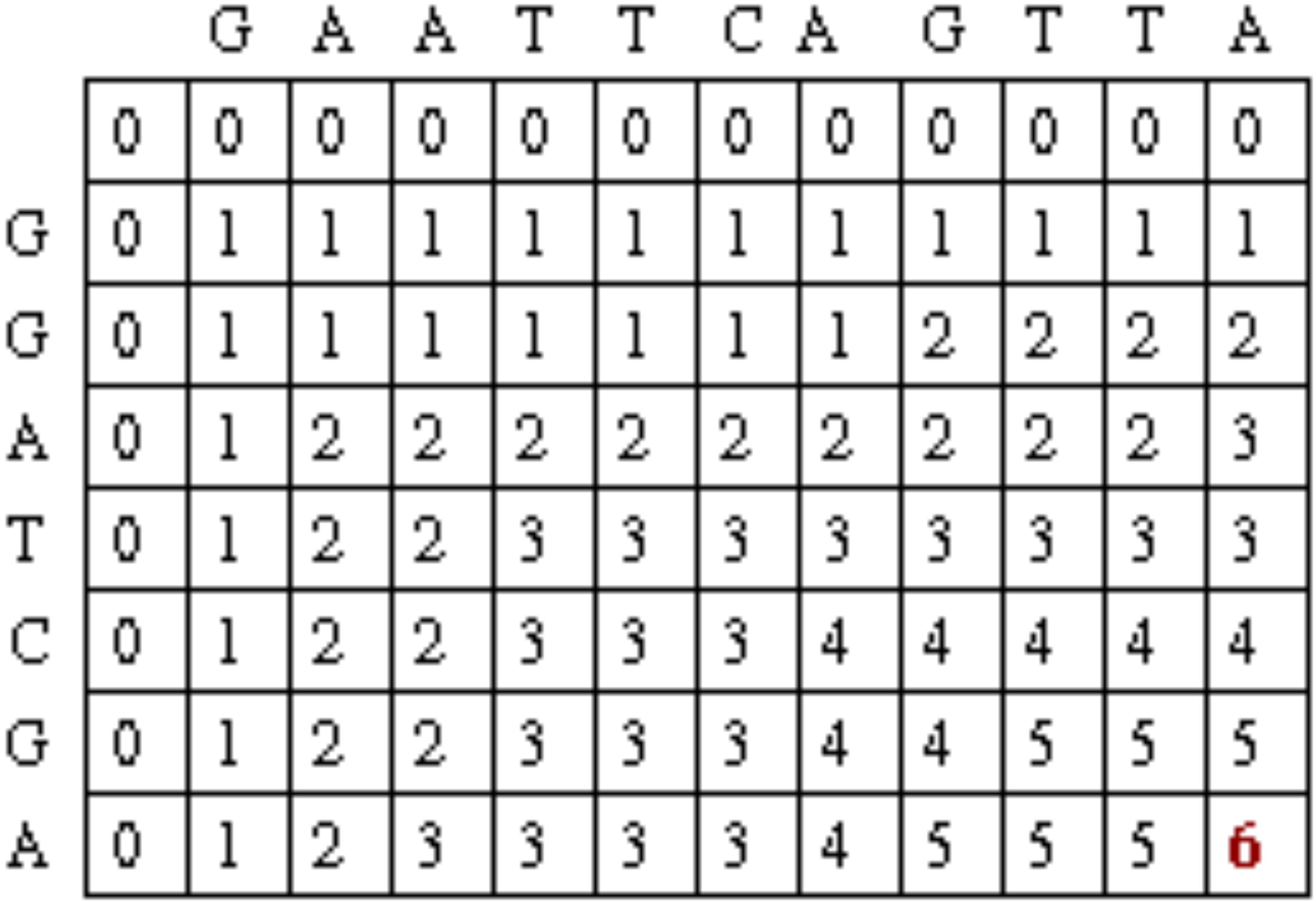
		G	A	A	T	T	C	A	G	T	T	A
	0	0	0	0	0	0	0	0	0	0	0	0
G	0	1	1	1	1	1	1	1	1	1	1	1
G	0	1	1									
A	0	1	2									
T	0	1	2									
C	0	1	2									
G	0	1	2									
A	0	1	2									

Column 3 with answers



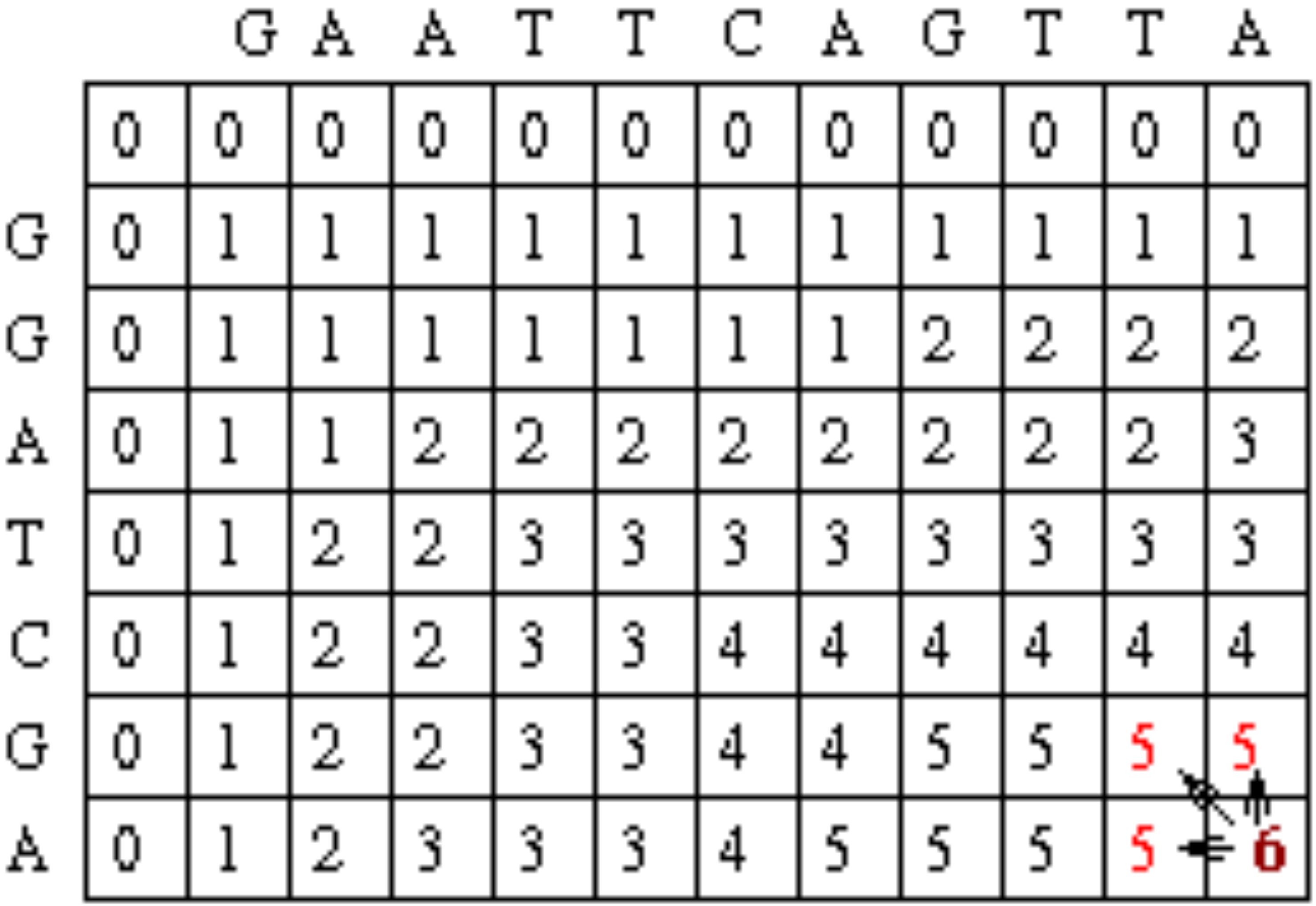
		G	A	A	T	T	C	A	G	T	T	A
	0	0	0	0	0	0	0	0	0	0	0	0
G	0	1	1	1	1	1	1	1	1	1	1	1
G	0	1	1	1								
A	0	1	2	2								
T	0	1	2	2								
C	0	1	2	2								
G	0	1	2	2								
A	0	1	2	3								

Fill in rest of matrix with answers



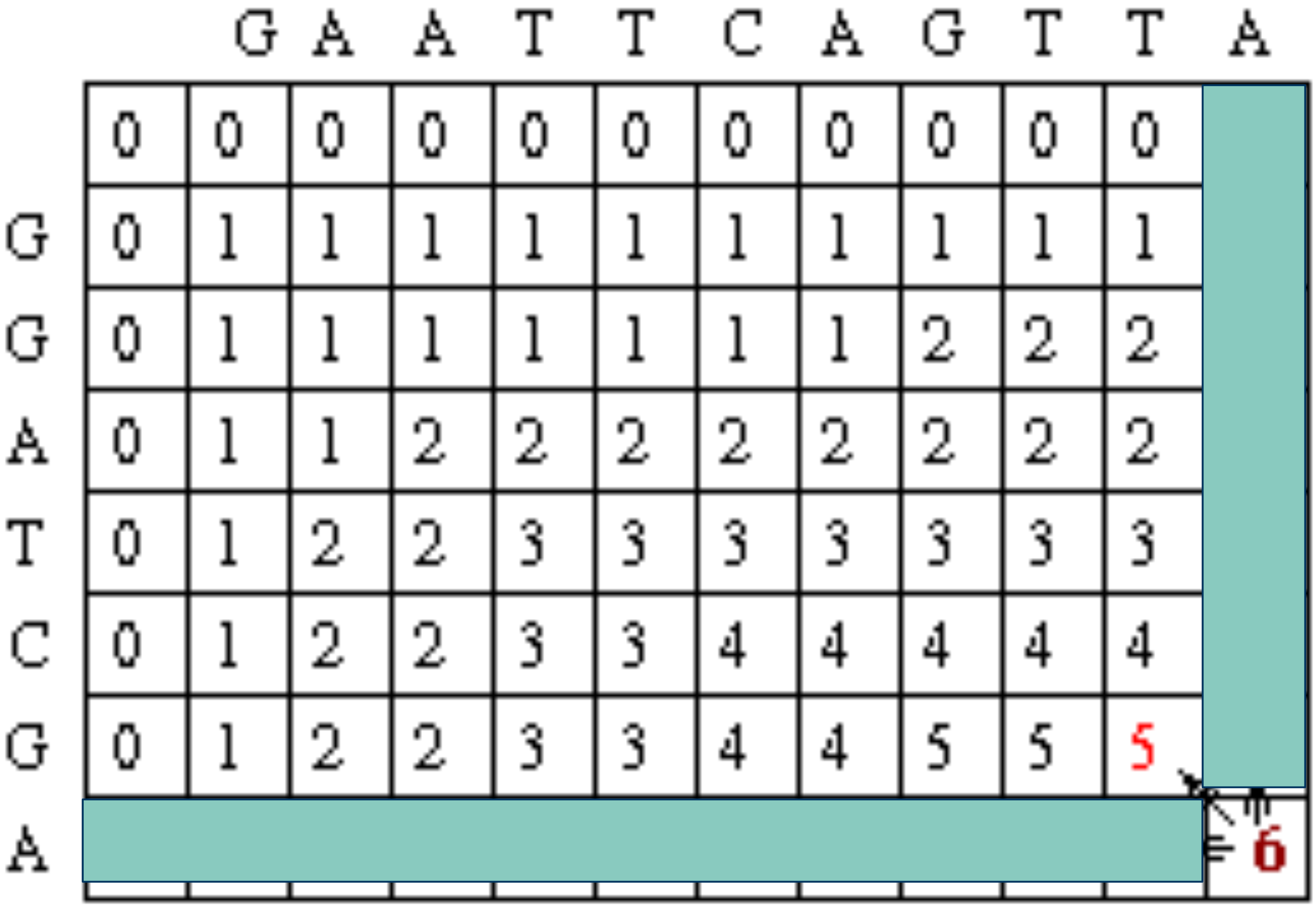
		G	A	A	T	T	C	A	G	T	T	A
	0	0	0	0	0	0	0	0	0	0	0	0
G	0	1	1	1	1	1	1	1	1	1	1	1
G	0	1	1	1	1	1	1	1	2	2	2	2
A	0	1	2	2	2	2	2	2	2	2	2	3
T	0	1	2	2	3	3	3	3	3	3	3	3
C	0	1	2	2	3	3	3	4	4	4	4	4
G	0	1	2	2	3	3	3	4	4	5	5	5
A	0	1	2	3	3	3	3	4	5	5	5	6

Traceback step:
Position at current cell and look at direct predecessors



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

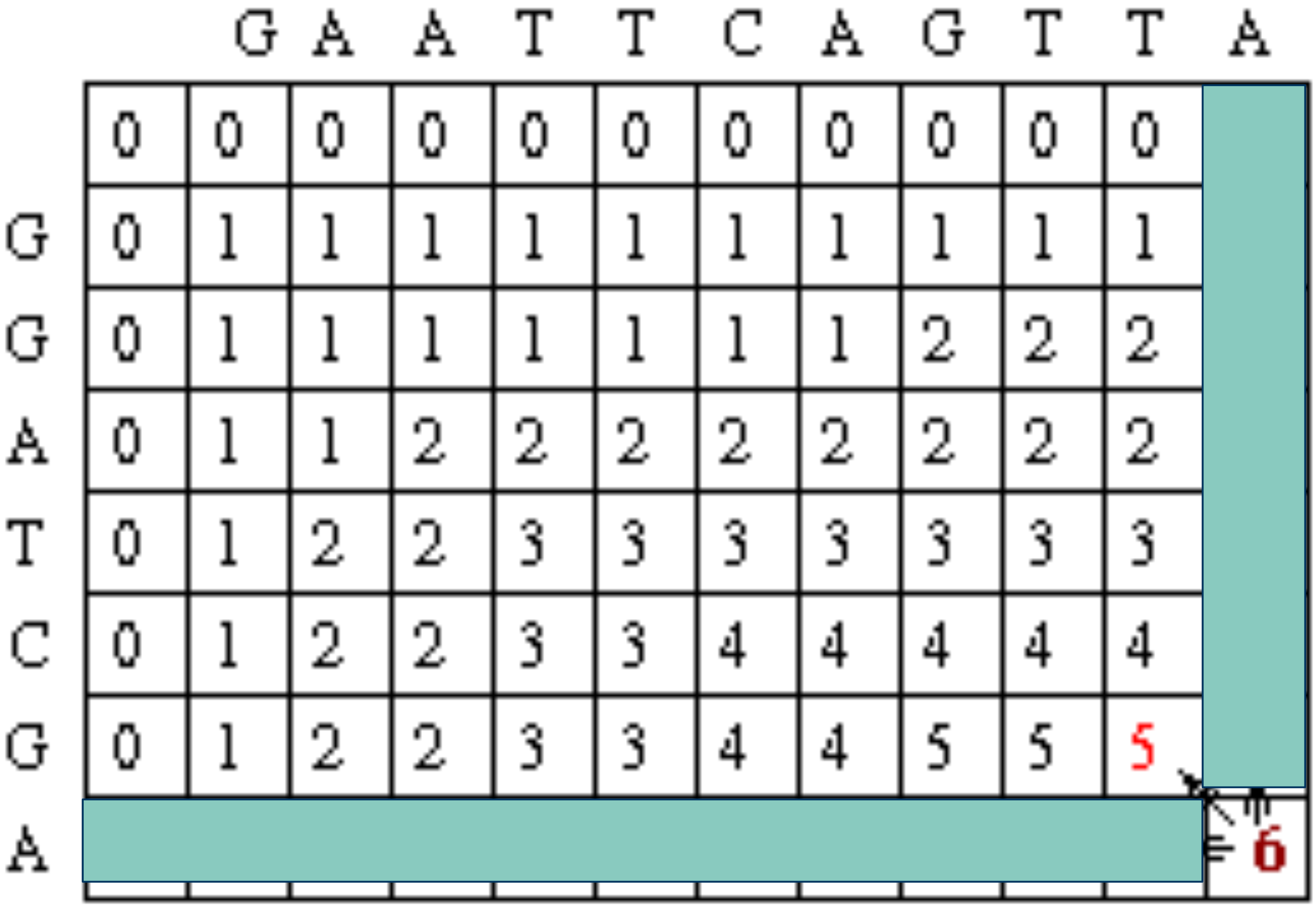
Traceback step:
Position at current cell and look at direct predecessors



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

```
Seq#1  A
       |
Seq#2  A
```

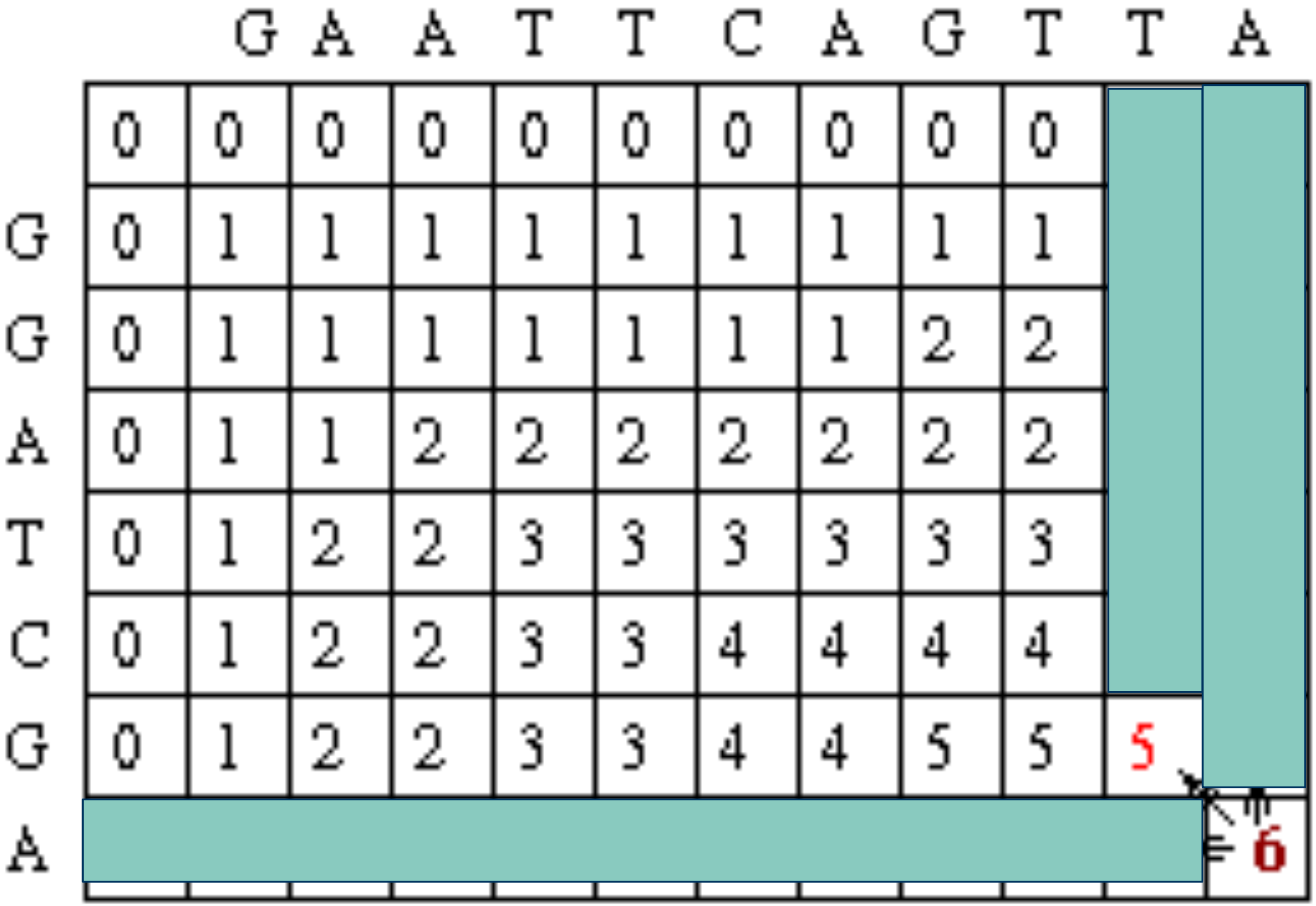
Traceback step:
Position at current cell and look at direct predecessors



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

Traceback step:

Position at current cell and look at direct predecessors

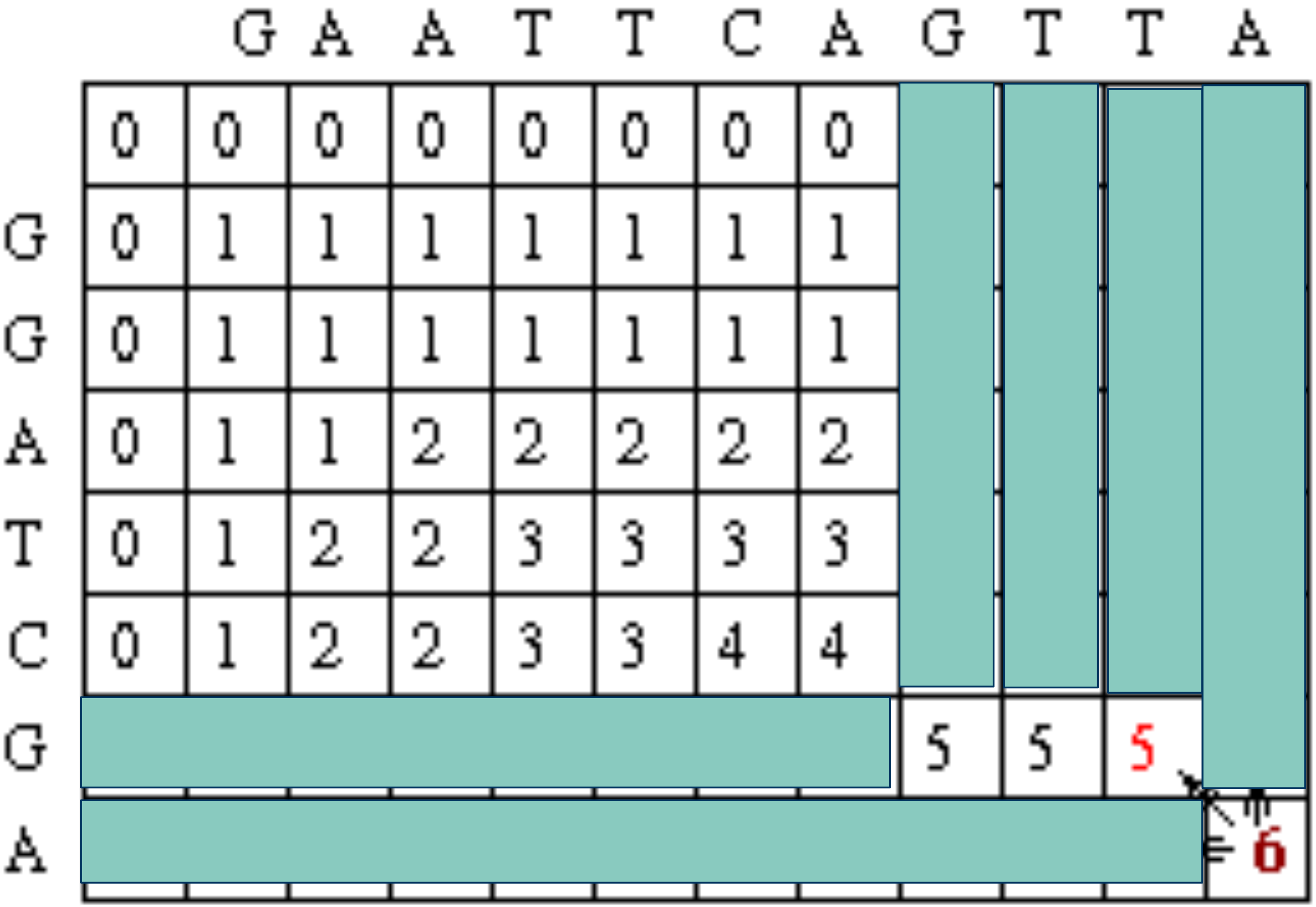


		G	A	A	T	T	C	A	G	T	T	A
	0	0	0	0	0	0	0	0	0	0		
G	0	1	1	1	1	1	1	1	1	1		
G	0	1	1	1	1	1	1	1	2	2		
A	0	1	1	2	2	2	2	2	2	2		
T	0	1	2	2	3	3	3	3	3	3		
C	0	1	2	2	3	3	4	4	4	4		
G	0	1	2	2	3	3	4	4	5	5	5	
A												6

Traceback step:
Position at current cell and look at direct predecessors

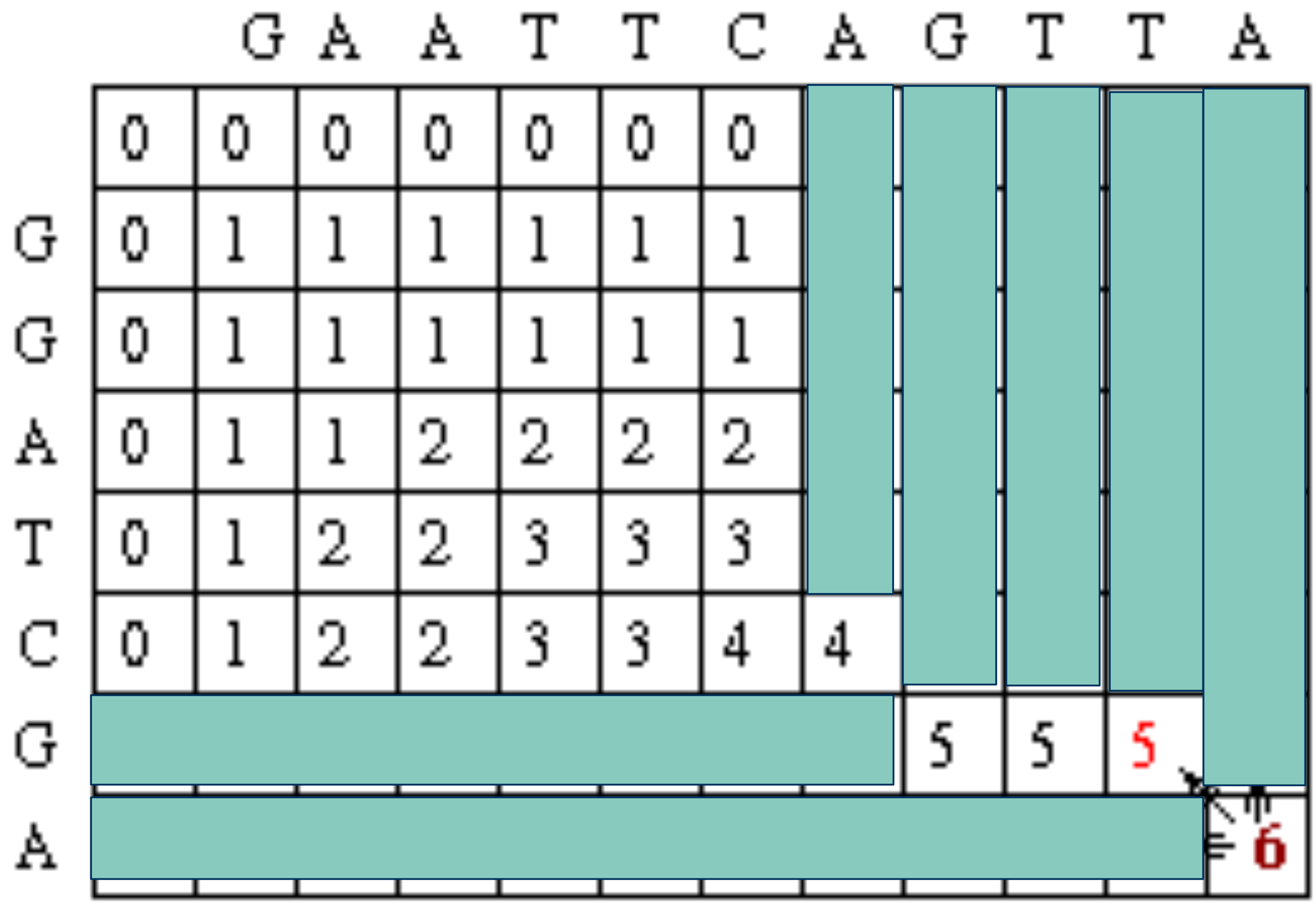
		G	A	A	T	T	C	A	G	T	T	A
	0	0	0	0	0	0	0	0	0			
G	0	1	1	1	1	1	1	1	1			
G	0	1	1	1	1	1	1	1	2			
A	0	1	1	2	2	2	2	2	2			
T	0	1	2	2	3	3	3	3	3			
C	0	1	2	2	3	3	4	4	4			
G	0	1	2	2	3	3	4	4	5	5	5	
A												6

Traceback step:
Position at current cell and look at direct predecessors



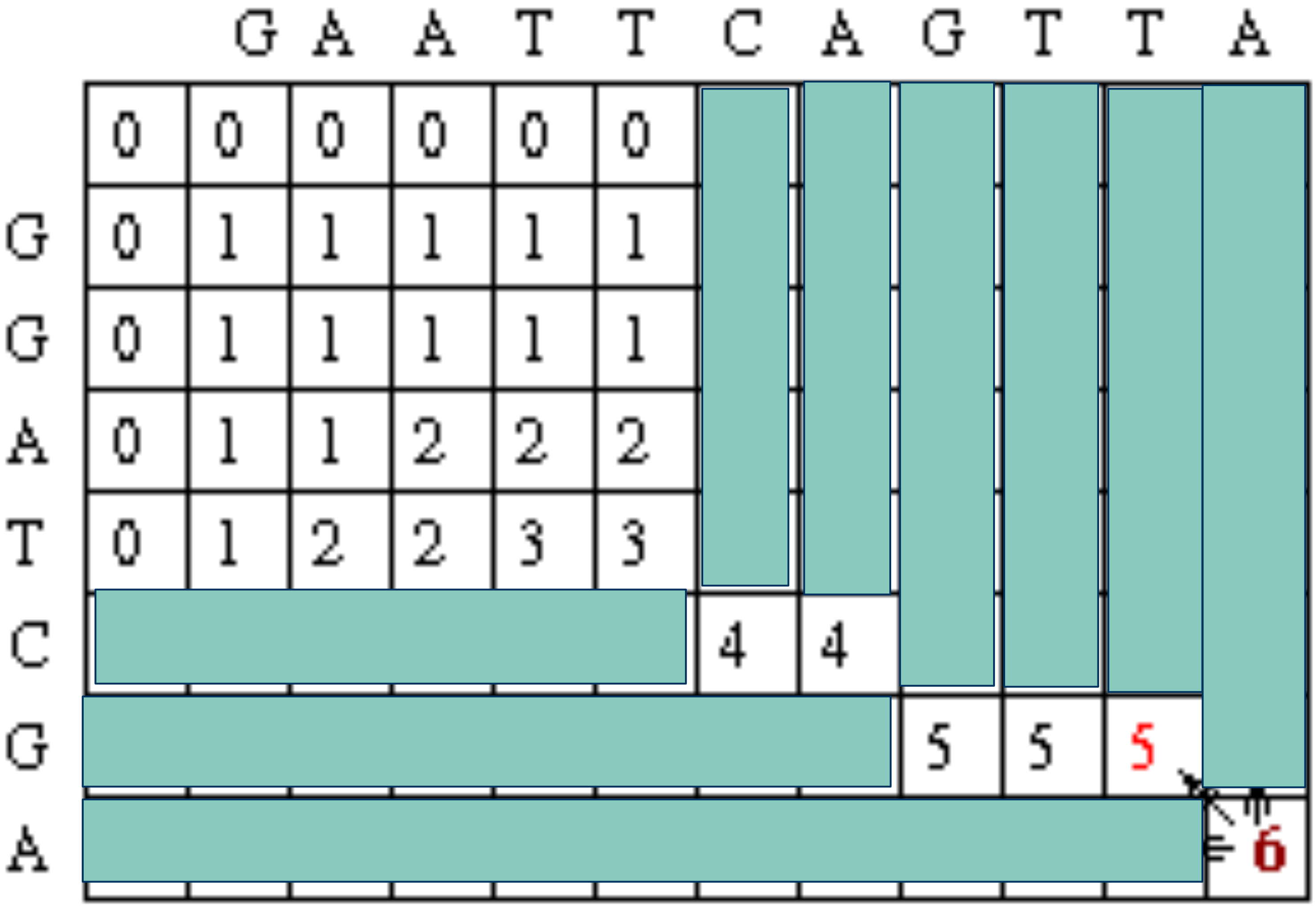
		G	A	A	T	T	C	A	G	T	T	A
	0	0	0	0	0	0	0	0				
G	0	1	1	1	1	1	1	1				
G	0	1	1	1	1	1	1	1				
A	0	1	1	2	2	2	2	2				
T	0	1	2	2	3	3	3	3				
C	0	1	2	2	3	3	4	4				
G									5	5	5	
A												6

Traceback step:
Position at current cell and look at direct predecessors



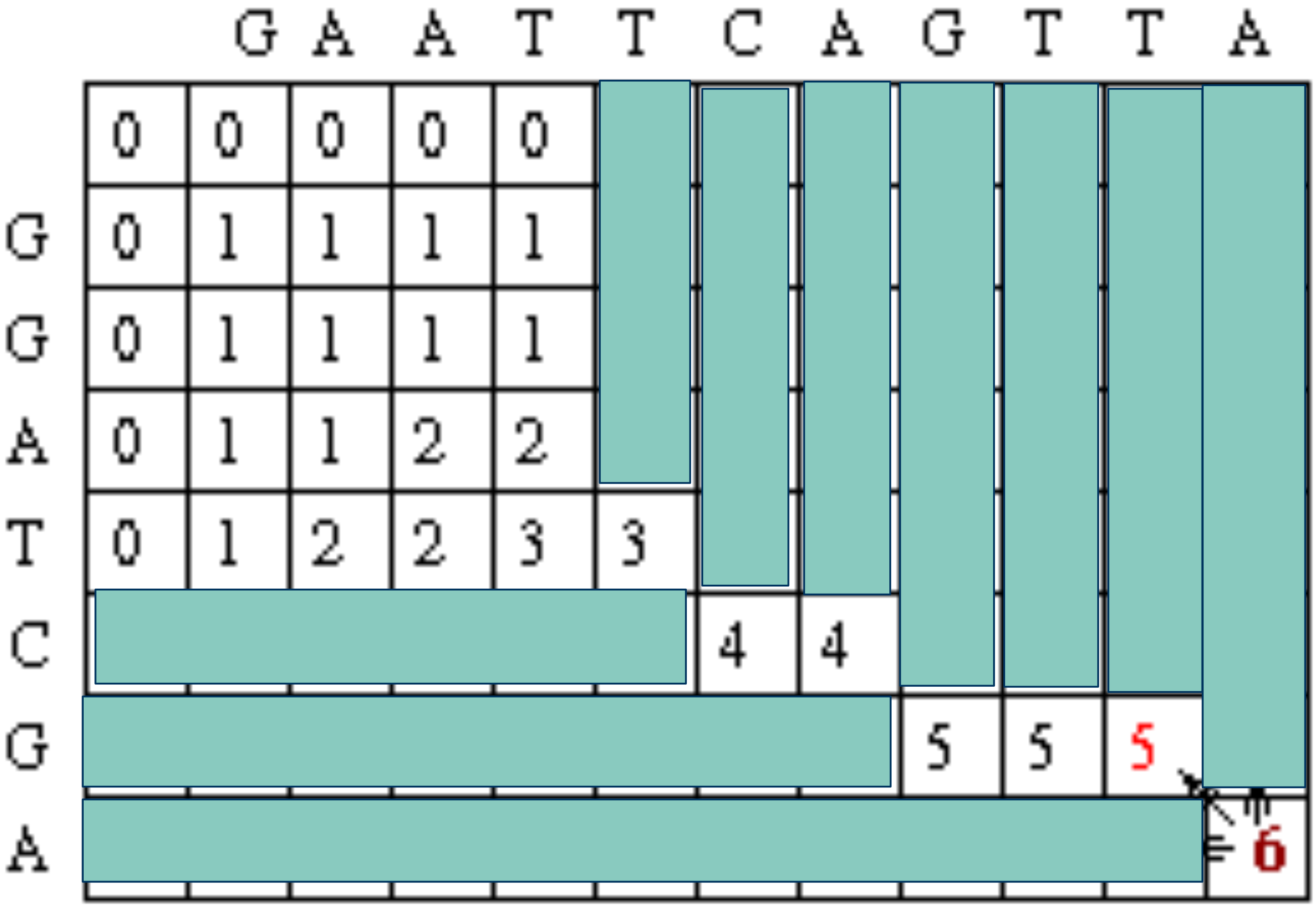
		G	A	A	T	T	C	A	G	T	T	A
	0	0	0	0	0	0	0					
G	0	1	1	1	1	1	1					
G	0	1	1	1	1	1	1					
A	0	1	1	2	2	2	2					
T	0	1	2	2	3	3	3					
C	0	1	2	2	3	3	4	4				
G									5	5	5	
A												6

Traceback step:
Position at current cell and look at direct predecessors



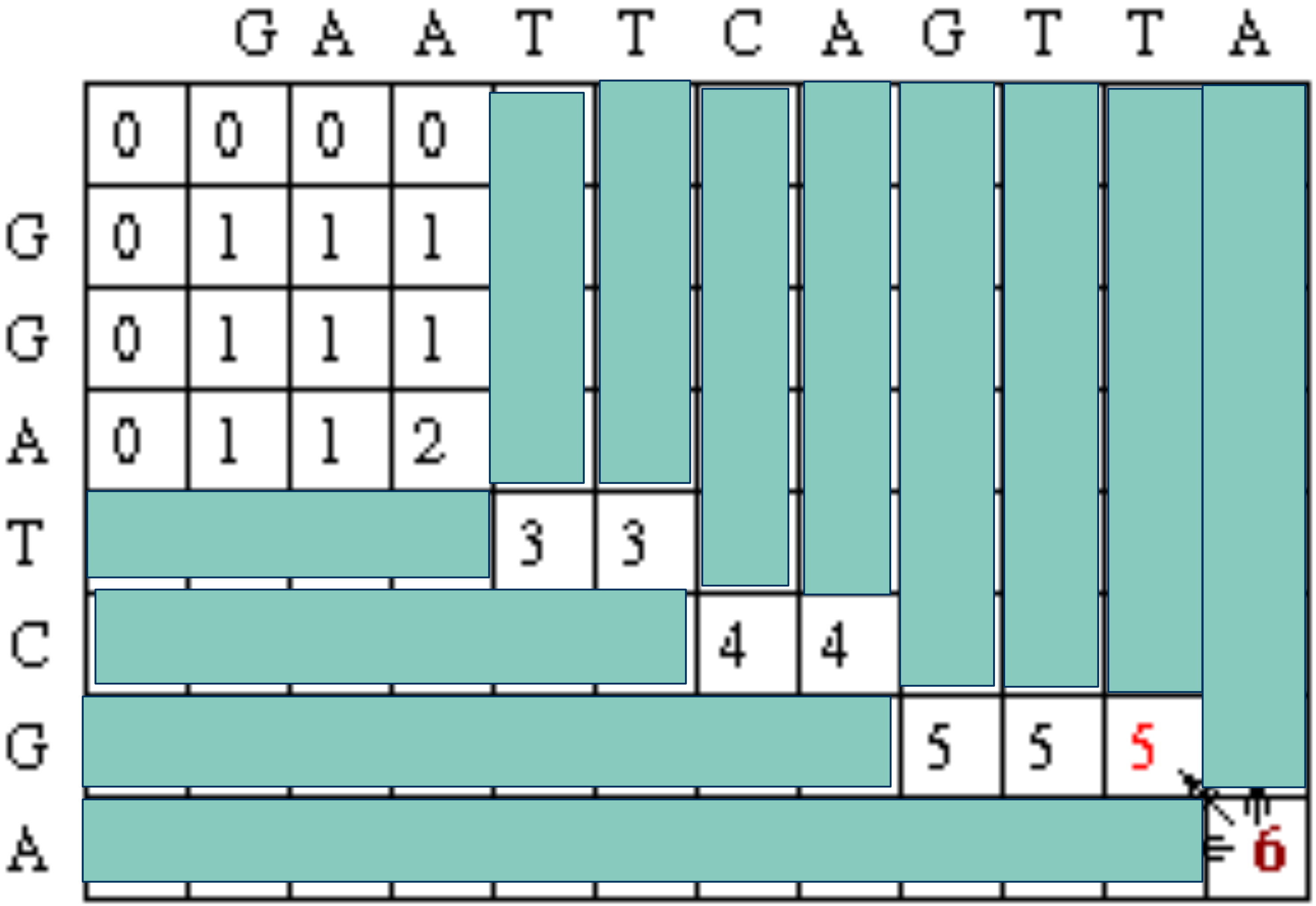
		G	A	A	T	T	C	A	G	T	T	A
	0	0	0	0	0	0						
G	0	1	1	1	1	1						
G	0	1	1	1	1	1						
A	0	1	1	2	2	2						
T	0	1	2	2	3	3						
C							4	4				
G									5	5	5	
A												6

Traceback step:
Position at current cell and look at direct predecessors



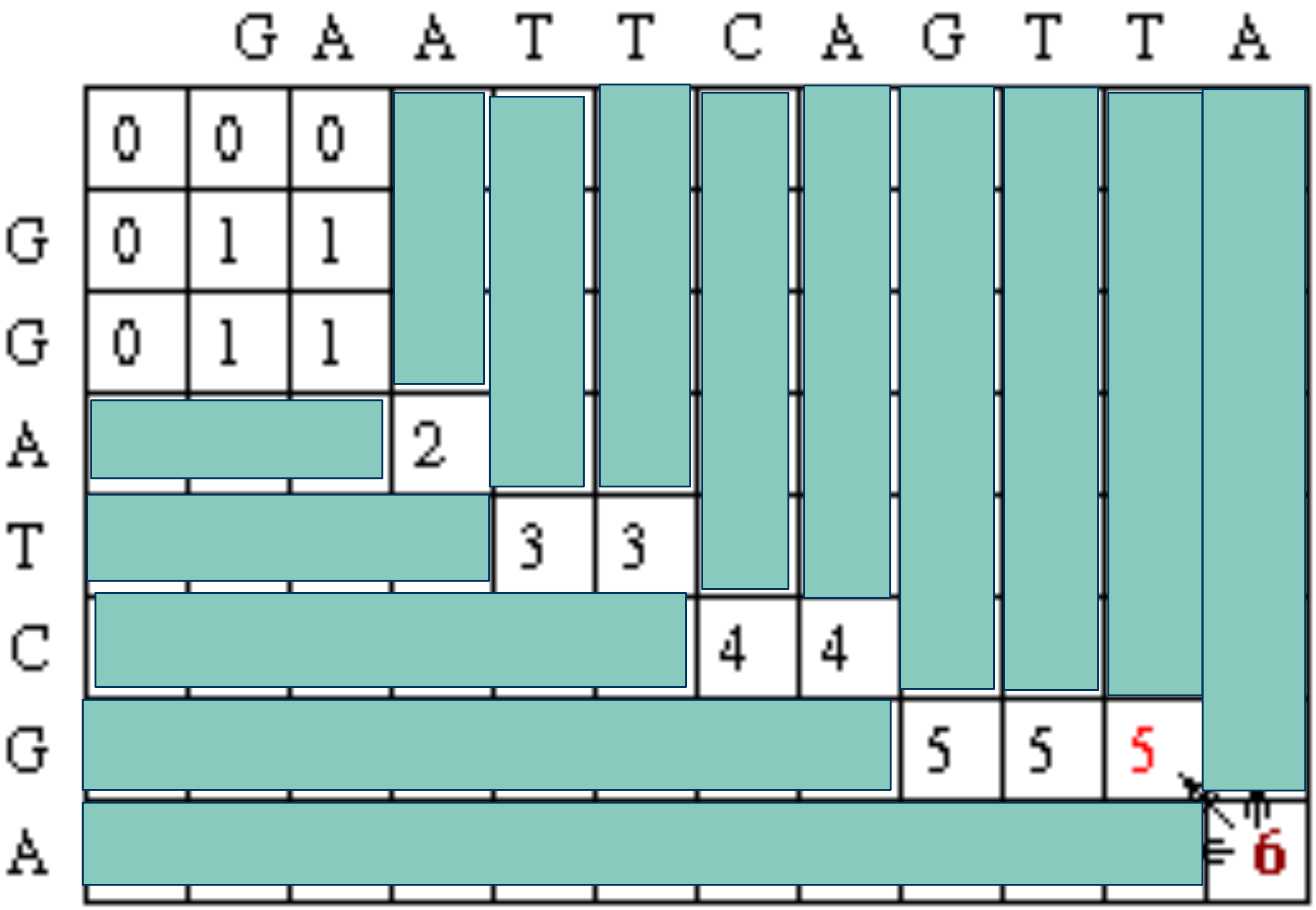
		G	A	A	T	T	C	A	G	T	T	A
	0	0	0	0	0							
G	0	1	1	1	1							
G	0	1	1	1	1							
A	0	1	1	2	2							
T	0	1	2	2	3	3						
C							4	4				
G									5	5	5	
A												6

Traceback step:
Position at current cell and look at direct predecessors



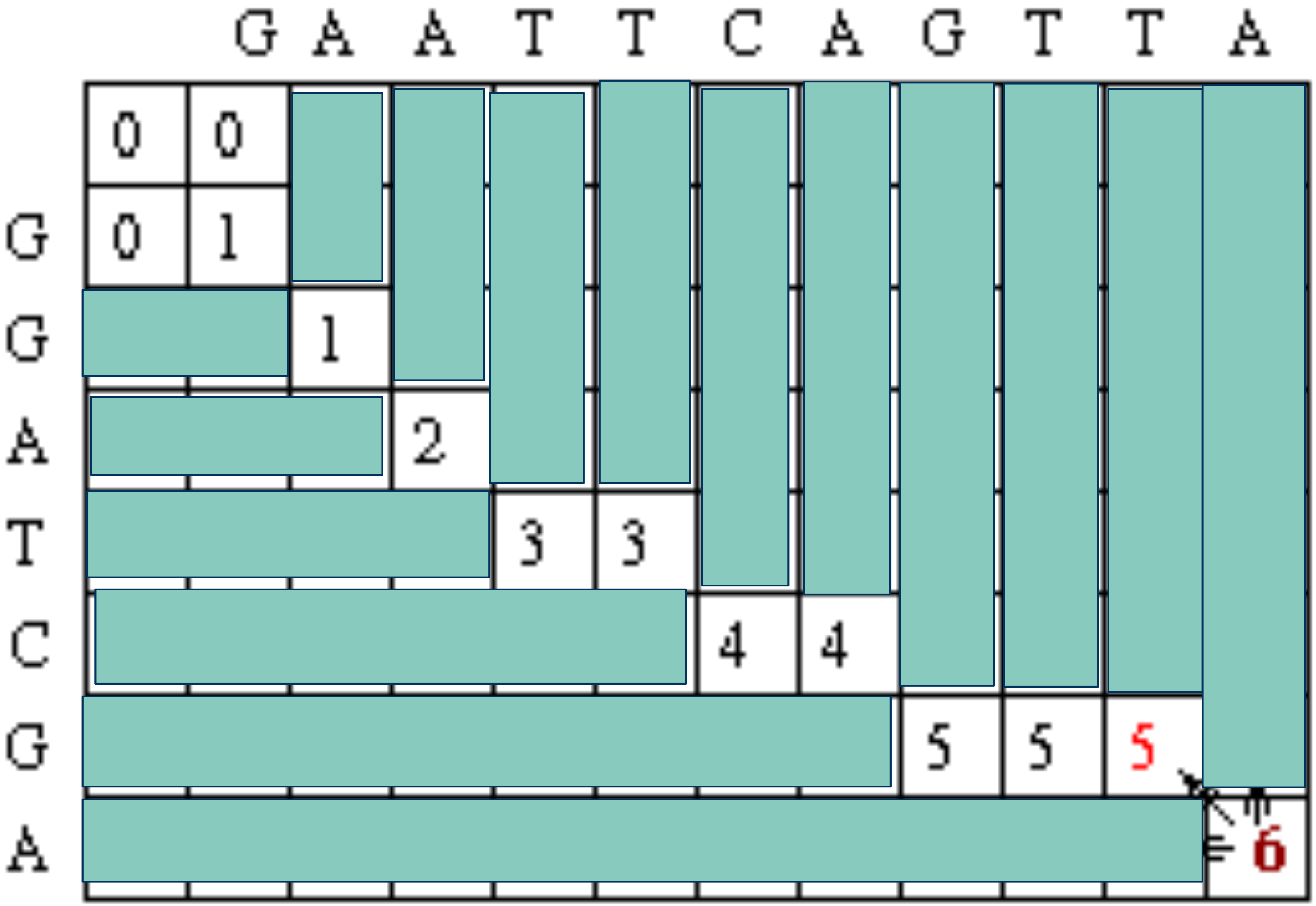
		G	A	A	T	T	C	A	G	T	T	A
	0	0	0	0								
G	0	1	1	1								
G	0	1	1	1								
A	0	1	1	2								
T					3	3						
C							4	4				
G									5	5	5	
A												6

Traceback step:
Position at current cell and look at direct predecessors



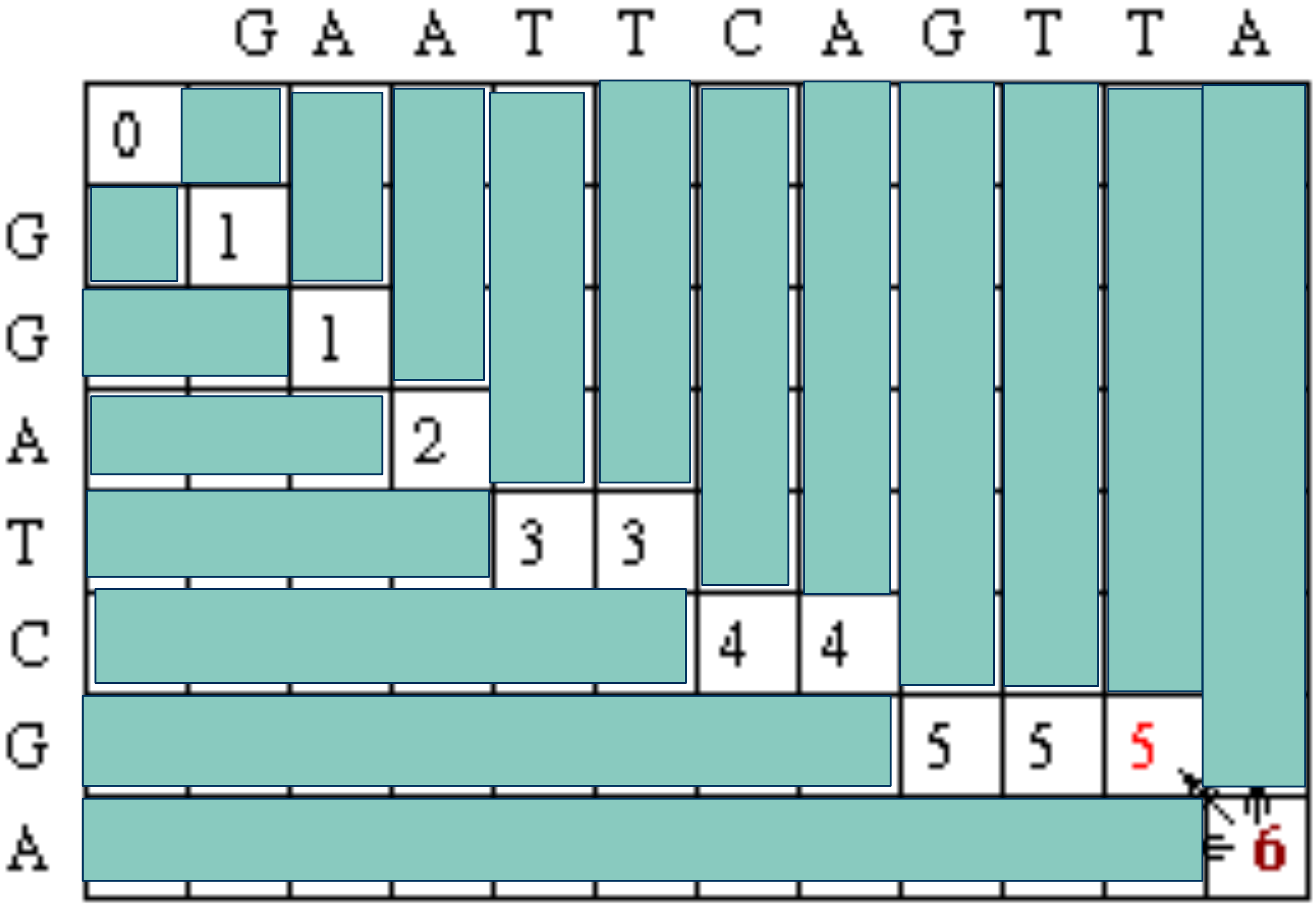
		G	A	A	T	T	C	A	G	T	T	A
	0	0	0									
G	0	1	1									
G	0	1	1									
A				2								
T					3	3						
C							4	4				
G									5	5	5	
A												6

Traceback step:
Position at current cell and look at direct predecessors

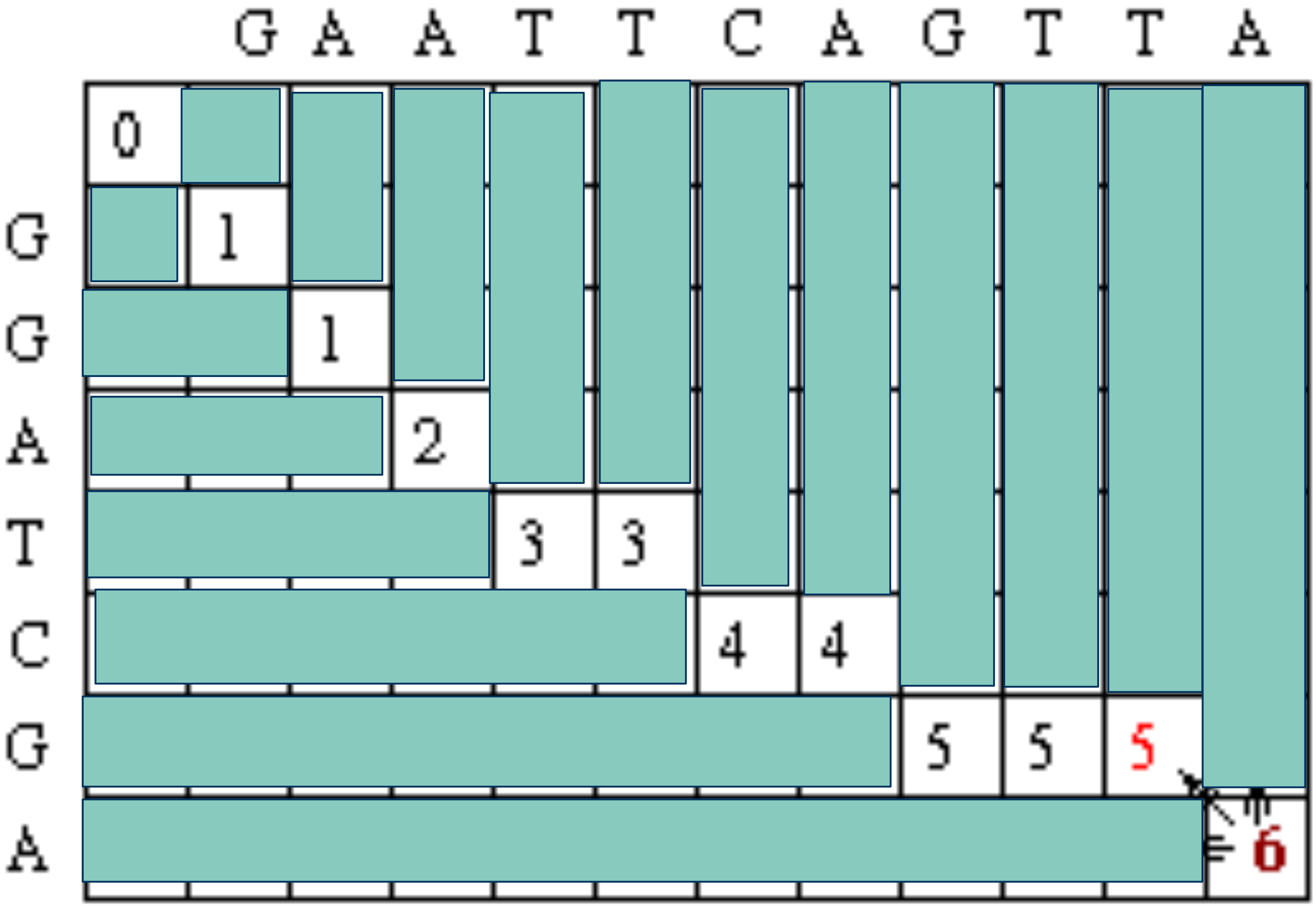


		G	A	A	T	T	C	A	G	T	T	A
	0	0										
G	0	1										
G			1									
A				2								
T					3	3						
C							4	4				
G									5	5	5	
A												6

Traceback step:
Position at current cell and look at direct predecessors



Traceback step:
Position at current cell and look at direct predecessors



```
Seq#1   G A A T T C A G T T A
        |   | |   |   |     |
Seq#2   G G A T - C - G - - A
```

Pseudocode

```
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)
  }
```

Traceback

```
AlignmentA ← ""
AlignmentB ← ""
i ← length(A)
j ← length(B)
while (i > 0 or j > 0)
{
    if (i > 0 and j > 0 and F(i,j) == F(i-1,j-1) + S($A_i$, $B_j$))
    {
        AlignmentA ← $A_i$ + AlignmentA
        AlignmentB ← $B_j$ + AlignmentB
        i ← i - 1
        j ← j - 1
    }
    else if (i > 0 and F(i,j) == F(i-1,j) + d)
    {
        AlignmentA ← $A_i$ + AlignmentA
        AlignmentB ← "-" + AlignmentB
        i ← i - 1
    }
    else (j > 0 and F(i,j) == F(i,j-1) + d)
    {
        AlignmentA ← "-" + AlignmentA
        AlignmentB ← $B_j$ + AlignmentB
        j ← j - 1
    }
}
```

Scoring alignments - substitution matrices

$$\begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix}$$

Identity matrix

Log Odds Ratios

$$S_{i,j} = \log \frac{p_i \cdot M_{i,j}}{p_i \cdot p_j} = \log \frac{M_{i,j}}{p_j} = \log \frac{observed\ frequency}{expected\ frequency}$$

PAM matrix

One of the first amino acid substitution matrices, the PAM (Point Accepted Mutation) matrix was developed by Margaret Dayhoff in the 1970s. This matrix is calculated by observing the differences in closely related proteins. The PAM1 matrix estimates what rate of substitution would be expected if 1% of the amino acids had changed. The PAM1 matrix is used as the basis for calculating other matrices by assuming that repeated mutations would follow the same pattern as those in the PAM1 matrix, and multiple substitutions can occur at the same site. Using this logic, Dayhoff derived matrices as high as PAM250. Usually the PAM 30 and the PAM70 are used.

A matrix for more distantly related sequences can be calculated from a matrix for closely related sequences by taking the second matrix to a power. For instance, we can roughly approximate the WIKI2 matrix from the WIKI1 matrix by saying $W_2=W_1^2$ $W_2=W_1^2$ where W_1 is WIKI1 and W_2 is WIKI2. This is how the PAM250 matrix is calculated.

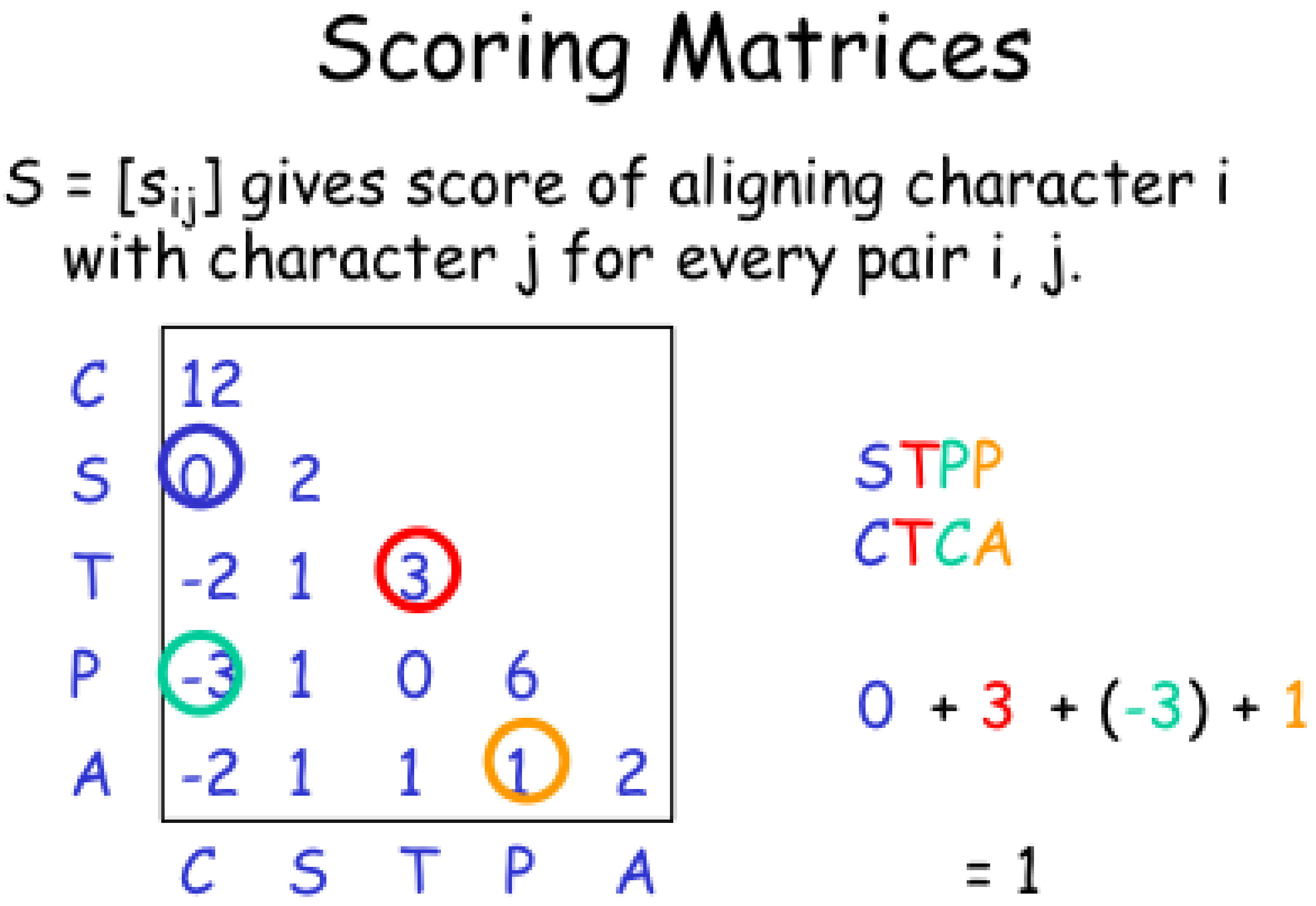
Blosum

Dayhoff's methodology of comparing closely related species turned out not to work very well for aligning evolutionarily divergent sequences. Sequence changes over long evolutionary time scales are not well approximated by compounding small changes that occur over short time scales. The BLOSUM (BLOck SUbstitution Matrix) series of matrices rectifies this problem. Henikoff constructed these matrices using multiple alignments of evolutionarily divergent proteins. The probabilities used in the matrix calculation are computed by looking at "blocks" of conserved sequences found in multiple protein alignments. These conserved sequences are assumed to be of functional importance within related proteins. To reduce bias from closely related sequences, segments in a block with a sequence identity above a certain threshold were clustered giving weight to each such cluster (Henikoff and Henikoff). For the BLOSUM62 matrix, this threshold was set at 62%. Pairs frequencies were then counted between clusters, hence pairs were only counted between segments less than 62% identical. One would use a higher numbered BLOSUM matrix for aligning two closely related sequences and a lower number for more divergent sequences.

It turns out that the BLOSUM62 matrix does an excellent job detecting similarities in distant sequences, and this is the matrix used by default in most recent alignment applications such as BLAST.

Pam vs identity

Pam vs identity



Scoring Matrices

$S = [s_{ij}]$ gives score of aligning character i with character j for every pair i, j.

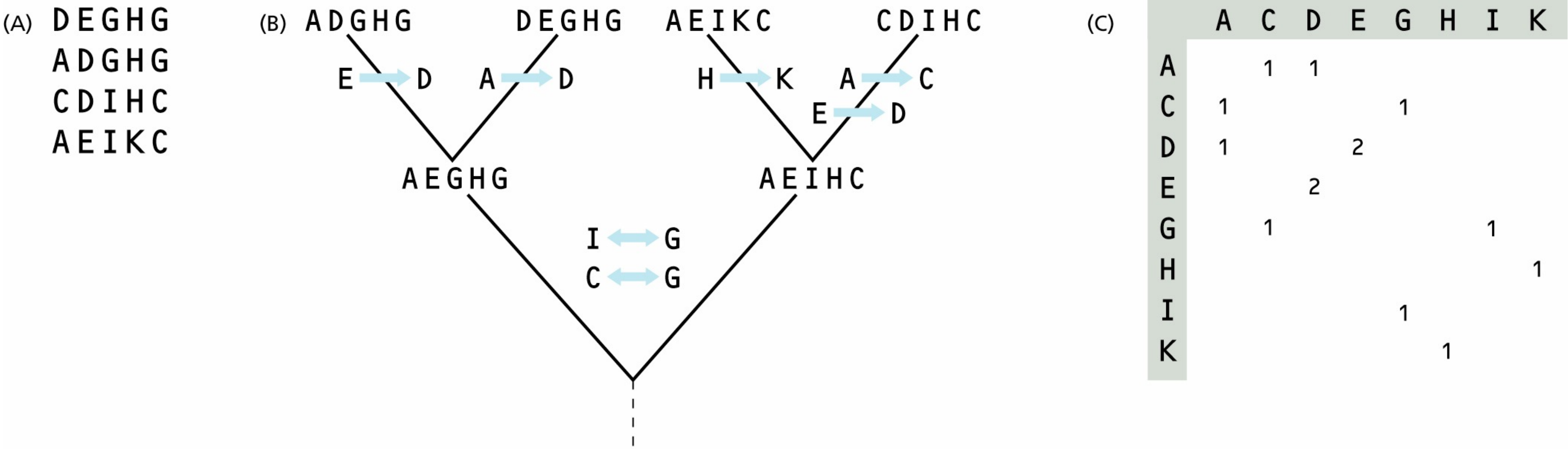
$$\begin{array}{c|ccccc}
C & 12 & & & & \\
S & 0 & 2 & & & \\
T & -2 & 1 & 3 & & \\
P & -3 & 1 & 0 & 6 & \\
A & -2 & 1 & 1 & 1 & 2 \\
\hline
 & C & S & T & P & A
\end{array}$$

STPP
CTCA

$0 + 3 + (-3) + 1$

$= 1$

Pam vs identity



(A)

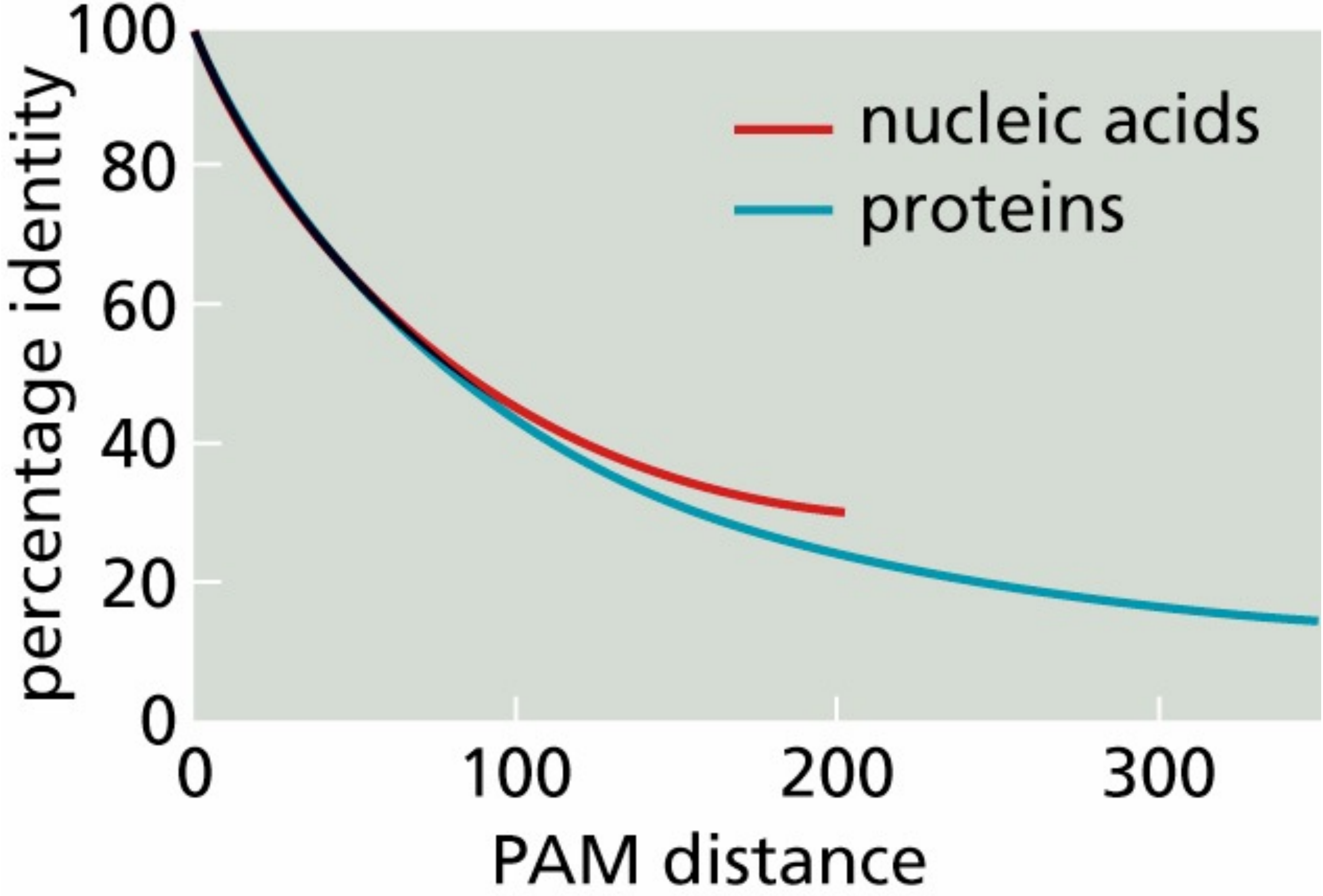
DEGHG
ADGHG
CDIHC
AEIKC

(B)

(C)

	A	C	D	E	G	H	I	K
A		1	1					
C	1				1			
D	1			2				
E			2					
G		1					1	
H								1
I					1			
K						1		

Pam vs identity

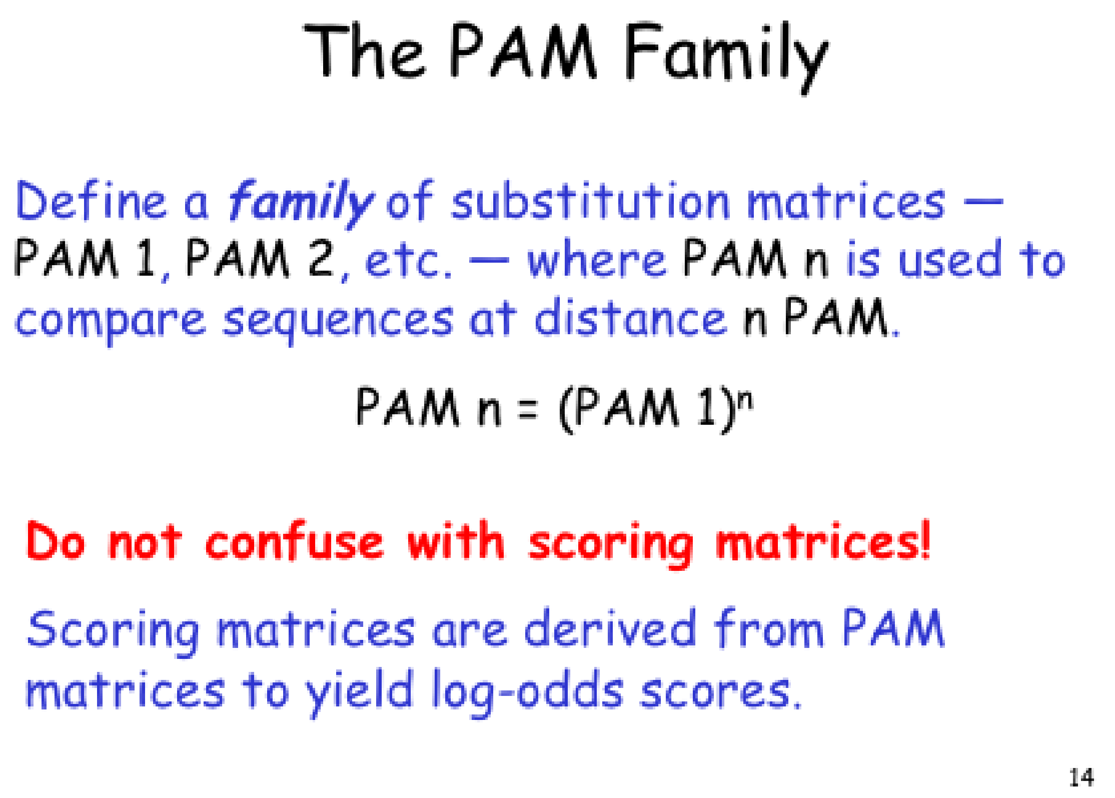


Point Accepted Mutations (PAM)

$$\mathrm{PAM}_n(i,j) = log\frac{f(i)M^n(i,j)}{f(i)f(j)} = log\frac{M^n(i,j)}{f(j)}$$

Point Accepted Mutations (PAM)

$$\mathrm{PAM}_n(i,j) = log\frac{f(i)M^n(i,j)}{f(i)f(j)} = log\frac{M^n(i,j)}{f(j)}$$



The PAM Family

Define a *family* of substitution matrices — PAM 1, PAM 2, etc. — where PAM n is used to compare sequences at distance n PAM.

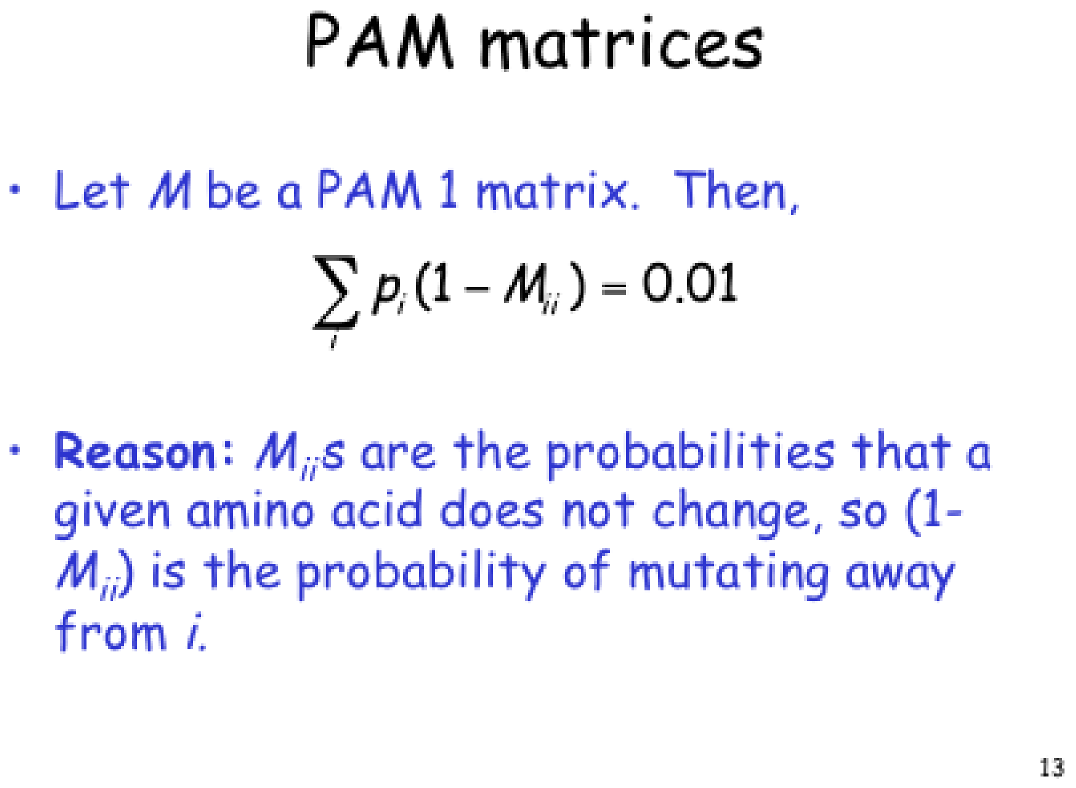
PAM n = (PAM 1)^n

Do not confuse with scoring matrices!

Scoring matrices are derived from PAM matrices to yield log-odds scores.

Point Accepted Mutations (PAM)

$$\mathrm{PAM}_n(i,j) = log\frac{f(i)M^n(i,j)}{f(i)f(j)} = log\frac{M^n(i,j)}{f(j)}$$



PAM matrices

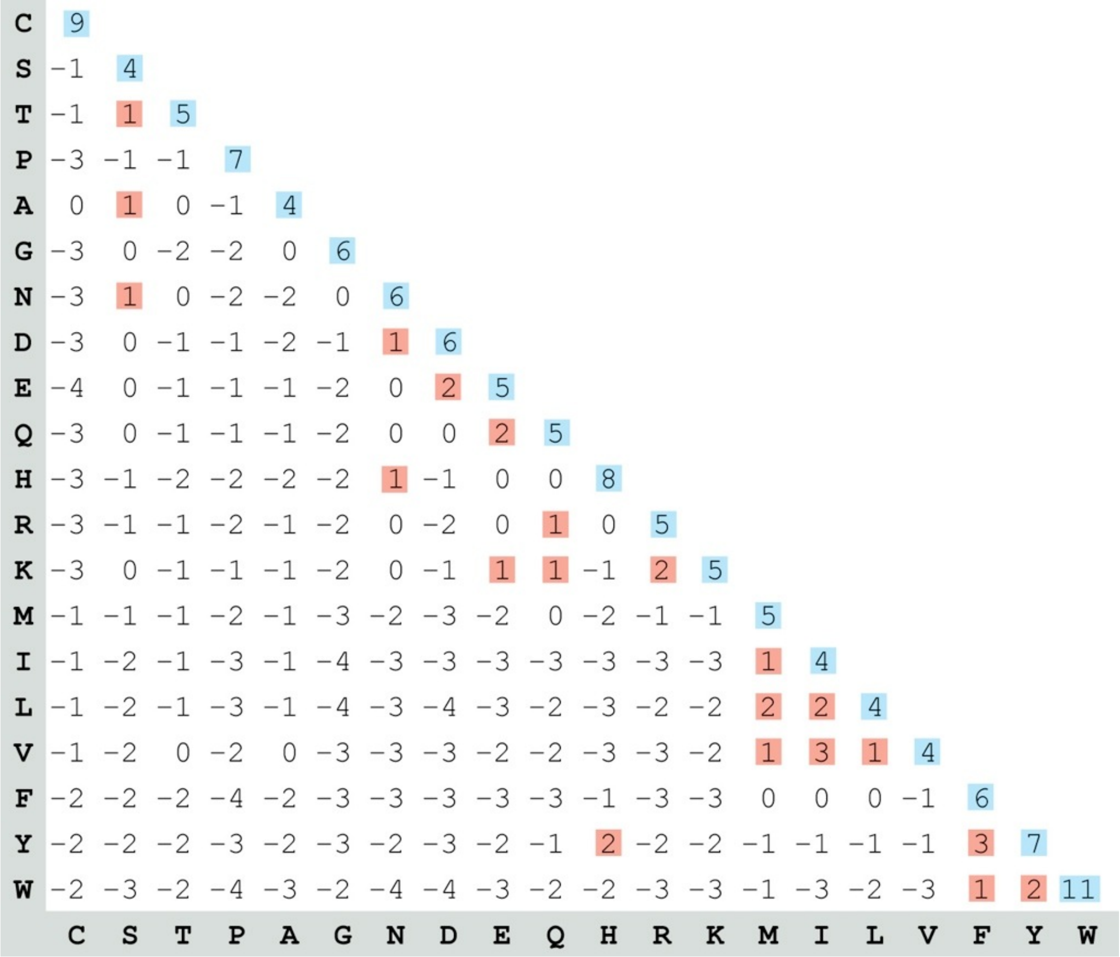
- Let M be a PAM 1 matrix. Then,

$$\sum_i p_i(1-M_{ii}) = 0.01$$

- **Reason:** M_{ii}s are the probabilities that a given amino acid does not change, so $(1-M_{ii})$ is the probability of mutating away from i.

Point Accepted Mutations (PAM)

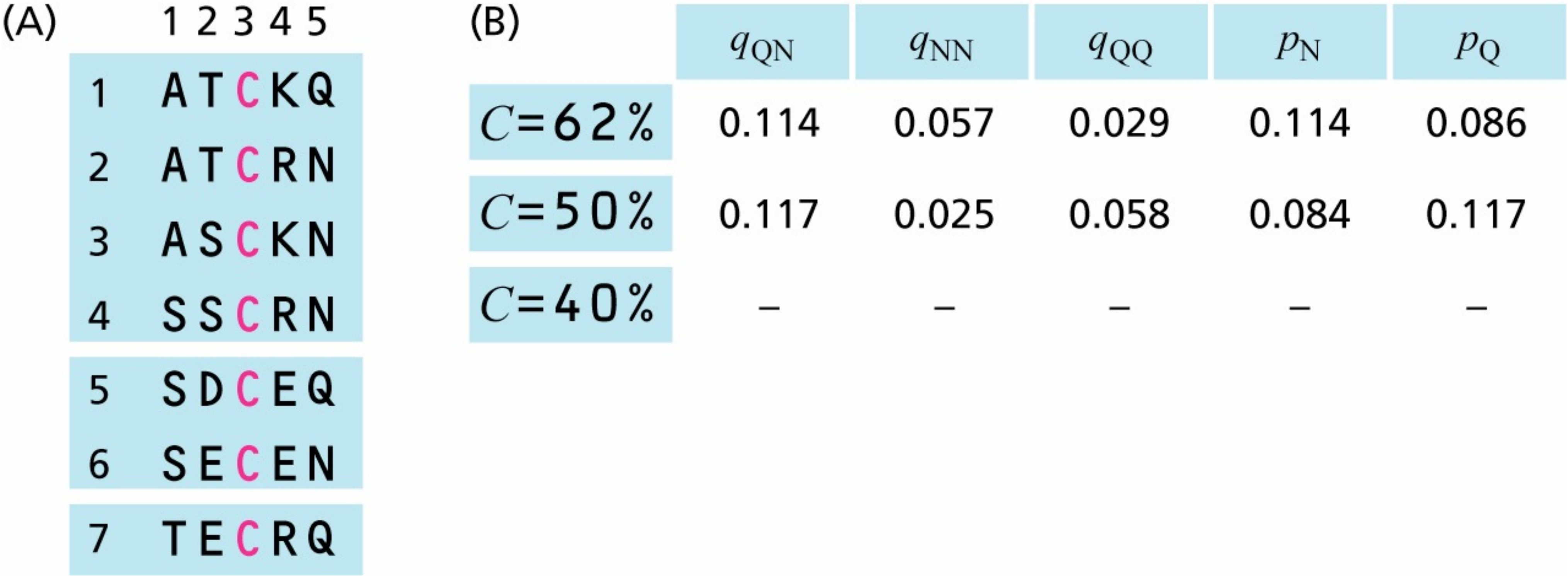
$$\mathrm{PAM}_n(i,j) = log\frac{f(i)M^n(i,j)}{f(i)f(j)} = log\frac{M^n(i,j)}{f(j)}$$



C	9																			
S	−1	4																		
T	−1	1	5																	
P	−3	−1	−1	7																
A	0	1	0	−1	4															
G	−3	0	−2	−2	0	6														
N	−3	1	0	−2	−2	0	6													
D	−3	0	−1	−1	−2	−1	1	6												
E	−4	0	−1	−1	−1	−2	0	2	5											
Q	−3	0	−1	−1	−1	−2	0	0	2	5										
H	−3	−1	−2	−2	−2	−2	1	−1	0	0	8									
R	−3	−1	−1	−2	−1	−2	0	−2	0	1	0	5								
K	−3	0	−1	−1	−1	−2	0	−1	1	1	−1	2	5							
M	−1	−1	−1	−2	−1	−3	−2	−3	−2	0	−2	−1	−1	5						
I	−1	−2	−1	−3	−1	−4	−3	−3	−3	−3	−3	−3	−3	1	4					
L	−1	−2	−1	−3	−1	−4	−3	−4	−3	−2	−3	−2	−2	2	2	4				
V	−1	−2	0	−2	0	−3	−3	−3	−2	−2	−3	−3	−2	1	3	1	4			
F	−2	−2	−2	−4	−2	−3	−3	−3	−3	−3	−1	−3	−3	0	0	0	−1	6		
Y	−2	−2	−2	−3	−2	−3	−2	−3	−2	−1	2	−2	−2	−1	−1	−1	−1	3	7	
W	−2	−3	−2	−4	−3	−2	−4	−4	−3	−2	−2	−3	−3	−1	−3	−2	−3	1	2	11
	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W

Blosum

Blosum



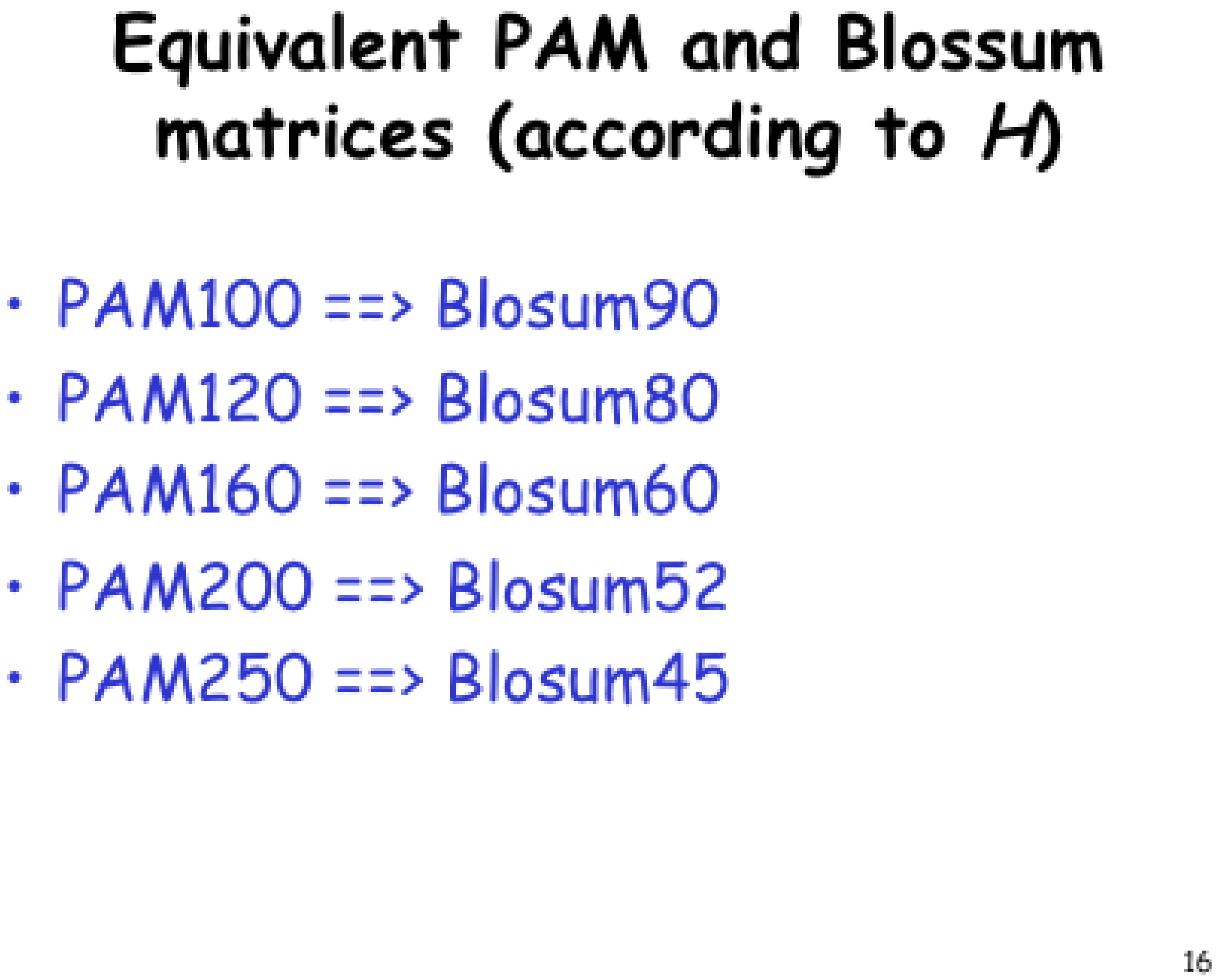
(A)

	1	2	3	4	5
1	A	T	C	K	Q
2	A	T	C	R	N
3	A	S	C	K	N
4	S	S	C	R	N
5	S	D	C	E	Q
6	S	E	C	E	N
7	T	E	C	R	Q

(B)

	q_{QN}	q_{NN}	q_{QQ}	p_N	p_Q
C=62%	0.114	0.057	0.029	0.114	0.086
C=50%	0.117	0.025	0.058	0.084	0.117
C=40%	–	–	–	–	–

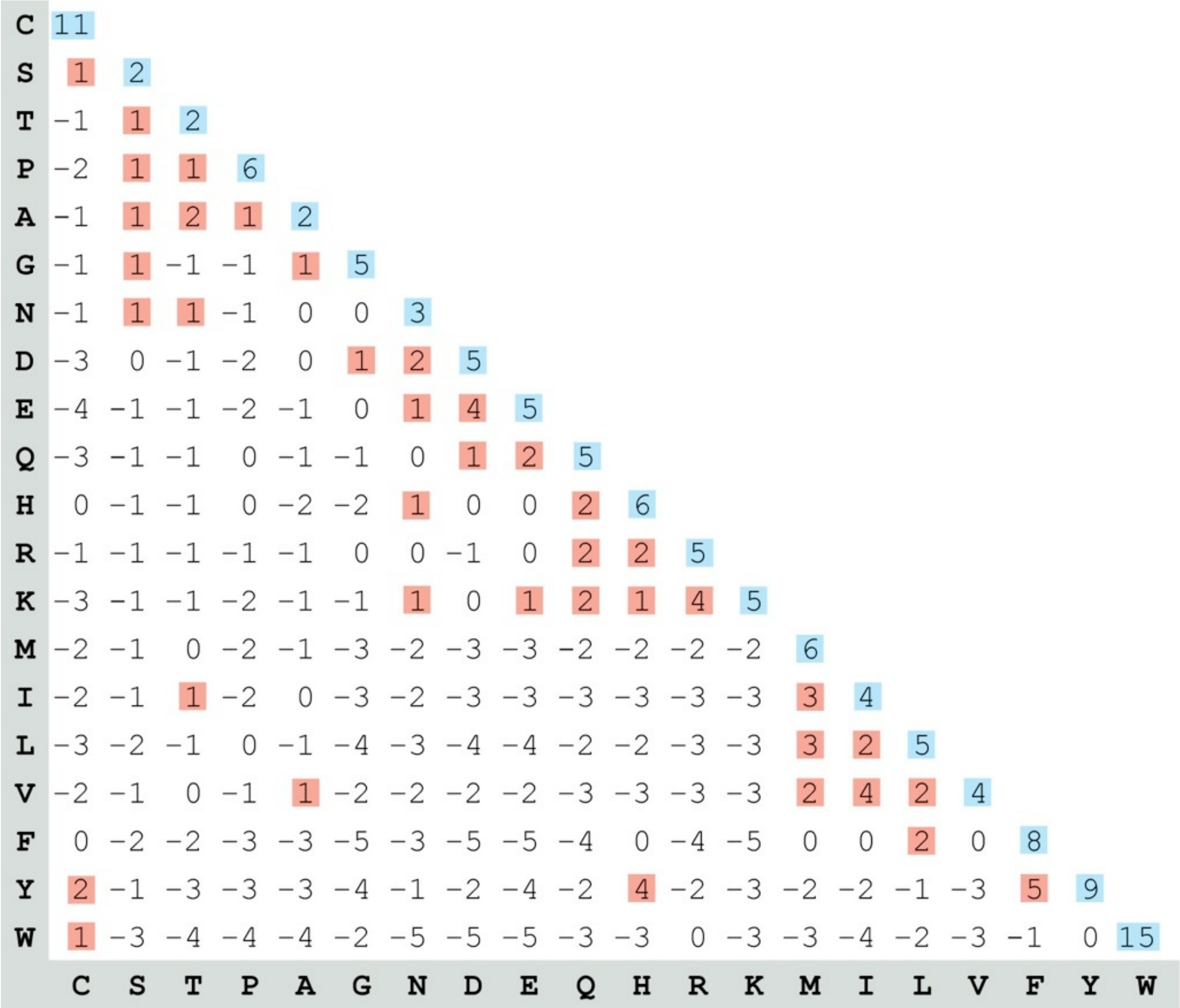
Blosum



Equivalent PAM and Blossum matrices (according to *H*)

- PAM100 ==> Blosum90
- PAM120 ==> Blosum80
- PAM160 ==> Blosum60
- PAM200 ==> Blosum52
- PAM250 ==> Blosum45

Blosum

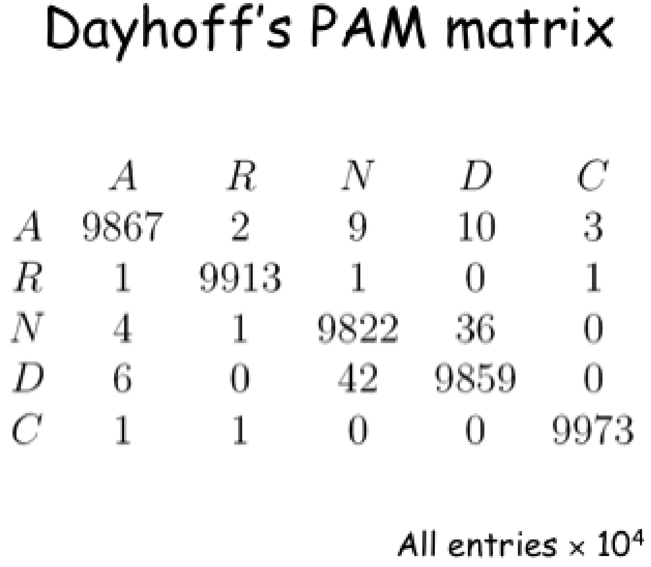


	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	11																			
S	1	2																		
T	−1	1	2																	
P	−2	1	1	6																
A	−1	1	2	1	2															
G	−1	1	−1	−1	1	5														
N	−1	1	1	−1	0	0	3													
D	−3	0	−1	−2	0	1	2	5												
E	−4	−1	−1	−2	−1	0	1	4	5											
Q	−3	−1	−1	0	−1	−1	0	1	2	5										
H	0	−1	−1	0	−2	−2	1	0	0	2	6									
R	−1	−1	−1	−1	−1	0	0	−1	0	2	2	5								
K	−3	−1	−1	−2	−1	−1	1	0	1	2	1	4	5							
M	−2	−1	0	−2	−1	−3	−2	−3	−3	−2	−2	−2	−2	6						
I	−2	−1	1	−2	0	−3	−2	−3	−3	−3	−3	−3	−3	3	4					
L	−3	−2	−1	0	−1	−4	−3	−4	−4	−2	−2	−3	−3	3	2	5				
V	−2	−1	0	−1	1	−2	−2	−2	−2	−3	−3	−3	−3	2	4	2	4			
F	0	−2	−2	−3	−3	−5	−3	−5	−5	−4	0	−4	−5	0	0	2	0	8		
Y	2	−1	−3	−3	−3	−4	−1	−2	−4	−2	4	−2	−3	−2	−2	−1	−3	5	9	
W	1	−3	−4	−4	−4	−2	−5	−5	−5	−3	−3	0	−3	−3	−4	−2	−3	−1	0	15

Difference between Pam and Blosum

- PAM matrices are based on an explicit evolutionary model (i.e. replacements are counted on the branches of a phylogenetic tree), whereas the BLOSUM matrices are based on an implicit model of evolution.
- The PAM matrices are based on mutations observed throughout a global alignment, this includes both highly conserved and highly mutable regions. The BLOSUM matrices are based only on highly conserved regions in series of alignments forbidden to contain gaps.
- The method used to count the replacements is different: unlike the PAM matrix, the BLOSUM procedure uses groups of sequences within which not all mutations are counted the same.
- Higher numbers in the PAM matrix naming scheme denote larger evolutionary distance, while larger numbers in the BLOSUM matrix naming scheme denote higher sequence similarity and therefore smaller evolutionary distance. Example: PAM150 is used for more distant sequences than PAM100; BLOSUM62 is used for closer sequences than BLOSUM50.

Nucleotide Matrices



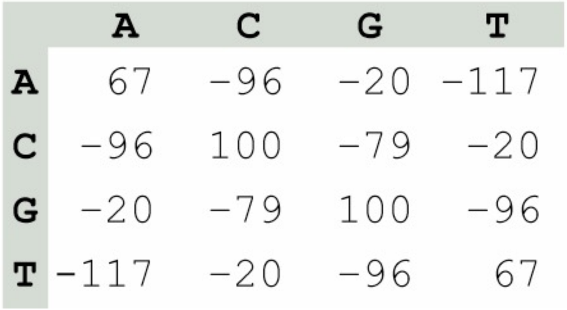
Dayhoff's PAM matrix

	A	R	N	D	C
A	9867	2	9	10	3
R	1	9913	1	0	1
N	4	1	9822	36	0
D	6	0	42	9859	0
C	1	1	0	0	9973

All entries × 10^4

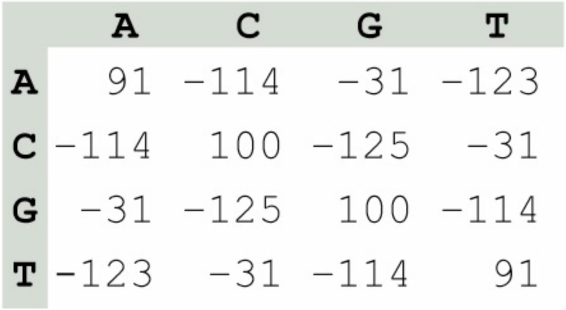
16

(A)



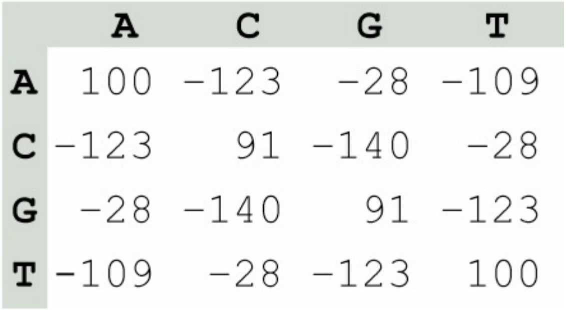
	A	C	G	T
A	67	−96	−20	−117
C	−96	100	−79	−20
G	−20	−79	100	−96
T	−117	−20	−96	67

(B)



	A	C	G	T
A	91	−114	−31	−123
C	−114	100	−125	−31
G	−31	−125	100	−114
T	−123	−31	−114	91

(C)

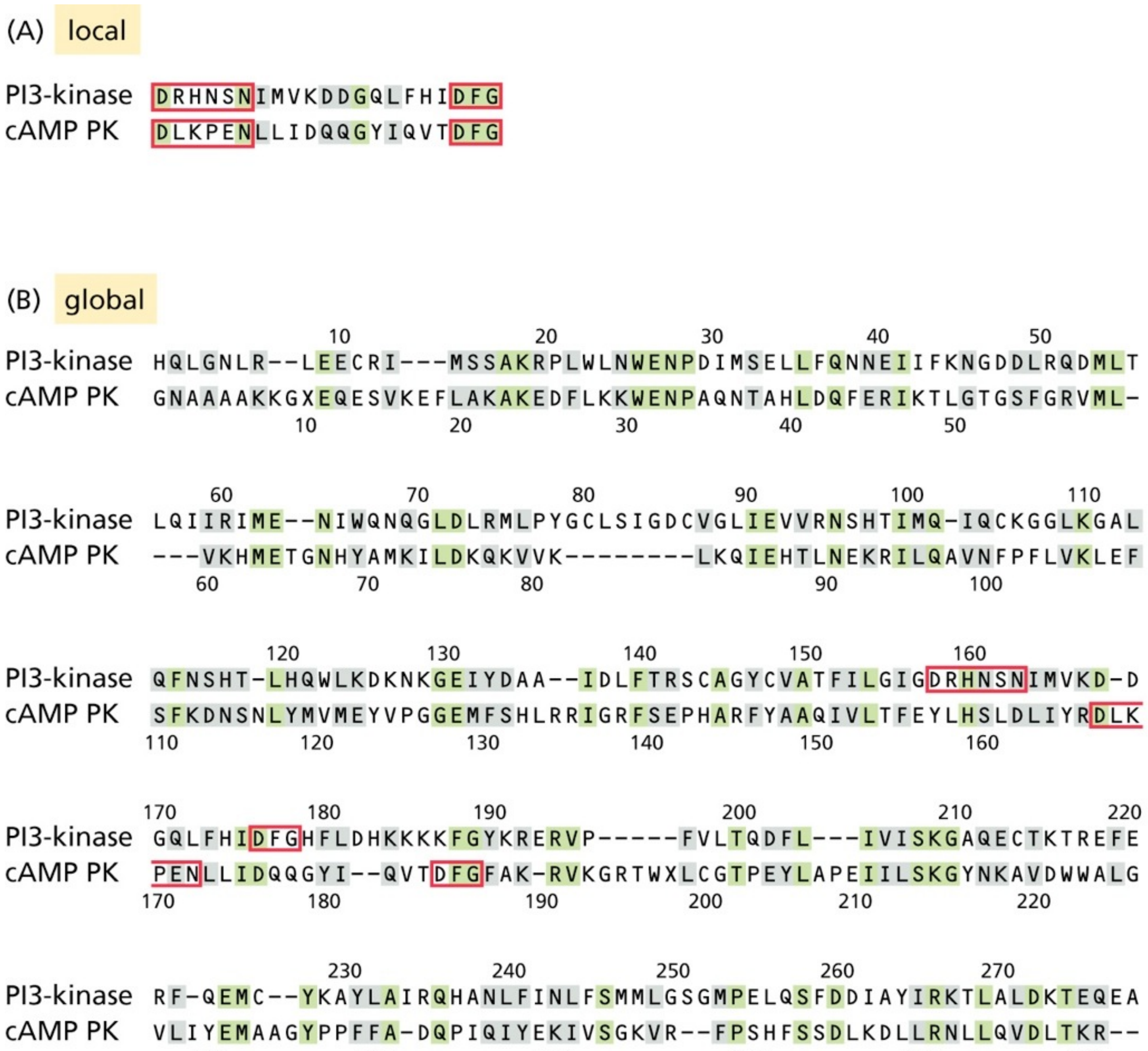


	A	C	G	T
A	100	−123	−28	−109
C	−123	91	−140	−28
G	−28	−140	91	−123
T	−109	−28	−123	100

Gap models

- Gap-extension
- Gap opening cost

Local and global



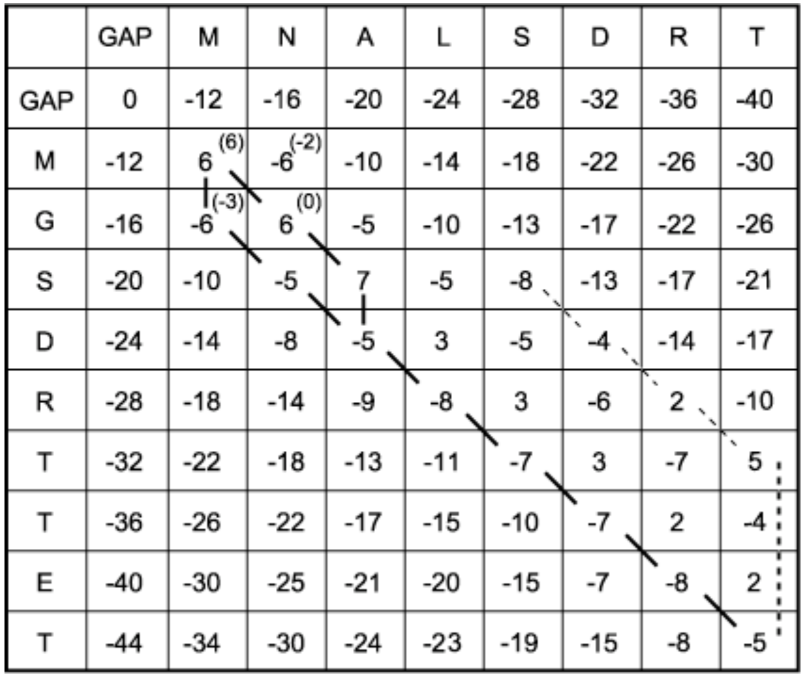
(A) local

```
PI3-kinase  DRHNSNIMVKDDGQLFHIDFG
cAMP PK     DLKPENLLIDQQGYIQVTDFG
```

(B) global

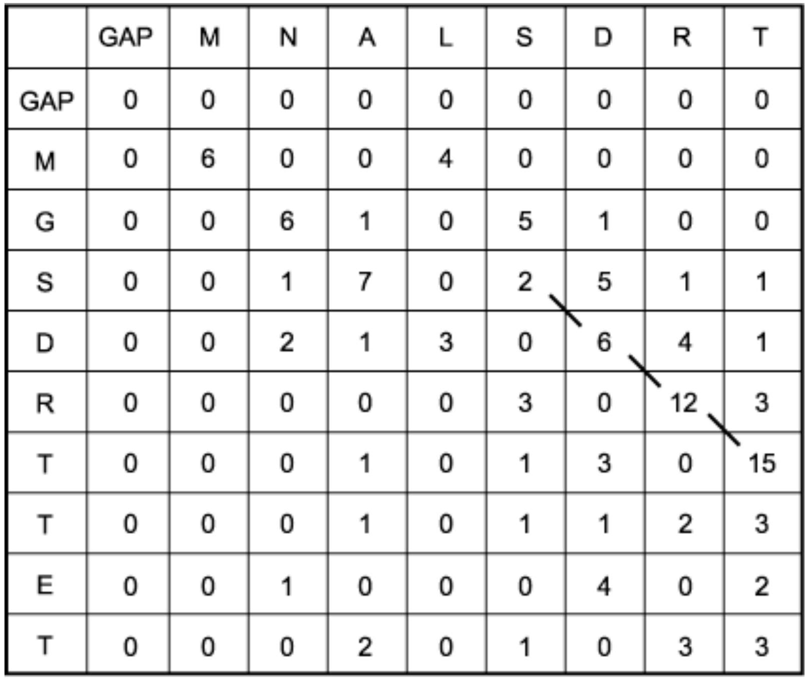
```
                      10          20        30        40        50
PI3-kinase  HQLGNLR--LEECRI---MSSAKRPLWLNWENPDIMSELLFQNNEIIFKNGDDLRQDMLT
cAMP PK     GNAAAAKKGXEQESVKEFLAKAKEDFLKKWENPAQNTAHLDQFERIKTLGTGSFGRVML-
                    10       20        30        40        50

               60          70        80        90        100       110
PI3-kinase  LQIIRIME--NIWQNQGLDLRMLPYGCLSIGDCVGLIEVVRNSHTIMQ-IQCKGGLKGAL
cAMP PK     ---VKHMETGNHYAMKILDKQKVVK--------LKQIEHTLNEKRILQAVNFPFLVKLEF
              60        70        80                90       100

                  120       130         140       150       160
PI3-kinase  QFNSHT-LHQWLKDKNKGEIYDAA--IDLFTRSCAGYCVATFILGIGDRHNSNIMVKD-D
cAMP PK     SFKDNSNLYMVMEYVPGGEMFSHLRRIGRFSEPHARFYAAQIVLTFEYLHSLDLIYRDLK
           110       120       130       140       150       160

           170       180       190            200          210       220
PI3-kinase  GQLFHIDFGHFLDHKKKKFGYKRERVP-----FVLTQDFL---IVISKGAQECTKTREFE
cAMP PK     PENLLIDQQGYI--QVTDFGFAK-RVKGRTWXLCGTPEYLAPEIILSKGYNKAVDWWALG
           170       180          190       200      210        220

                      230       240       250       260       270
PI3-kinase  RF-QEMC--YKAYLAIRQHANLFINLFSMMLGSGMPELQSFDDIAYIRKTLALDKTEQEA
cAMP PK     VLIYEMAAGYPPFFA-DQPIQIYEKIVSGKVR--FPSHFSSDLKDLLRNLLQVDLTKR--
```

Global alignment Needleman-Wunch



	GAP	M	N	A	L	S	D	R	T
GAP	0	-12	-16	-20	-24	-28	-32	-36	-40
M	-12	6$^{(6)}$	-6$^{(-2)}$	-10	-14	-18	-22	-26	-30
G	-16	-6$^{(-3)}$	6$^{(0)}$	-5	-10	-13	-17	-22	-26
S	-20	-10	-5	7	-5	-8	-13	-17	-21
D	-24	-14	-8	-5	3	-5	-4	-14	-17
R	-28	-18	-14	-9	-8	3	-6	2	-10
T	-32	-22	-18	-13	-11	-7	3	-7	5
T	-36	-26	-22	-17	-15	-10	-7	2	-4
E	-40	-30	-25	-21	-20	-15	-7	-8	2
T	-44	-34	-30	-24	-23	-19	-15	-8	-5

Local alignment Smith Waterman

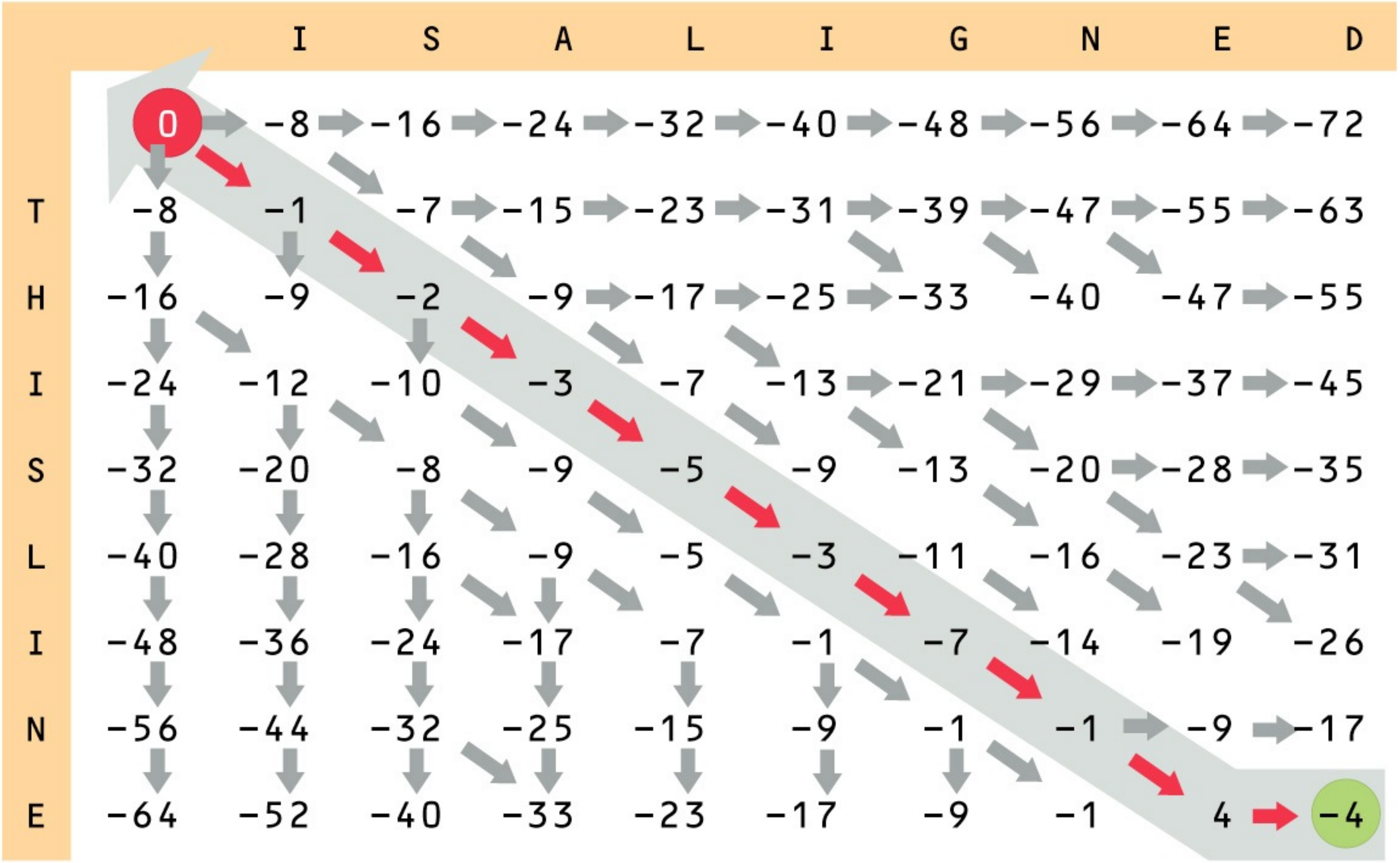


	GAP	M	N	A	L	S	D	R	T
GAP	0	0	0	0	0	0	0	0	0
M	0	6	0	0	4	0	0	0	0
G	0	0	6	1	0	5	1	0	0
S	0	0	1	7	0	2	5	1	1
D	0	0	2	1	3	0	6	4	1
R	0	0	0	0	0	3	0	12	3
T	0	0	0	1	0	1	3	0	15
T	0	0	0	1	0	1	1	2	3
E	0	0	1	0	0	0	4	0	2
T	0	0	0	2	0	1	0	3	3

Different alignments

Different alignments

(A)

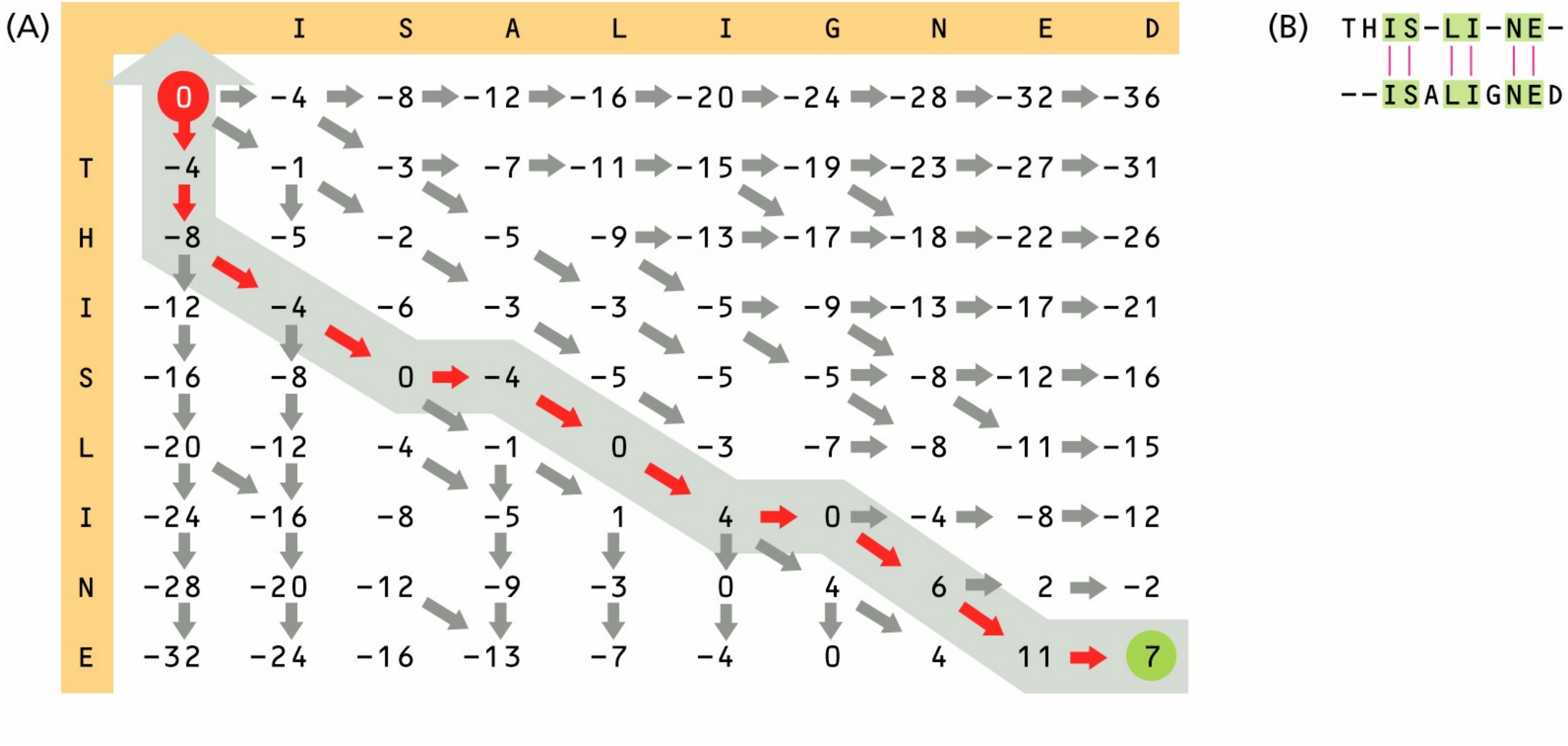


		I	S	A	L	I	G	N	E	D
	0	−8	−16	−24	−32	−40	−48	−56	−64	−72
T	−8	−1	−7	−15	−23	−31	−39	−47	−55	−63
H	−16	−9	−2	−9	−17	−25	−33	−40	−47	−55
I	−24	−12	−10	−3	−7	−13	−21	−29	−37	−45
S	−32	−20	−8	−9	−5	−9	−13	−20	−28	−35
L	−40	−28	−16	−9	−5	−3	−11	−16	−23	−31
I	−48	−36	−24	−17	−7	−1	−7	−14	−19	−26
N	−56	−44	−32	−25	−15	−9	−1	−1	−9	−17
E	−64	−52	−40	−33	−23	−17	−9	−1	4	−4

(B)

```
THISLINE-
      ||
ISALIGNED
```

Different alignments

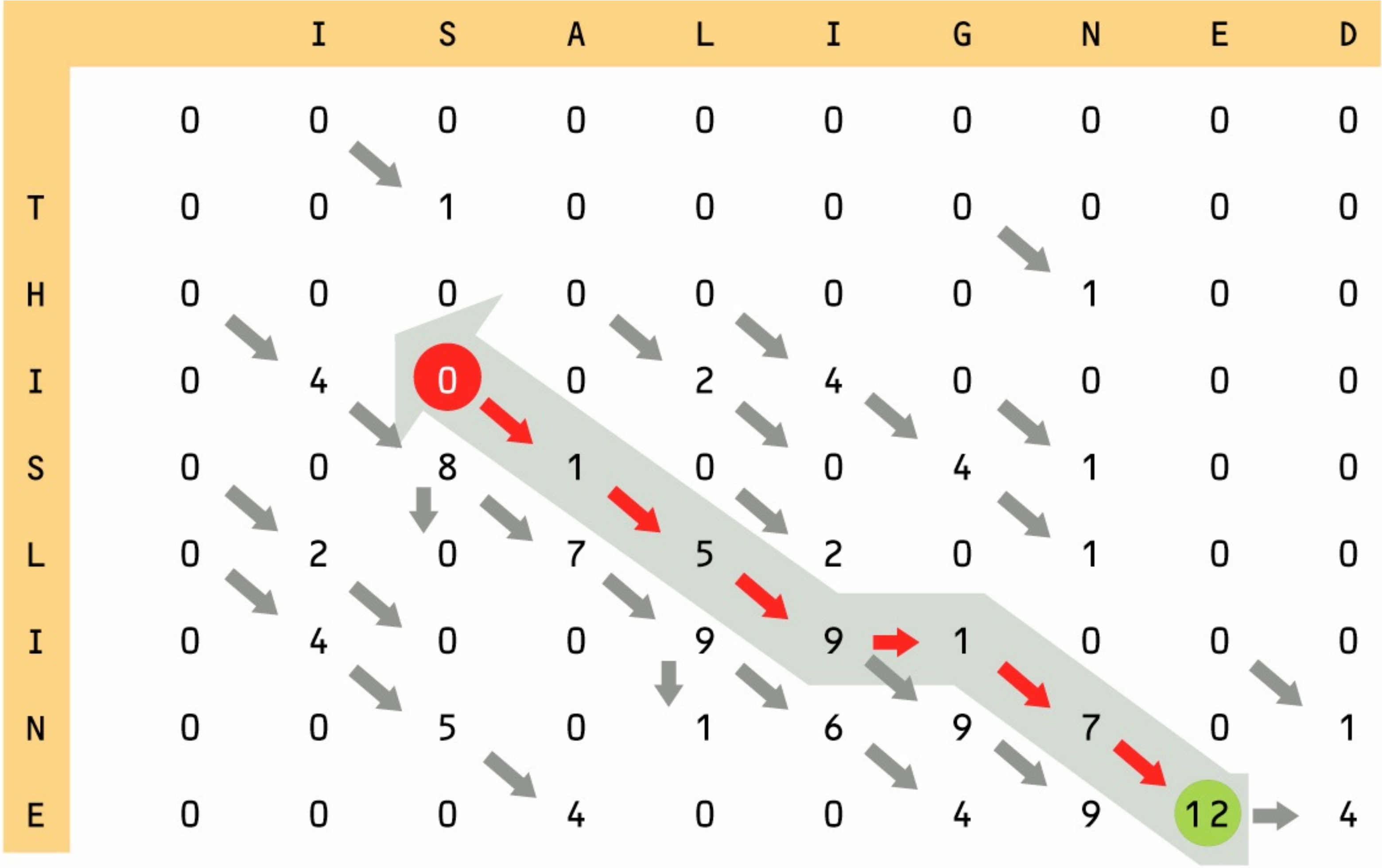


		I	S	A	L	I	G	N	E	D
	0	-4	-8	-12	-16	-20	-24	-28	-32	-36
T	-4	-1	-3	-7	-11	-15	-19	-23	-27	-31
H	-8	-5	-2	-5	-9	-13	-17	-18	-22	-26
I	-12	-4	-6	-3	-3	-5	-9	-13	-17	-21
S	-16	-8	0	-4	-5	-5	-5	-8	-12	-16
L	-20	-12	-4	-1	0	-3	-7	-8	-11	-15
I	-24	-16	-8	-5	1	4	0	-4	-8	-12
N	-28	-20	-12	-9	-3	0	4	6	2	-2
E	-32	-24	-16	-13	-7	-4	0	4	11	7

```
THIS-LI-NE-
  || || ||
--ISALIGNED
```

Different alignments

(A)

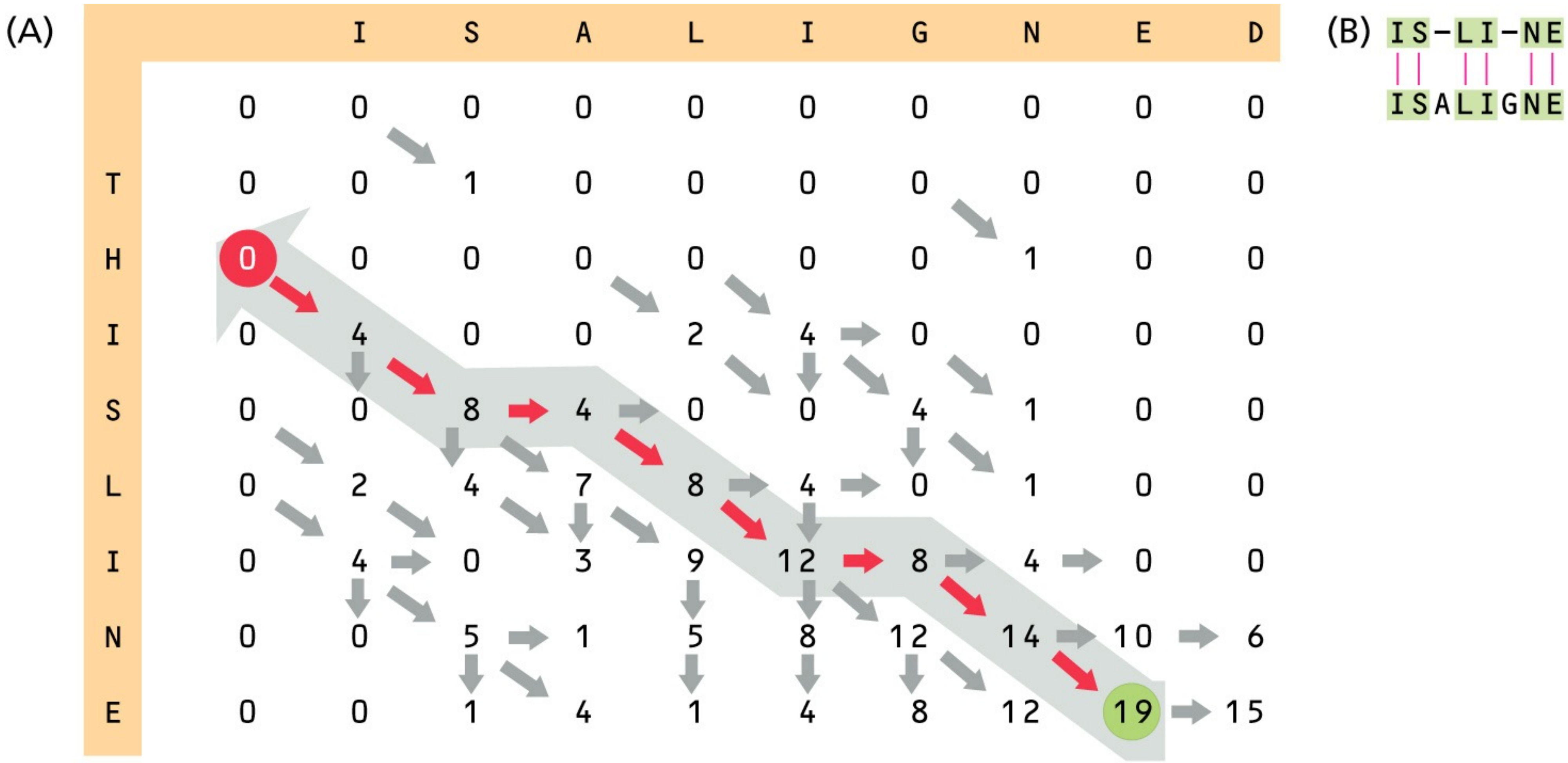


		I	S	A	L	I	G	N	E	D
	0	0	0	0	0	0	0	0	0	0
T	0	0	1	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	1	0	0
I	0	4	0	0	2	4	0	0	0	0
S	0	0	8	1	0	0	4	1	0	0
L	0	2	0	7	5	2	0	1	0	0
I	0	4	0	0	9	9	1	0	0	0
N	0	0	5	0	1	6	9	7	0	1
E	0	0	0	4	0	0	4	9	12	4

(B)

```
SLI-NE
 || ||
ALIGNE
```

Different alignments



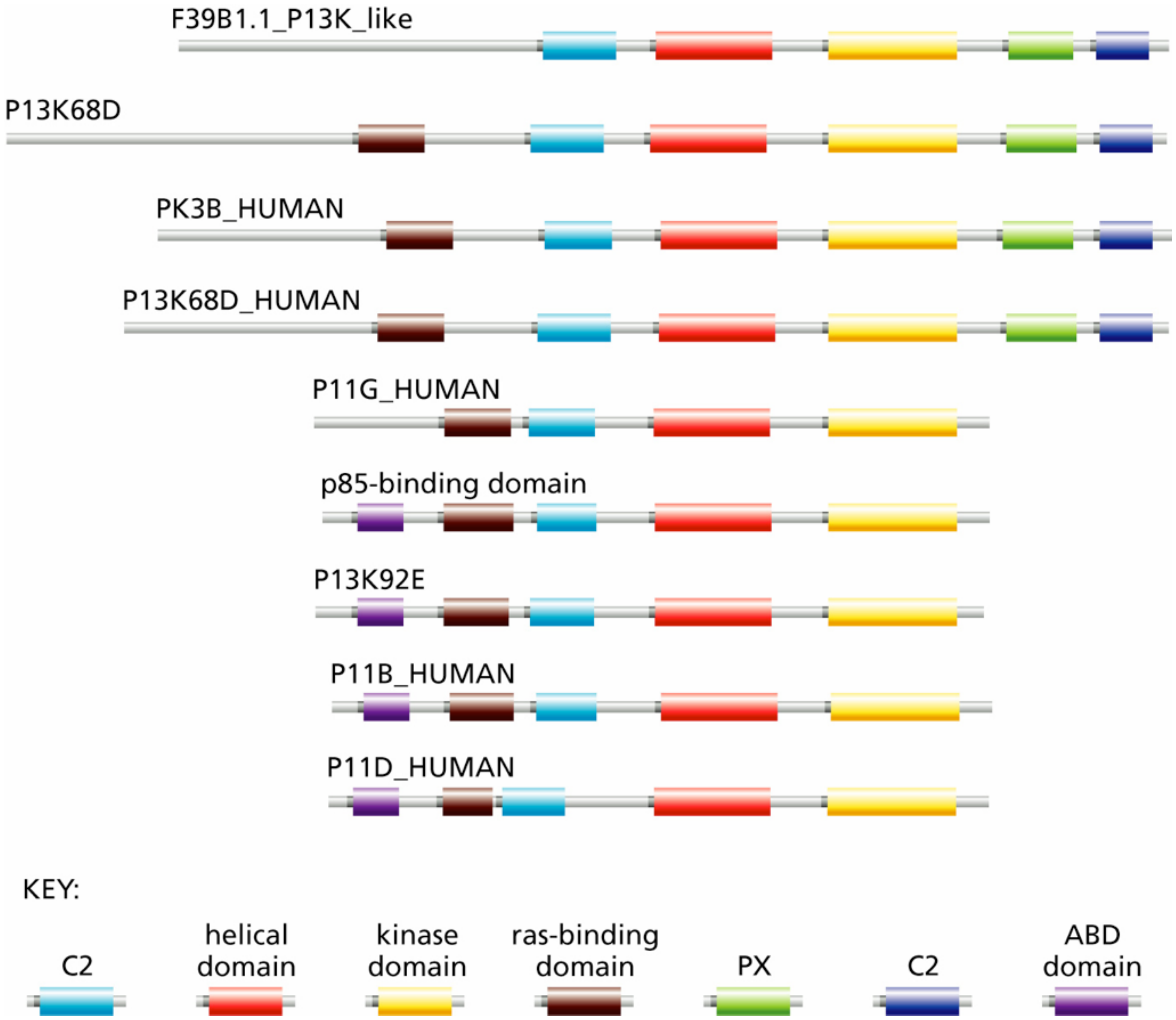
(A)

		I	S	A	L	I	G	N	E	D
	0	0	0	0	0	0	0	0	0	0
T	0	0	1	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	1	0	0
I	0	4	0	0	2	4	0	0	0	0
S	0	0	8	4	0	0	4	1	0	0
L	0	2	4	7	8	4	0	1	0	0
I	0	4	0	3	9	12	8	4	0	0
N	0	0	5	1	5	8	12	14	10	6
E	0	0	1	4	1	4	8	12	19	15

(B)

```
IS-LI-NE
||  || ||
ISALIGNE
```

Multidomain proteins



Next lecture

- O(nm) is too slow. How to speed up
- When is a "score" significant.