Intro to Alignment Algorithms:

Global and Local

Sequence Comparison

Biomolecular sequences

- DNA sequences (string over 4 letter alphabet {A, C, G, T})
- RNA sequences (string over 4 letter alphabet {ACGU})
- Protein sequences (string over 20 letter alphabet {Amino Acids})

Sequence similarity helps in the discovery of genes, and the prediction of structure and function of proteins.

The Basic Similarity Analysis Algorithm

Global Similarity

- Scoring Schemes
- Edit Graphs
- Alignment = Path in the Edit Graph
- The Principle of Optimality
- The Dynamic Programming Algorithm
- The Traceback

Sequence Alignment

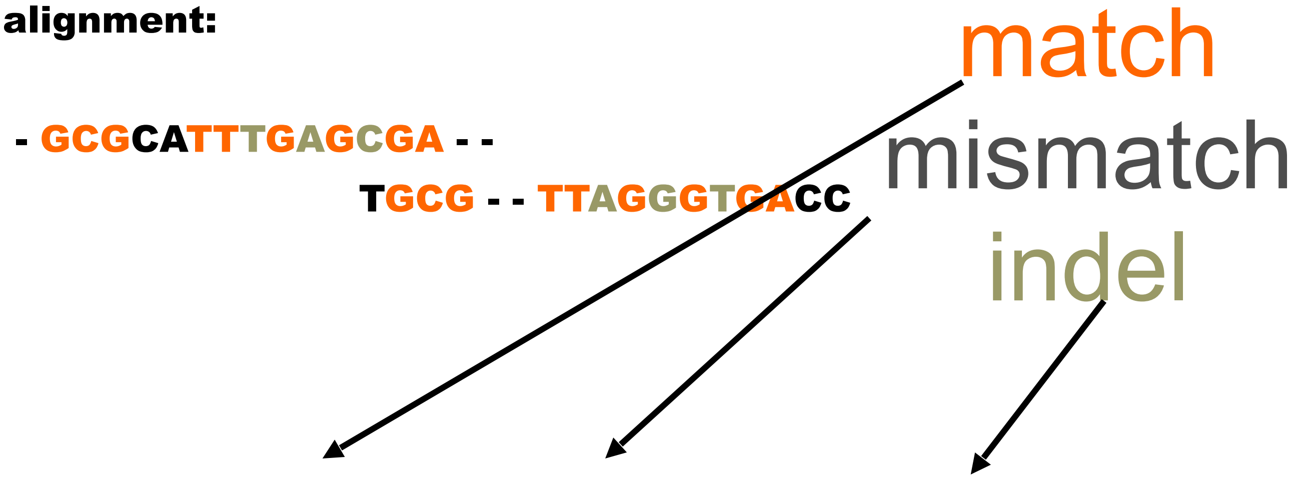
Input: two sequences over the same alphabet

Output: an alignment of the two sequences

Example:

- **GCGCATTTGAGCGA**
- **TGCGTTAGGGTGACCA**

A possible alignment:



- GCGCATTTGAGCGA - -

TGCG - - TTAGGGTGACC

Consider two sequences

$$X = x_1 x_2 ... x_n$$

x_i, y_j belong to Σ

$$Y = y_1 y_2 ... y_m$$

Over the alphabet

$$\Sigma = \{A, C, G, T\}$$

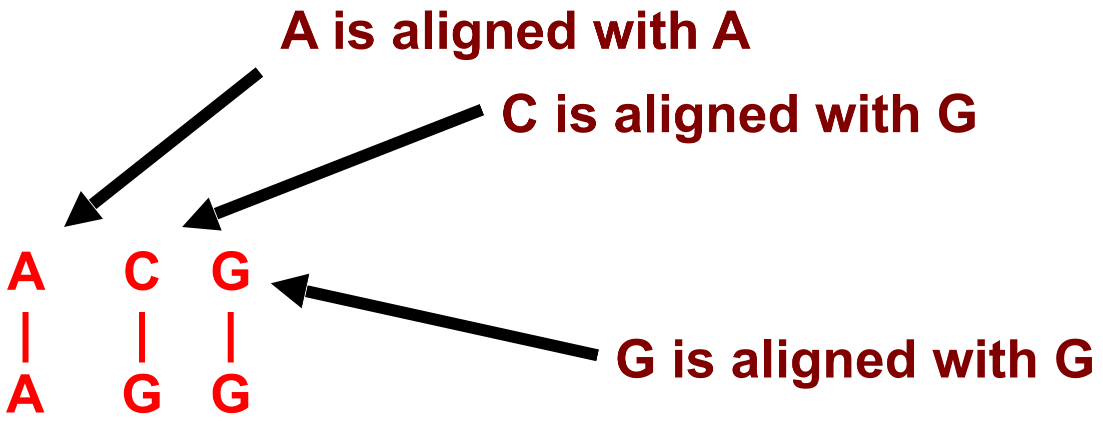
Scoring Schemes

Unit-score

δ	A	C	G	T	-
A	1	0	0	0	0
C	0	1	0	0	0
G	0	0	1	0	0
T	0	0	0	1	0
-	0	0	0	0	0

Alignment

ACG
| | |
AGG



A is aligned with A

C is aligned with G

G is aligned with G

A C G
| | |
A G G

Unit-cost

$$\text{Score} = \delta(A,A) + \delta(C,G) + \delta(G,G)$$

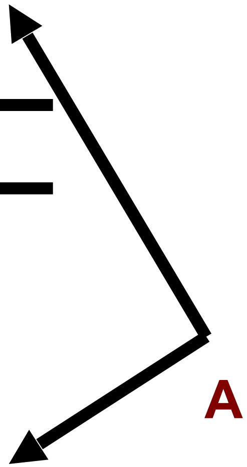
$$= 1 + 0 + 1 = 2$$

Gaps

"-" is the gap symbol

	ACATGGAAT ACAGGAAAT	ACAT GG - AAT ACA - GG AAAT
SCORE	7	8

	AAAGGG GGGAAA	- - - AAAGGG GGGAAA - - -
SCORE	0	3



OPTIMAL ALIGNMENTS

$\delta(x,y)$ = the score for aligning x with y

$\delta(x,-)$ = the score for aligning x with -

$\delta(-,y)$ = the score for aligning - with y

Alignment

A-CG - G
ATCGTG

Score

$$\delta_{(A,A)} + \delta_{(-,T)} + \delta_{(C,C)} + \delta_{(G,G)} + \delta_{(-,T)} + \delta_{(G,G)}$$

THE SUM OF THE SCORES OF THE PAIRWISE ALIGNED SYMBOLS

Scoring Scheme

Dayhoff score

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

...

PTIPLSRLFDNAMLRAHRLHQ
SAIENQRLFNIAVSRVQHLHL

Partial alignment for Monkey and Trout somatotropin proteins

Scoring Functions

Mutations= Substitutions, Insertions, Deletions

Scoring function = a sum of a terms each for a pair of aligned residues, and for each gap

The meaning = log of the relative likelihood that the sequences are related, compared to being unrelated

Identities and conservative substitutions are **Positive terms**

Non-conservative substitutions are **Negative terms**

The Edit Graph

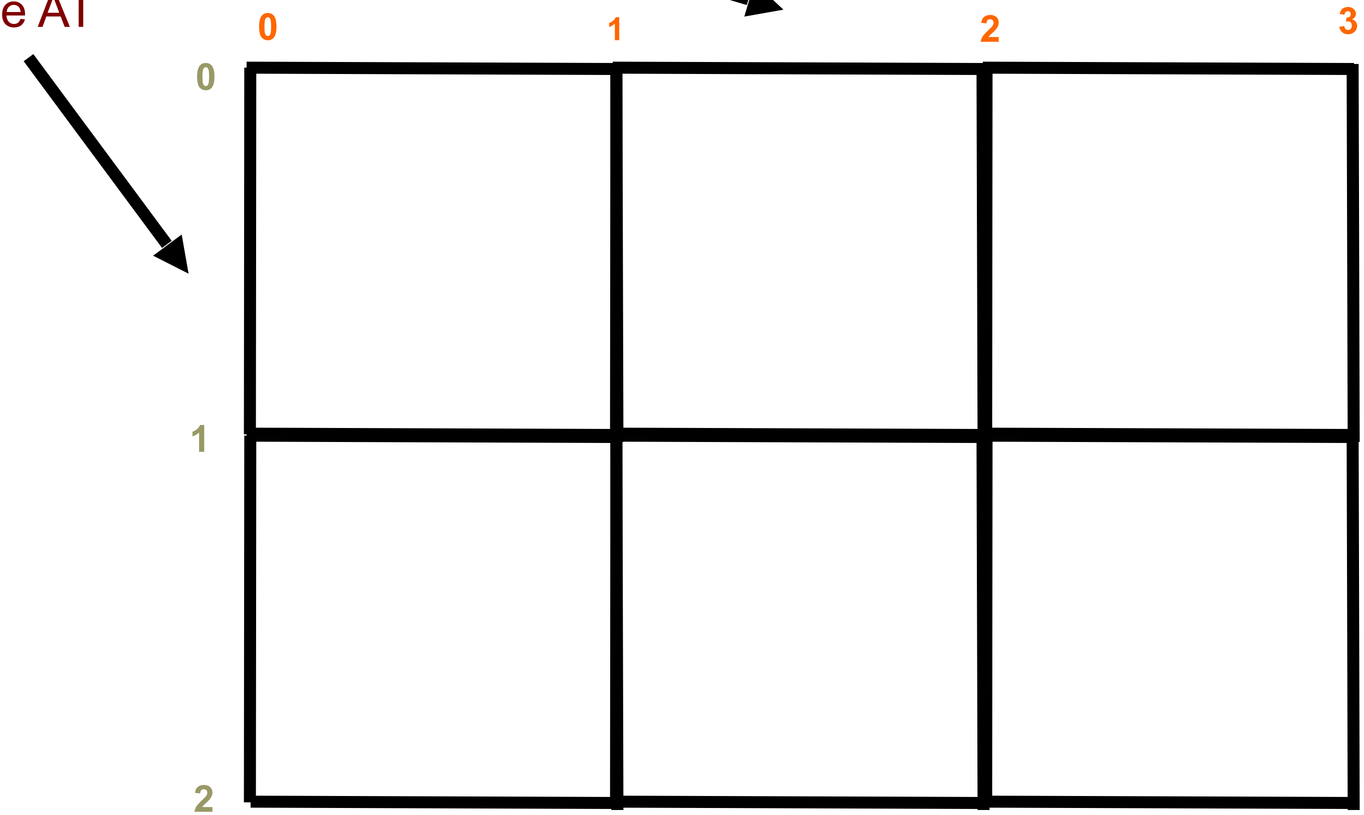
Suppose that we want to align AGT with AT

We are going to construct a graph where alignments between the two sequences correspond to paths between the begin and and end nodes of the graph.

This is the Edit Graph

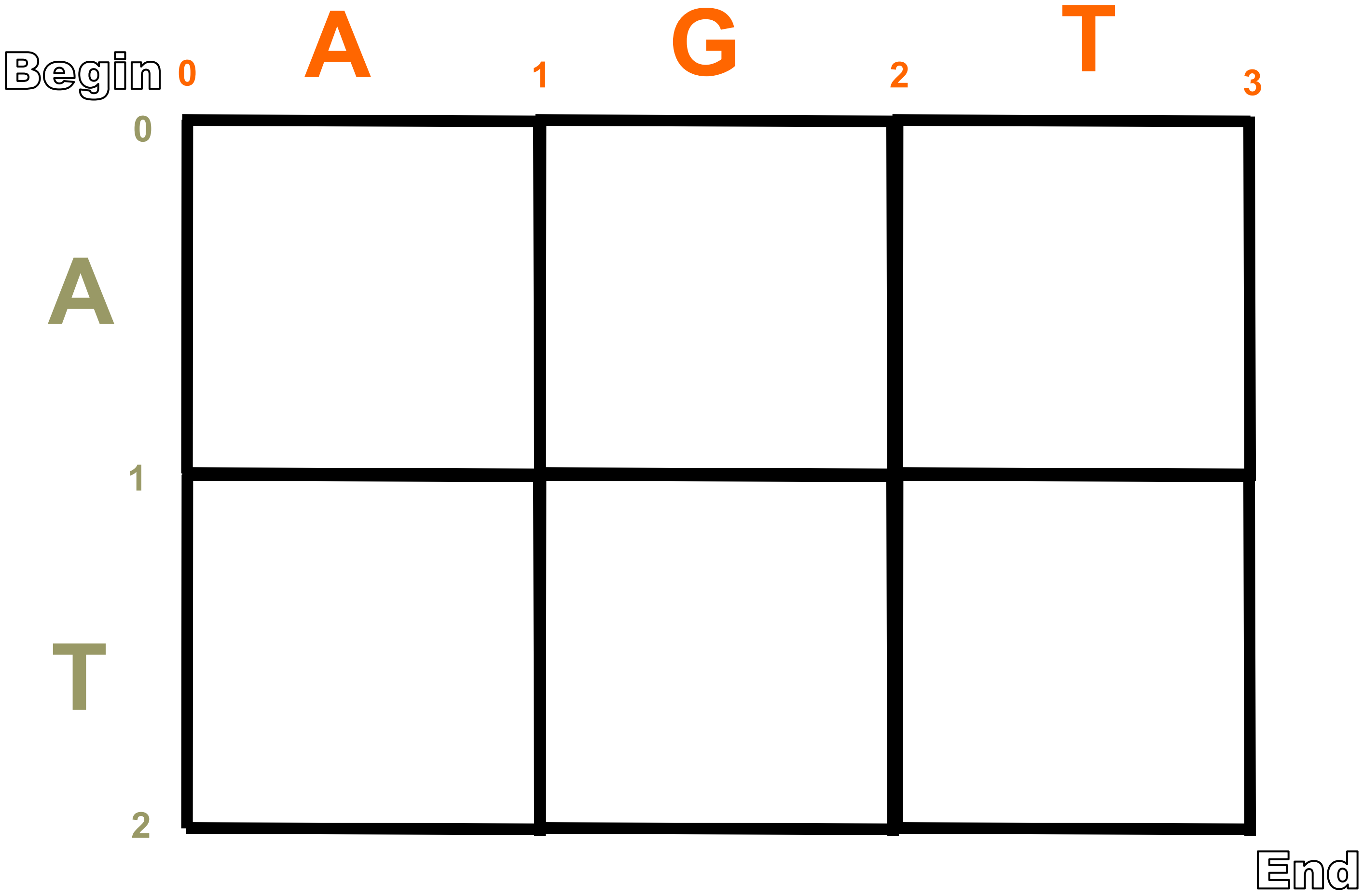
The sequence AGT

The sequence AT

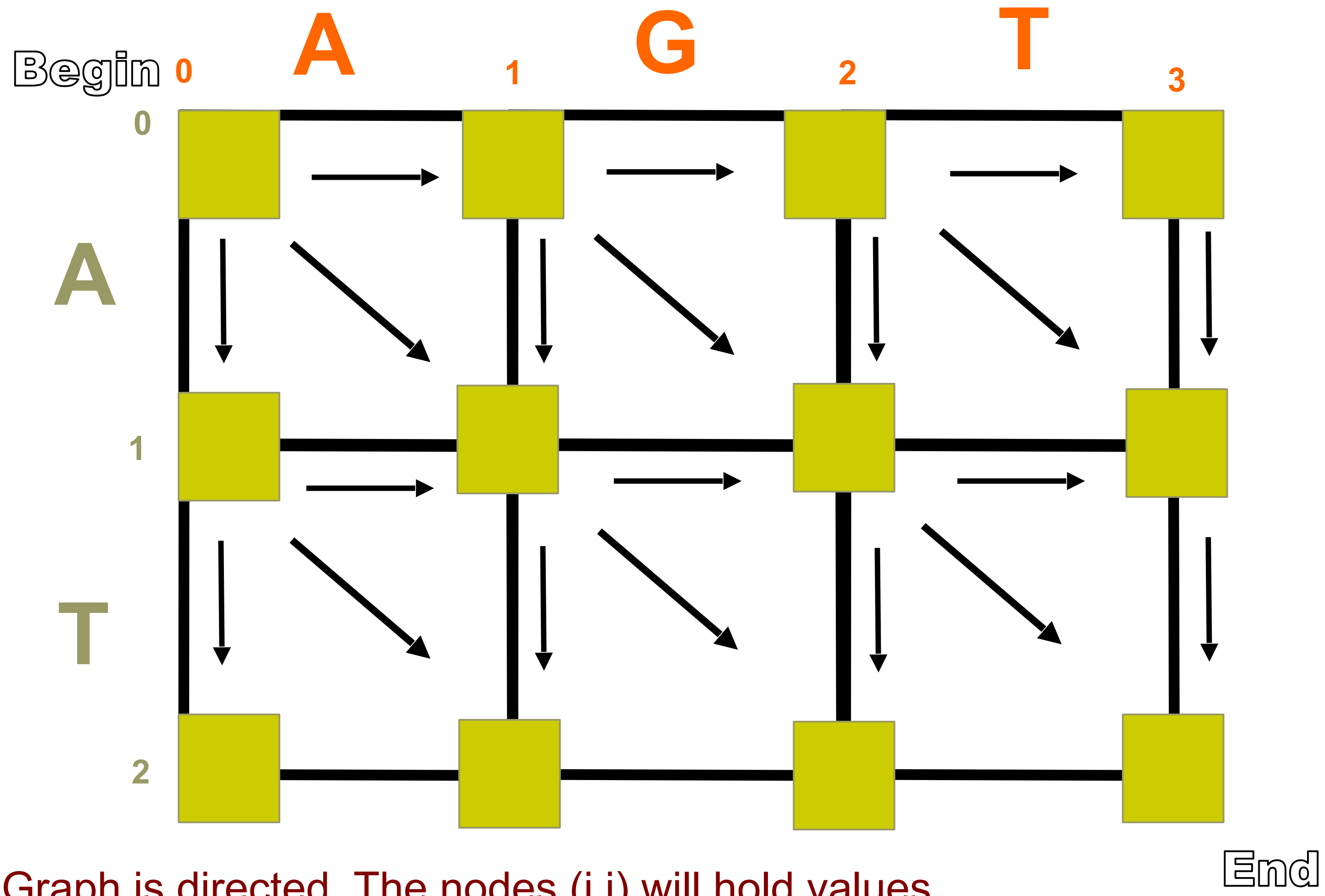


AGT has length 3
AT has length 2

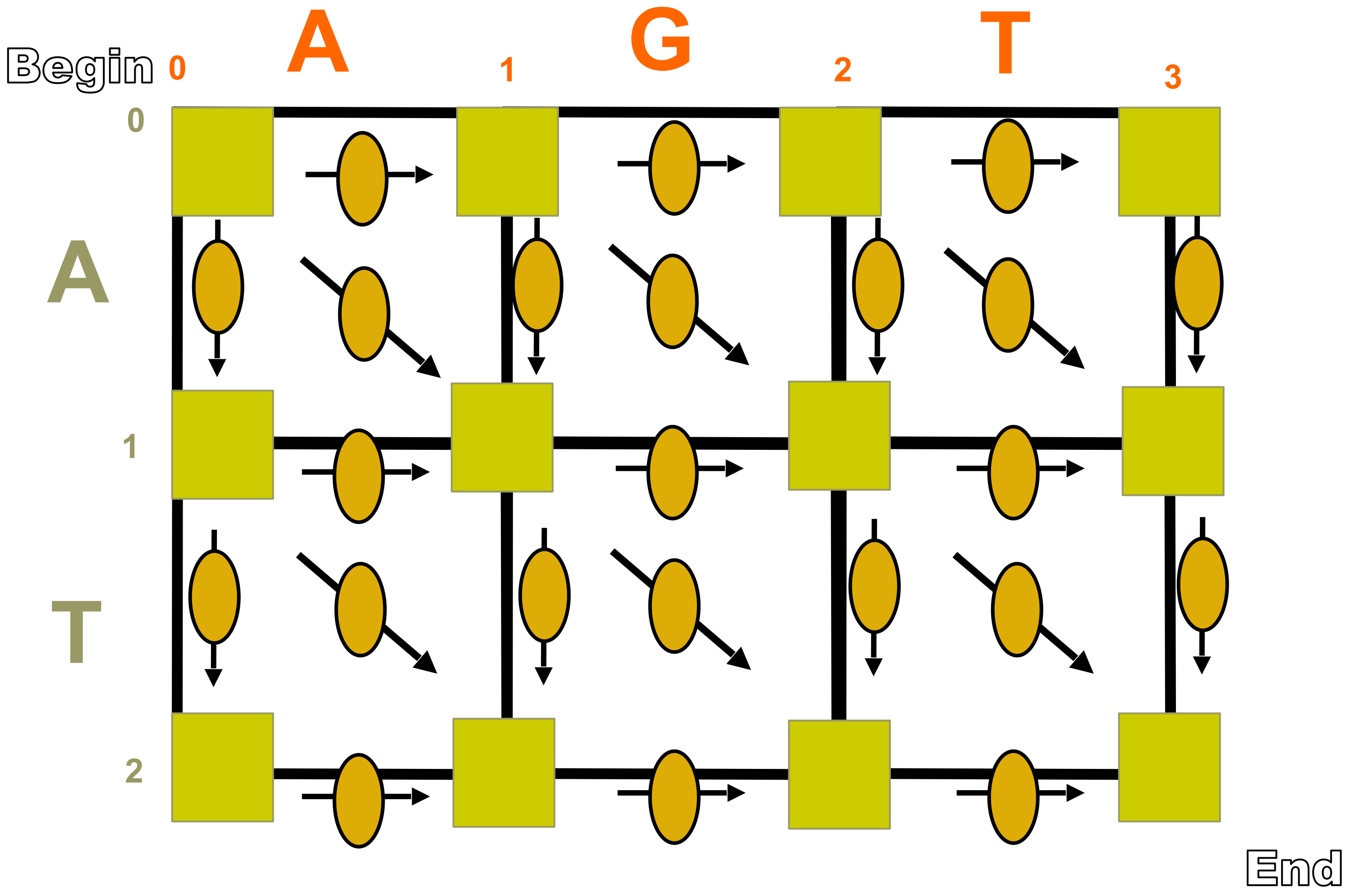
The Edit graph has (3+1)*(2+1) nodes



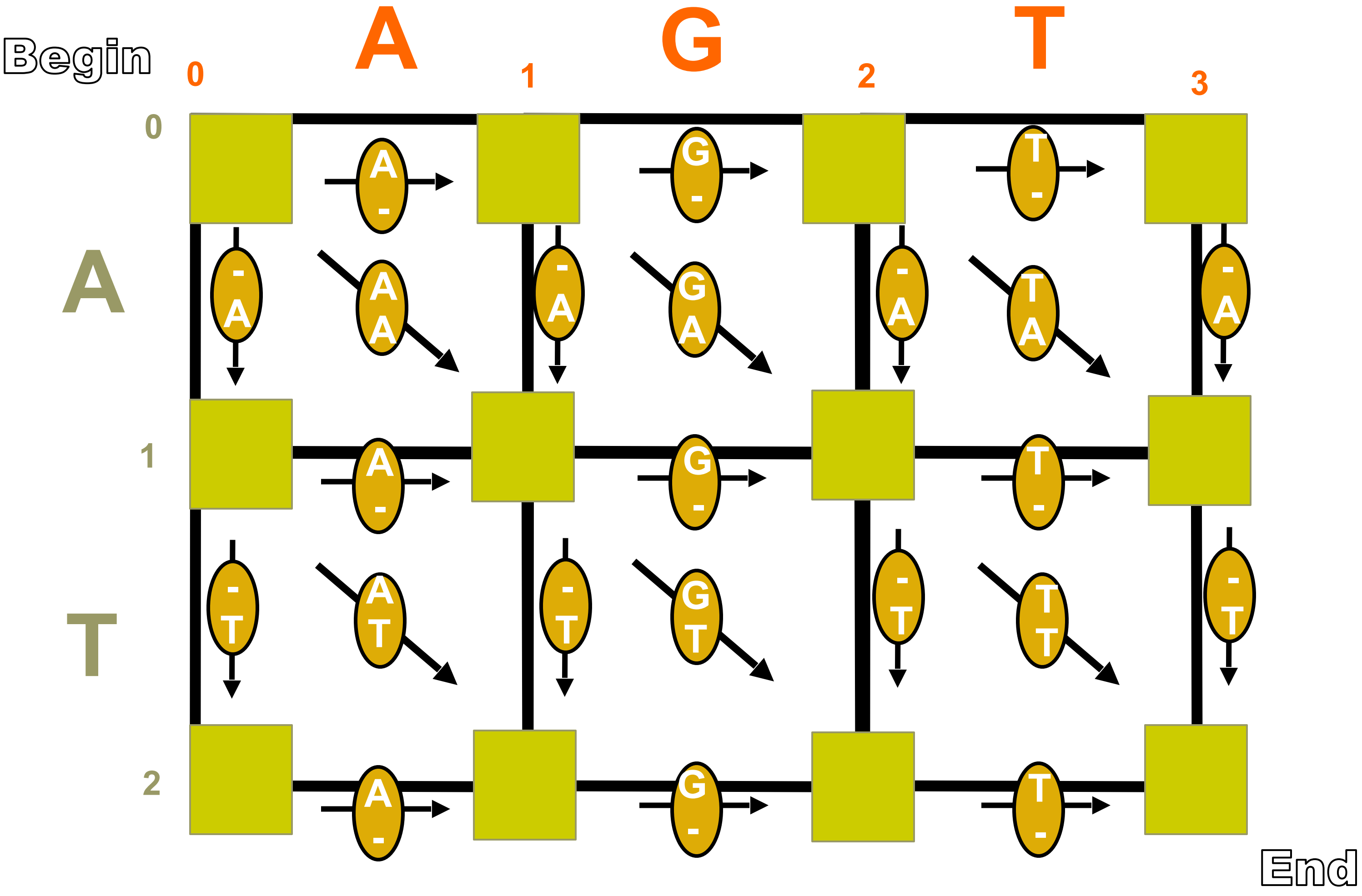
AGT indexes the columns, and AT indexes the rows of this “table”



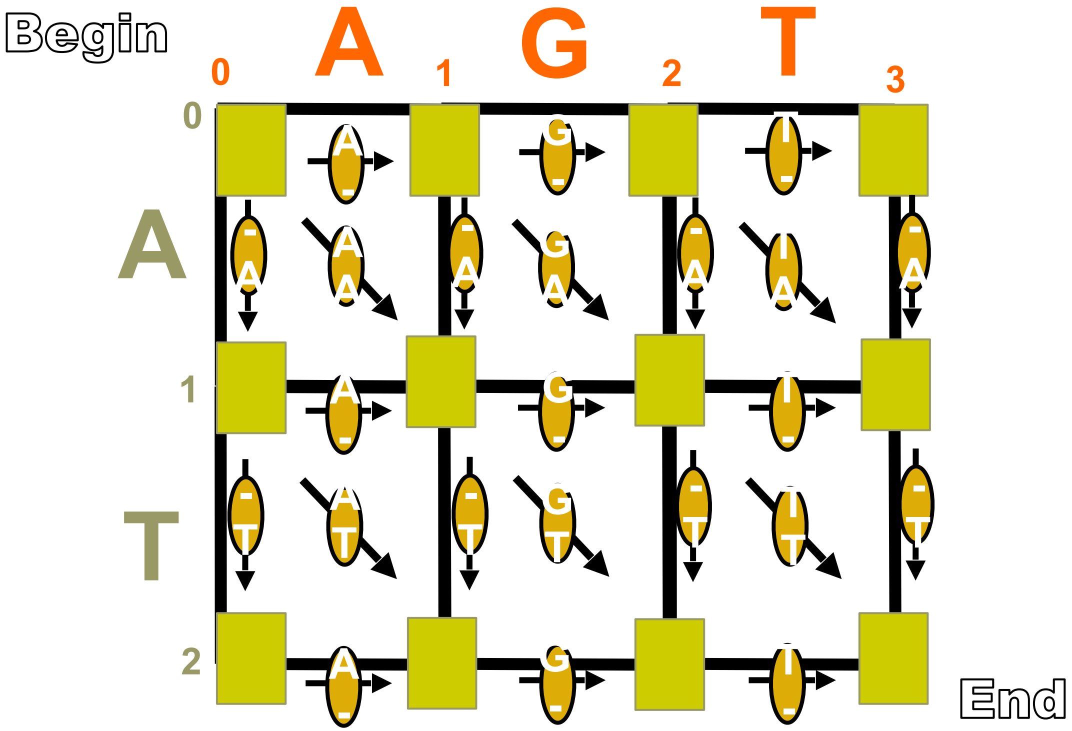
The Graph is directed. The nodes (i,j) will hold values.


Begin
0
A
1
G
2
T
3
0
A
1
T
2
End

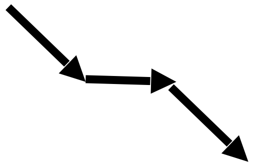
Directed edges get as labels pairs of aligned letters.



Alignment = Path in the Edit Graph



AGT
A-T



Every path from Begin to End corresponds to an alignment

Every alignment corresponds to a path between Begin and End

The Principle of Optimality

The optimal answer to a problem is expressed in terms of optimal answer for its sub-problems

Dynamic Programming

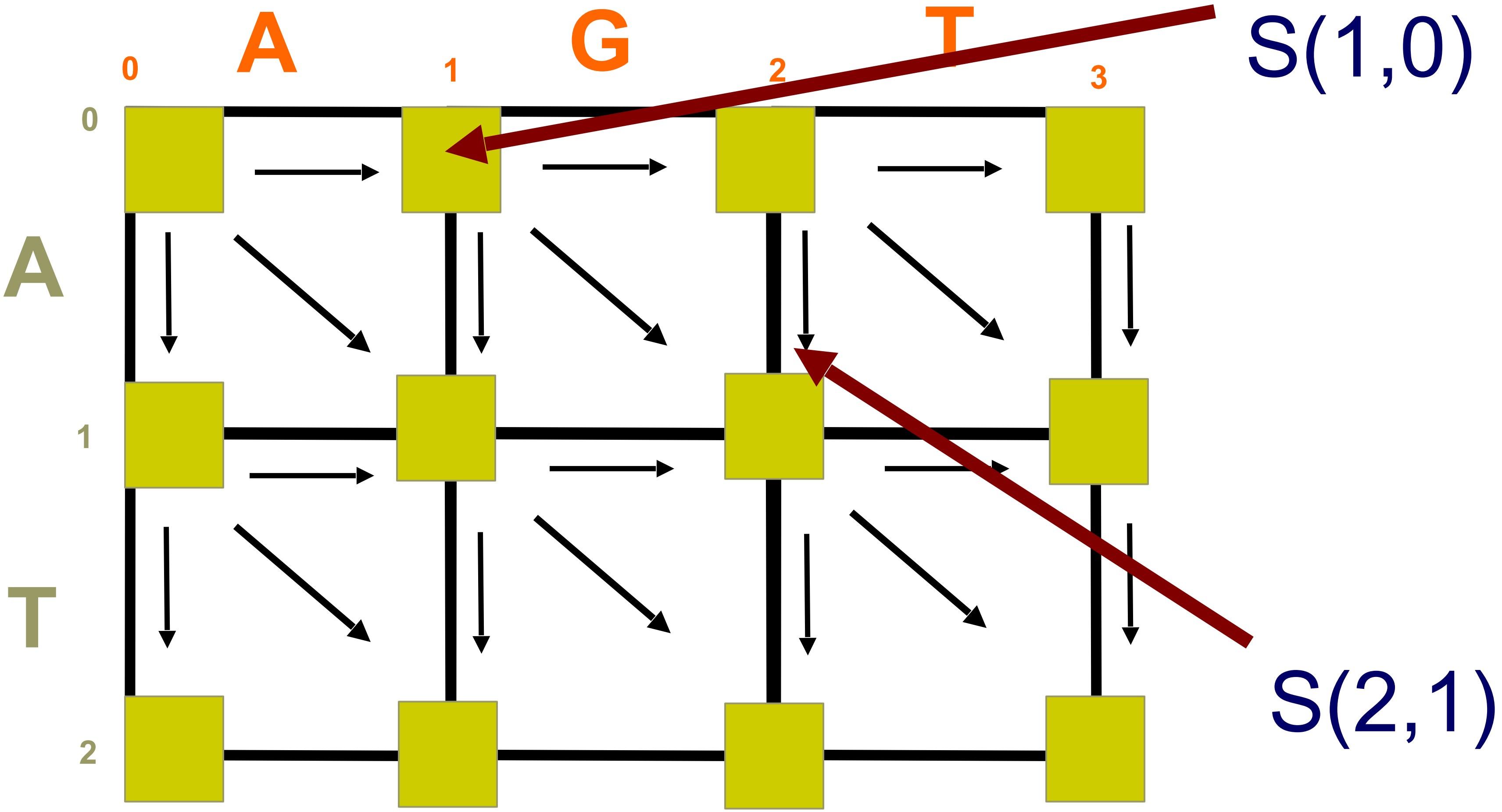
Given: Two sequences X and Y
Find: An optimal alignment of X with Y

Part 1: Compute first the optimal alignment score

Part 2: Construct optimal alignment

We are looking for the optimal alignment = maximal score path in the Edit Graph from the Begin vertex to the End vertex

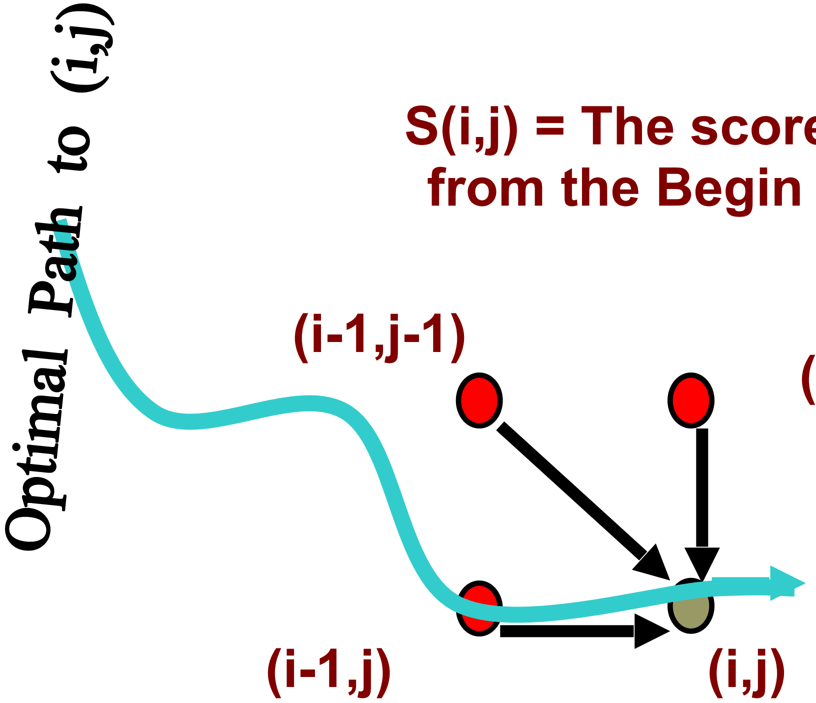
The DP Matrix S(i,j)



The DP Matrix

Matrix S =[S(i,j)]

S(i,j) = The score of the maximal cost path from the Begin Vertex and the vertex (i,j)



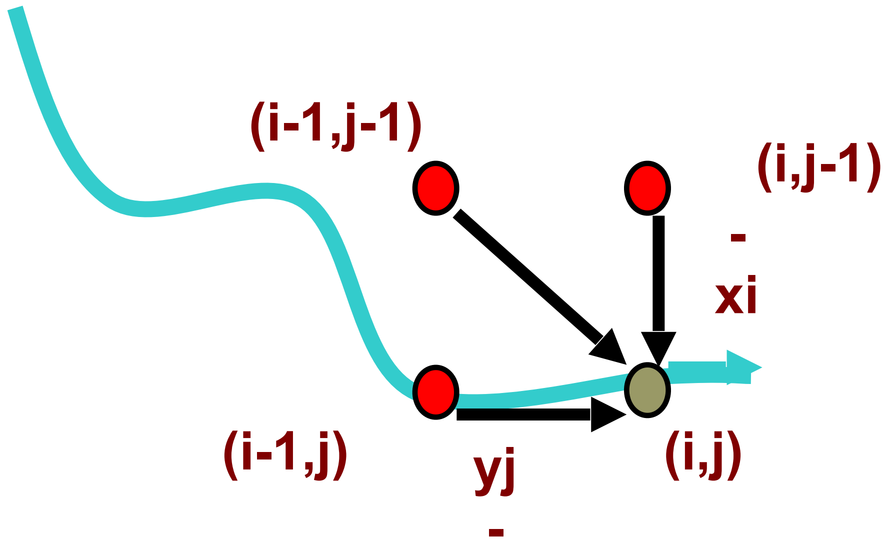
The optimal path to ● (i,j) must pass through one of the vertices

● (i-1,j)

● (i,j-1)

● (i-1,j-1)

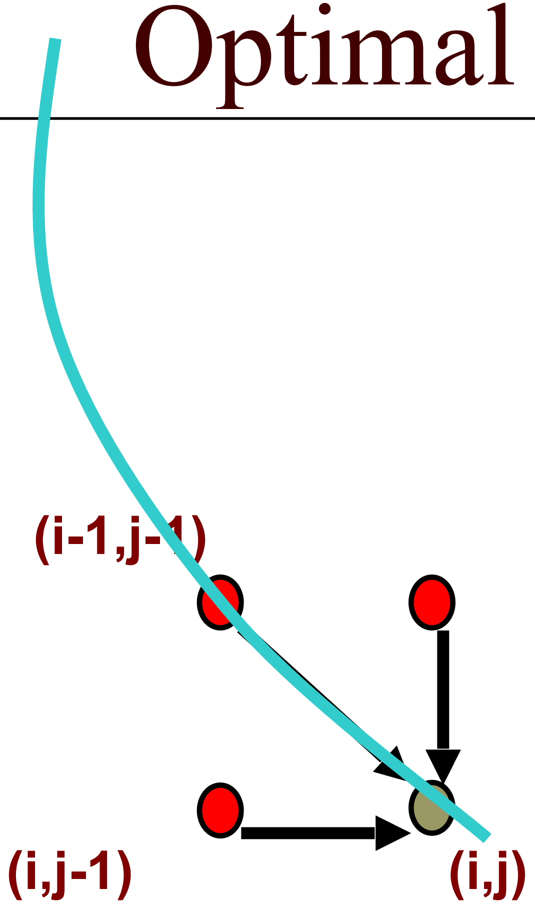
Opt path



S(i-1,j) + δ (- , yj)

Optimal path to (i-1,j) + δ(- , yj)

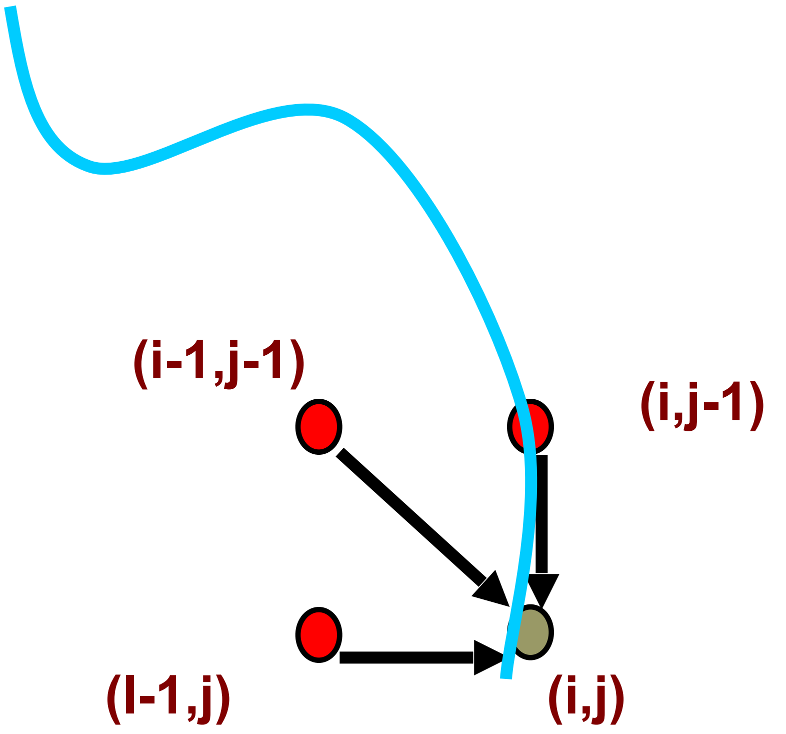
Optimal path



S(i-1,j-1) + δ (xi , yj)

Optimal path to (i-1,j-1)
+ δ (xi,yj)

Optimal path



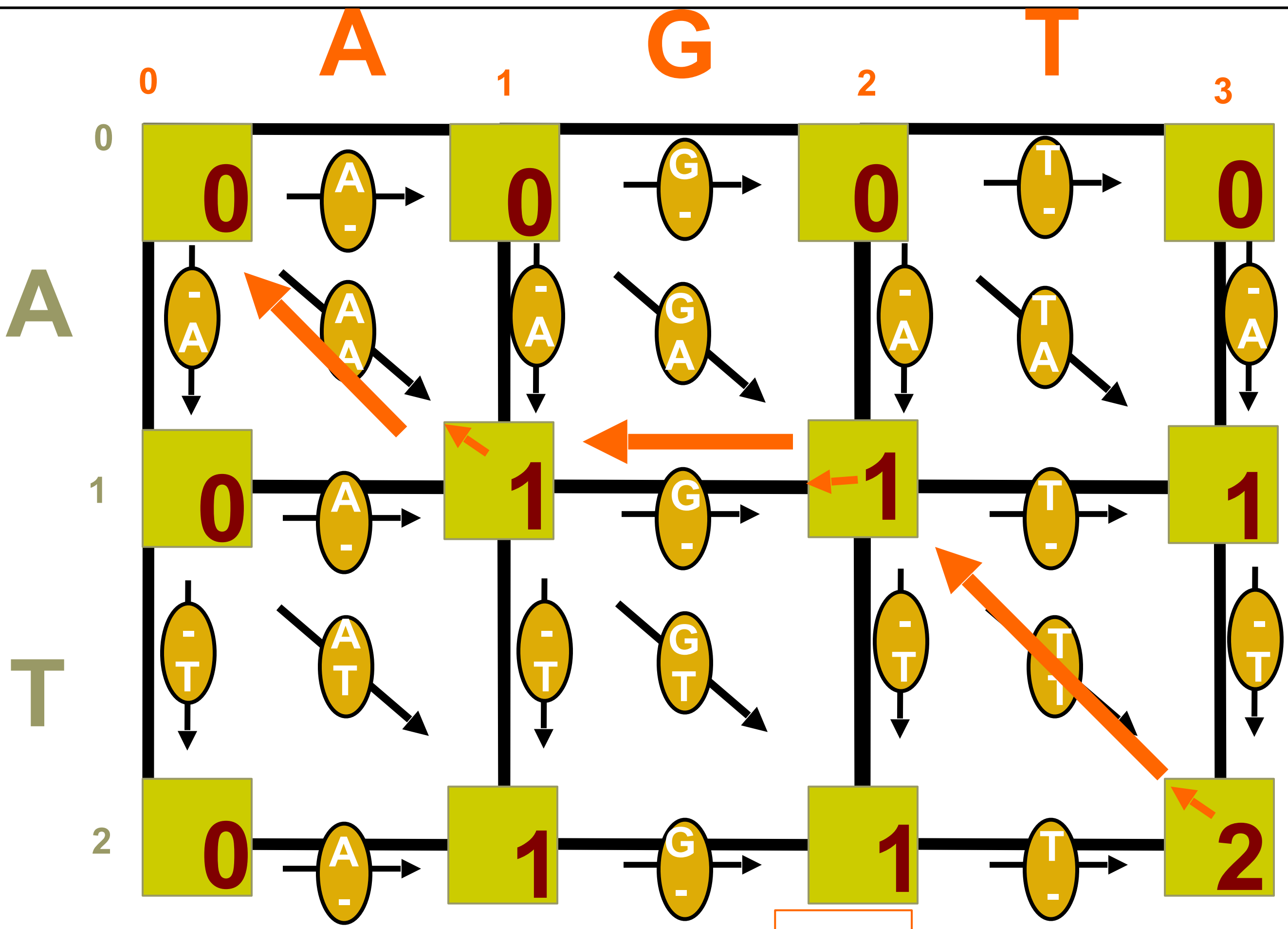
S(i,j-1) + δ (xi, -)

Optimal path to (i,j-1)
+ δ (xi,-)

The Basic ALGORITHM

$$S(i,j) = \text{MAX} \begin{cases} S(i-1, j-1) + \delta\ (xi, yj) \\ S(i-1, j) + \delta\ (xi, -) \\ S(i, j-1) + \delta\ (-, yj) \end{cases}$$

Optimal Alignment and Tracback



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

Optimal Alignment

AGT
A - T

The Basic ALGORITHM: Local Similarity

We add this

$$S(i,j) = \text{MAX} \begin{cases} 0, \\ S(i-1, j-1) + \delta\ (xi, yj), \\ S(i-1, j) + \delta\ (xi, -), \\ S(i, j-1) + \delta\ (-, yj) \end{cases}$$

General Scoring Schemes

Assumptions

1. Independence of mutations at different sites

Additive scoring scheme

2. Gaps of any length are considered one mutation

All of the efficient alignment algorithms -- employing on the dynamic programming method --are based fundamentally on the of the fact that the scoring function is additive.