Sequence alignment

Correspondence between bases of two DNA sequences, or between amino acids of two protein sequences

Alignment : 2 x k matrix (k ≥ m, n)

V = ACCTGGTAAA n = 10

W = ACTGCGTATA m = 10

8 matches
1 mismatches
1 deletions
1 insertions

V	A	C	C	T	G	—	G	T	A	A	A
W	A	C	—	T	G	C	G	T	A	T	A

“Goodness” of alignments

Given two sequences, there are many possible alignments

```
ATTTTCCC
ATTTACGC
```

distance=2

```
ATTT-TCCC
ATTTA-CGC
```

distance=3

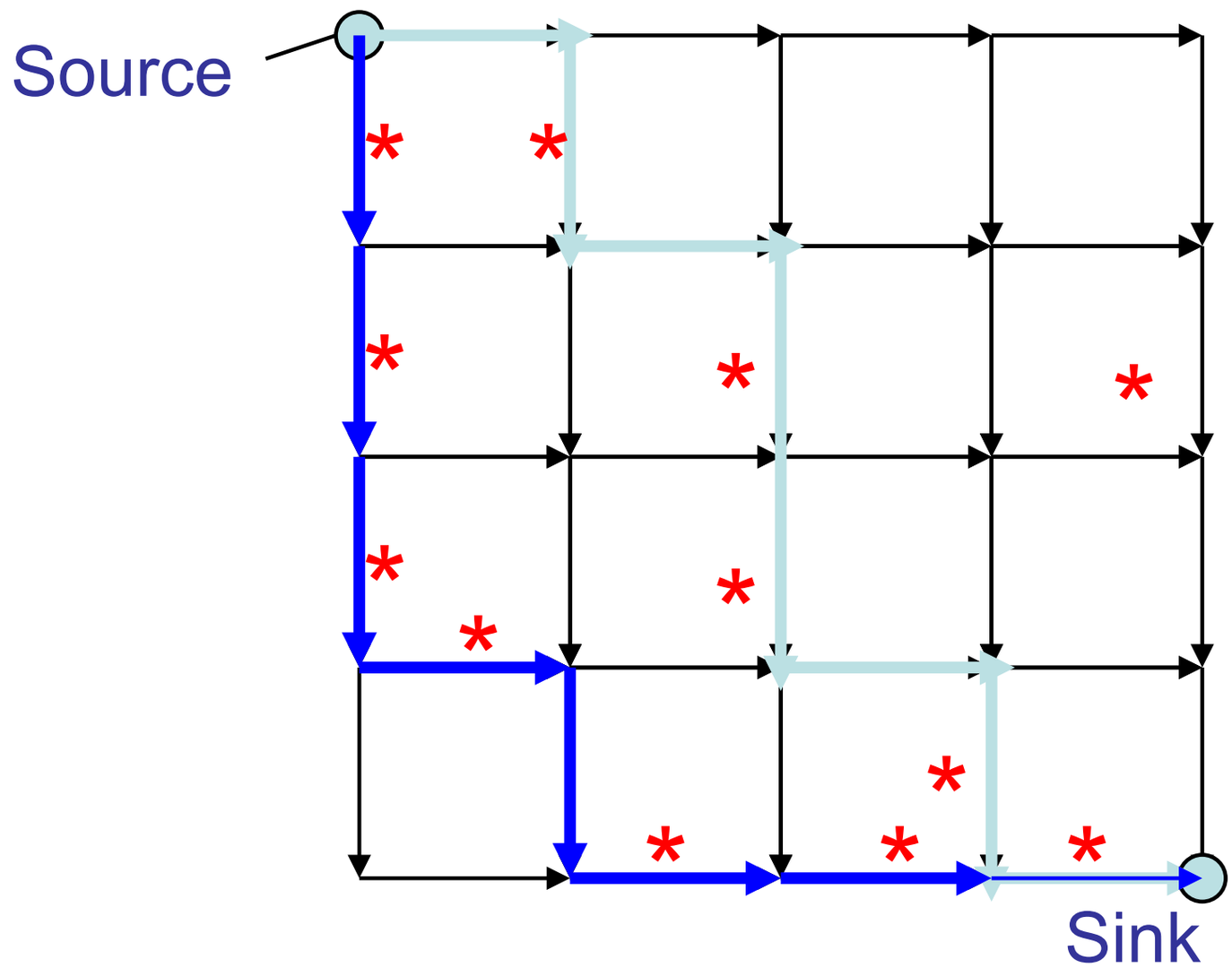
```
ATTTTCCC--------
--------ATTTACGC
```

distance=16

Edit distance: the total number of substitutions, insertions and deletions needed to transform one sequence to another

Manhattan tourist problem

Imagine seeking a path (from source to sink) to travel (only eastward and southward) with the most number of attractions (*) in the Manhattan grid



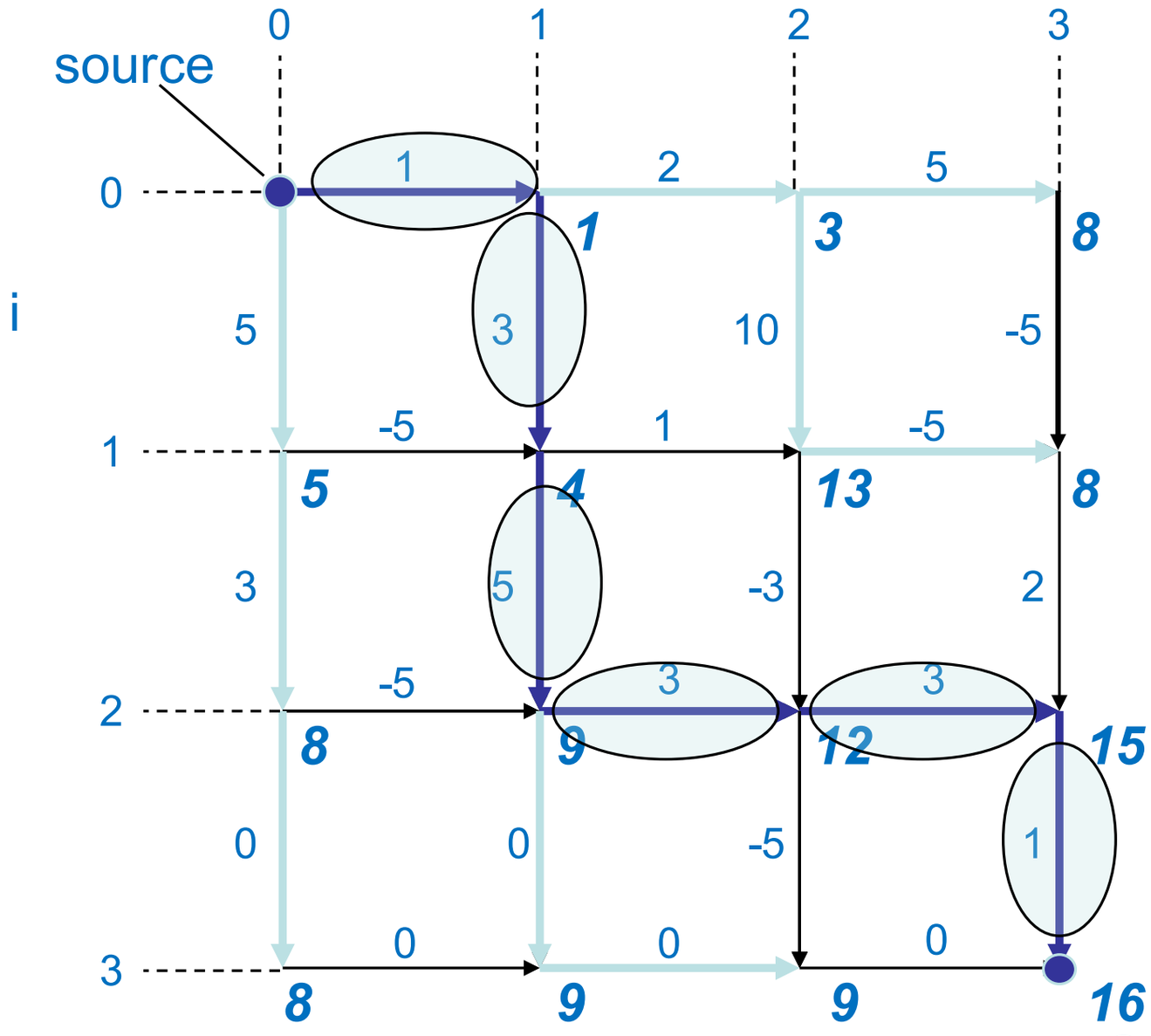
Recursive algorithm -> Dynamic programming

Function **MT**(*n,m*)

1. *x = MT(n-1,m)+*
   weight of the edge from *(n-1,m) to (n,m)*
2. *y = MT(n,m-1)+*
   weight of the edge from *(n,m-1) to (n,m)*
3. **return** *max{x,y}*

MT(x, y) returns the “most weighted” path from point (x, y) to the “sink”.

How to find the optimal path



$S_{3,3} = 16$

- Start from Sink.
- Find which of the two edges gave the "max". Take it.
- Repeat.

Recipe

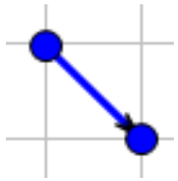
1. Identify subproblems

2. Write down recursions

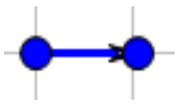
3. Make it dynamic-programming!

The edit distance problem

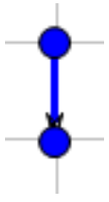
Match

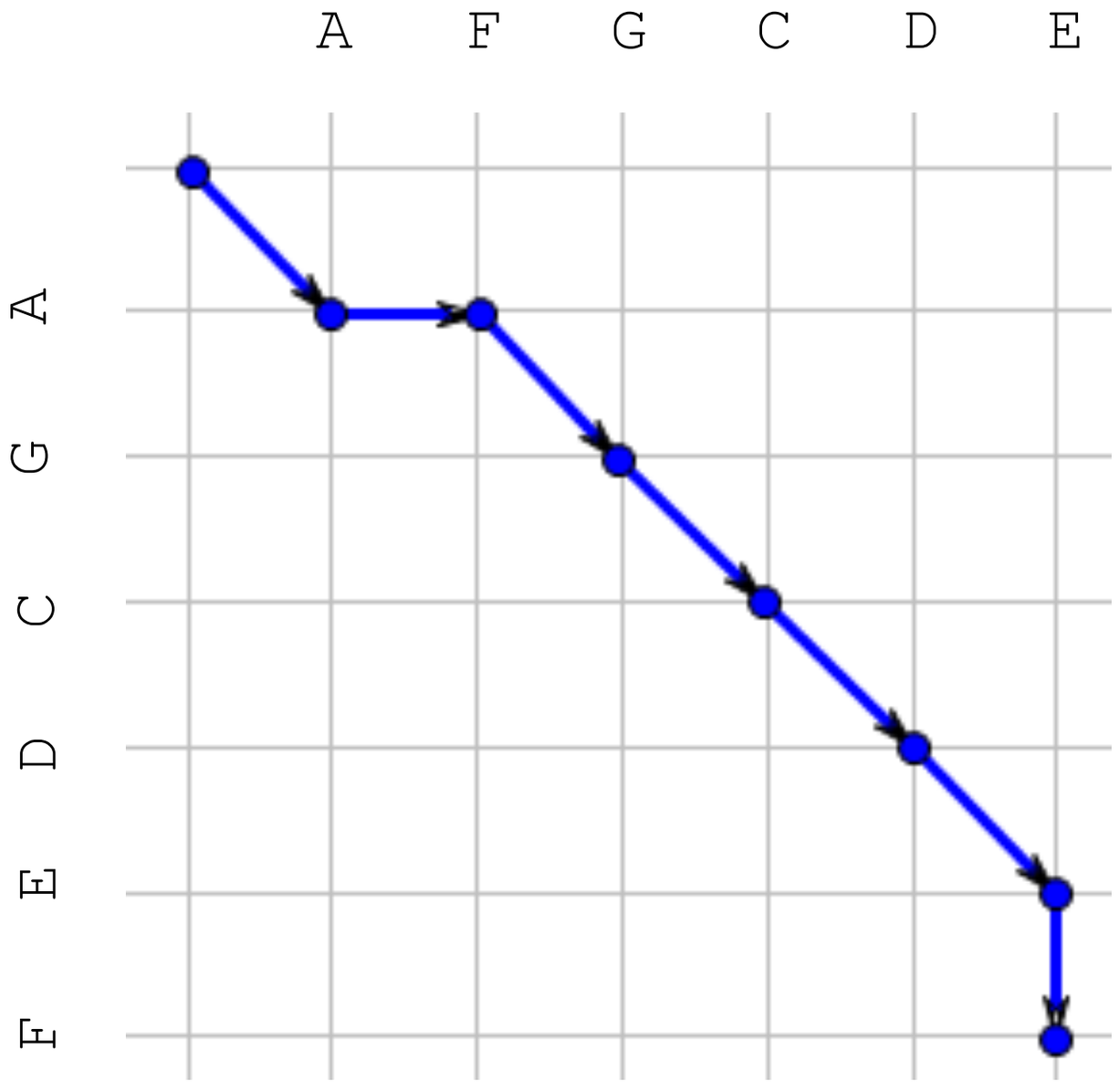


Insertion_X



Insertion_Y





A-GCDEF
AFGCDE-

Minimum Edit Distance

For sequence X and Y

- Initialization

  $D(i,0) = i$

  $D(0,j) = j$

- Recurrence Relation:

  For each $i = 1 \ldots M$

  For each $j = 1 \ldots N$

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 1; & \text{if } X(i) \neq Y(j) \\ 0; & \text{if } X(i) = Y(j) \end{cases} \end{cases}$$

- Termination:

  $D(N,M)$ is distance

Optimal alignment

- Base conditions:

  `D(i,0) = i` `D(0,j) = j`

- Recurrence Relation:

  `For each i = 1…M`

  `For each j = 1…N`

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 & \text{deletion} \\ D(i,j-1) + 1 & \text{insertion} \\ D(i-1,j-1) + \begin{cases} 1; & \text{if } X(i) \neq Y(j) \quad \text{substitution} \\ 0; & \text{if } X(i) = Y(j) \quad \text{match} \end{cases} \end{cases}$$

$$ptr(i,j) = \begin{cases} \text{LEFT} & \text{insertion} \\ \text{DOWN} & \text{deletion} \\ \text{DIAG} & \text{substitution} \quad \text{match} \end{cases}$$

Termination:

`D(N,M) is distance`

Complexity

- Time: O(nm)
- Space: O(nm)
- Backtrace O(n+m)

Is the edit distance the best way?

For sequence X and Y

- Initialization

  $D(i,0) = i$

  $D(0,j) = j$

- Recurrence Relation:

  For each $i = 1 \ldots M$

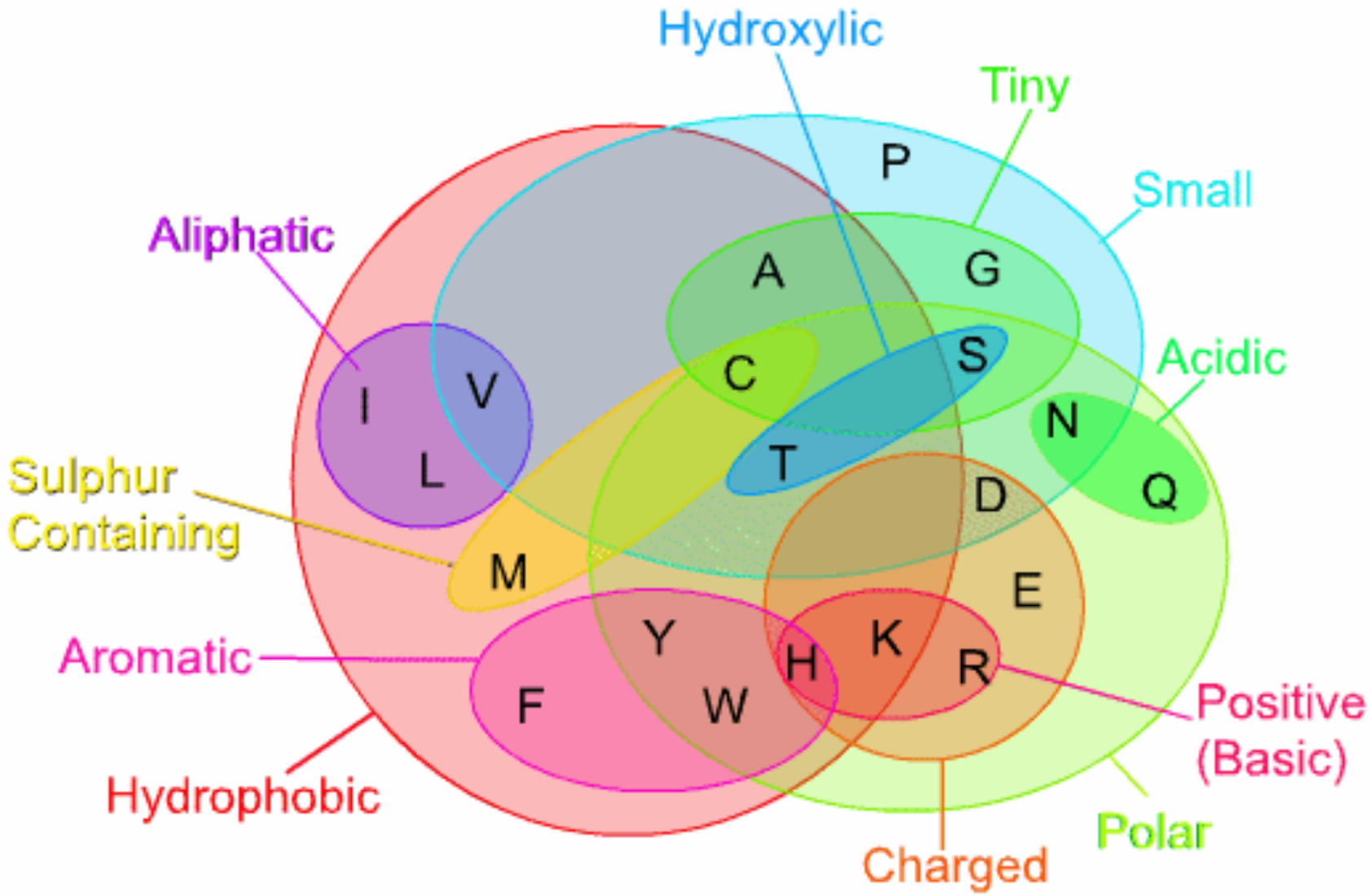
  For each $j = 1 \ldots N$

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 1; & \text{if } X(i) \neq Y(j) \\ 0; & \text{if } X(i) = Y(j) \end{cases} \end{cases}$$

- Termination:

  $D(N,M)$ is distance

Amino acids can share similar properties



Amino Acids

A alanine (ala)
R arginine (arg)
N asparagine (asn)
D aspartic acid (asp)
C cysteine (cys)
Q glutamine (gln)
E glutamic acid (glu)
G glycine (gly)
H histidine (his)
I isoleucine (ile)
L leucine (leu)
K lysine (lys)
M metioneine (met)
F phenyalanine (phe)
P proline (pro)
S serine (ser)
T threonine (thr)
W trytophan (trp)
Y tyrosine (tyr)

Weighted edit distance

- To generalize scoring for DNA/RNA, consider a 4x4 scoring matrix **S**.

- In the case of an amino acid sequence alignment, the scoring matrix would be a 20x20 size.

- The addition of d is to include the score for comparison of a gap character “-”.

- Two questions:
  - (a) What should **S** be?
  - (b) How do we find optimal scoring alignment?

Weighted edit distance

- To generalize scoring for DNA/RNA, consider a (4+1) x(4+1) scoring matrix **S**.

- In the case of an amino acid sequence alignment, the scoring matrix would be a (20+1)x(20+1) size.

- The addition of d is to include the score for comparison of a gap character “-”.

- Two questions:
  - (a) What should **S** be?
  - (b) How do we find optimal scoring alignment?

Traditionally, people tend to maximize the alignment score with a negative gap penalty score

BLOcks SUbstitution Matrix (BLOSUM)

	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
Ala	4																			
Arg	-1	5																		
Asn	-2	0	6																	
Asp	-2	-2	1	6																
Cys	0	-3	-3	-3	9															
Gln	-1	1	0	0	-3	5														
Glu	-1	0	0	2	-4	2	5													
Gly	0	-2	0	-1	-3	-2	-2	6												
His	-2	0	1	-1	-3	0	0	-2	8											
Ile	-1	-3	-3	-3	-1	-3	-3	-4	-3	4										
Leu	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4									
Lys	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5								
Met	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5							
Phe	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6						
Pro	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7					
Ser	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4				
Thr	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5			
Trp	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11		
Tyr	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	
Val	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

amino acids

BLOcks SUbstitution Matrix (BLOSUM)

Introduced by Henikoff & Henikoff in 1992

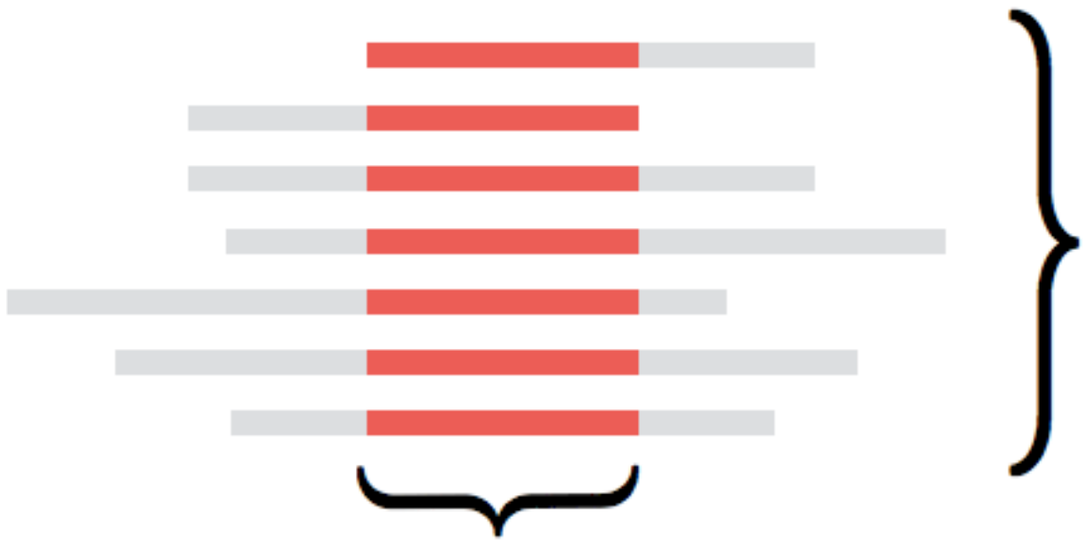
Start with the BLOCKS database

1. Look for conserved (gapless, >=62% identical) regions in alignments.

2. Count all pairs of amino acids in each column of the alignments.

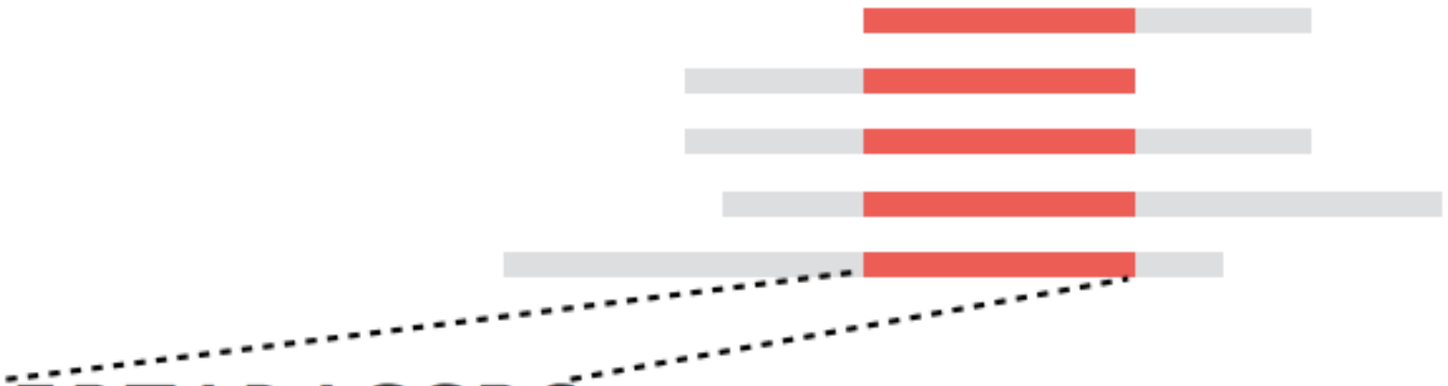
3. Use amino acid pair frequencies to derive “score” for a mutation/replacement

1. Look for conserved (gapless) regions in alignments.



sequences too similar are “clustered” & represented by either a single sequence, or a weighted combination of the cluster members

2. Count all pairs of amino acids in each column of the alignments.

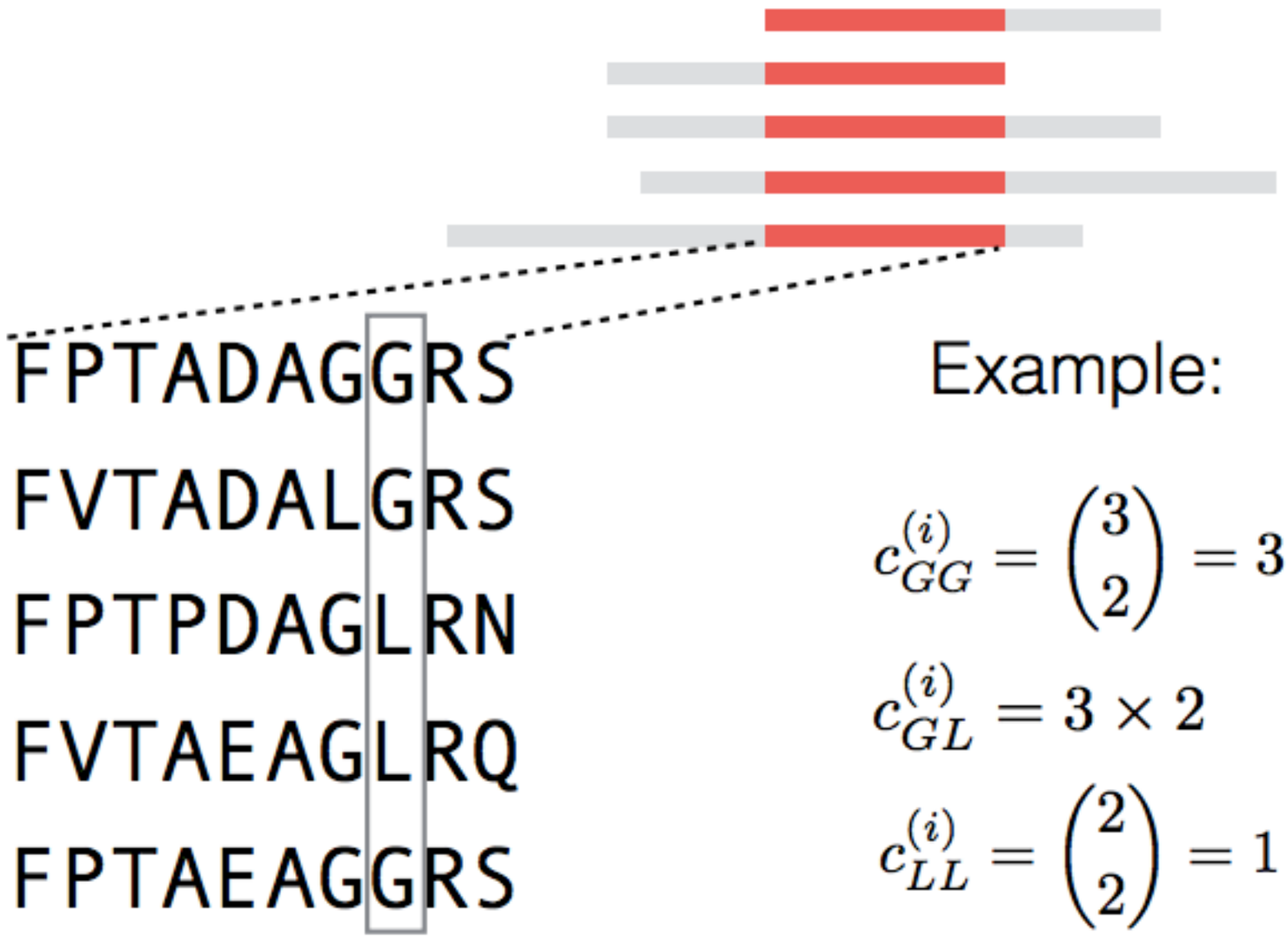


FPTADAGGRS
FVTADALGRS
FPTPDAGLRN
FVTAEAGIRQ
FPTAEAGGRS

$$c_{AB}^{(i)} = \begin{cases} \binom{c_A^{(i)}}{2} & \text{if } A = B \\ c_A^{(i)} \times c_B^{(i)} & \text{otherwise} \end{cases}$$

$c_A^{(i)} =$ num. of occurrences of A in column i

2. Count all pairs of amino acids in each column of the alignments.



Example:

$$c_{GG}^{(i)} = \binom{3}{2} = 3$$

$$c_{GL}^{(i)} = 3 \times 2$$

$$c_{LL}^{(i)} = \binom{2}{2} = 1$$

In this column, there are 3 ways to pair G with G, 6 potential ways to pair G with L and 1 potential way to pair L with L.

3. Use amino acid pair frequencies to derive “score” for a mutation/replacement

Total # of potential align. between A & B: $c_{AB} = \sum_i c_{AB}^{(i)}$

Total number of pairwise char. alignments: $T = \sum_{A \geq B} c_{AB}$

Normalized frequency of aligning A & B: $q_{AB} = \frac{c_{AB}}{T}$

3. Use amino acid pair frequencies to derive “score” for a mutation/replacement

Probability of occurrence of amino acid A in any {A,B} pair:

$$p_A = q_{AA} + \sum_{A \neq B} q_{AB}$$

Expected likelihood of each {A,B} pair, assuming independence:

$$e_{AB} = \begin{cases} (p_A)(p_B) = (p_A)^2 & \text{if } A = B \\ (p_A)(p_B) + (p_B)(p_A) = 2(p_A)(p_B) & \text{otherwise} \end{cases}$$

How to get a good substitution score?

Hypothesis we wish to test; two amino acids are correlated because they are homologous.

$$\text{score} = \log \text{ odds ratio} = s_{AB} \propto \log\left(\frac{\text{observed}}{\text{expected}}\right)$$

Null hypothesis; two amino acids occur independently (and are uncorrelated and unrelated).

$$\text{score} = \log \text{ odds ratio} = s_{AB} \propto \log\left(\frac{\text{observed}}{\text{expected}}\right)$$

$$\text{score} = \log \text{ odds ratio} = \; s_{AB} = \text{Round}\left(\left(\frac{1}{\lambda}\right)\log_2\left(\frac{q_{AB}}{e_{AB}}\right)\right)$$

Scaling factor used to produce scores that can be rounded to integers; set to 0.5 in H&H ‘92.

Henikoff, S.; Henikoff, J.G. (1992). "Amino Acid Substitution Matrices from Protein Blocks". PNAS 89 (22): 10915–10919.

TODO: compute weighted edit distance?