Sequence alignment Jan 17 2022

Steve Rozen steverozen@gmail.com

Global alignment example

```
ACAAGT-
|  |  ||
ATA-GTA
```

Match +1
Mismatch -1
Gap -1

Alignment Score: 4 – 3 = 1

Local alignment example (same 2 sequences)

```
ACAAGT
   | | |
 ATAGTA
```

Match +1
Mismatch -1
Gap -1

Alignment Score: 3

Global versus Local

```
ACAAGT-
|  |  ||
ATA-GTA
```

Must match all of both sequences

Alignment Score: 4 – 3 = 1

```
ACAAGT
   |||
 ATAGTA
```

No need to match all of both sequences

Alignment Score: 3

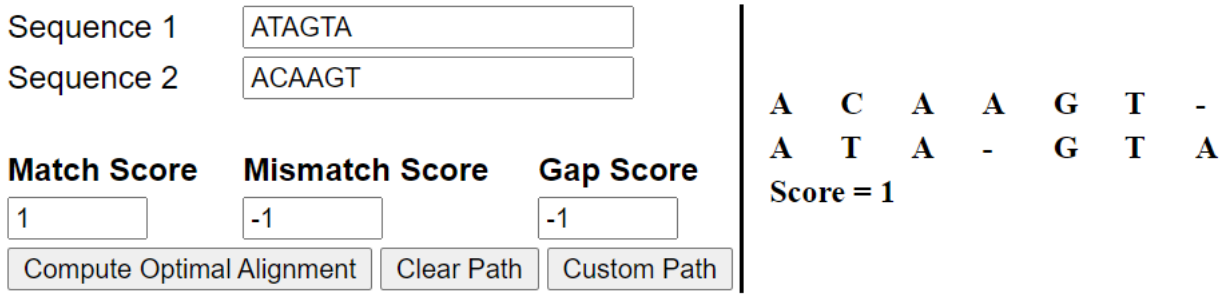
How to compute a global alignment
Needleman-Wunsch algorithm

Well-known example of “dynamic programming”
in which problem is decomposed into sub-problems, and optimum of larger problem
is computed from optima of sub-problems

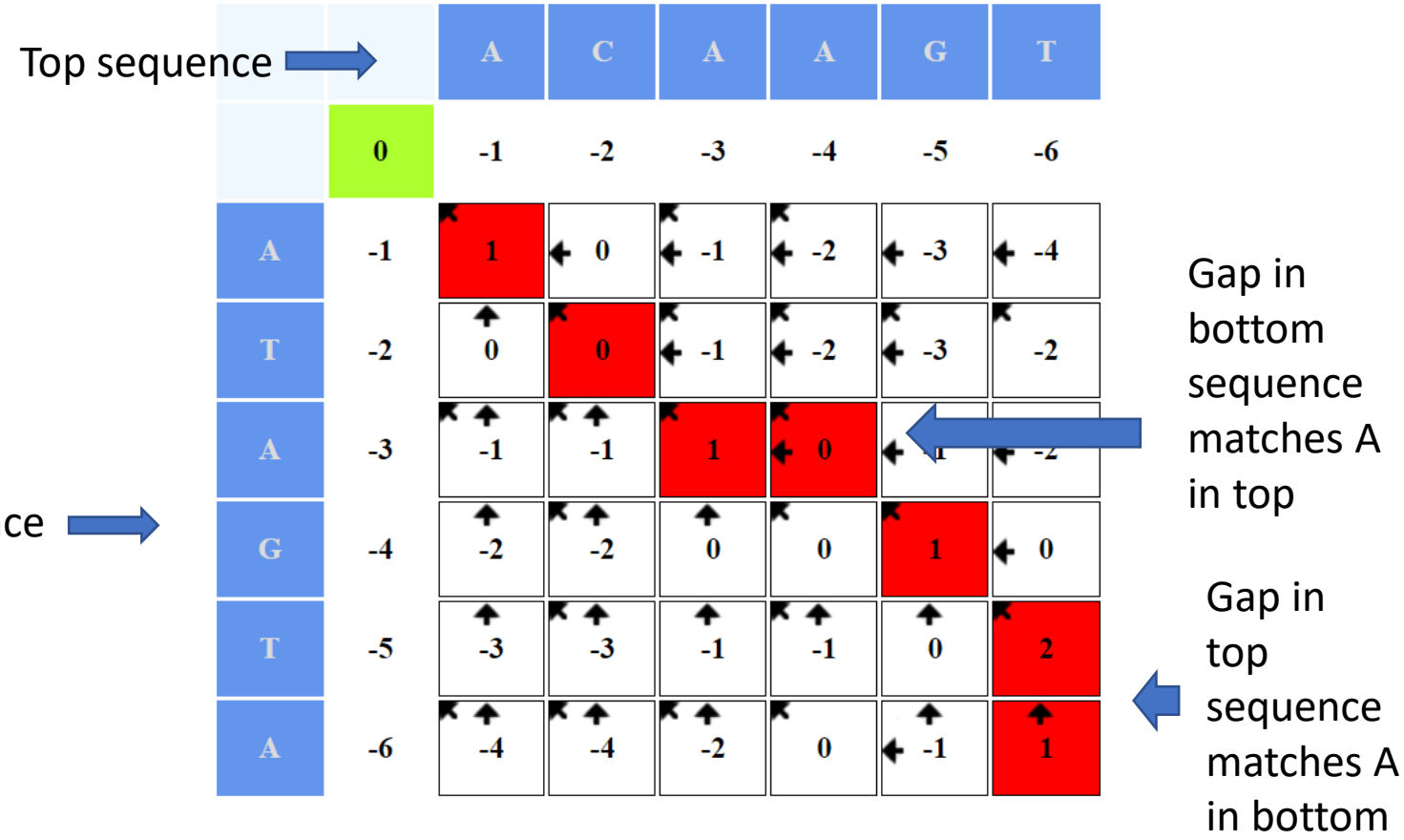
```
ACAAGT-
|  |  ||
ATA-GTA
```

Interactive demo:
https://bioboot.github.io/bimm143_W20/class-material/nw/

(In case there is a problem with connecting to web site https://bioboot.github.io/bimm143_W20/class-material/nw/)




```
ACAAGT-
|  |  ||
ATA-GTA
```



Note: There can be > 1 optimal alignment

```
ACAAGT-
|  |  ||
ATA-GTA
```

```
ACAAGT-        ACAAGT-
|    |||       |    |||
AT-AGTA        A-TAGTA
```

All alignment scores: 4 – 3 = 1

How to do a local alignment
Smith-Waterman algorithm

```
ACAAGT
   | | |
 ATAGTA
```

Match +1
Mismatch -1
Gap -1

Alignment Score: 3

Interactive demo:
https://gtuckerkellogg.github.io/pairwise/demo/

How to do a local alignment Smith-Waterman algorithm (backup https://gtuckerkellogg.github.io/pairwise/demo/)

ACAAGT
| | |
ATAGTA

TOP sequence: ACAAGT

BOTTOM sequence: ATAGTA

Alignment type: Needleman-Wunsch | Smith-Waterman

Top sequence →

Algorithm Parameters

Scoring Matrix: User-defined / Standard

match: 1

mismatch: -1

Linear gap penalty: 1

Bottom sequence →

Local protein alignment score: 3

AGT
AGT

Dynamic programming matrix visualisation

Paths for optimal alignments are indicated in red

Bottom sequence ↓ \ Top sequence →		A	C	A	A	G	T
	0	0	0	0	0	0	0
A	0	1	0	1	1	0	0
T	0	0	0	0	0	0	1
A	0	1	0	1	1	0	0
G	0	0	0	0	0	2	1
T	0	0	0	0	0	1	3
A	0	1	0	1	1	0	2

No negative entries

Why did we choose the DNA scoring system match = 1, mismatch = -1, gap = -1?

- Mainly for simplicity of explanation
- Why would one want to a nucleotide / nucleotide alignment?
- Short read alignment: E.g. in BWA-MEM2, matches are extended using a restricted local alignment (restricted because short-read aligners are only interested in quite good matches). Default match = 1, mismatch = -4 (based on estimated sequence error rates [?]). Gap opening = -6, gap extension = -1
- Check candidate PCR / RT-PCR primer pairs for mispriming (priming from unintended sequences) PRIMER-BLAST
- Evolutionary relationships between closely related species

Protein sequence alignment

- Historically focused on evolutionary relationship among proteins
- Amino acids have different physical properties, so some mutations are better tolerated over evolutionary time than others
- Scoring matrices were developed by observing how frequent or rare different amino acid substitutions were over evolutionary time
- There two common *series* of amino acid scoring matrices:
- BLOSUM (BLOcks Substitution Matrix)
- PAM (Point Accepted Mutations)

BLOSUM62

	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

What the scores mean

- Roughly: the tcore of each pair of amino acids is the scaled log of the ratio of the frequency of mutations between the two amino acids over the products of the overall frequencies of the two amino acids.
- In detail: Henikoff and Henikoff created a database of ungapped alignments of conserved regions of protein sequences (the BLOCKS) database (https://www.pnas.org/content/pnas/89/22/10915.full.pdf)
- See next slide

BLOSUM matrices

The Dayhoff matrices have been one of the mainstays of sequence comparison techniques, but they do have their limitations. The entries in $S(1)$ arise mostly from short time interval substitutions, and raising $S(1)$ to a higher power, to give for instance a PAM250 matrix, does not capture the true difference between short time substitutions and long term ones [Gonnet, Cohen & Benner 1992]. The former are dominated by amino acid substitutions that arise from single base changes in codon triplets, for example L ↔ I, L ↔ V or Y ↔ F, whereas the latter show all types of codon changes.

Since the PAM matrices were made, databases have been formed containing

multiple alignments of more distantly related proteins, and these can be used to derive score matrices more directly. One such set of score matrices that is widely used is the BLOSUM matrix set [Henikoff & Henikoff 1992]. In detail, they were derived from a set of aligned, ungapped regions from protein families called the BLOCKS database [Henikoff & Henikoff 1991]. The sequences from each block were clustered, putting two sequences into the same cluster whenever their percentage of identical residues exceeded some level $L\%$. Henikoff & Henikoff then calculated the frequencies A_{ab} of observing residue a in one cluster aligned against residue b in another cluster, correcting for the sizes of the clusters by weighting each occurrence by $1/(n_1 n_2)$, where n_1 and n_2 are the respective cluster sizes.

From the A_{ab}, they estimated q_a and p_{ab} by $q_a = \sum_b A_{ab} / \sum_{cd} A_{cd}$, i.e. the fraction of pairings that include an a, and $p_{ab} = A_{ab} / \sum_{cd} A_{cd}$, i.e. the fraction of pairings between a and b out of all observed pairings. From these they derived the score matrix entries using the standard equation $s(a,b) = \log p_{ab}/q_a q_b$ (2.3). Again, the resulting log-odds score matrices were scaled and rounded to the nearest integer value. The matrices for $L = 62$ and $L = 50$ in particular are widely used for pairwise alignment and database searching, BLOSUM62 being standard for ungapped matching, and BLOSUM50 being perhaps better for alignment with gaps [Pearson 1996]. BLOSUM62 is scaled so that its values are in half-bits, i.e. the log-odds values were multiplied by $2/\log 2$, and BLOSUM50 is given in third-bits. Note that lower L values correspond to longer evolutionary time, and are applicable for more distant searches.

pg 43/44, sDurbin, Eddy, Krogh, Mitchison, Biological Sequence Analysis, 1998, Cambridge University Press

BLOSUM62

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