

A simple alignment

- Let us try to align two short nucleotide sequences:
 - AATCTATA and AAGATA
- Without considering any gaps (insertions/deletions) there are 3 possible ways to align these sequences

```

AATCTATA   AATCTATA   AATCTATA
AAGATA     AAGATA     AAGATA
    
```

- Which one is better?

Sequence alignment

Bioinformatics Fall 2008

1

Scoring the alignments

- We need to have a scoring mechanism to evaluate alignments
 - match score
 - mismatch score
- We can have the total score as:

$$\sum_{i=1}^n \text{match or mismatch score at position } i$$

- For the simple example, assume a match score of 1 and a mismatch score of 0:

```

AATCTATA   AATCTATA   AATCTATA
AAGATA     AAGATA     AAGATA
  4         1         3
    
```

Sequence alignment

Bioinformatics Fall 2008

2

Good alignments require gaps

- Maximal consecutive run of spaces in alignment
 - Matching mRNA (cDNA) to DNA
 - Shortening of DNA/protein sequences
 - Slippage during replication
 - Unequal crossing-over during meiosis
 - ...
- We need to have a scoring function that also considers gaps

Sequence alignment

Bioinformatics Fall 2008

3

Simple alignment with gaps

- Considering gapped alignments vastly increases the number of possible alignments:

```

AATCTATA   AATCTATA   AATCTATA   more?
AAG-AT-A   AA-G-ATA   AA--GATA
  1         3         3
    
```

$$f(m,n) = \text{number of alignments of two strings of length } m,n$$

$$= f(m,n-1) + f(m-1,n) + f(m-1,n-1)$$

- If gap penalty is -1 what will be the new scores?

Sequence alignment

Bioinformatics Fall 2008

4

More complicated gap penalties

- Nature favors small number of long gaps compared to large number of short gaps.
- How do we adjust our scoring scheme to account for this fact above?

By having different gap opening and gap extension penalties.

- Choices of gap penalties
 - Linear
 - Affine
 - Gap open penalty
 - Gap extension penalty
 - Convex
 - Arbitrary

Sequence alignment

Bioinformatics Fall 2008

5

Optimal sequence alignment

- Based on dynamic programming
- General idea of dynamic programming
 - Solve smaller sub-problems, remember these solutions in a table, and use them to solve the larger problem
 - Example of “coin change problem”
 - Finding the best path from source to sink in a grid

Sequence alignment

Bioinformatics Fall 2008

6

Global sequence alignment (Needleman-Wunsch)

- Edit distance, $d(a,b)$ = distance between symbols a and b
 - Linear gap model
- $D(i,j)$ = edit distance between $\alpha(1:i)$ and $\beta(1:j)$
- Recurrence relation
 - $D(i,0) = \sum d(\alpha(k),-), 1 \leq k \leq i$ $\beta(1:0) = \text{null} = -$
 - $D(0,j) = \sum d(-, \beta(k)), 1 \leq k \leq j$
 - $D(i,j) = \min [D(i-1,j) + d(\alpha(i),-), \text{--- } i \text{ opposite gap}$
 $D(i,j-1) + d(-, \beta(j)), \text{--- } j \text{ opposite gap}$
 $D(i-1,j-1) + d(\alpha(i), \beta(j))] \text{--- } i \text{ opposite } j$

Sequence alignment

Bioinformatics Fall 2008

7

Example

α : G C - A G - A - G C A C G
 β : G C T G G A A G G C A - T

Linear gap model
 Match = 0
 Mismatch = 1

	--	G	C	T	G	G	A	G	G	C	A	T	
--	0	1	2	3	4	5	6	7	8	9	10	11	12
G	1	0	1	2	3	4	5	6	7	8	9	10	11
C	2	1	0	1	2	3	4	5	6	7	8	9	10
A	3	2	1	1	2	3	3	4	5	6	7	8	9
G	4	3	2	2	1	2	3	4	4	5	6	7	8
A	5	4	3	3	2	2	2	3	4	5	6	6	7
G	6	5	4	4	3	2	3	3	3	4	5	6	7
C	7	6	5	5	4	3	3	4	4	4	4	5	6
A	8	7	6	6	5	4	3	3	4	5	5	4	5
C	9	8	7	7	6	5	4	4	4	5	5	5	5
G	10	9	8	8	7	6	5	5	4	4	5	6	6

Sequence alignment

Bioinformatics Fall 2008

8

Similarity instead of distance

- $s(a,b)$ = score of aligning symbols a and b
- $S(i,j)$ = optimal similarity of $\alpha(1:i)$ and $\beta(1:j)$
- Recurrence relation
 - $S(i,0) = \sum s(\alpha(k),-), 1 \leq k \leq i$
 - $S(0,j) = \sum s(-, \beta(k)), 1 \leq k \leq j$
 - $S(i,j) = \max [S(i,j-1) + s(-, \beta(j)),$
 $S(i-1,j) + s(\alpha(i),-),$
 $S(i-1,j-1) + s(\alpha(i), \beta(j))]$
- Assume linear gap penalty

Sequence alignment

Bioinformatics Fall 2008

9

The BLOSUM45 Matrix

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	5	-2	-1	-2	-1	-1	0	-2	-1	-1	-1	-2	-1	1	0	-2	-2	0		
R	-2	7	0	-1	-3	1	0	-2	0	-3	-2	3	-1	-2	-2	-1	-2	-1	-2	-1
N	-1	0	6	-2	0	0	0	1	-2	-3	0	-2	-2	-2	1	0	-4	-2	-3	
D	-2	-1	2	7	-3	0	2	-1	0	-4	-3	0	-3	-4	-1	0	-1	-4	-2	-3
C	-1	-3	-2	-3	12	-3	-3	-3	-3	-2	-3	-2	-2	-4	-1	-1	-5	-3	-1	
Q	-1	0	0	-3	6	2	-2	1	-2	2	1	0	-4	-1	0	-1	-2	-1	-3	
E	-1	0	0	2	-3	2	6	-2	0	-3	-2	1	-2	-3	0	0	-1	-3	-2	-3
G	-2	0	-1	-3	-2	7	-2	-4	-3	-2	-2	-3	-2	0	-2	-2	-3	-3		
H	-2	0	1	0	-3	1	0	-2	10	-3	-2	-1	0	-2	-2	-1	-2	-3	-2	
I	-1	-3	-2	-4	-3	-2	-3	-4	-3	5	2	-3	2	0	-2	-2	-1	-2	0	3
L	-1	-2	-3	-2	-2	-3	-2	-3	2	2	5	2	1	-3	-3	-2	0	0	1	
K	-1	3	0	0	-3	1	1	-2	-1	-3	-3	5	-1	-3	-1	-1	-1	-2	-1	-2
M	-1	-2	-3	-2	0	-2	-2	0	2	2	-1	6	0	-2	-2	-1	-2	0	1	
F	-2	-3	-2	-4	-2	-4	-3	-2	0	1	-3	0	8	-3	-2	-1	1	3	0	
P	-1	-2	-2	-1	-4	-1	0	-2	-2	-3	-1	-2	-3	9	-1	-1	-3	-3		
S	-1	1	0	-1	0	0	0	-1	-2	-3	-1	-2	-2	-1	4	2	-4	-2	-1	
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-1	2	5	-3	-1	0	
W	-2	-2	-4	-4	-5	-2	-3	-2	-3	-2	-2	-2	1	-3	-4	-3	15	3	-3	
Y	-2	-1	-2	-2	-3	-1	-2	-3	2	0	-1	0	3	-2	-1	3	8	-1		
V	0	-2	-3	-3	-1	-3	-3	-3	3	1	-2	1	0	-3	-1	0	-3	-1	5	

Sequence alignment

Bioinformatics Fall 2008

10

score(H,P) = -2, gap penalty = -8

	H	E	A	G	A	W	G	H	E	E	
P	0	-8	-16	-24	-32	-40	-48	-56	-64	-72	-80
A	-8	-2									
W	-16										
H	-24										
E	-32										
A	-40										
A	-48										
E	-56										

Sequence alignment

Bioinformatics Fall 2008

11

score(E,P) = 0, score(E,A) = -1, score(H,A) = -2

	H	E	A	G	A	W	G	H	E	E	
P	0	-8	-16	-24	-32	-40	-48	-56	-64	-72	-80
A	-8	-2	-8								
W	-16		-10	-3							
H	-24										
E	-32										
E	-40										
A	-48										
E	-56										

Sequence alignment

Bioinformatics Fall 2008

12

Optimal alignment: HEAGAWGHE - E
- P - - A W - HEAE

		H	E	A	G	A	W	G	H	E	E
P	0	-8	-16	-24	-32	-40	-48	-56	-64	-72	-80
A	-8	-2	-8	-16	-24	-33	-42	-49	-57	-65	-73
W	-16	-10	-3	-4	-12	-19	-28	-36	-44	-52	-60
H	-24	-18	-11	-6	-7	-15	-4	-12	-21	-29	-37
E	-32	-14	-18	-13	-8	-9	-12	-6	-2	-11	-19
A	-40	-22	-8	-16	-16	-9	-12	-14	-6	-4	-5
E	-48	-30	-16	-3	-11	-11	-12	-12	-14	-4	2
E	-56	-38	-24	-11	-6	-12	-14	-15	-12	-8	2

Score of the best alignment

Sequence alignment Bioinformatics Fall 2008 13

Local sequence alignment (Smith-Waterman)

- $S(i,j)$ = optimal local similarity among suffixes of $\alpha(1:i)$ and $\beta(1:j)$
- Recurrence relation
 - $S(i,0) = 0$
 - $S(0,j) = 0$
 - $S(i,j) = \max [0, S(i,j-1) + s(-, \beta(j)), S(i-1,j) + s(\alpha(i), -), S(i-1,j-1) + s(\alpha(i), \beta(j))]$
- Assume linear gap model

Sequence alignment Bioinformatics Fall 2008 14

Example

α 's subsequence: G C A G A G C A
 β 's subsequence: G A A G - G C A

Linear gap model
Match = +5
Mismatch = -4

		β											
	--	G	C	T	G	G	A	A	G	G	C	A	T
--	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	5	1	0	5	5	1	0	5	5	1	0	0
C	0	1	10	6	2	1	1	0	1	1	10	6	2
A	0	0	6	6	2	0	6	6	2	0	6	15	11
G	0	5	2	2	11	7	3	2	11	7	3	11	11
A	0	1	1	0	7	7	11	8	7	7	3	8	7
G	0	5	1	0	5	11	7	7	13	12	8	4	4
C	0	0	10	6	2	7	7	3	9	8	17	13	9
A	0	0	6	6	2	3	11	12	8	5	13	22	18
C	0	0	5	2	2	0	7	8	8	4	18	18	18
G	0	5	1	1	7	7	5	4	13	13	14	14	14

Sequence alignment Bioinformatics Fall 2008 15

Affine gaps: d (open), e (extend)

- Four dynamic programming terms
 - $G(i,j)$ is the optimal local similarity when $\alpha(i)$ is aligned with $\beta(j)$
 - $E(i,j)$ is the optimal local similarity when $\alpha(i)$ is aligned to the left of $\beta(j)$
 - $F(i,j)$ is the optimal local similarity when $\alpha(i)$ is aligned to the right of $\beta(j)$
 - $S(i,j)$ is the optimal score of aligning $\alpha(1:i)$ and $\beta(1:j)$
 - Case of global alignment here
- Recurrences
 - $S(i,j) = \max [E(i,j), F(i,j), G(i,j)]$
 - $G(i,j) = S(i-1,j-1) + s(\alpha(i), \beta(j))$
 - $E(i,j) = \max [S(i,j-1) + d, E(i,j-1) + e]$
 - $F(i,j) = \max [S(i-1,j) + d, F(i-1,j) + e]$
- Base case
 - $S(0,0) = 0, S(i,0) = E(i,0) = d + (i-1)*e$
 - $S(0,j) = F(0,j) = d + (j-1)*e$

Why not $G(i,j-1)$ instead of $S(i,j-1)$?

$\alpha(i) -$
 $\beta(i-1) - \beta(0)$
 $\alpha(0) -$
 $\beta(i-1) \beta(0)$
 $\alpha(0) - -$
 $\beta(i-1) \beta(0)$

Sequence alignment Bioinformatics Fall 2008 16

Typical parameters

- DNA
 - Match = +2
 - Mismatch = -3
 - Gaps: open = -5, extension = -2
- Protein
 - BLOSUM/PAM
 - Gaps: open = -11, extension = -1

Sequence alignment Bioinformatics Fall 2008 17

Complexity

- $O(mn)$ time
- $O(mn)$ space
 - $O(\max(m,n))$ if only distance value is needed
- More complicated "divide-and-conquer" algorithm that doubles time complexity and uses $O(\min(m,n))$ space [Hirschberg, JACM 1977]
- Convex gaps: $O(mn \log(m+n))$ time
- Arbitrary gaps: $O(mn (m+n))$ time

Sequence alignment Bioinformatics Fall 2008 18

Time and space bottlenecks

- Comparing two one-megabase genomes.
- Space:
 - An entry: 4 bytes;
 - Table: $4 * 10^6 * 10^6 = 4$ T bytes memory.
- Time:
 - 1 GHz CPU: 10 M entries/second;
 - 10^{12} entries: 10^5 seconds > 1 day.

Sequence alignment

Bioinformatics Fall 2008

19

Banded global alignment

- Two sequences differ by at most w ($w \ll n$).
- w-band algorithm: $O(wn)$ time and space.
- Example: $w = 4$
 - Linear gap penalty
 - Match = +1
 - Mismatch = -1

		A	C	C	A	C	A	C	A
	0	-1	-2	-3					
A	-1	1	0	-1	-2				
C	-2	0	2	1	0	-1			
A	-3	-1	1	1	2	1	0		
C		-2	0	2	1	3	2	1	
C			-1	1	1	2	2	3	2
A				0	1	1	2	2	4
T						0	0	1	3
A							-1	1	0
									2

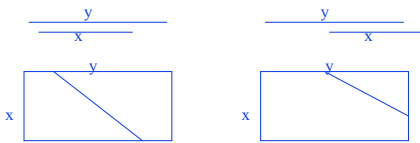
Sequence alignment

Bioinformatics Fall 2008

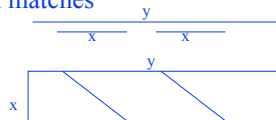
20

Other kinds of alignments

- Overlaps



- Repeated matches



Sequence alignment

Bioinformatics Fall 2008

21

Statistics of local alignment

- Expected score should be negative.
- $\sum q_a q_b s(a,b) < 0$, assume no gaps
- $\sum q_a q_b \log(p_{ab}/q_a q_b) < 0$, for log odds score
- $-\sum q_a q_b \log(q_a q_b/p_{ab}) < 0$ $H = 0$ only when $q^2 = p$
- $-H(q^2 || p) < 0$, since H is positive
- Relative entropy between the product distribution and the joint distribution
 - Also called the Kullback-Leibler distance

Sequence alignment

Bioinformatics Fall 2008

22

Suffix tree and suffix array

- Data structures for *exact* matches

Sequence alignment

Bioinformatics Fall 2008

23

Suffix tree

- Rooted tree of all suffixes of a given string
 - Unique label at each edge
 - No two edges out of node share the first character
 - Every path ends in a suffix
 - Add special termination symbol (xabxa)
 - Number of leaves = length of string
 - Every non-leaf node has at least two children
 - Linear time construction
 - Linear time exact matches

Sequence alignment

Bioinformatics Fall 2008

24

Implementing suffix tree

- Alphabet array at internal node
 - $O(m |\Sigma|)$ space, m is size of database
 - $O(n)$ time for exact match, n is size of query
- Sorted linked list
 - $O(m)$ space
 - $O(n |\Sigma|)$ time
- Binary search at internal node
 - $O(m)$ space
 - $O(n \log |\Sigma|)$ time

Sequence alignment

Bioinformatics Fall 2008

25

Applications of suffix tree

- Finding longest repeat
 - Linear time
- Longest common substring problem
 - Generalized suffix tree
 - Leaf labels indicate the corresponding string
 - Example: sanddollar, handler
 - Linear time
- Common substrings of more than two strings
 - $l(k)$ = length of longest substring common to at least k strings
 - Example: sanddollar, handler, sandlot, grand, pantry
 - $l(2) = 4$ (sand), $l(3) = 3$ (and), $l(4) = 3$ (and), $l(5) = 2$ (an)
 - Build a generalized suffix tree
 - At each internal node, maintain a count of the number of distinct string identifiers at leaves
- Multiple alignment
 - Find anchor strings

Sequence alignment

Bioinformatics Fall 2008

26

Suffix array

- More space efficient than suffix trees
- A suffix array for a string x of length m is an array of size m that specifies the lexicographic ordering of the suffixes of x .
 - Example of a suffix array (mississippi)
- $O(m)$ space
- Lookup query
 - Binary search
 - $O(n \log m)$ time; n is the size of the query
 - Can reduce time to $O(n + \log m)$ using a more efficient implementation

Sequence alignment

Bioinformatics Fall 2008

27