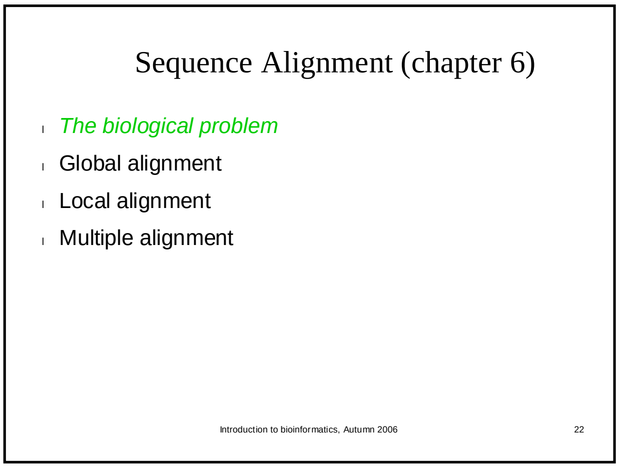
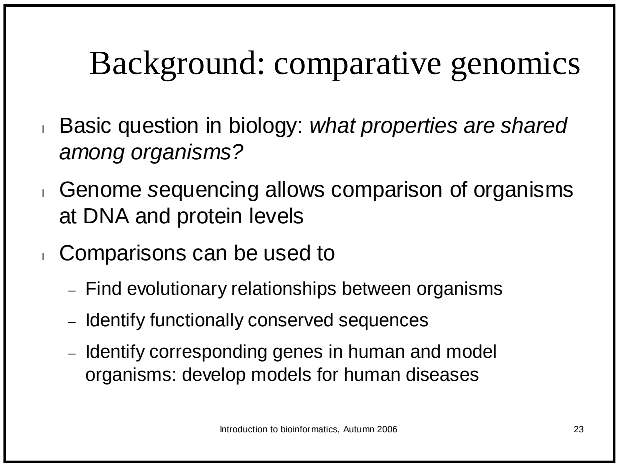


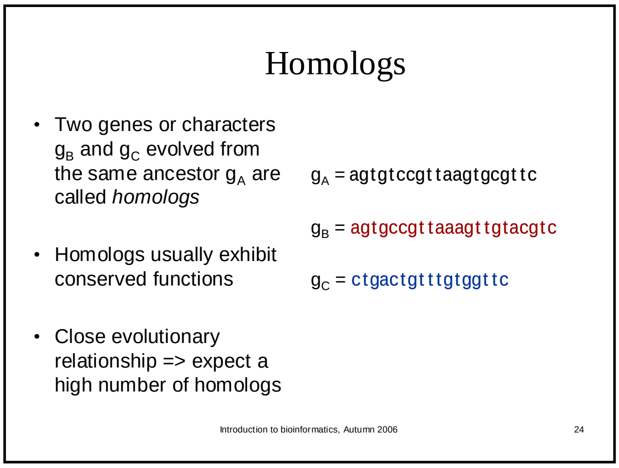
Sequence Alignment (chapter 6)

- *The biological problem*
- Global alignment
- Local alignment
- Multiple alignment



Background: comparative genomics

- Basic question in biology: *what properties are shared among organisms?*
- Genome sequencing allows comparison of organisms at DNA and protein levels
- Comparisons can be used to
  - Find evolutionary relationships between organisms
  - Identify functionally conserved sequences
  - Identify corresponding genes in human and model organisms: develop models for human diseases



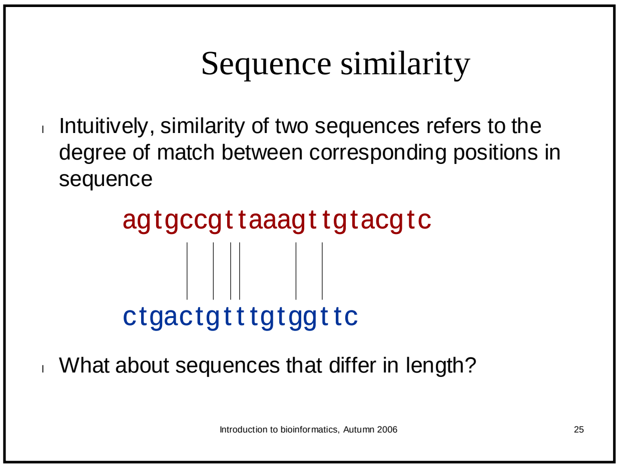
Homologs

- Two genes or characters g_B and g_C evolved from the same ancestor g_A are called *homologs*
- Homologs usually exhibit conserved functions
- Close evolutionary relationship => expect a high number of homologs

g_A = agtgtccgttaagtgcgttc

g_B = agtgccgttaaagttgtacgtc

g_C = ctgactgtttgtggttc



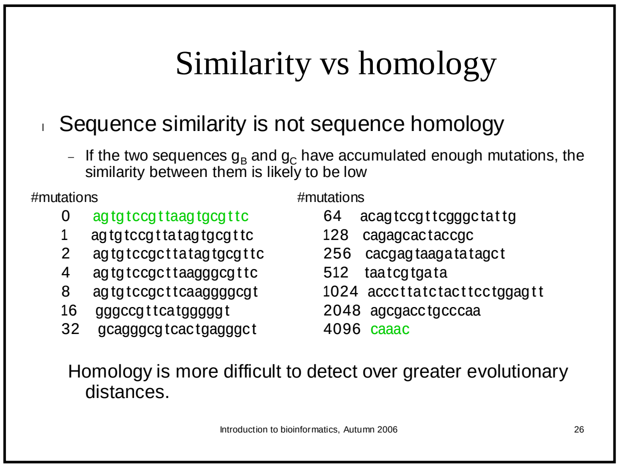
Sequence similarity

- Intuitively, similarity of two sequences refers to the degree of match between corresponding positions in sequence

agtgccgttaaagttgtacgtc

ctgactgtttgtggttc

- What about sequences that differ in length?

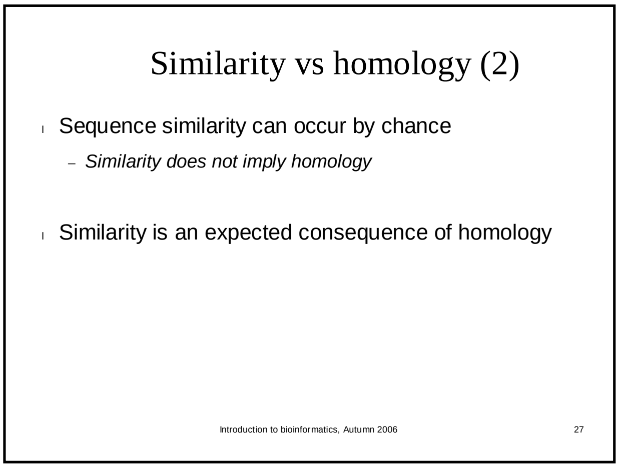


Similarity vs homology

- Sequence similarity is not sequence homology
  - If the two sequences g_B and g_C have accumulated enough mutations, the similarity between them is likely to be low

#mutations		#mutations	
0	agtgtccgttaagtgcgttc	64	acagtccgttcgggctattg
1	agtgtccgttatagtgcgttc	128	cagagcactaccgc
2	agtgtccgcttatagtgcgttc	256	cacgagtaagatatagct
4	agtgtccgcttaagggcgttc	512	taatcgtgata
8	agtgtccgcttcaaggggcgt	1024	acccttatctacttcctggagtt
16	gggccgttcatggggt	2048	agcgacctgcccaa
32	gcagggcgtcactgagggct	4096	caaac

Homology is more difficult to detect over greater evolutionary distances.

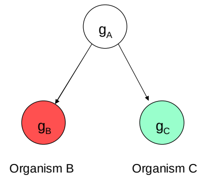


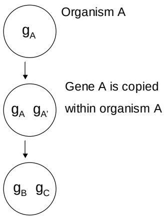
Similarity vs homology (2)

- Sequence similarity can occur by chance
  - *Similarity does not imply homology*
- Similarity is an expected consequence of homology

Orthologs and paralogs

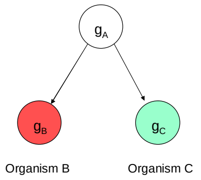
- We distinguish between two types of homology
  - Orthologs: homologs from two different species
  - Paralogs: homologs within a species

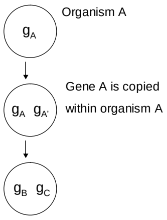




Orthologs and paralogs (2)

- Orthologs typically retain the original function
- In paralogs, one copy is free to mutate and acquire new function (no selective pressure)





Sequence alignment

- Alignment specifies which positions in two sequences match

```
acgtctag   acgtctag   acgtctag
||             |||||  || |||||
actctag-   -actctag   ac-tctag
```

2 matches
5 mismatches
1 not aligned

5 matches
2 mismatches
1 not aligned

7 matches
0 mismatches
1 not aligned

Mutations: Insertions, deletions and substitutions

Indel: insertion or deletion of a base with respect to the ancestor sequence

```
acgtctag
   |||||
-actctag
```

Mismatch: substitution (point mutation) of a single base

- Insertions and/or deletions are called *indels*
  - *We can't tell whether the ancestor sequence had a base or not at indel position*

Problems

- What sorts of alignments should be considered?
- How to score alignments?
- How to find optimal or good scoring alignments?
- How to evaluate the statistical significance of scores?

In this course, we discuss the first three problems.

Course *Biological sequence analysis* tackles all four in-depth.

Sequence Alignment (chapter 6)

- The biological problem
- *Global alignment*
- Local alignment
- Multiple alignment

Global alignment

- Problem: find optimal scoring alignment between two sequences (Needleman & Wunsch 1970)
- We give score for each position in alignment

– Identity (match)	+1	**WHAT**
– Substitution (mismatch)	-µ	**||**
– Indel	-δ	**WH-Y**

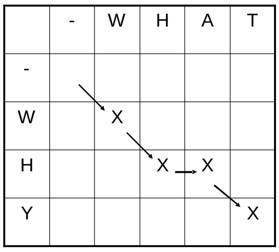
S(WHAT/WH-Y) = 1 + 1 – δ – µ

Representing alignments and scores

WHAT

||

WH-Y



	-	W	H	A	T
-					
W		X			
H			X	X	
Y					X

Representing alignments and scores

WHAT

||

WH-Y

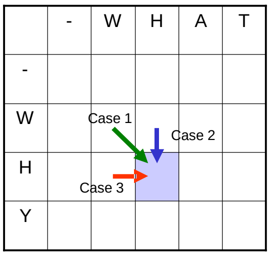
	-	W	H	A	T
-	0				
W		1			
H			2	2-δ	
Y					2-δ-µ

Global alignment score $S_{3,4}$ = 2-δ-µ

Dynamic programming

- How to find the optimal alignment?
- We use previous solutions for optimal alignments of smaller subsequences
- This general approach is known as dynamic programming

Filling the alignment matrix



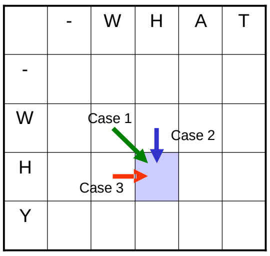
Consider the alignment process at shaded square.

Case 1. Align H against H (match or substitution).

Case 2. Align H in WHY against – (indel) in WHAT.

Case 3. Align H in WHAT against – (indel) in WHY.

Filling the alignment matrix (2)



Scoring the alternatives.

Case 1. $S_{2,2} = S_{1,1} + s(2, 2)$

Case 2. $S_{2,2} = S_{1,2} – δ$

Case 3. $S_{2,2} = S_{2,1} – δ$

s(i, j) = 1 for matching positions,

s(i, j) = - µ for substitutions.

Choose the case (path) that yields the maximum score.

Keep track of path choices.

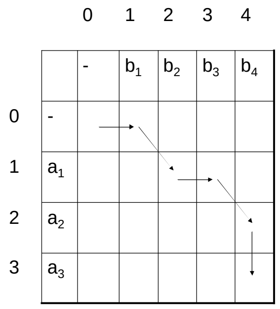
Global alignment: formal development

$A = a_1a_2a_3\ldots a_n$,
$B = b_1b_2b_3\ldots b_m$

b_1	b_2	b_3	b_4	-
-	a_1	-	a_2	a_3

- Any alignment can be written as a unique path through the matrix
- Score for aligning A and B up to positions i and j:

$S_{i,j} = S(a_1a_2a_3\ldots a_i, b_1b_2b_3\ldots b_j)$



		0	1	2	3	4
		-	b_1	b_2	b_3	b_4
0	-					
1	a_1					
2	a_2					
3	a_3					

Scoring partial alignments

- Alignment of $A = a_1a_2a_3\ldots a_n$ with $B = b_1b_2b_3\ldots b_m$ can end in three ways
  - Case 1: $(a_1a_2\ldots a_{i-1})\ a_i$
    $(b_1b_2\ldots b_{j-1})\ b_j$
  - Case 2: $(a_1a_2\ldots a_{i-1})\ a_i$
    $(b_1b_2\ldots b_j)$ -
  - Case 3: $(a_1a_2\ldots a_i)$ –
    $(b_1b_2\ldots b_{j-1})\ b_j$

Scoring alignments

- Scores for each case:
  - Case 1: $(a_1a_2\ldots a_{i-1})\ a_i$
    $(b_1b_2\ldots b_{j-1})\ b_j$
  - Case 2: $(a_1a_2\ldots a_{i-1})\ a_i$
    $(b_1b_2\ldots b_j)$ –
  - Case 3: $(a_1a_2\ldots a_i)$ –
    $(b_1b_2\ldots b_{j-1})\ b_j$

$$s(a_i, b_j) = \begin{cases} +1 \text{ if } a_i = b_j \\ -\mu \text{ otherwise} \end{cases}$$

$$s(a_i, -) = s(-, b_j) = -\delta$$

Scoring alignments (2)

- First row and first column correspond to initial alignment against indels:
  $S(i, 0) = -i\ \delta$
  $S(0, j) = -j\ \delta$
- Optimal global alignment score $S(A, B) = S_{n,m}$

		0	1	2	3	4
		-	b_1	b_2	b_3	b_4
0	-	0	$-\delta$	-2δ	-3δ	-4δ
1	a_1	$-\delta$				
2	a_2	-2δ				
3	a_3	-3δ				

Algorithm for global alignment

```
Input sequences A, B, n = |A|, m = |B|
Set $S_{i,0}$ := $-\delta i$ for all i
Set $S_{0,j}$ := $-\delta j$ for all j
for i := 1 to n
  for j := 1 to m
    $S_{i,j}$ := max{$S_{i-1,j} - \delta$, $S_{i-1,j-1} + s(a_i,b_j)$, $S_{i,j}-1 - \delta$}
  end
end
```

Algorithm takes O(nm) time and space.

Global alignment: example

$\mu = 1$
$\delta = 2$

	-	T	G	G	T	G
-	0	-2	-4	-6	-8	-10
A	-2					
T	-4					
C	-6					
G	-8					
T	-10					?

Global alignment: example (2)

$\mu = 1$

$\delta = 2$

```
ATCGT-
 | ||
-TGGTG
```

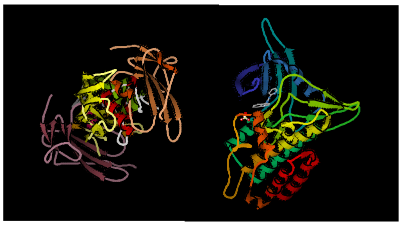
	-	T	G	G	T	G
-	0	-2	-4	-6	-8	-10
A	-2	-1	-3	-5	-7	-9
T	-4	-1	-2	-4	-4	-6
C	-6	-3	-2	-3	-5	-5
G	-8	-5	-2	-1	-3	-4
T	-10	-7	-4	-3	0	-2

Sequence Alignment (chapter 6)

- The biological problem
- Global alignment
- *Local alignment*
- Multiple alignment

Local alignment: rationale

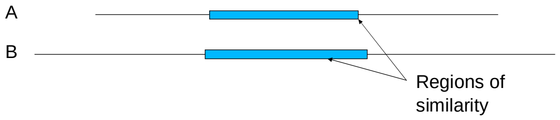
- Otherwise dissimilar proteins may have local regions of similarity
  -> Proteins may share a function



Human bone morphogenic protein receptor type II precursor (left) has a 300 aa region that resembles 291 aa region in TGF-β receptor (right).

The shared function here is protein kinase.

Local alignment: rationale



A

B

Regions of similarity

- Global alignment would be inadequate
- Problem: find the highest scoring *local* alignment between two sequences
- Previous algorithm with minor modifications solves this problem (Smith & Waterman 1981)

From global to local alignment

- Modifications to the global alignment algorithm
  - Look for the highest-scoring path **in** the alignment matrix (not necessarily through the matrix)
  - Allow preceding and trailing indels without penalty

Scoring local alignments

$A = a_1a_2a_3 \ldots a_n$, $B = b_1b_2b_3 \ldots b_m$

Let I and J be intervals (substrings) of A and B, respectively: $I \subset A$, $J \subset B$

Best local alignment score:

$$M(A,B) = \max\{S(I,J) : I \subset A, J \subset B\}$$

where S(I, J) is the score for substrings I and J.

Allowing preceding and trailing indels

- First row and column initialised to zero:

$M_{i,0} = M_{0,j} = 0$

		0	1	2	3	4
		-	b_1	b_2	b_3	b_4
0	-	0	0	0	0	0
1	a_1	0				
2	a_2	0				
3	a_3	0				

```
b1 b2 b3
-  -  a1
```

Recursion for local alignment

- $M_{i,j}$ = max {
  $M_{i-1,j-1} + s(a_i, b_j)$,
  $M_{i-1,j} - \delta$,
  $M_{i,j-1} - \delta$,
  0
  }

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

Finding best local alignment

- Optimal score is the highest value in the matrix

$M(A, B) = \max\{S(I, J) : I \subset A, J \subset B\}$

$= \max_{i,j} M_{i,j}$

- Best local alignment can be found by backtracking from the highest value in M

	-	T	G	G	T	G
-	0	0	0	0	0	0
A	0	0	0	0	0	0
T	0	1	0	0	1	0
C	0	0	0	0	0	0
G	0	0	1	1	0	1
T	0	1	0	0	🟩 2	0

Local alignment: example

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

Local alignment: example

Scoring

Match: +2

Mismatch: -1

Indel: -2

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

```
C T - A A
C T C A A
```

Non-uniform mismatch penalties

- We used uniform penalty for mismatches:

  s('A', 'C') = s('A', 'G') = … = s('G', 'T') = μ

- Transition mutations (A->G, G->A, C->T, T->C) are approximately twice as frequent than transversions (A->T, T->A, A->C, G->T)
  - use non-uniform mismatch penalties

	A	C	G	T
A	1	-1	-0.5	-1
C	-1	1	-1	-0.5
G	-0.5	-1	1	-1
T	-1	-0.5	-1	1

Gaps in alignment

- Gap is a succession of indels in alignment

```
C T - - - A A
C T C G C A A
```

- Previous model scored a length k gap as w(k) = -kδ
- Replication processes may produce longer stretches of insertions or deletions
  - In coding regions, insertions or deletions of codons may preserve functionality

Gap open and extension penalties (2)

- We can design a score that allows the penalty opening gap to be larger than extending the gap:

$$w(k) = -\alpha - \beta(k - 1)$$

- Gap open cost α, Gap extension cost β
- Our previous algorithm can be extended to use w(k) (not discussed on this course)

Sequence Alignment (chapter 6)

- The biological problem
- Global alignment
- Local alignment
- *Multiple alignment*

Multiple alignment

- Consider a set of n sequences on the right
  - Orthologous sequences from different organisms
  - Paralogs from multiple duplications
- How can we study relationships between these sequences?

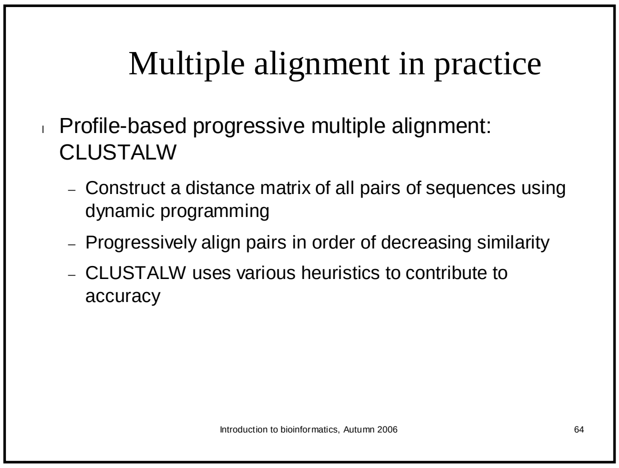
```
aggcgagctgcgagtgcta
cgttagattgacgctgac
ttccggctgcgac
gacacggcgaacgga
agtgtgcccgacgagcgaggac
gcgggctgtgagcgcta
aagcggcctgtgtgcccta
atgctgctgccagtgta
agtcgagccccgagtgc
agtccgagtcc
actcggtgc
```

Optimal alignment of three sequences

- Alignment of $A = a_1a_2\ldots a_i$ and $B = b_1b_2\ldots b_j$ can end either in $(-, b_j)$, (a_i, b_j) or $(a_i, -)$
- $2^2 - 1 = 3$ alternatives
- Alignment of A, B and $C = c_1c_2\ldots c_k$ can end in $2^3 - 1$ ways: $(a_i, -, -)$, $(-, b_j, -)$, $(-, -, c_k)$, $(-, b_j, c_k)$, $(a_i, -, c_k)$, $(a_i, b_j, -)$ or (a_i, b_j, c_k)
- Solve the recursion using three-dimensional dynamic programming matrix: $O(n^3)$ time and space
- Generalizes to n sequences but impractical with moderate number of sequences

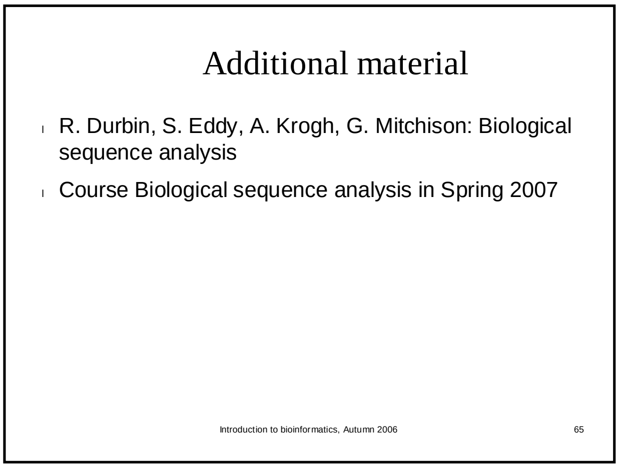
Multiple alignment in practice

- In practice, real-world multiple alignment problems are usually solved with heuristics
- Progressive multiple alignment
  - Choose two sequences and align them
  - Choose third sequence w.r.t. two previous sequences and align the third against them
  - Repeat until all sequences have been aligned
  - Different options how to choose sequences and score alignments



Multiple alignment in practice

- Profile-based progressive multiple alignment: CLUSTALW
  - Construct a distance matrix of all pairs of sequences using dynamic programming
  - Progressively align pairs in order of decreasing similarity
  - CLUSTALW uses various heuristics to contribute to accuracy



Additional material

- R. Durbin, S. Eddy, A. Krogh, G. Mitchison: Biological sequence analysis
- Course Biological sequence analysis in Spring 2007