Sequence Alignment

Why do we need to compare?

In General

- Apples and oranges
- This way we could align two different audio recordings of a piece of music by alignment of musical notes and rhythms.
- Identify plagiarism.

In Biology (Bioinformatics)

Here, we will look at the DNA (A, C, G and T), RNA (A, C, G and U) and protein (A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y) sequences.

We can compare two sequences by placing them above each other in rows and comparing them character by character.

Why do we need to compare?

So, why do we compare biological sequences?

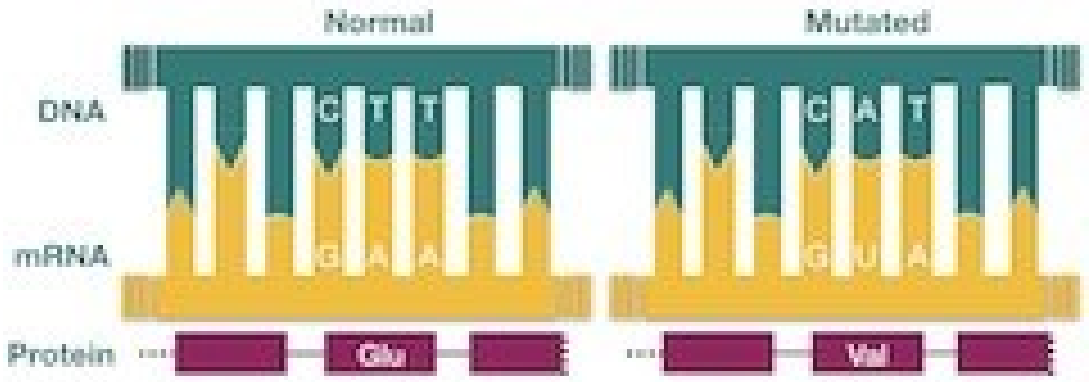
- We can identify causes of genetic diseases by comparing sequences from healthy and unhealthy individuals.

- A comparison of multiple gene sequences from several species can recognize sequence stretches preserved or similar among species; thus, hinting about the possibility that these ***conserved regions*** have a ***related function in organisms***.

- Importantly, all sequence database searches involve comparison of sequences ***to detect a similarity*** to a search sequence.

Pair-wise sequence alignment

- Pair-wise sequence alignment is one of the fundamental means in bioinformatics to assess a degree of similarity as well as to find differences between two sequences.



- Pair-wise alignments are also an essential element in genome assembly pipelines and alignment of entire genome sequences can identify genes duplications and deletions.

Similarity. The similarity measure can tell us whether or not two sequences have the ***same function***. Finding similar sequences in databases is one of the first steps, and probably the most informative step, in ***identifying newly determined sequences***.

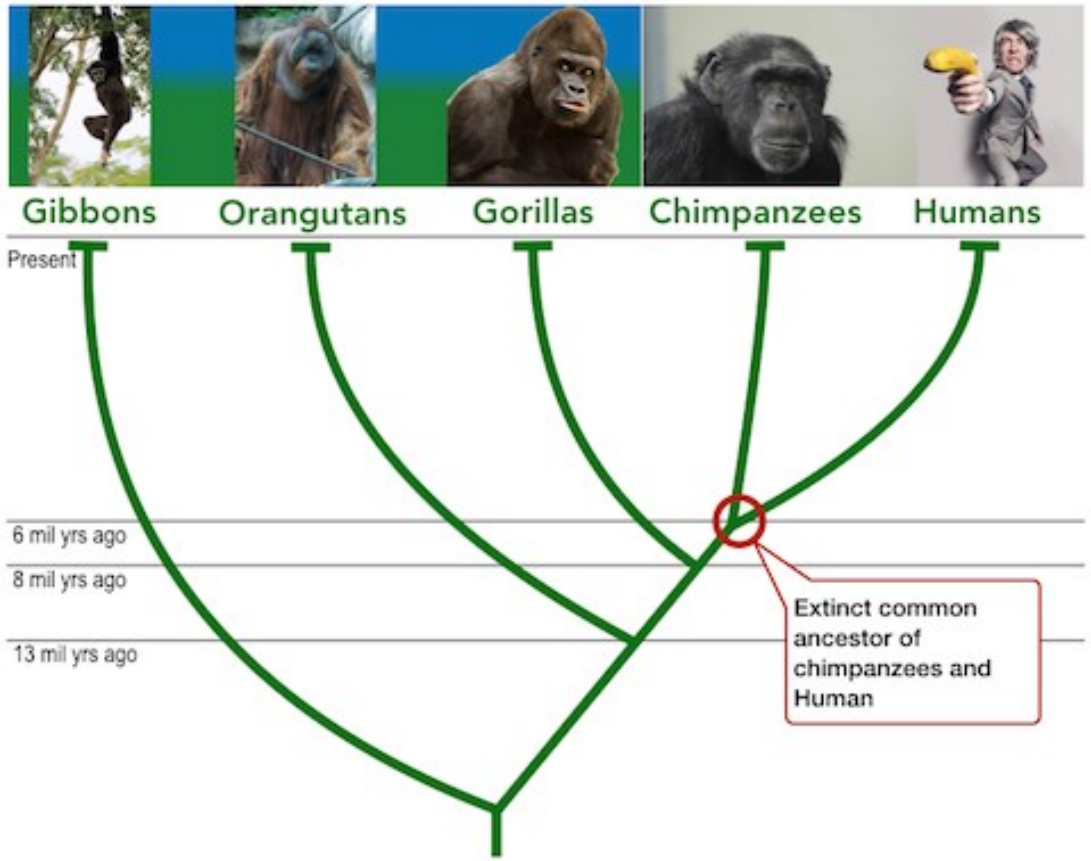
Dissimilarity. It in turn can unveil the point mutations and thus, reveal causes of genetic diseases. Genomic differences between individuals give insight into individuals' tolerance or intolerance of a specific medication and also aid in the proper dosage of medicines. Variation among individual genomic sequences is the fundament of precision medicine or personalized medicine.

Pairwise sequence alignment

Why are sequences similar?

1. **By chance (random)**

- Sequences can be similar by random chance alone, but usually, then the similarity tends to be low. A high degree of similarity implies the sequences to originate from a common ancestor.



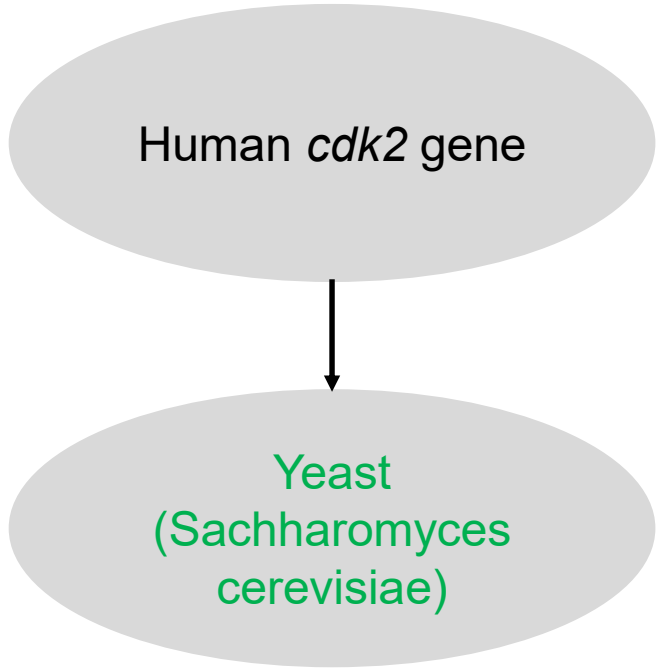
Pair-wise sequence alignment

2. Common ancestry (guided)

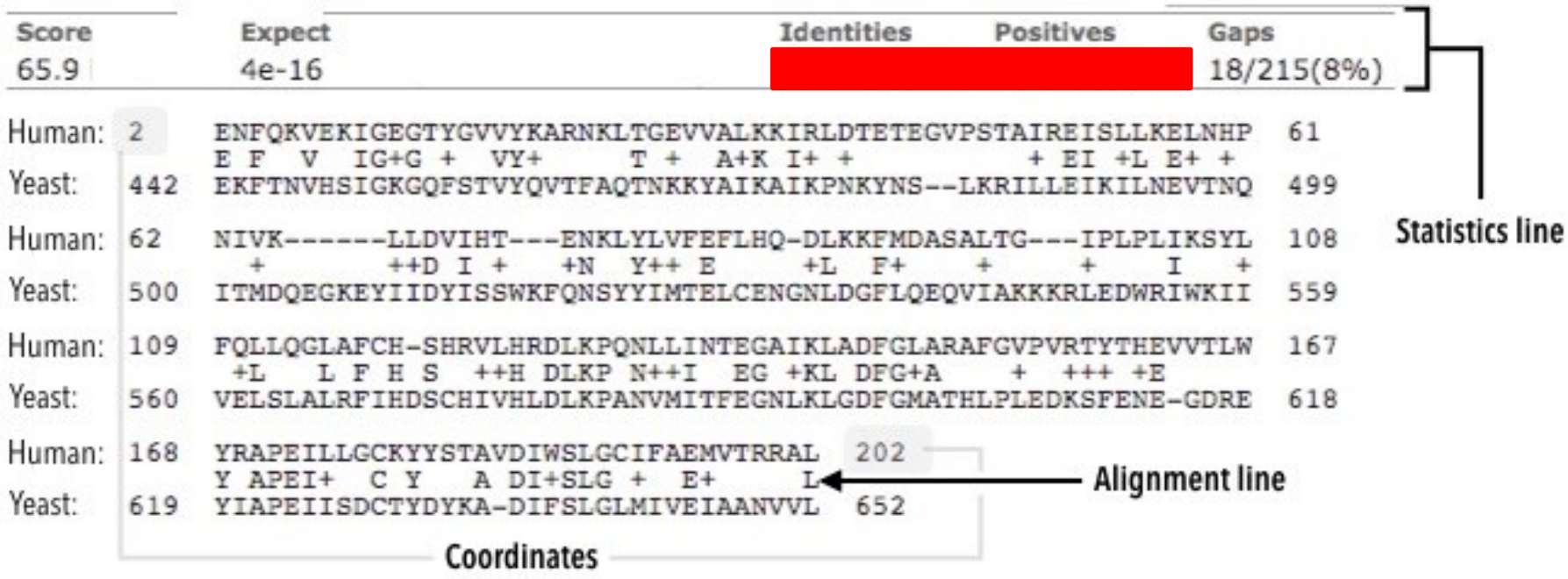
- All living organisms share DNA, which originates from a universal common ancestor. For example, fission yeast (*Schizosaccharomyces pombe*) and humans had a common ancestor some one billion years ago.
- To give some idea about the timescale, Earth formed about 4.5 billion years ago, animals appeared roughly only 600 million years ago, and dinosaurs went extinct mere 66 million years ago.

Sir Paul Maxime Nurse
Nobel Prize 2001 in Physiology or Medicine

Cyclin-dependent PK, CDK2 (a clock gene which controls the cell cycle)



Pairwise sequence alignment



- A comparison of human and chimpanzee genomes reveal them to differ in average 1.2%.
- We, humans, are on average 99.9% identical, a minuscule 0.1% difference between individuals.
- On the other hand, given the length of the human genome, over three billion base pairs, 0.1% amounts to about the difference of three million base pairs.

Some concepts in sequence alignment

Example of an alignment:

Sequence 1: ATGAAGCGTGC(11)

Sequence 2: ATGAAGAGTGCA(12)

ATGAAGCGTGT
ATGAGAGTCGAT

By inserting gaps, the sequences become the same length, and by choosing proper positions, additional letters will align.

Match and Mismatch

ATGAAGCGT_G_T
ATGA_GAGTCGAT

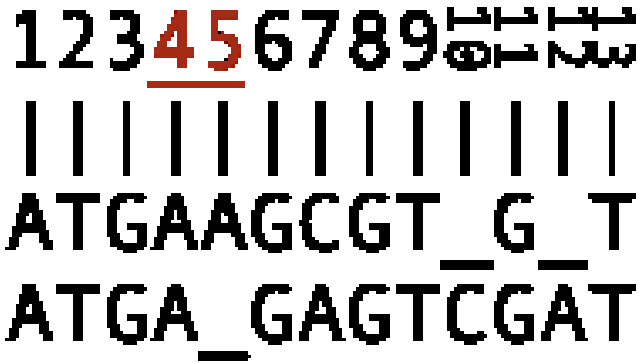
Gap

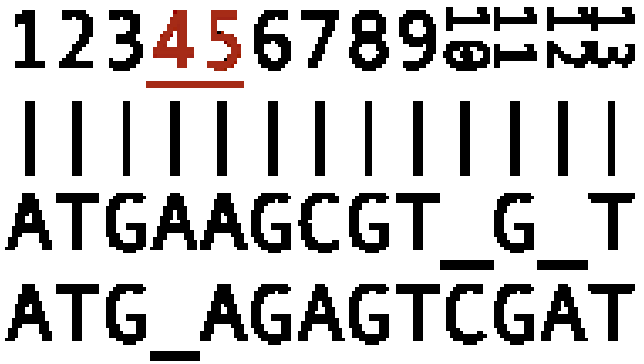
In this alignment, nine letters out of the total 13 are aligned and matching in the columns.

Some concepts in sequence alignment

- The number of matching characters is the same, independent of placement of the gap in position four or five.

Insertion





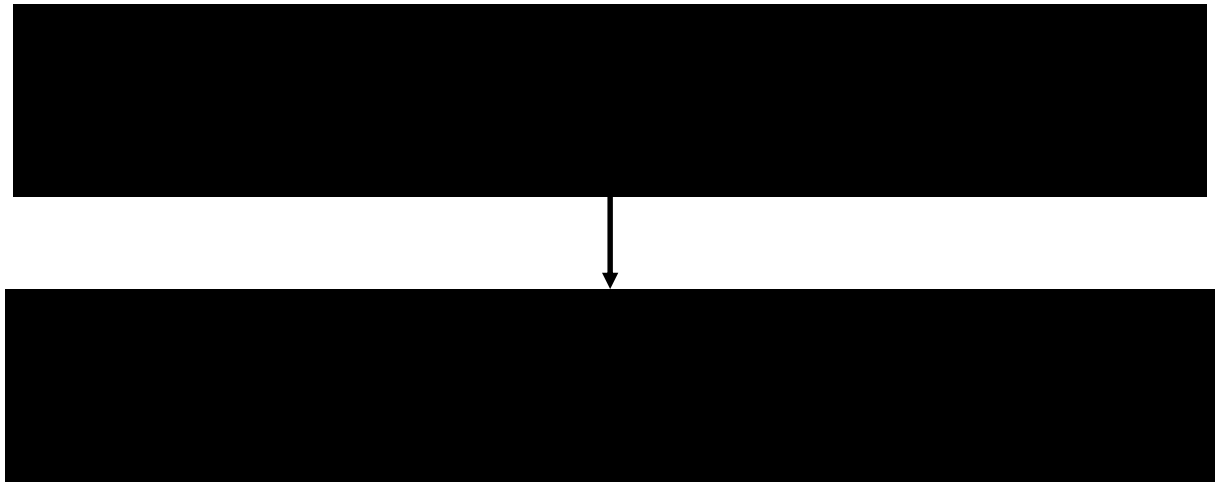
Deletion

These two alignments are considered to be separate and different alignments, although it is not possible to draw conclusions where to place the gap based on a pair-wise alignment.

Note that we are not allowed to place gaps in both sequences in the same column.

Some concepts in sequence alignment

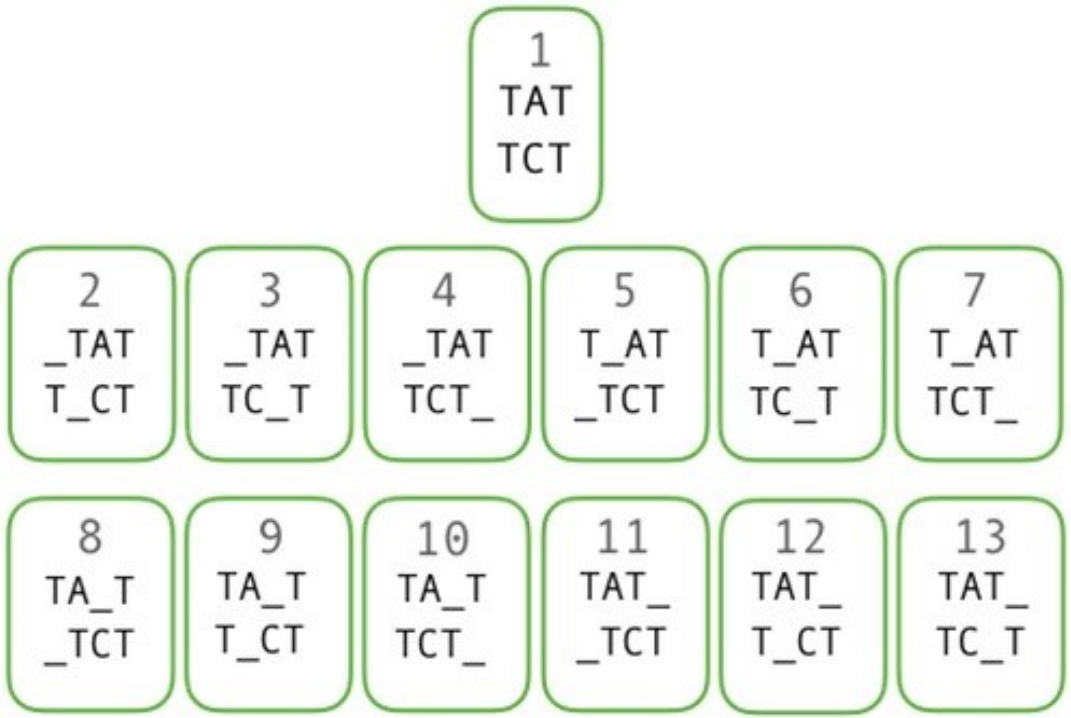
Local alignment



Global alignment

Match, Mismatch (or Substitution), Gap, Insertion, Deletion (Indel), Global and Local

Number of possible alignments



m, *n*	No. of possible alignments
1,1	3
2,2	13
3,3	63
4,4	321
5,5	1683
6,6	8989
7,7	48639
8,8	265729
9,9	1462563
10,10	8097453

Pair-wise sequence alignment

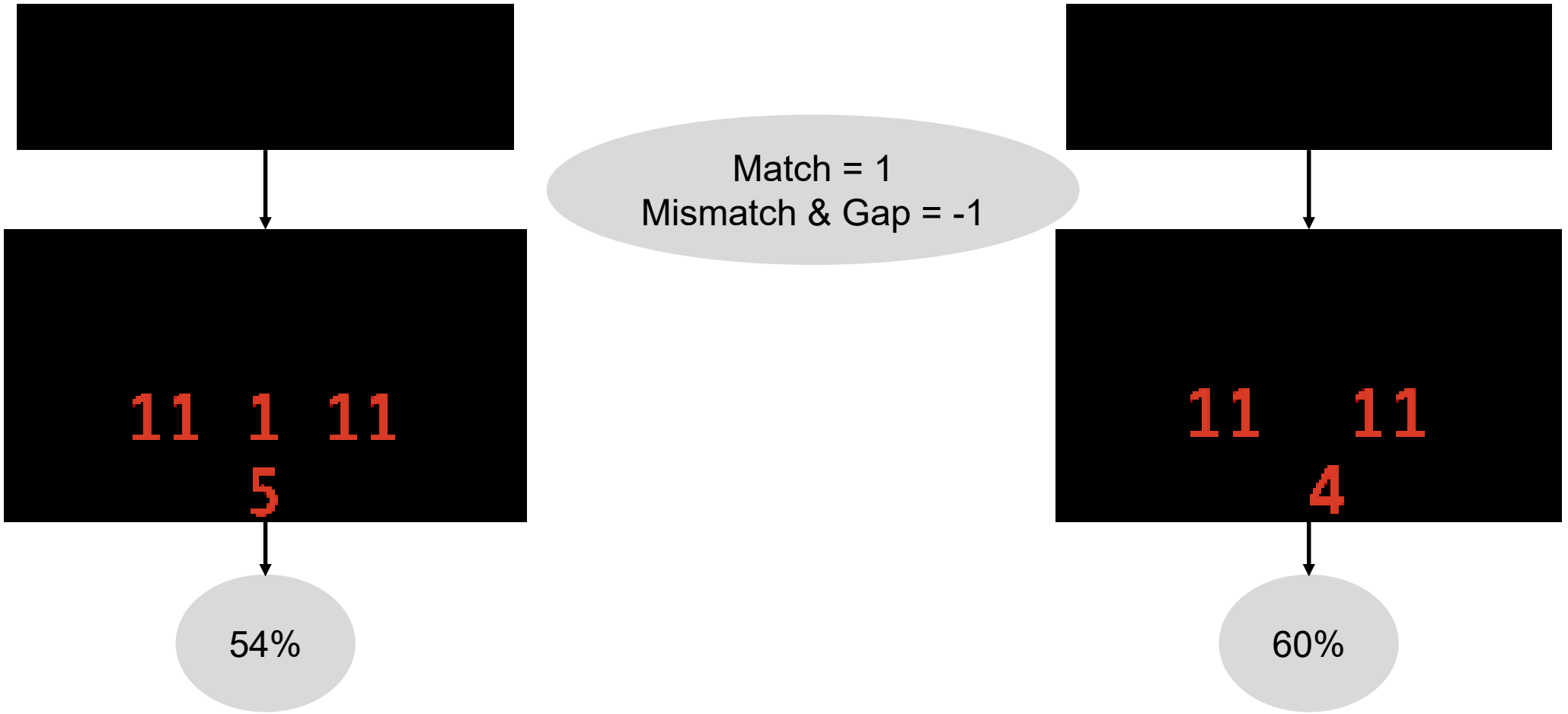
Can you tell which alignment is better?

How to assess distinct sequence alignments?

Scoring of sequence alignments

- For the demonstration purposes, let us start with a simple one and score each match with +1 and both mismatches (substitutions) and gaps (indels) with -1.
- The total alignment score is the sum of matches, mismatches and gaps.
- Proper alignment, in general, is the one having the least number of substitutions and indels.
- Alignment algorithms aim to maximize the alignment score and this way, the number of gaps and indels also get minimized, given that matches have a positive score, substitution and indels a negative score.

Scoring of sequence alignments



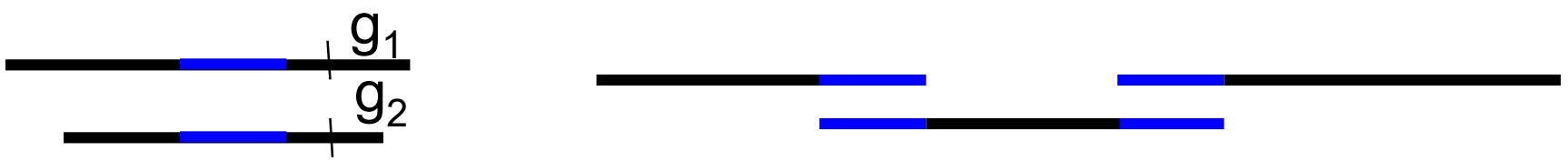
So, what does this score mean?

Significance of sequence alignment

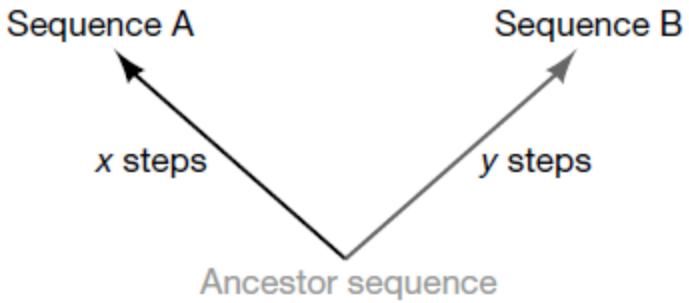
Sequence alignment is useful for discovering functional, structural, and evolutionary information in biological sequences.

Locate similar subsequences in DNA allows to identify (e.g.) regulatory elements.

Locate DNA sequences that might overlap: helps in sequence assembly.



The alignment indicates the changes that could have occurred between the two homologous sequences and a common ancestor sequence during evolution.

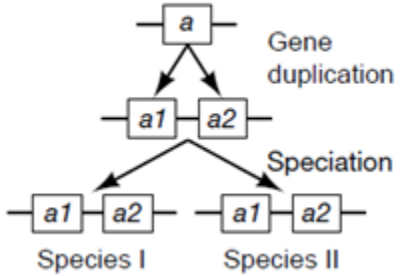


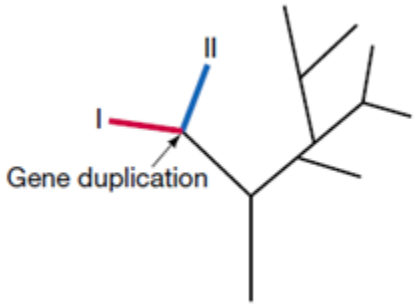
Significance of sequence alignment

Sequence similarity may be an indicator of several possible types of ancestor relationships, or there may be no ancestor relationship at all.

Paralogy and Orthology

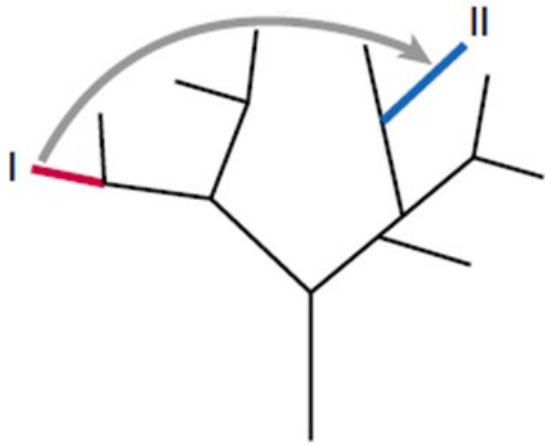
a1 in species I and a1 in species II are orthologs (they share a common ancestor). Similarly, a2 in species I and a2 in species II are orthologs. However, the a1 genes are paralogs to the a2 genes because they arose from a gene duplication event.





Xenologous or Horizontal Gene Transfer

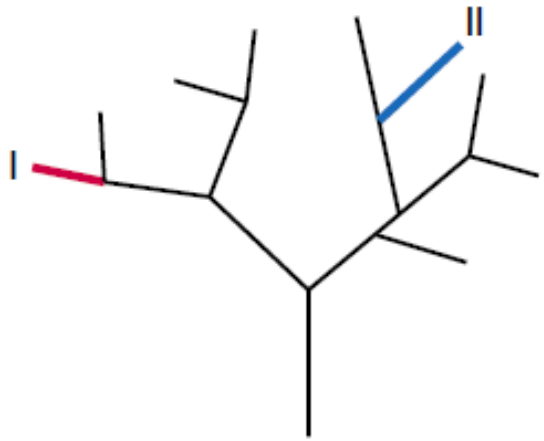
Genes in species I and II are related through the transfer of genetic material between species, even though the two species are separated by a long evolutionary distance. Although the transfer is shown between outer branches of the evolutionary tree, it could also have occurred in lower-down branches, thus giving rise to a group of organisms with the transferred gene. Such genes are known as xenologous or horizontally transferred genes.



Significance of sequence alignment

Analogy

A gene in species I and a different gene in species II have converged on the same function by separate evolutionary paths. Such analogous genes, or genes that result from convergent evolution, include proteins that have a similar active site but within a different backbone sequence.



Similarity versus Homology

- Similarity refers to the likeness or % identity between 2 sequences
- Similarity means sharing a statistically significant number of bases or amino acids
- Similarity does not imply homology

- Homology refers to shared ancestry
- Two sequences are homologous if they are derived from a common ancestral sequence
- Homology usually implies similarity

Sequence Alignment

Before we can make statements about two sequences, we have to produce a pairwise sequence alignment.

What is the optimal alignment between two sequences?

Quantitative? Match/mismatch? Gaps/Extension?

Is an optimal alignment always significant?

Random sequences?

Input: two sequences over the same alphabet

Output: an **alignment** of the two sequences

Example:

- GCGCATGGATTGAGCGA
- TGCGCCATTGATGACCA

A possible alignment:

- GCGC - ATGGATTGAGCGA

TGCGCCATTGAT – GACC - A

- GCGC - ATGGATTGAGCGA

TGCGCCATTGAT – GACC – A

Three elements:

- Perfect matches
- Mismatches
- Insertions & deletions (**indel**)

Choosing Alignments

There are many possible alignments
For example, compare:

-GCGC - ATGGATTGAGCGA
TGCGCCATTGAT – GACC - A

to

- - - - - -GCGCATGGATTGAGCGA
TGCGCC - - - - ATTGATGACCA - -

Which one is better?

Example

-GCGC - ATGGATTGAGCGA
TGCGCCATTGAT – GACC – A
Score = (+1x13) + (-1x2) + (-2 x 4) = 3

- - - - - -GCGCATGGATTGAGCGA
TGCGCC - - - - ATTGATGACCA - -
Score = (+1x5) + (-1x6) + (-2 x 11) = -23

what is “similar” enough to be relevant ?

The optimal score

- The **optimal (maximal) score** between two sequences is the maximal score of all alignments of these sequences, namely,

$$d(s_1, s_2) = \max_{\text{alignment of } s_1 \& s_2} \text{score}(\text{alignment})$$

- Computing the maximal score or actually finding an alignment that yields the maximal score are closely related tasks with similar algorithms.

Challenges

Without gaps, there are N+M-1 possible alignments between sequences of length N and M.

```
AGTT    AGTT    AGTT    AGTT      AGTT      AGTT
ACT      ACT      ACT      ACT    ACT       ACT
```

With gaps and mismatches, there are about N^M possible alignments.

```
AG      AG -     A - G     AG -
AC      - AC     AC -      A - C
```

What is a good alignment?

AGGCTAGTT and AGCGAAGTTT

```
AGGCTAGTT-
AGCGAAGTTT
```

6 matches, 3 mismatches, 1 gap

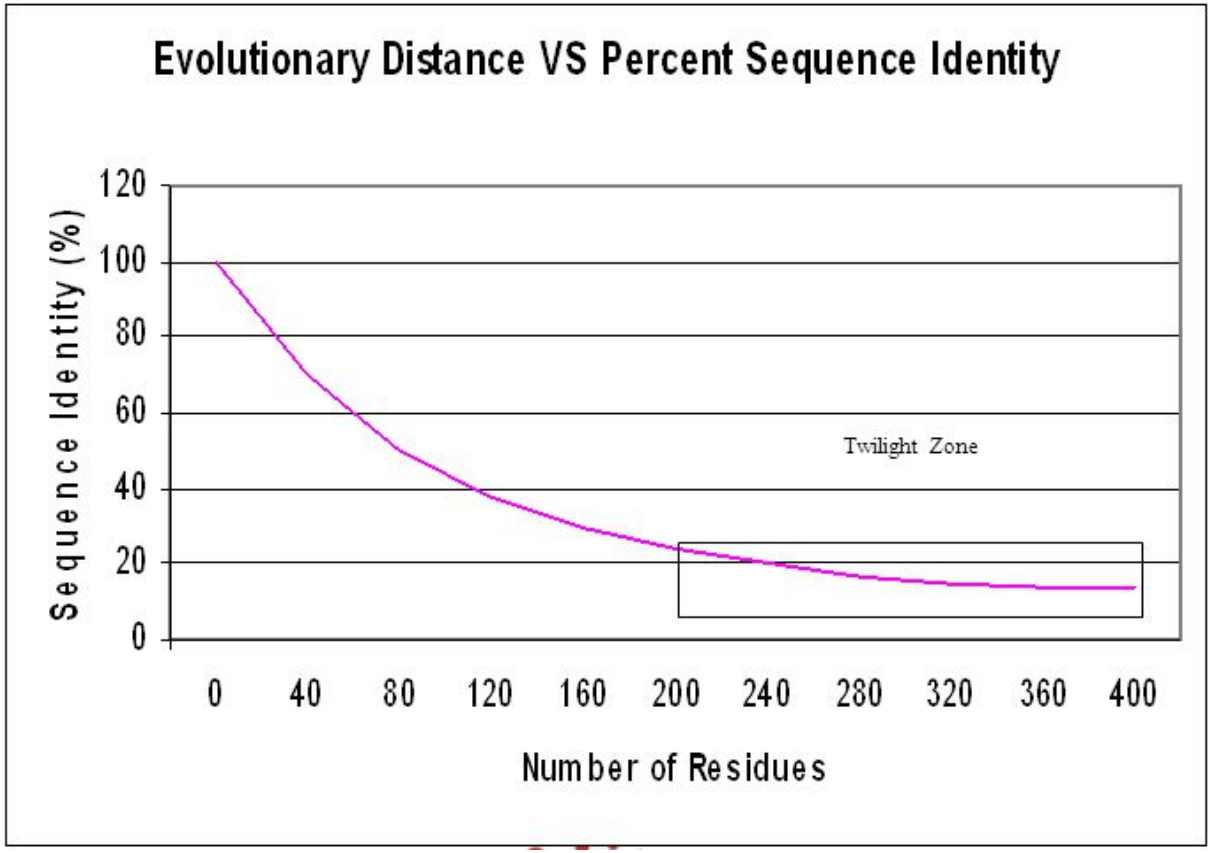
```
AGGCTA-GTT-
AG-CGAAGTTT
```

7 matches, 1 mismatch, 3 gaps

```
AGGC-TA-GTT-
AG-CG-AAGTTT
```

7 matches, 0 mismatches, 5 gaps

Doolittle's Rules of Thumb



- If two sequence are > 100 residues and > 25% identical, they are likely related.
- If two sequences are 15-25% identical they may be related, but more tests are needed.
- If two sequences are < 15% identical they are probably not related.
- If you need more than 1 gap for every 20 residues the alignment is suspicious

Methods of Alignment

- **By hand** - slide sequences on two lines of a word processor
- **Dot plot**
  - with windows
- **Rigorous mathematical approach**
  - Dynamic programming (slow, optimal)
- **Heuristic methods (fast, approximate)**
  - BLAST and FASTA
    - Word matching and hash tables

Align by Hand

Informal:

C T C **T A G C A** T T A G

G **T G C A** C C C A

Note: Prone to error for large sequences, Not systematic or quantitative.

Formal:

C T C **T A G C A** T T A G

G **T - G C A** C C C A

You still need some kind of scoring system to find the best alignment

Dotplot

- A dotplot gives an overview of all possible alignments

Sequence 1 →

Sequence 2 ↓

	T	A	C	A	T	T	A	C	G	T	A	C
A		1		1			1				1	
T	1				1	1				1		
T	1				1	1				1		
C			1					1				1
A		1		1			1				1	
C			1					1				1
A		1		1			1				1	
T	1				1	1				1		
A		1		1			1				1	

Dotplot

- Dotplot can also be used to find direct or indirect repeats within the sequences

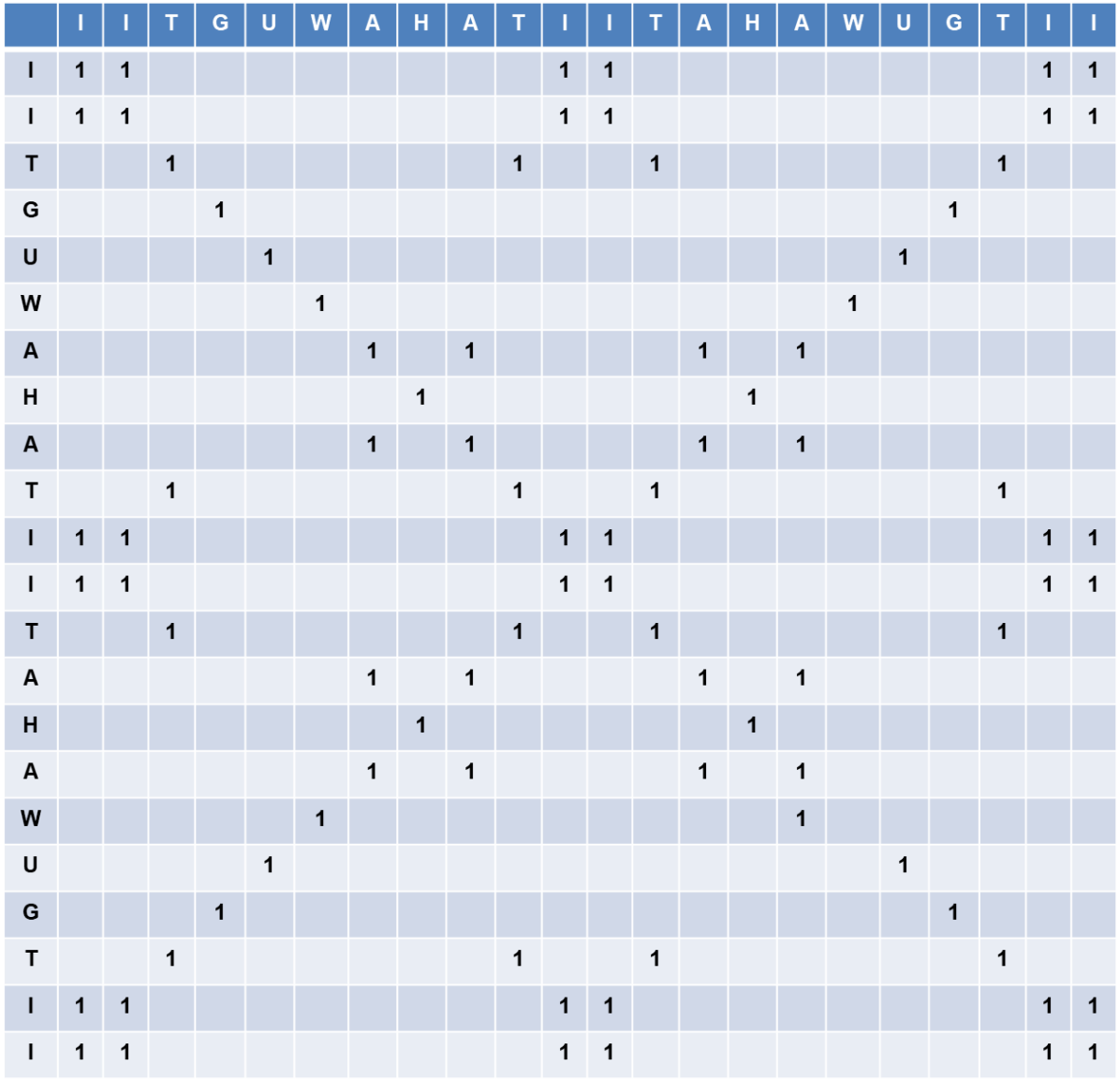
	A	B	R	A	C	A	D	A	B	R	A	C	A	D	A	B	R	A
A	1			1		1		1			1		1		1			1
B		1							1							1		
R			1							1							1	
A	1			1		1		1			1		1		1			1
C					1							1						
A	1			1		1		1			1		1		1			1
D							1							1				
A	1			1		1		1			1		1		1			1
B		1							1							1		
R			1							1							1	
A	1			1		1		1			1		1		1			1
C					1							1						
A	1			1		1		1			1		1		1			1
D							1							1				
A	1			1		1		1			1		1		1			1
B		1							1							1		
R			1							1							1	
A	1			1		1		1			1		1		1			1

Exercise: Find the internal repeats in the sequence using dot plot:

MASAMGPASSAMESCTASAMI
ITGASSAMESP

Dotplot

- Dotplot can also be used to find direct or indirect repeats within the sequences.



	I	I	T	G	U	W	A	H	A	T	I	I	T	A	H	A	W	U	G	T	I	I
I	1	1									1	1									1	1
I	1	1									1	1									1	1
T			1							1			1							1		
G				1															1			
U					1													1				
W						1											1					
A							1		1					1		1						
H								1							1							
A							1		1					1		1						
T			1							1			1							1		
I	1	1									1	1									1	1
I	1	1									1	1									1	1
T			1							1			1							1		
A							1		1					1		1						
H								1							1							
A							1		1					1		1						
W						1										1						
U					1													1				
G				1															1			
T			1							1			1							1		
I	1	1									1	1									1	1
I	1	1									1	1									1	1

Dynamic Programming

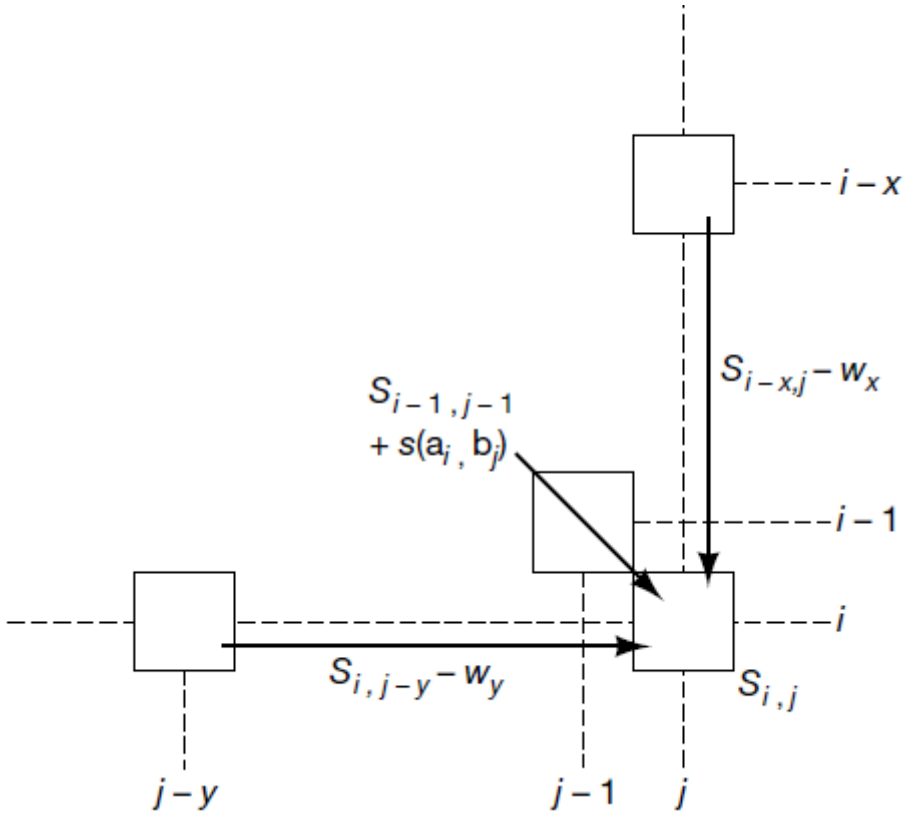
- DP provides the best (optimal) alignment between two sequences.
- Includes matches, mismatches and gaps to maximize the number of matched characters.
- Score: match, mismatch, gap (non-affine vs affine)
- DP compares every pair of character in the two sequences and generates an alignment, which is the best or optimal.

- Each alignment has its own score and thus several alignments can have identical scores. However, intelligent manipulation of some parameters may discriminate the alignment with similar scores.

- **Global alignment: Needleman-Wunsch Algorithm**

- **Local alignment: Smith-Waterman Algorithm**

Formal description of the dynamic programming algorithm



Derivation of the dynamic programming algorithm

```
1.  SCORE OF NEW      =    SCORE OF PREVIOUS +  SCORE OF NEW
    ALIGNMENT               ALIGNMENT (A)        ALIGNED PAIR

V  D  S  -  C  Y          V  D  S  -  C            Y
V  E  S  L  C  Y          V  E  S  L  C            Y

      15                =        8              +     7


II. SCORE OF          =    SCORE OF PREVIOUS +  SCORE OF NEW
    ALIGNMENT (A)           ALIGNMENT (B)        ALIGNED PAIR

V  D  S  -  C             V  D  S  -               C
V  E  S  L  C             V  E  S  L               C

      8                 =       -1              +     9

III. REPEAT REMOVING ALIGNED PAIRS UNTIL END OF ALIGNMENT IS REACHED.
```

Example of using the dynamic programming algorithm to align sequences

1.

	gap	a1	a2	a3	a4
gap	0	1 gap	2 gaps	3 gaps	4 gaps
b1	1 gap				
b2	2 gaps				
b3	3 gaps				
b4	4 gaps				

2a.

	gap	a1	a2	a3	a4
gap	0	1 gap	2 gaps	3 gaps	4 gaps
b1	1 gap	s11			
b2	2 gaps				
b3	3 gaps				
b4	4 gaps				

2c.

	gap	a1	a2	a3	a4
gap	0	1 gap	2 gaps	3 gaps	4 gaps
b1	1 gap	s11	s21		
b2	2 gaps	s12			
b3	3 gaps				
b4	4 gaps				

2d.

	gap	a1	a2	a3	a4
gap	0	1 gap	2 gaps	3 gaps	4 gaps
b1	1 gap	s11	s21		
b2	2 gaps	s12	s22		
b3	3 gaps				
b4	4 gaps				

2b.

	gap	a1	a2	a3	a4
gap	0	1 gap	2 gaps	3 gaps	4 gaps
b1	1 gap	s11			
b2	2 gaps	s12			
b3	3 gaps				
b4	4 gaps				

3. Part of trace back matrix

	gap	a1	a2	a3	a4
gap	0	1 gap	2 gaps	3 gaps	4 gaps
b1	1 gap	s11	s21	s31	s41
b2	2 gaps	s12	s22	s32	s42
b3	3 gaps	s13	s23	s33	s43
b4	4 gaps	s14	s24	s34	s44

4. Trace back matrix

	gap	a1	a2	a3	a4
gap	0	1 gap	2 gaps	3 gaps	4 gaps
b1	1 gap	s11	s21 B	s31	s41
b2	2 gaps	s12	s22	s32	s42
b3	3 gaps	s13	s23 A	s33	s43
b4	4 gaps	s14	s24	s34	s44

Alignment A:

a1	a2	a3	a4
b1	b2	b3	b4

Alignment B:

a1	a2	a3	a4	–
b1	–	b2	b3	b4

Dynamic Programming Can Provide Global or Local Sequence Alignments

Global alignment: Needleman-Wunsch Algorithm

$$S(0,0) = 0$$

$$S(i,j) = \max \begin{cases} S(i-1, j-1) + w(x_i, y_j) \\ S(i-1, j) + Gap\_Penalty \\ S(i, j-1) + Gap\_Penalty \end{cases}$$

Time complexity

Space: *O(mn)*

Time: *O(mn)*

- Filling the matrix *O(mn)*
- Backtrace *O(m+n)*

Example

Find an optimal sequence alignment between two sequences:
1. GAAGA 2. GTTTAAG

Score: match = +1, mismatch = -1, gap penalty = -3

		G	A	A	G	A
	0	-3	-6	-9	-12	-15
G	-3	1	-2	-5	-8	-11
T	-6	-2	0	-3	-6	-9
T	-9	-5	-3	-1	-4	-7
T	-12	-8	-6	-4	-2	-5
A	-15	-11	-7	-5	-5	-1
A	-18	-14	-10	-6	-6	-4
G	-21	-17	-13	-9	-5	-7

Possible optimal alignments

1. GAAGA - -
   GT TT AAG
   Score = -7
2. G – A – AGA
   G T T T AAG
   score = -7
3. G - - AAGA
   GTTT AAG
   score = -7
4. G – AAGA –
   G T T T AAG
   score = -7
5. GAAA – G -
   GTTT A AG
   score = -7
6. GAA – GA -
   GTT T AAG
   score = -7

Exercise: Find the global alignment between two sequences CTCAGTGT and GTCAGTT

Local alignment: Smith-Waterman Algorithm

The values in the first row and first column are set to zero

$$S(i,j) = \max \begin{cases} 0 \\ S(i-1, j-1) + w(x_i, y_j) \\ S(i-1, j) + Gap\_Penalty \\ S(i, j-1) + Gap\_Penalty \end{cases}$$

Find the maximum score obtained in the entire matrix and start trace backing from there.

It is possible that the maximum scores can be present in more than one cell, in that case there may be possibility of two or more alignments and the best alignment by scoring it.

Cross-check: No negative values in the matrix.

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

Score: match = +1
Mismatch = -1
Gap penalty = -3

Optimal alignment:

AAG
AAG
Score = 3

Exercise: Find the local alignment between GATGTAGTCT and CTAGAGCTAG

Score: match = +1
Mismatch = -1
Gap penalty = -3

Overlap Alignment

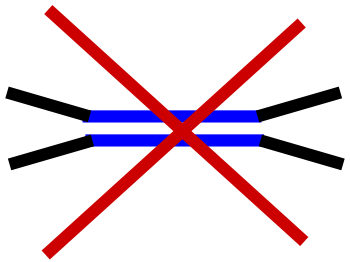
Consider the following problem:

1. Find the most significant overlap between two sequences S, T ?
2. Possible overlap relations:

a.

b.

Difference from local alignment: alignment between the endpoints of the two sequences.

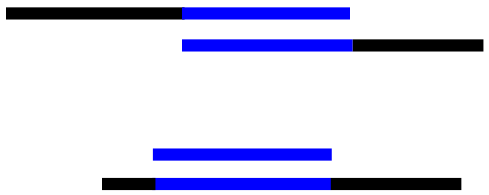


Formally: given *S[1..n]* , *T[1..m]* find *i,j* such that:

d= max { D(S[1..i], T[j..m]), D(S[i..n],T[1..j]),

D(S[1..n],T[i..j]), D(S[i..j],T[1..m])

}



Solution: Same as Global alignment except we do not penalise overhanging ends.

Initialization: S[0,0] = 0, *S[i,0]=0, S[0,j]=0 for all i and j.*

Recurrence: as in global alignment

$$S[i,j] = \max \begin{cases} S[i-1, j-1] + w(x_i, y_j) \\ S[i-1, j] + gap\_penalty \\ S[i, j-1] + gap\_penalty \end{cases}$$

Score: maximum value at the bottom line and rightmost line

Example

S = PAWHEAE
T = HEAGAWGHEE

Scoring scheme:
Match: +4
Mismatch: -1
Indel: -5

		H	E	A	G	A	W	G	H	E	E
	0	0	0	0	0	0	0	0	0	0	0
P	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
A	0	-1	-2	3	-2	3	-2	-2	-2	-2	-2
W	0	-1	-2	-2	2	-2	7	2	-3	-3	-3
H	0	4	-1	-3	-3	1	2	6	6	1	-4
E	0	-1	8	3	-2	-4	0	1	5	10	5
A	0	-1	3	12	7	2	-3	-1	0	5	9
E	0	-1	3	7		6	1	-4	-2	4	9

The best overlap is:

PAWHEAE - - - -- --
- - - HEAGAWGHEE

Exercise

Scoring scheme :
Match: +4
Mismatch: -1
Indel: -2

		H	E	A	G	A	W	G	H	E	E
	0	0	0	0	0	0	0	0	0	0	0
P	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
A	0	-1	-2	3	1	3	1	-1	-2	-2	-2
W	0	-1	-2	-1	2	1	7	5	3	1	-1
H	0	4	2	0	0	1	5	6	9	6	4
E	0	2	8	6	4	2	3	4	7	13	11
A	0	0	6	12	10	8	6	4	5	11	12
E	0	-1	4	10	11	9	7	5	3	9	15

- - - PAW - HEAE
HEAGAWGHEE -

Limitations of Dynamic Programming

- When handling a large number of sequences, Dynamic Programming algorithms are too slow to be practical.

- For example, Genbank contains 654,057,069,549 bases i.e. ~650 billion bases. If we were to use a Dynamic Programming algorithm to search the database with a sequence length of 1,000 bases, the algorithm is required to make 654,057,069,549x1,000 comparisons, which is in total 654,057,069,549,000, i.e., about ~650 trillion comparisons.

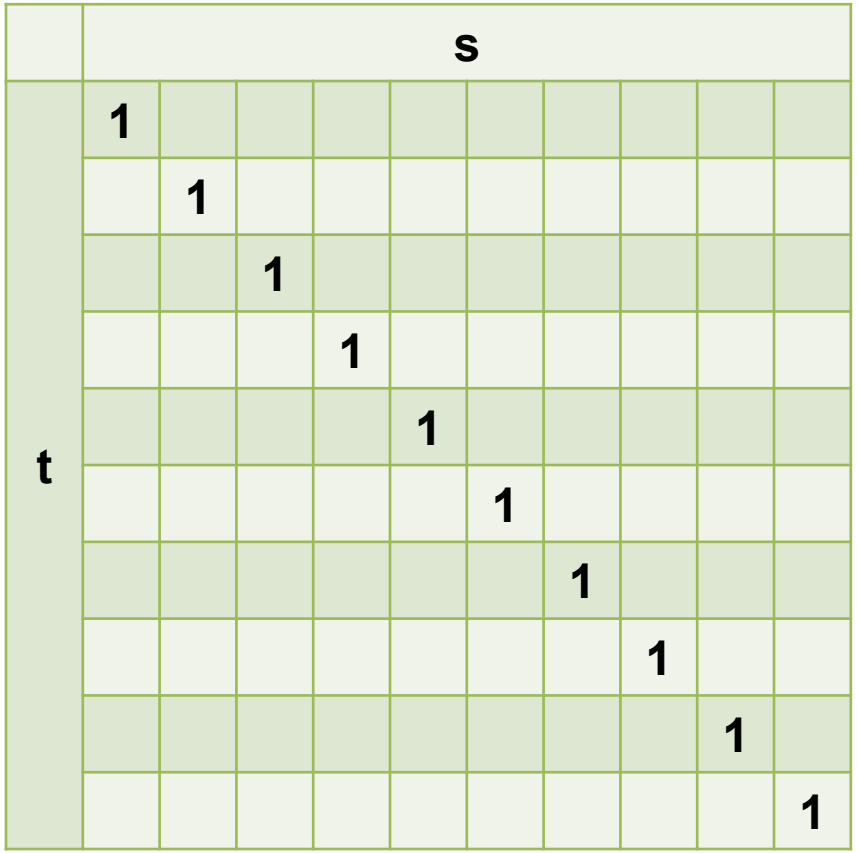
- A modern fast mainframe digital computer can perform 1,00,00,00,000 (10^9) operations per second. The time taken to compare ~650 trillion would be approximately 650,000 seconds (= 10,833 mins or 180 hrs or 7.5 days).

- That is for a single sequence. Many people are simultaneously searching the same database.

So, what is the solution?

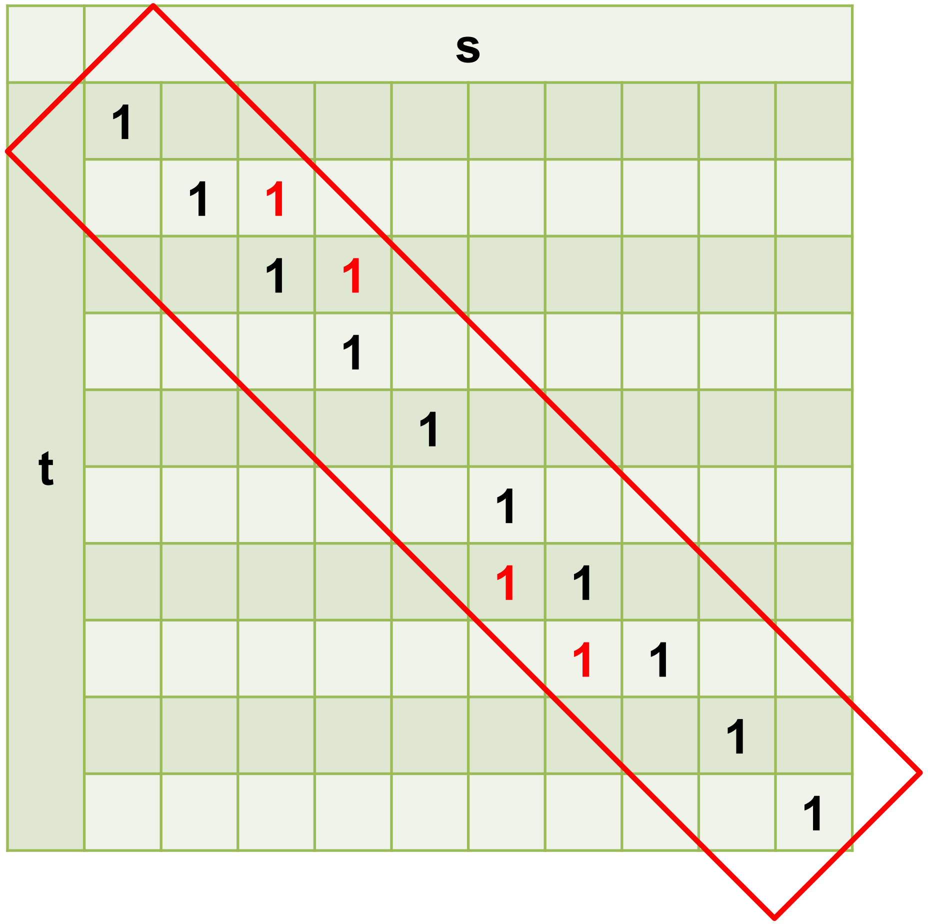
Banded Dynamic Programming

- Suppose that we have two strings *s[1..n]* and *t[1..m]* such that $n \approx m$. If the optimal alignment of s and t has few gaps, then path of the alignment will be close to diagonal.



Banded Dynamic Programming

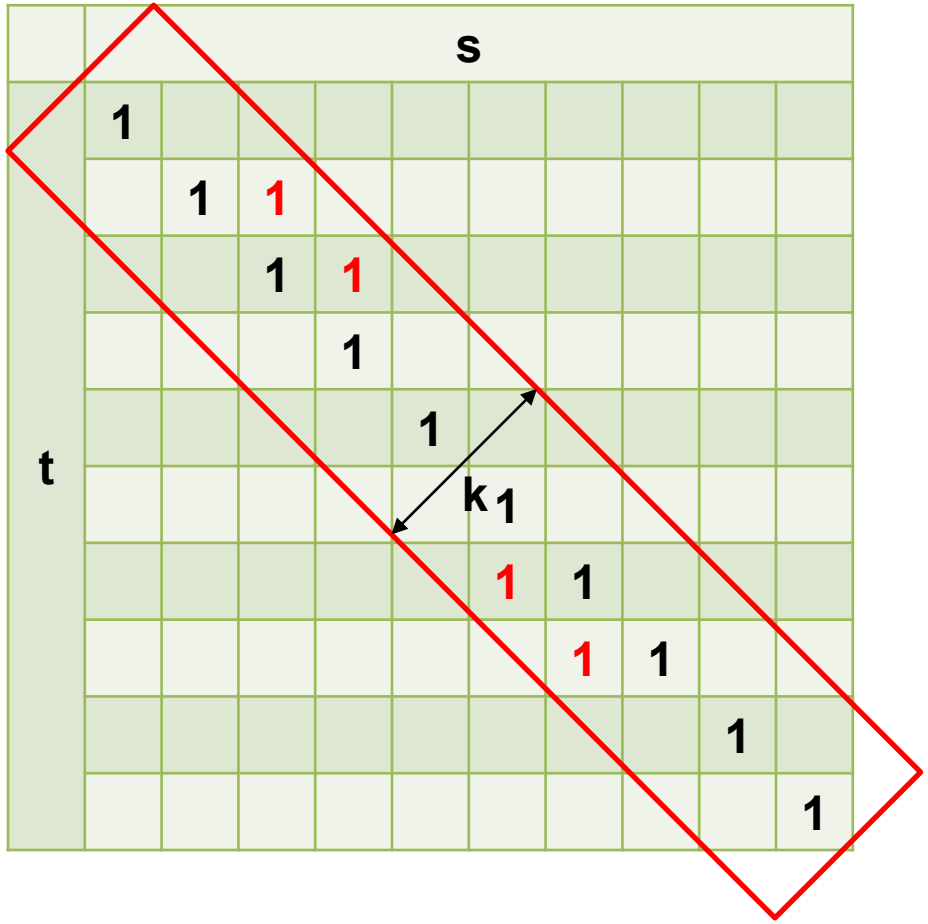
- Suppose that we have two strings *s[1..n]* and *t[1..m]* such that $n \approx m$. If the optimal alignment of s and t has few gaps, then path of the alignment will be close to diagonal.



Banded Dynamic Programming

- Suppose that we have two strings *s[1..n]* and *t[1..m]* such that $n \approx m$. If the optimal alignment of s and t has few gaps, then path of the alignment will be close to diagonal.

- To find such a path, it suffices to search in a diagonal band of the matrix.



- If the diagonal band consists of *k* diagonals (width *k)*, then dynamic programming takes *O(kn)*, *m*uch faster than $O(n^2)$ of standard DP.

Thank You