Pairwise sequence comparison

Dotplots and dynamic programming

Adapted from the courses of the Bonsai team,

CRIStAL UMR 9189

Sylvain.legrand@univ-lille.fr

Introduction

Why compare two sequences?

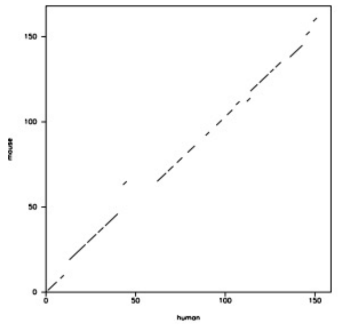
- **Assemble** a set of sequenced DNA fragments (fragment assembly)
- **Search for homology** (between genes, mRNA, proteins…)
- Find **similar regions** (protein domains)
- Identify **intron/exon positions** (comparison of a gene and its mRNA(s))

How to compare two sequences?

- **2 approaches**

  - **Dotplots** (Dot-matrix plots)

  A **graphical method** for the comparison of two sequences or a sequence against itself

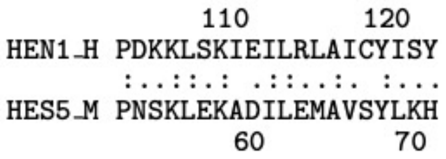


  - **Alignment**

  A **text comparison method**

  → Using **dynamic programming** → optimal alignment

  → Using **heuristic** methods (Blast) → Fast, useful when classic methods are too slow. But favours speed at the expense of optimality or precision.



```
                   110         120
HEN1_H  PDKKLSKIEILRLAICYISY
        :..::.: .::..:. :...
HES5_M  PNSKLEKADILEMAVSYLKH
                     60          70
```

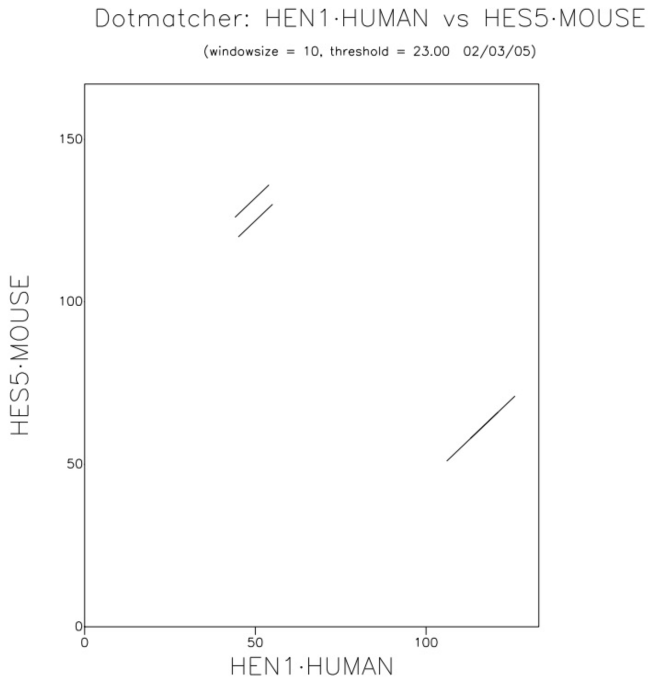
Dotplots

Dotplots

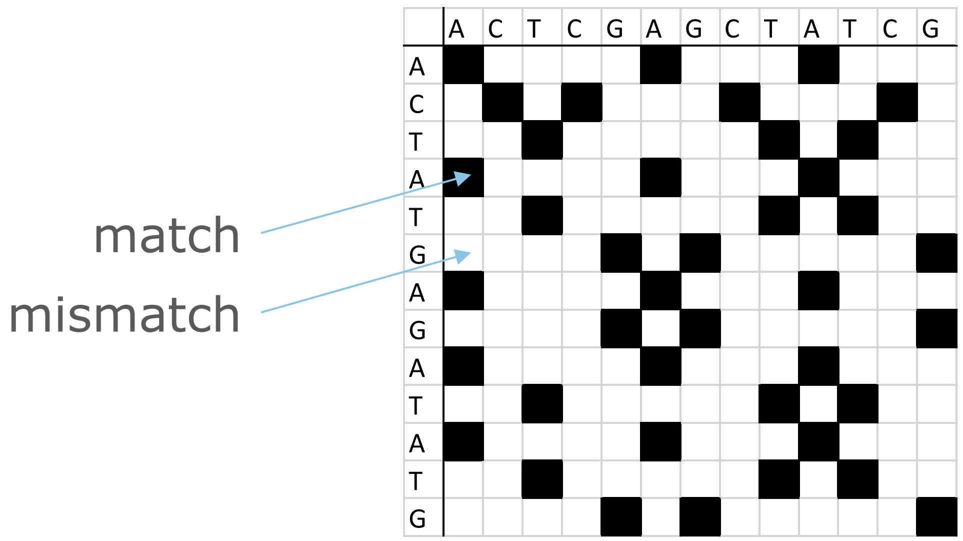
- A **graphical tool** for the comparison of two sequences

- **Method**
  - Put the two sequences along the axes of a matrix (x, y)
  - Draw a point where there is a match between the two sequences

- **A diagonal** (a series of points) represents a **similar region**

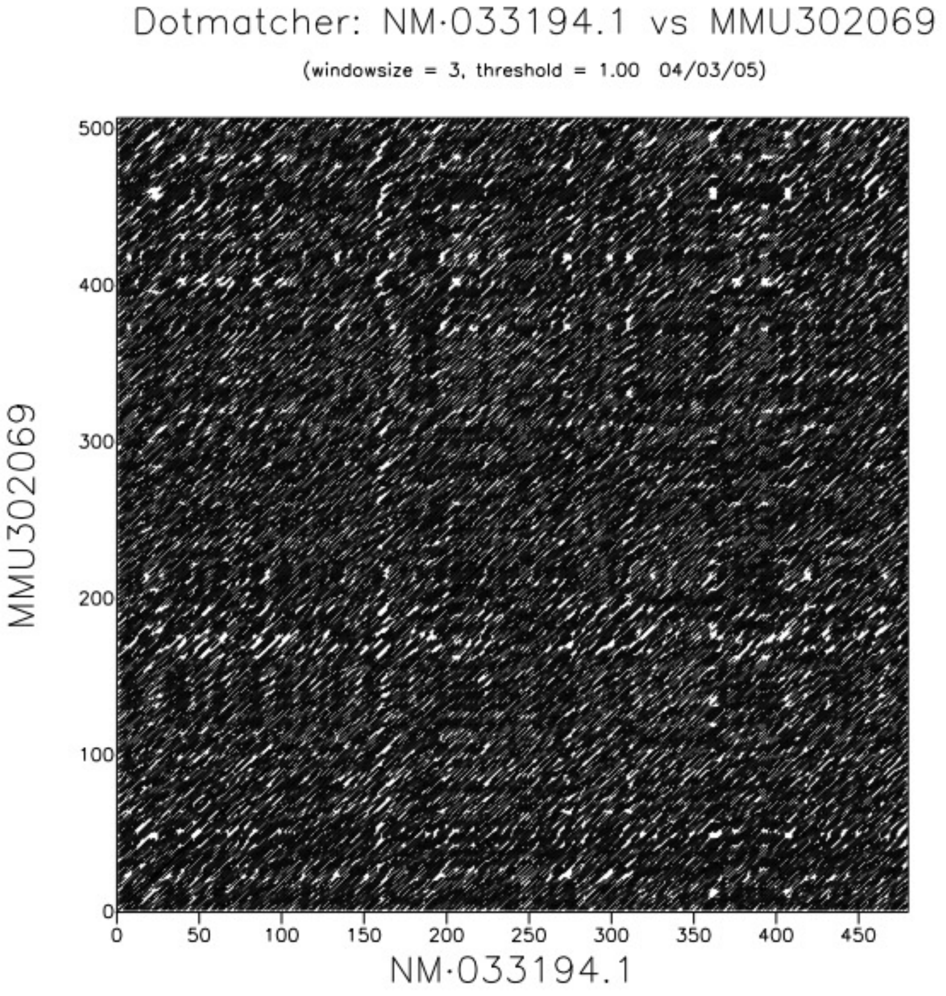


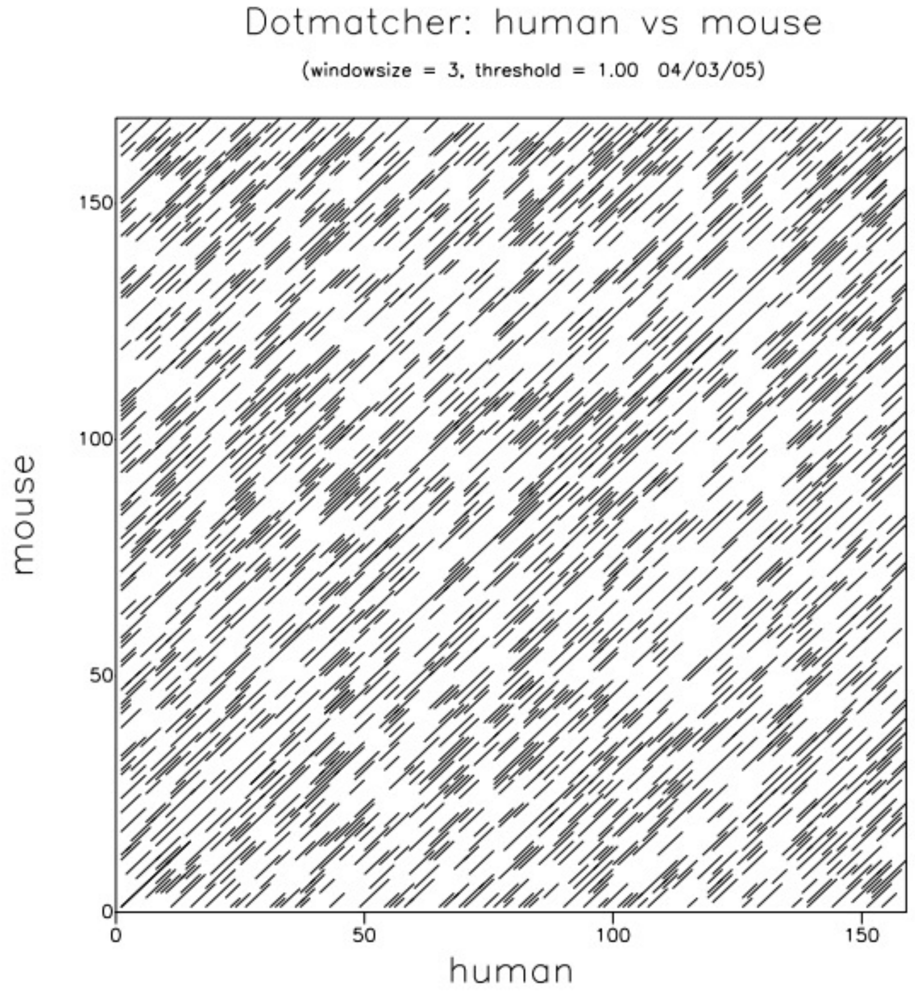
Dotplot example



	A	C	T	C	G	A	G	C	T	A	T	C	G
A	■					■				■			
C		■		■				■				■	
T			■						■		■		
A	■					■				■			
T			■						■		■		
G					■		■						■
A	■					■				■			
G					■		■						■
A	■					■				■			
T			■						■		■		
A	■					■				■			
T			■						■		■		
G					■		■						■

Dotplot example





Two genes

The related proteins

→ Noise problem (too low filtering)

Université de Lille

Dotplot filtering

- **Using a word** of size `k`
  - a sliding window of a defined length (`k`) that moves through the matrix comparing the two sequences

  - only represent exact windows: high selectivity/low sensitivity

  - Example of software: **dottup**

  - Example with `k=3`

	A	C	T	C	G	A	G	C	T	A	T	C	G
A	■					■				■			
C		■		■				■				■	
T			■						■		■		
A	■					■				■			
C		■		■				■				■	
G					■		■						■
A	■					■				■			
G					■		■						■
A	■					■				■			
T			■						■		■		
A	■					■				■			
T			■						■		■		
G					■		■						■

→

	A	C	T	C	G	A	G	C	T	A	T	C	G
A	■												
C		■						■					
T			■						■				
A										■			
C				■									
G					■								
A						■							
G							■						
A													
T									■				
A										■			
T											■		
G													

Dotplot filtering

- **Using** a **sliding window** and **score threshold** (Maizel & Lenk - 1981)

  - a sliding window of a defined length (`k`) that moves through the matrix comparing the two sequences

  - Represent windows with a score ≥ `S`: high selectivity/high sensitivity

  - Software example: **dotmatcher**

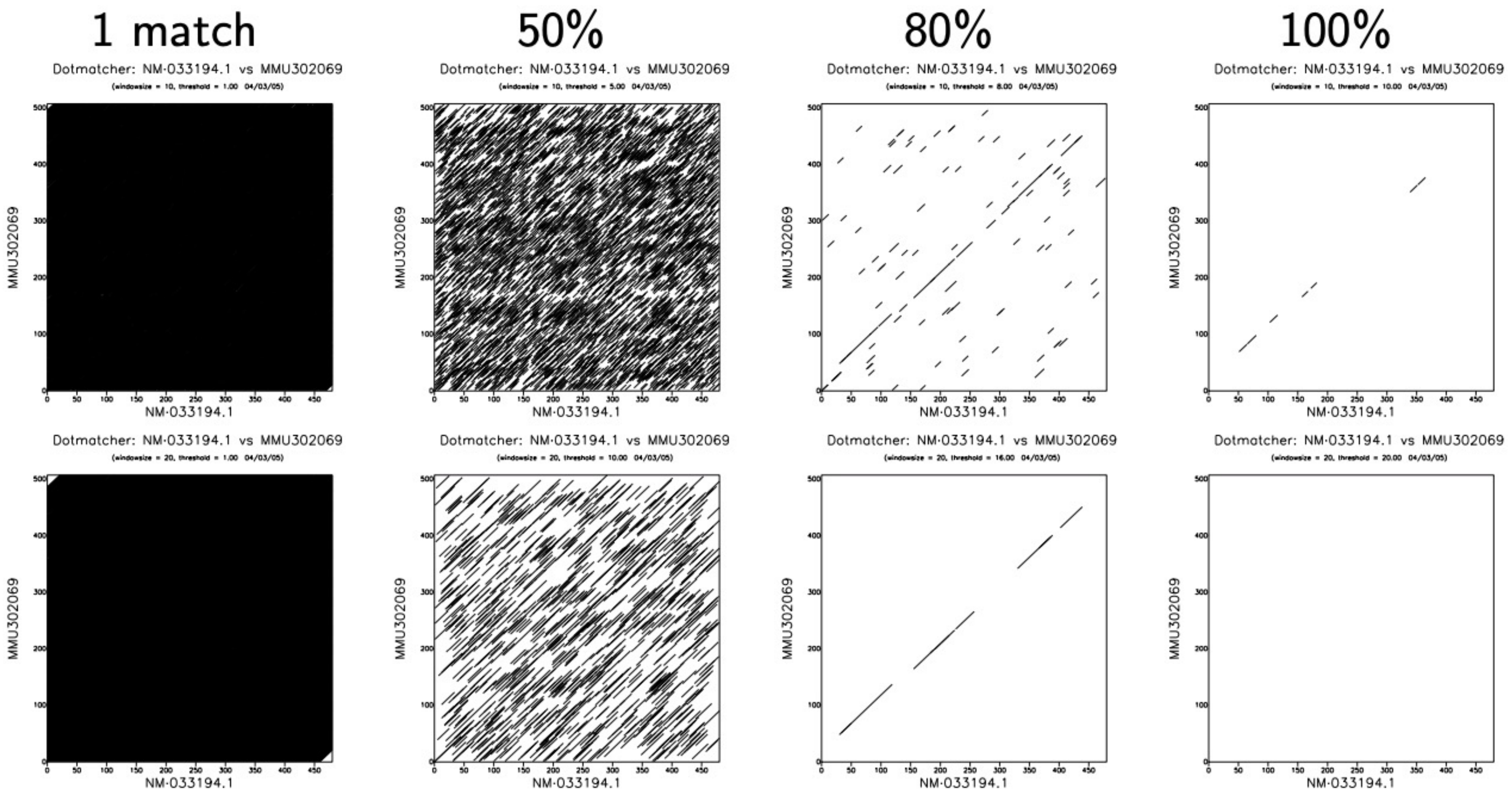
  - Example with `k=4 and 75% identity`

	A	C	T	C	G	A	G	C	T	A	T	C	G
A	■						■						
C		■						■					
T			■						■				
A				■						■			
C				■							■		
G					■								
A						■							
G							■						
A								■					
T									■				
A										■			
T											■		
G												■	

Université de Lille

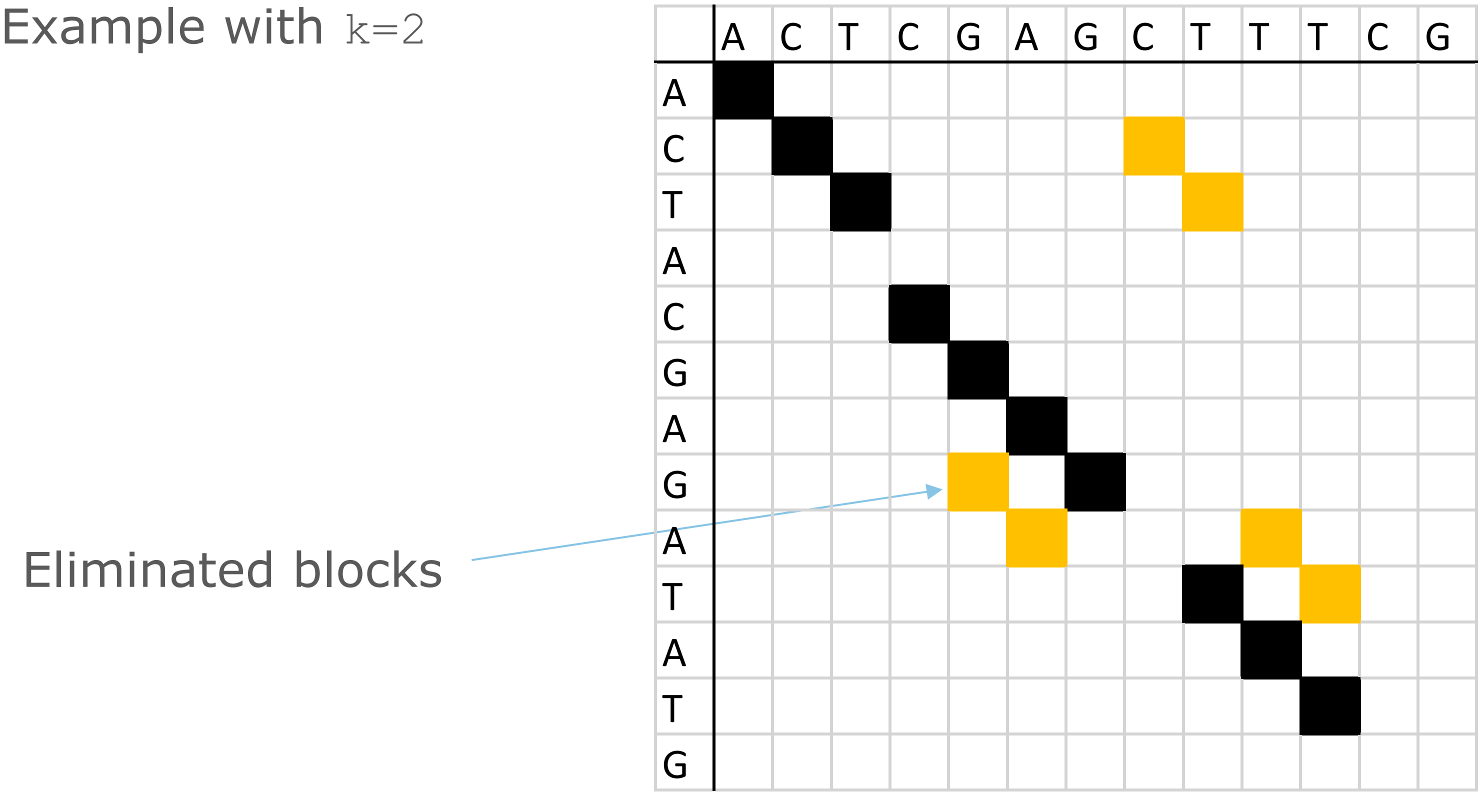
Dotmatcher, examples

- K= 10 and 20, threshold from 1% to 100%

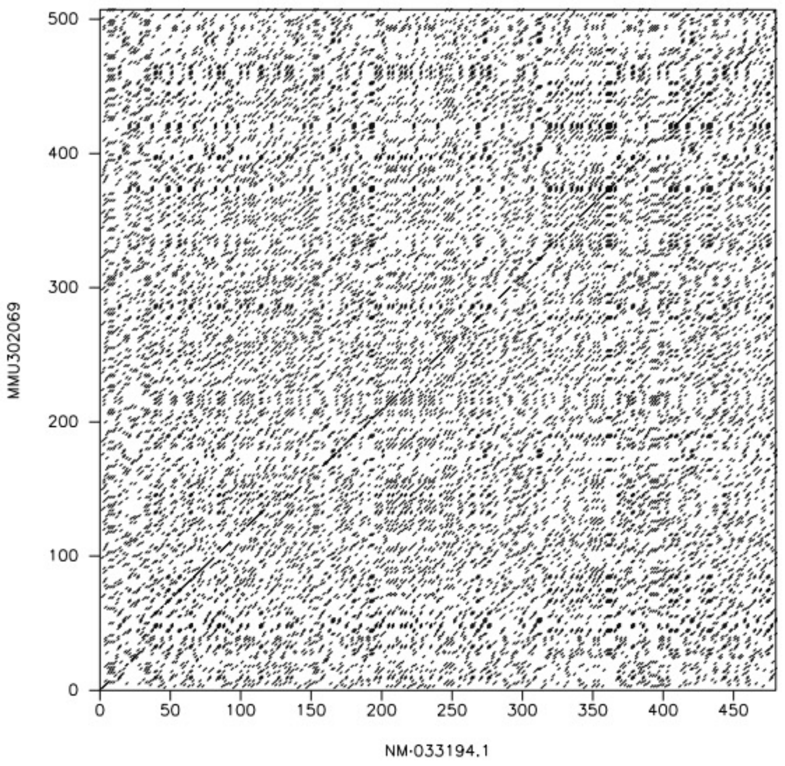


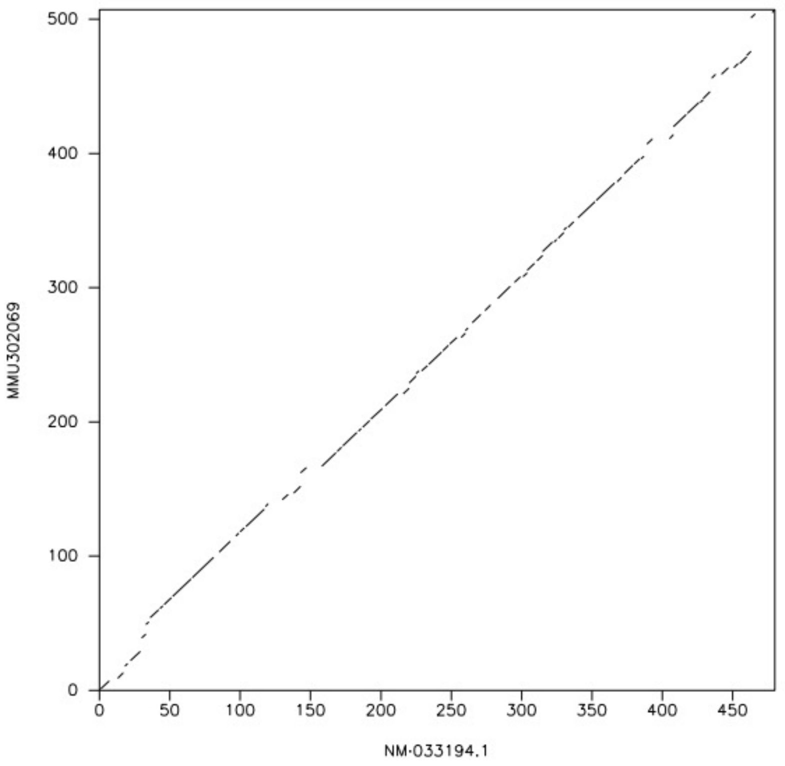
Dotplot filtering

- **By eliminating overlapping blocks**
  - Observe the overall resemblance
  - Software example: **dotpath**
  - Example with `k=2`



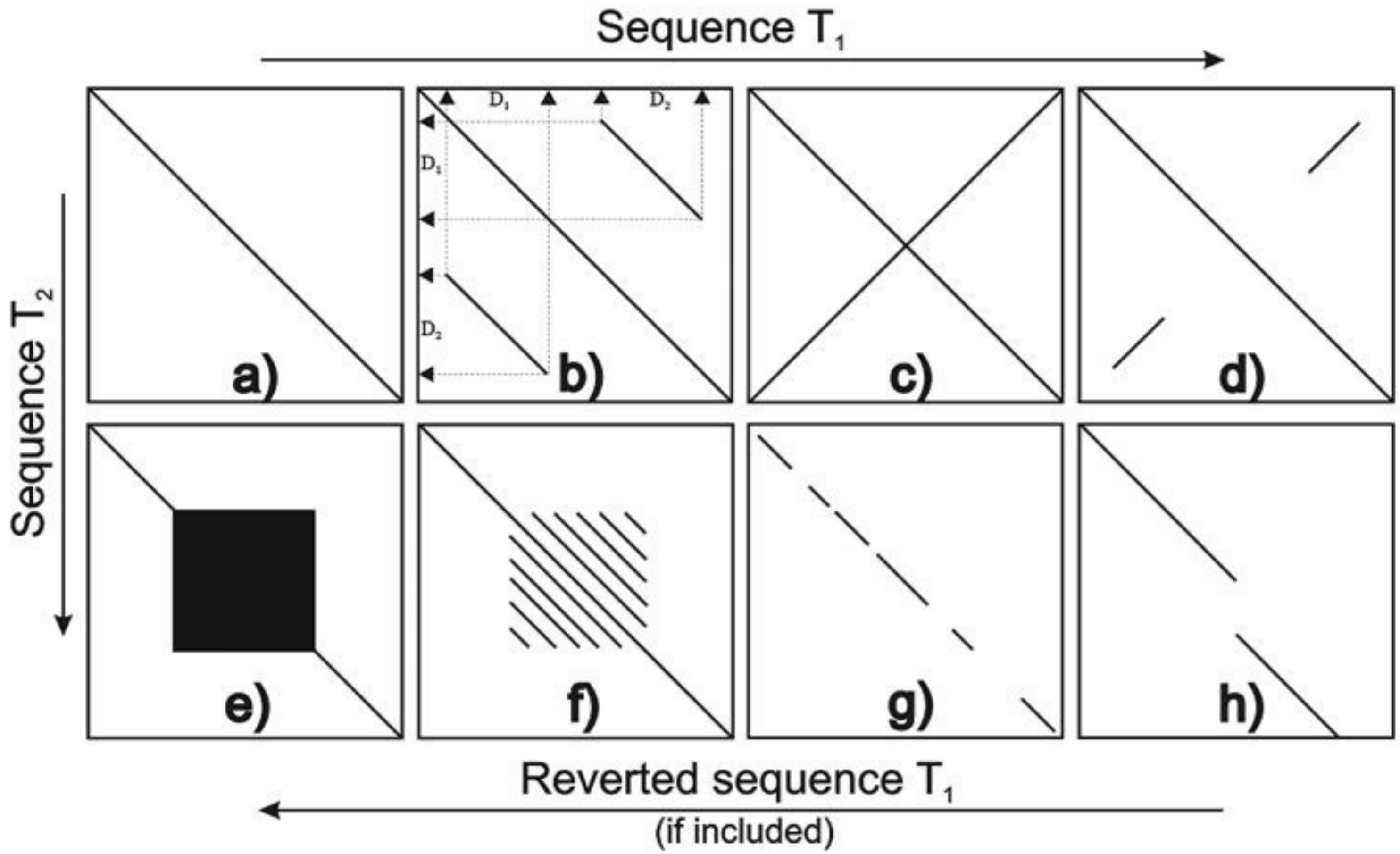
Dotpath, examples





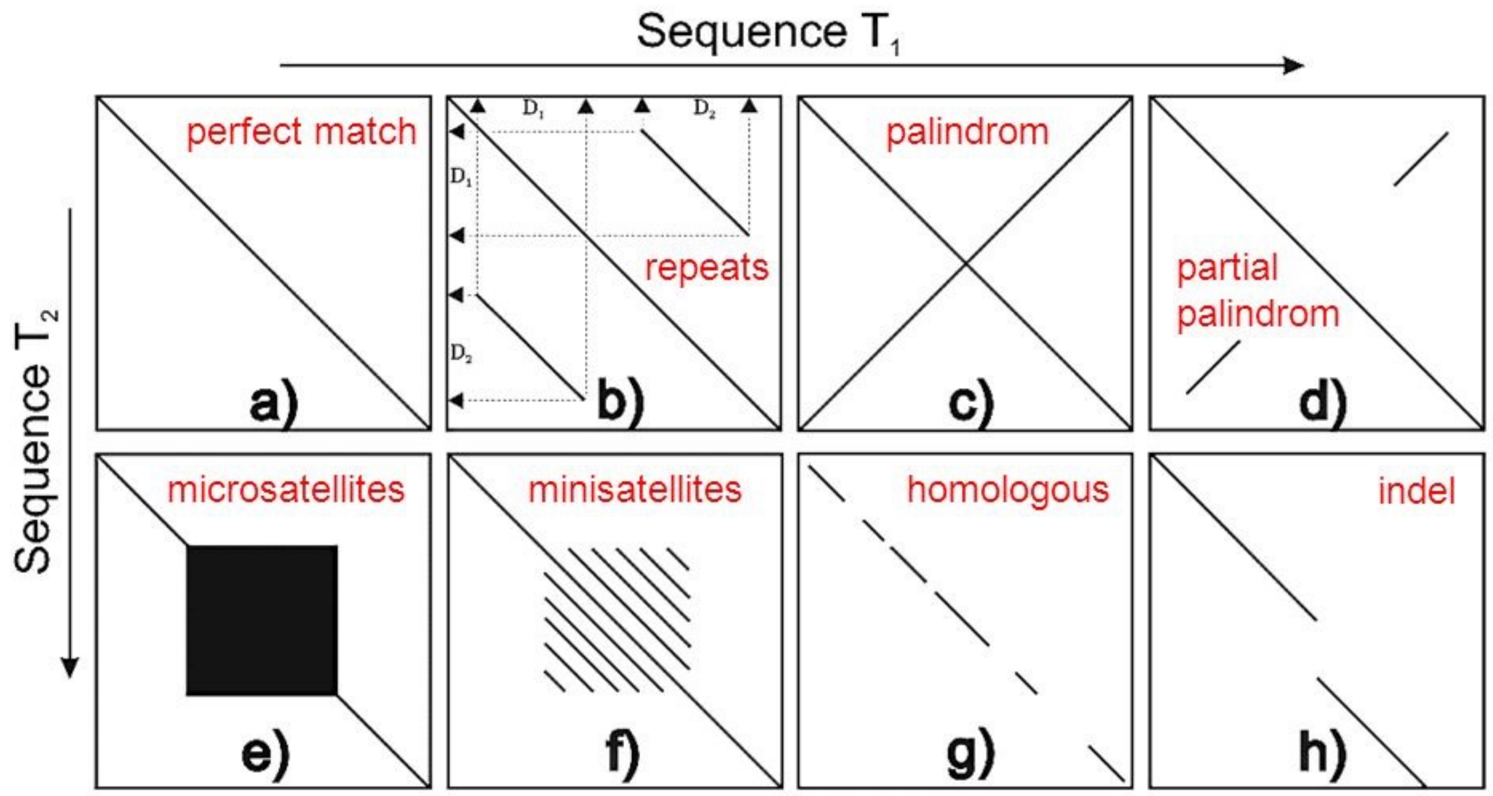
dotpath finds all matches of size wordsize *or greater between two sequences. It then reduces the matches found to the minimal set of long matches that do not overlap. This is a way of finding the (nearly) optimal path aligning two sequences. It is not the true optimal path as produced by the algorithms used in water or needle, but for very closely related sequences it will produce the same result and will work well with very long sequences*

Interpretation of dotplots



http://www.code10.info/index.php?option=com_content&view=article&id=64:inroduction-to-dot-plots&catid=52:cat_coding_algorithms_dot-plots&Itemid=76

Interpretation of dotplots



Dotplots, summary

- **Advantages**
  - Simple
  - Very informative

- **Cons**
  - Identification: no automatic detection method
  - Interpretation: no objective measurement

→ Need for a quantitative measure of similarity

Pairwise alignment

Pairwise alignment

- **2+1 types of alignments**
  - **global**: align **every residue** in the two sequences.
  When the two **sequences are similar** and of roughly equal size

  - **local**: match on **sub-sequences**
  For **dissimilar sequences** that are suspected to contain **regions of similarity**

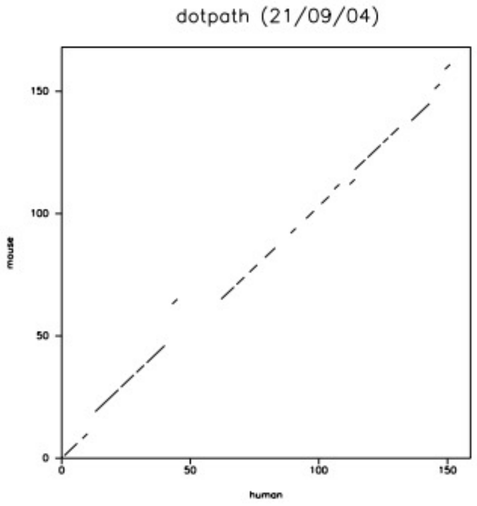
  - **semi-global**: derived from global
  When you align a **short** sequence **against** a **longer** one or when the **downstream** part of one sequence **overlaps** with the **upstream** part of the other sequence

Example global alignment

- Example: heat shock protein beta 9 from human and mouse

```
>human
MQRVGNTFSN ESRVASRCPS VGLAERNRVA TMPVRLLRDS PAAQEDNDHA RDGFQMKLDA
HGFAPEELVV QVDGQWLMVT GQQQLDVRDP ERVSYRMSQK VHRKMLPSNL SPTAMTCCLT
PSGQLWVRGQ CVALALPEAQ TGPSPRLGSL GSKASNLTR
>mouse
MQRVGSSFST GQREPGENRV ASRCPSVALA ERNQVATLPV RLLRDEVQGN GCEQPSFQIK
VDAQGFAPED LVVRIDGQNL TVTGQRQHES NDPSRGRYRM EQSVHRQMQL PPTLDPAAMT
CSLTPSGHLW LRGQNKCLPP PEAQTGQSQK PRRGGPKSSL QNESVKNP
```

```
human   1 MQRVGNTFS------NESRVASRCPSVGLAERNRVATMPVRLLRDSPAAQ
          |||||::||      .|:|||||||||.|||||:|||:|||||||.   .
mouse   1 MQRVGSSFSTGQREPGENRVASRCPSVALAERNQVATLPVRLLRDE---V

human  45 EDNDHARDGFQMKLDAHGFAPEELVVQVDGQWLMVTGQQQLDVRDPERVS
          :.|...:..||:|:||.|||||:|||::|||.|.||||:|.:..||.|..
mouse  48 QGNGCEQPSFQIKVDAQGFAPEDLVVRIDGQNLTVTGQRQHESNDPSRGR

human  95 YRMSQKVHRKM-LPSNLSPTAMTCCLTPSGQLWVRGQCVALALPEAQTG
          |||.|.|||:| ||..|.|.||||.|||||.||:|||...|..||||||.
mouse  98 YRMEQSVHRQMQLPPTLDPAAMTCSLTPSGHLWLRGQNKCLPPPEAQTGQ

human 144 S--PRLGSLGSKASNLTR-----      159
          |  ||.|  |.|:|....
mouse 148 SQKPRRG--GPKSSLQNESVKNP      168
```

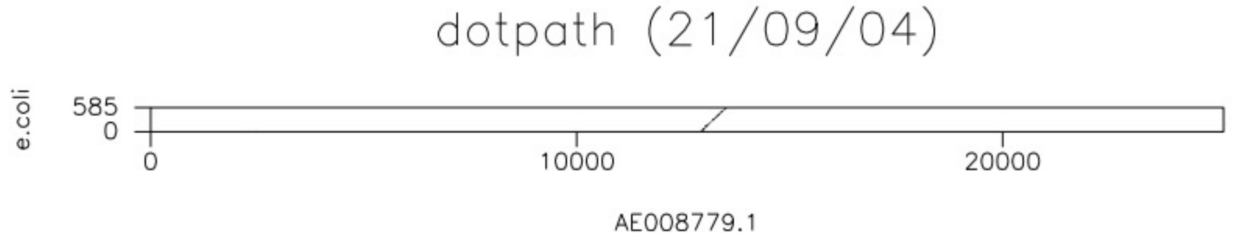


Example semi-global alignment

- Example: alignment of a genomic region and a tRNA

AE008779.1 : *Salmonella typhimurium LT2, section 83 of 220 of the complete genome. 25184 bp*

e.coli : *Escherichia coli peptidyl tRNA hydrolase. 585 bp*



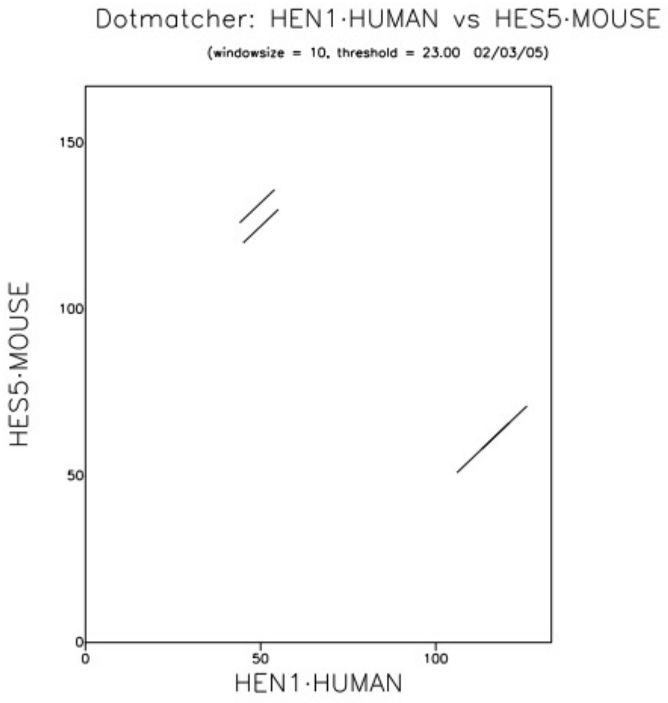
```
AE008779.1 12901 TCAGGACAAAAAACGTGGCAATTAAATTGATTGTTGGTCTGGCGAATCCC   13201 TGGATCTCCCTCCGGGCGTCGCGAAATTTAAACTTGGCGGCGGCCACGGC
                               |||.|.||||||||||||||.||.||||||||.|||         |||||||.|||||.||||||||.|||||||||.|.|||||.|||||.||.
e.coli               1               gtgacgattaaattgattgtcggcctggcgaacccc     287 tggatctgcctcctggcgtcgccaaatttaaattgggcggtggccatggt

AE008779.1 12951 GGTGCGGAATATGCCGCGACGCGACACAATGCAGGCGCATGGTACGTCGA   13251 GGCCACAATGGTCTGAAAGACATCATCAGCAAGCTGGGCAATAATCCCAA
                 |||||.|||||.|||||.||||||||.|||||.||.||.||||.|||.||         ||.||||||||.|||||||||||||||||.||..||||.|||||.||.||
e.coli          37 ggtgctgaatacgccgcaacgcgacataatgctggtgcctggttcgttga     337 ggtcacaatggactgaaagacatcatcagtaaattgggtaataaccctaa

AE008779.1 13001 TTTACTGGCGGAGCGCCTGCGCGCGCCGTTGCGTGAAGAGCCTAAATTCT   13301 CTTTCACCGATTACGCGTTGGAATTGGTCATCCAGGCGATAAAAATAAAG
                 .||||||||.|||||..|||||||.|||.||||.||||||.|||||||||         |||||||||.||||||.|.|||||.||||||||.||||||||||||||||
e.coli          87 cttactggcagagcgtttgcgcgctccgctgcgcgaagaggctaaattct     387 ctttcaccgtttacgcatcggaatcggtcatccgggcgataaaaataaag

AE008779.1 13051 TTGGCTATACCTCACGCATCACGCTGGAAGGGGAAGATGTTCGCCTGCTG   13351 TTGTTGGTTTCGTGCTGGGTAAACCCCCTGTTTCTGAACAAAAATTAATT
                 ||||.|||||.||.||..||||.||.|.|||.||||||||.||||||.|.         ||||.|||||.|||.|.||.|||||.||||||..||||||.||.||||||
e.coli         137 ttggttatacttcgcgagtcactcttggaggcgaagatgtccgcctgtta     437 ttgtcggttttgtgttaggcaaaccgcctgttagtgaacagaagttaatt

AE008779.1 13101 GTACCCACCACGTTCATGAACCTCAGTGGTAAAGCAGTTGGCGCAATGGC   13401 GATGAGGCCATTGACGAAGCGGCACGCTGTACGGAATTGTGGTTCAAAGA
                 ||.||.||.||.||.|||||.|||||.||.||||.||||..||.|||||         |||||.|||||||||||||||||.||.|||||.|||.|||||||.|.|||
e.coli         187 gtcccgactacatttatgaatctcagcggcaaagccgttgcggcgatggc     487 gatgaagccattgacgaagcggcgcgttgtactgaaatgtggtttacaga

AE008779.1 13151 CAGTTTTTACCGTATTCAGCCGGACGAAATTTTGGTCGCTCACGACGAGC   13451 GGGTCTGGCCAAAGCAACAAGCCGTTTGCATACCTTTAAGGCGCAATAAC
                 ||||||||.|||.|||.|.||||||||||||.||||.||.||||||||.|         .||..||.||||||||||.|.|||.|||||..|||||||.|||||||||
e.coli         237 cagtttttccgcattaatccggacgaaattctggtggcccacgacgaac     537 tggcttgaccaaagcaacgaaccgattgcacgcctttaaagcgcaataa
```

Example local alignment

- Example: 2 dissimilar sequences with a conserved domain

```
>HEN1_HUMAN
MMLNSDTMELDLPPTHSETESGFSDCGGGAGPDGAGPGGPGGGQARGPEPGEPGRKDLQHLSREERRRRR
RATAKYRTAHATRERIRVEAFNLAFAELRKLLPTLPPDKKLSKIEILRLAICYISYLNHVLDV
>HES5_MOUSE
MAPSTVAVEMLSPKEKNRLRKPVVEKMRRDRINSSIEQLKLLLEQEFARHQPNSKLEKADILEMAVSYLK
HSKAFAAAAGPKSLHQDYSEGYSWCLQEAVQFLTLHAASDTQMKLLYHFQRPPAPAAPAKEPPAPGAAPQ
PARSSAKAAAAAVSTSRQPACGLWRPW
```

```
             110       120
HEN1_H PDKKLSKIEILRLAICYISY
       :..::.: .::..:. :...
HES5_M PNSKLEKADILEMAVSYLKH
               60        70
```



Pairwise alignment

- **Data**:
  - A pair of sequences (DNA / protein)
  - A scoring system: how to count what is similar?

- **goal**:
  - To determine the degree of similarity (best score)
  - Show similarity (better alignment)

- **Describes** the **resemblance** through **3 operations** (point mutations)
  - Insertion
  - Deletion
  - Identity/substitution (match/mismatch)

- **Measures similarity by giving weight** to each operation
  - Positive weight ("reward") to the good parts of the alignment (matching of two identical or close letters)
  - Negative (or zero) weight ("penalty") associated with bad (matching of two unrelated letters, mismatch)

Scoring system

- Score (or weight) for an **identity/substitution**
  - **Substitution matrix** → See dedicated course
  - s(a, b) = alignment score of nucleotides a and b

$$\begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix} \longrightarrow$$

	A	**C**	**G**	**T**
A	2	0	0	0
C	0	2	0	0
G	0	0	2	0
T	0	0	0	2

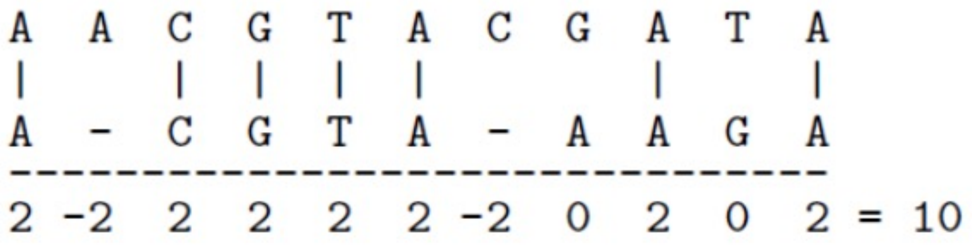
More complicated for proteins !

- Score (or weight) of an **indel** (insertion/deletion) : -2 for example per indel

Alignment score

- **Alignment score** = sum of the scores of the elementary events

- For **example** :



```
A  A  C  G  T  A  C  G  A  T  A
|     |  |  |  |        |     |
A  -  C  G  T  A  -  A  A  G  A
-------------------------------
2 -2  2  2  2  2 -2  0  2  0  2 = 10
```

Why we need a smart algorithm

- 2 sequences of **length n**: **max. length of the alignment 2n**
- Example with the sequences **TA** and **CA**
- **Naive algorithm**: **enumerate all the alignments**
  (match=+1 mismatch=-1 indel = -1)
  i= number of overlapping nucleotides

	TA--			
i=0	--CA -4			
	TA-	TA-	TA-	T-A
i=1	-CA -3	-CA -3	C-A -3	-CA -1
	TA			
i=2	CA 0			

Why we need a smart algorithm

- **Maximum number of alignments** (sequences of length n)

$$\frac{(2n)!}{(n)!^2}$$ → For 2 sequences of length 100: 2.10^{57} alignments

- Using **dynamic programming**: matrix representation, **complexity n²**
  → For 2 sequences of length 100: 10 000 calculations

	T	A
C		
A		

Dynamic programming

- The concept was developed by **Richard Bellman** in the 1950s
- Aims to **simplify a complicated problem** by **breaking it** down into **simpler sub-problems** in a **recursive** way
- See “**Pyramide de nombres**”: https://fr.wikipedia.org/wiki/Programmation_dynamique#opc
- Widely used in **bioinformatics**: sequence alignment, protein folding, protein binding, nucleic acid structures…

Global alignment

- Algorithm by **Needleman & Wunsch**, **1970**
- For **nucleic acids** or **proteins**
- **Dynamic programming**
- **Optimal** alignment

Needleman Wunsch algorithm

- **Sequences**
  - Sequence A: `ATT`
  - Sequence B: `TTC`

- Define a **scoring table**:
  - match: +1 $\rightarrow$ $s(a,b) = +1 \text{ if } a = b$
  - mismatch: -1 $\rightarrow$ $s(a,b) = -1 \text{ if } a \neq b$
  - indel: -1 $\rightarrow$ $s(a,-) = s(-,b) = -1$

- Build **scoring matrix** by following these rules
  - Matrix initialization : 0
  - $S_{i,J} = \text{MAX} \begin{bmatrix} S_{i-1,J-1}+s(a,b) \\ S_{i-1,J}+s(a,-) \\ S_{i,J-1}+s(-,b) \end{bmatrix} = \begin{bmatrix} S(\nwarrow)+\text{match or mismatch} \\ S(\leftarrow)+\text{indel} \\ S(\uparrow)+\text{indel} \end{bmatrix}$

Needleman Wunsch algorithm

	-	A	T	T
-				
T				
T				
C				

match: +1
mismatch: -1
indel: -1

$$S_{i,J} = \text{MAX} \begin{array}{l} S(\nwarrow)\text{+match/mis.} \\ S(\leftarrow)\text{+indel} \\ S(\uparrow)\text{+indel} \end{array}$$

Needleman Wunsch algorithm

	-	A	T	T
-	0	→ -1	→ -2	→ -3
T	↓ -1			
T	↓ -2			
C	↓ -3			

match: +1
mismatch: -1
indel: -1

$$S_{i,j} = \text{MAX} \begin{array}{l} S(\nwarrow)+\text{match/mis.} \\ S(\leftarrow)+\text{indel} \\ S(\uparrow)+\text{indel} \end{array}$$

Université de Lille

Needleman Wunsch algorithm

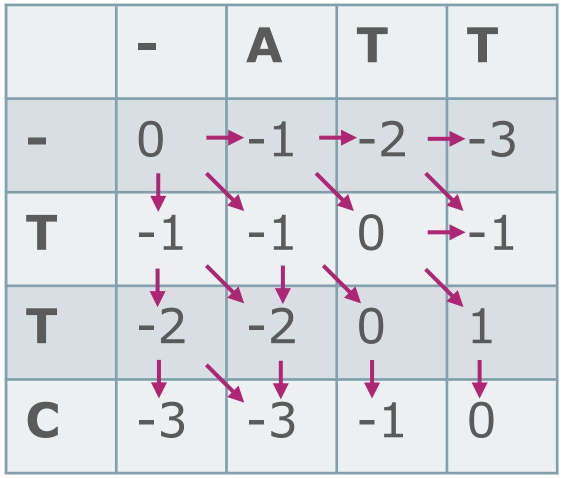
	-	**A**	**T**	**T**
-	0	-1	-2	-3
T	-1	-1		
T	-2			
C	-3			

match: +1
mismatch: -1
indel: -1

$$S_{i,j} = \mathrm{MAX} \begin{array}{l} S(\nwarrow)+\text{match/mis.} \\ S(\leftarrow)+\text{indel} \\ S(\uparrow)+\text{indel} \end{array}$$

MAX (0-1;-1-1;-1-1)=MAX (-1;-2;-2)=-1

Needleman Wunsch algorithm

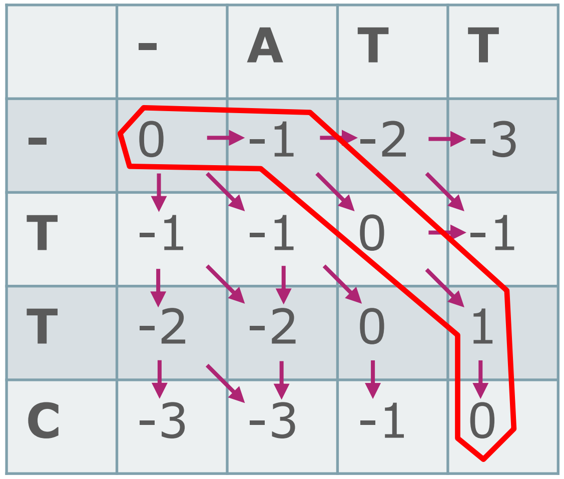


	-	A	T	T
-	0	-1	-2	-3
T	-1	-1	0	-1
T	-2	-2	0	1
C	-3	-3	-1	0

match: +1
mismatch: -1
indel: -1

$$S_{i,j} = \text{MAX} \begin{array}{l} S(\nwarrow)+\text{match/mis.} \\ S(\leftarrow)+\text{indel} \\ S(\uparrow)+\text{indel} \end{array}$$

Needleman Wunsch algorithm



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

. Backtracking from the lower-right corner

. Reconstructing alignment

```
ATT-
-TTC
```

Needleman Wunsch algorithm

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

Other example from https://www.youtube.com/watch?v=BYdTqq8AGgc

Sequence A: `ATCGGAG`
Sequence B: `ATGGCAA`

match: +1
mismatch: -1
indel: -1

$$S_{i,j} = \text{MAX} \begin{array}{l} S(\nwarrow)\text{+match/mis.} \\ S(\leftarrow)\text{+indel} \\ S(\uparrow)\text{+indel} \end{array}$$

Université de Lille

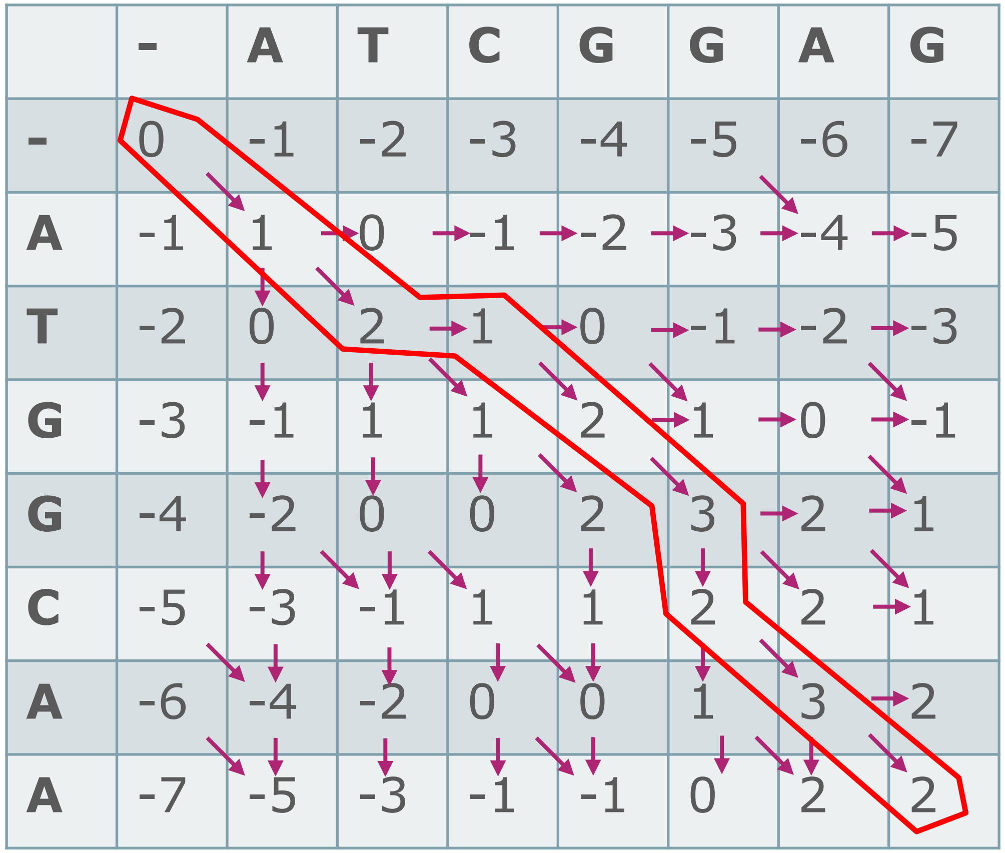
Needleman Wunsch algorithm

	-	A	T	C	G	G	A	G
-	0	-1	-2	-3	-4	-5	-6	-7
A	-1	↘1	→0	→-1	→-2	→-3	↘→-4	→-5
T	-2	↓0	↘2	→1	→0	→-1	→-2	→-3
G	-3	↓-1	↓1	↘1	↘2	↘→1	→0	↘→-1
G	-4	↓-2	↓0	↓0	↘2	↘3	→2	↘→1
C	-5	↓-3	↘↓-1	↘1	↓1	↓2	↘2	↘→1
A	-6	↘↓-4	↓-2	↓0	↘↓0	↓1	↘3	→2
A	-7	↘↓-5	↓-3	↓-1	↘↓-1	↓0	↘↓2	↘2

match: +1
mismatch: -1
indel: -1

$$S_{i,j} = \text{MAX} \begin{array}{l} S(\nwarrow)+\text{match/mis.} \\ S(\leftarrow)+\text{indel} \\ S(\uparrow)+\text{indel} \end{array}$$

Needleman Wunsch algorithm



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

. Backtracking from the lower-right corner

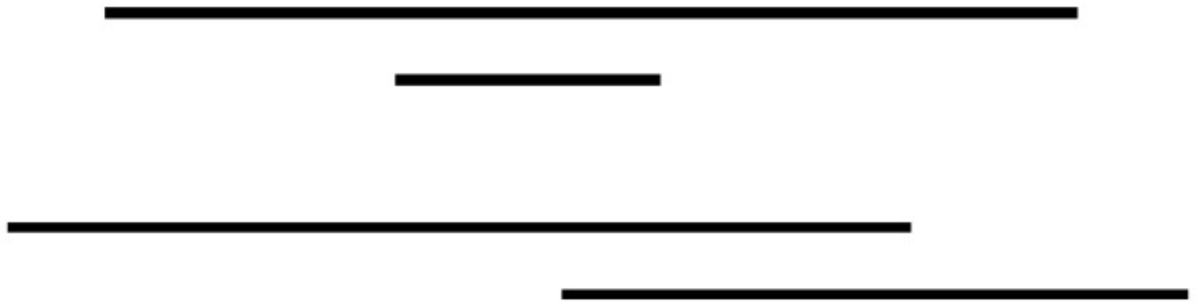
. Reconstructing alignment

```
ATCGG-AG
AT-GGCAA
```

Semi-global alignment

- A variant of global alignment that allows **for gaps at the beginning and/or the end** of one of the sequences

- Useful when you align a **short against a long** sequence or when the **downstream** part of one sequence **overlaps with** the **upstream** part of the other sequence



Local alignment

- Algorithm by **Smith & Waterman**, **1981**
- For **nucleic acids** or **proteins**
- Look for **subsequence matches**
- Modification of the **Needleman Wunsch** algorithm : negative scores are set to zero
- **Dynamic programming**
- **Optimal** alignment

Smith Waterman algorithm

- **Sequences**
  - Sequence A: `ACCGTGA`
  - Sequence B: `GTGAATA`

  Example from https://www.youtube.com/watch?v=BYdTqq8AGgc

- Define a **scoring table**:
  - match: +1 → $s(a,b) = +1 \text{ if } a = b$
  - mismatch: -1 → $s(a,b) = -1 \text{ if } a \neq b$
  - indel: -1 → $s(a,-) = s(-,b) = -1$

- Build **scoring matrix** by following these rules
  - Matrix initialization : 0; **No negative scores !**
  - $$S_{i,J} = \text{MAX} \begin{bmatrix} S_{i-1,J-1}+s(a,b) \\ S_{i-1,J}+s(a,-) \\ S_{i,J-1}+s(-,b) \\ 0 \end{bmatrix} = \begin{bmatrix} S(\nwarrow)+\text{match or mismatch} \\ S(\leftarrow)+\text{indel} \\ S(\uparrow)+\text{indel} \\ 0 \end{bmatrix}$$

Université de Lille

Smith Waterman algorithm

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

match: +1
mismatch: -1
indel: -1

$$S_{i,J} = \text{MAX} \begin{array}{l} S(\nwarrow)+\text{match/mis.} \\ S(\leftarrow)+\text{indel} \\ S(\uparrow)+\text{indel} \\ 0 \end{array}$$

Smith Waterman algorithm

	-	A	C	C	G	T	G	A
-	0	0	0	0	0	0	0	0
G	0	0	0	0	↘ 1	0	↘ 1	0
T	0	0	0	0	0	↘ 2	↘ 1	0
G	0	0	0	0	↘ 1	↘ 1	↘ 3	↘ 2
A	0	↘ 1	0	0	0	0	↘ 2	↘ 4
A	0	1	0	0	0	0	↓ 1	↘↓ 3
T	0	0	0	0	0	↘ 1	0	↓ 2
A	0	1	0	0	0	0	0	↓ 1

match: +1
mismatch: -1
indel: -1

$$S_{i,j} = \text{MAX} \begin{array}{l} S(\nwarrow)+\text{match/mis.} \\ S(\leftarrow)+\text{indel} \\ S(\uparrow)+\text{indel} \\ 0 \end{array}$$

Smith Waterman algorithm

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

. Backtracking from max score in the matrix until zero

. Reconstructing local alignment

```
GTGA
GTGA
```

Université de Lille

Gaps penalties

- The **different algorithms** use **different functions** to calculate gap penalties

- The simplest function: **linear** function: `g × l`
  - `g`: indel penalty
  - `l`: length of the gap

- More realistic functions
  - **Affine** functions: `o + e × l`
    → `o`: gap opening penalty
    → `e`: gap extension penalty

  - **Logarithmic** function

Influence of the scoring system, exercise

	A	B	C
Match Cost	1	1	1
Mismatch Cost	-1	-1	-1
Gap Open Penalty (o)	0	1	4
Gap Extension Penalty (e)	0	0.1	0.1

AL1 :

```
AT-GCGGGACA-TG
|  |||   |  ||
A-GGCG---C-CTG
```

7 matches, 0 mismatches, 5 ogaps, 2 egaps

AL2 :

```
ATGCGGGACATG
|.|||   |.||
AGGCG---CCTG
```

7 matches, 2 mismatches, 1 ogap, 2 egaps

AL3 :

```
ATGCGGGACATG
   .||..|.||
---AGGCGCCTG
```

5 matches, 4 mismatches, 1 ogap, 2 egaps

For each scoring system (A, B or C), determine which alignment (AL1, AL2 or AL3) will be returned

Université de Lille

Choose the right gap penalties

- Little *a priori* knowledge
- Data specificity
- **Typical values** for an affine gap function

```
0.5 < o < 5.0
0.05 < e < 1.0
```

- Always take (in absolute value) `o > 1/2` substitution

Assess the quality of an alignment

```
A C C T G A C G T A A G C
| | | | | | | | | | | | |
A C C T G A C G T A A G C
```

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

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

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

Assess the quality of an alignment

- **Robustness of the score when changing the parameters**
  - Alignment is doubtful if small changes (about 10%) in penalties of insertion/deletion significantly change this alignment

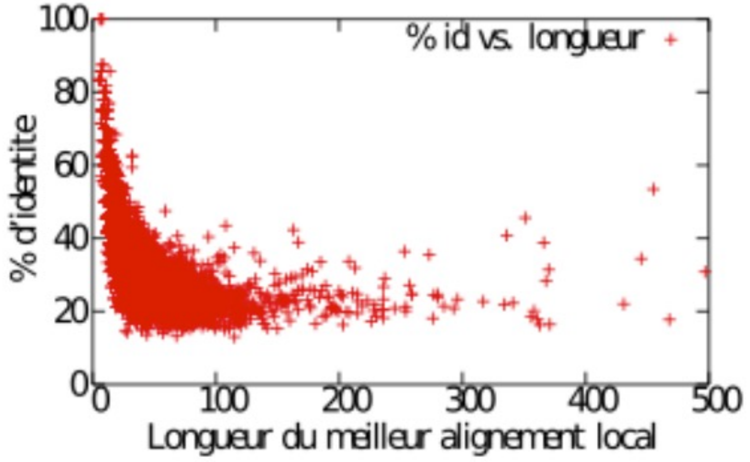
- **Frequency of gaps**
  - Alignment is doubtful if it requires more than one insertion on average for 20 amino acids

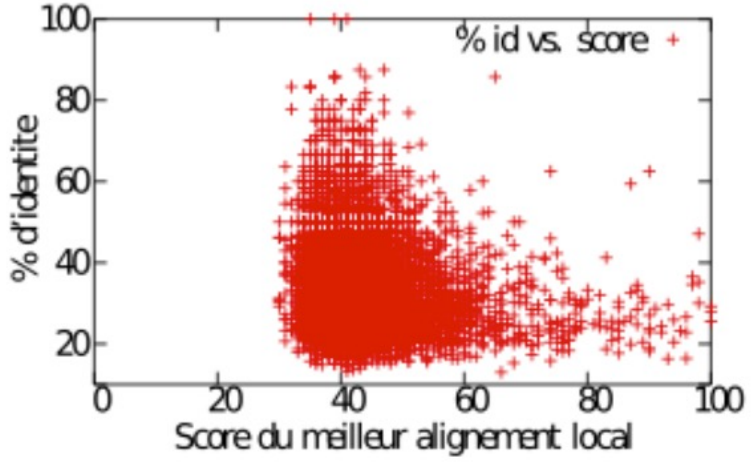
- Two nucleic sequences of at least 100 bases and 50% identities do not necessarily have a biological relationship

- Protein sequences of 100 or more residues, with at least 25% identity have certainly a common ancestor (Doolittle, 1990 - PDB).

Use % of identity?

- Depends on the **composition** of bases or amino acids
- Depends on the **length** of the sequences





Not a good idea!

Empirical approach

- **Score robustness test**
  - Two sequences : `u` and `v`
  - `S`: score of the alignment between `U` and `V`

- **Method**:
  - 1. Generation of 100 (or more) permutations of `V` (same length, same composition)
  - 2. Alignment with `U` $\rightarrow$ score calculation
  - 3. Distribution of alignment scores
  - 4. Where does `S` fit into this distribution?

Statistical approach

- **E-value**: number of times one expects to find a score alignment greater than `S` by chance when a sequence of length `n` is aligned with a length sequence `m`

- Describes the random noise that exists when aligning sequences increases proportionally with `n` and `m`
- Decreases exponentially as a function of the score `S`
- The closer the E-value is to 0, the more the similarity is significant

Université de Lille

Example

Human alpha haemoglobin (141 aa) vs. Human myoglobin (153 aa)

```
VLSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF-DLS-----HGSAQVKGHGKKVADALTNAVAHVDDMPNALSAL
 :: ..   : ..::::.:.  ..:.:.: :.: . :.: . : .: .:.     ..:...: :: .: .::.. . . ..  .....:
GLSDGEWQLVLNVWGKVEADIPGHGQEVLIRLFKGHPETLEKFDKFKHLKSEDEMKASEDLKKHGATVLTALGGILKKKGHHEAEIKPL

SDLHAHKLRVDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSKYR------
.. :: : ..    ....:.:.. .:... :..:.........: :.  .. ..:.:.
AQSHATKHKIPVKYLEFISECIIQVLQSKHPGDFGADAQGAMNKALELFRKDMASNYKELGFQG
```

Chicken lysozyme (129 aa) vs. Bovine ribonuclease (124 aa)

```
KVFGRCELAAAMKRHGLDNYRGYSLGNWVCAAKFESNFNTQATNRNTDGSTDYGILQINSRWWCNDGRTP--GSRNLCNIPCSALLSSD
:  .    :: ..:. .:.  . . ..  :.....:.  :..  . ... .. .. ....   :.. ...  .... :  ..:..  .:
KETA----AAKFERQHMDSSTSAASSSNYCNQMMKSRNLTKDRCKPVNTFVHESLADVQAV--CSQKNVACKNGQTNCYQSYSTMSITD

ITASVNCAKKIVSDGDGMNAWVAWRNRCKGTDVQAWIRGCRL
  .. ...:   .. .. .:       :.:.   .   . ..
CRET-GSSKYPNCAYKTTQANKHIIVACEGNPYVPVHFDASV
```

Université de Lille

Use of PRSS

PRSS : Probability of Random Shuffle Sequence

```
< 20     0      0:
  22     0      0:          one = represents 1 library sequences
  24     0      0:
  26     0      0:
  28     0      0:
  30     1      1:*
  32     3      3:==*
  34     9      7:======*==
  36    25     15:==============*==========
  38    37     25:========================*============
  40    29     34:=============================    *
  42    33     42:=================================        *
  44    51     46:=============================================*=====
  46    41     47:=========================================     *
  48    32     45:================================            *
  50    51     41:========================================*==========
  52    31     36:===============================    *
  54    24     31:========================      *
  56    30     26:=========================*====
  58    18     21:==================  *
  60    24     17:================*=======
  62    19     14:=============*=====
  64     4     11:====      *
  66    10      9:========*=
  68     5      7:===== *
  70     4      5:====*
  72     7      4:===*===
  74     3      3:==*
  76     2      3:==*
  78     3      2:=*=
  80     1      2:=*
  82     0      1:*
  84     1      1:*
  86     0      1:*
  88     1      1:*
  90     0      0:      unshuffled s-w score: 177
  92     1      0:=    For 500 sequences, a score >= 177 is expected 3.096e-06 times
  94     0      0:
```

Sylvain Legrand
Maître de Conférences
UMR CNRS 8198 EVO-ECO-PALEO
Evolution, Ecologie et Paléontologie
Université de Lille - Faculté des Sciences et Technologies
Bât SN2, bureau 208 - 59655 Villeneuve d'Ascq

sylvain.legrand@univ-lille.fr | http://eep.univ-lille.fr/
Tél. +33 (0)3 20 43 40 16