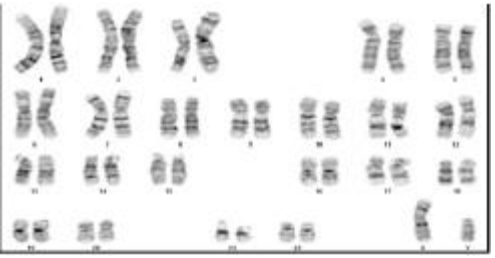
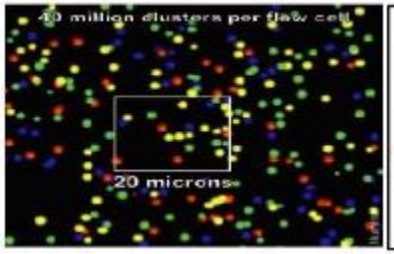
Bioinformatics: Introduction and Methods

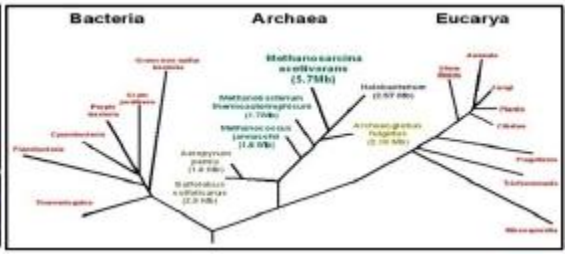
Kun Lu

Computer Science Department, Southwest University



TAACCCTAACCCTAACCCTAACCCTAACCCTA
CCTAACCCTAACCCTAACCCTAACCCTAACCC
CCCTAACCCCTAACCCTAACCCTAACCCTAAC
AACCCTAACCCTAACCCTAACCCCTAACCCTA
ACCCTAACCCCAACCCCAACCCCAACCCCAAC
CTACCCTAACCCTAACCCTAACCCTAACCCTA
ACCCTAACCCTAACCCTAACCCCTAACCCTAA



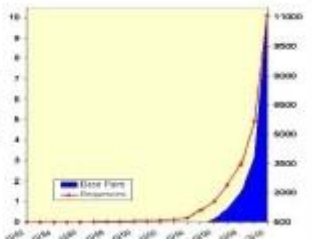


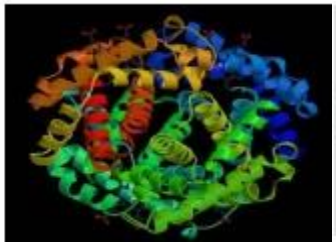
Sequence Alignment

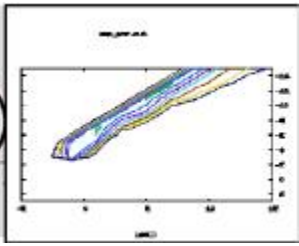
Kun Lu

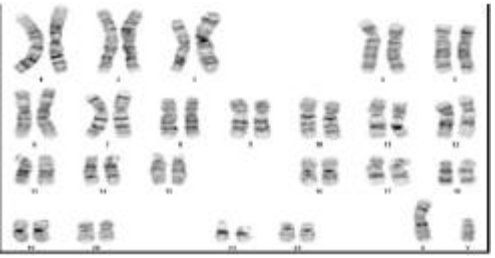
Computer Science Department

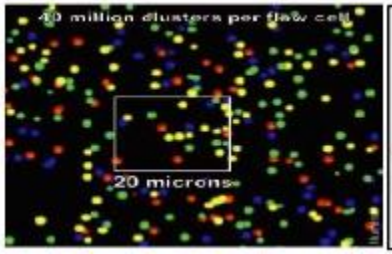
Southwest University

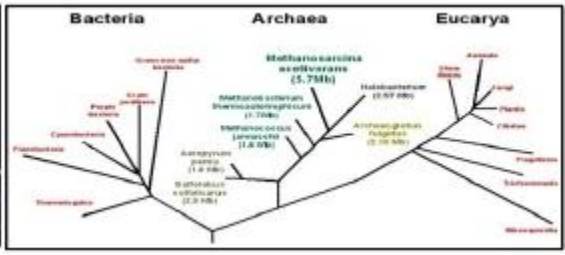










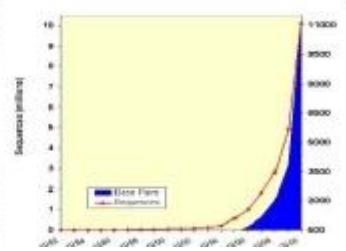


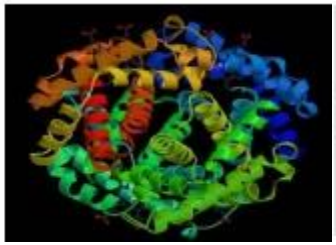
Unit 1: Essential Concepts

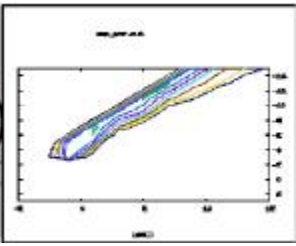
Kun Lu

Computer Science Department

Southwest University

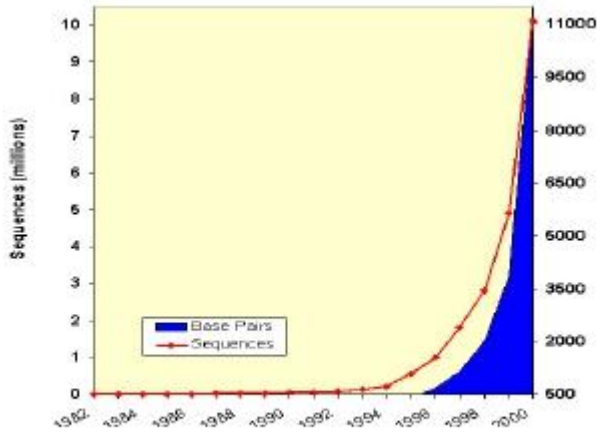




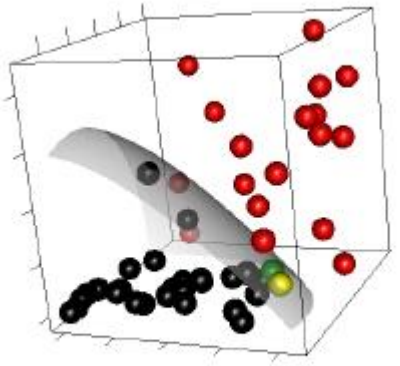


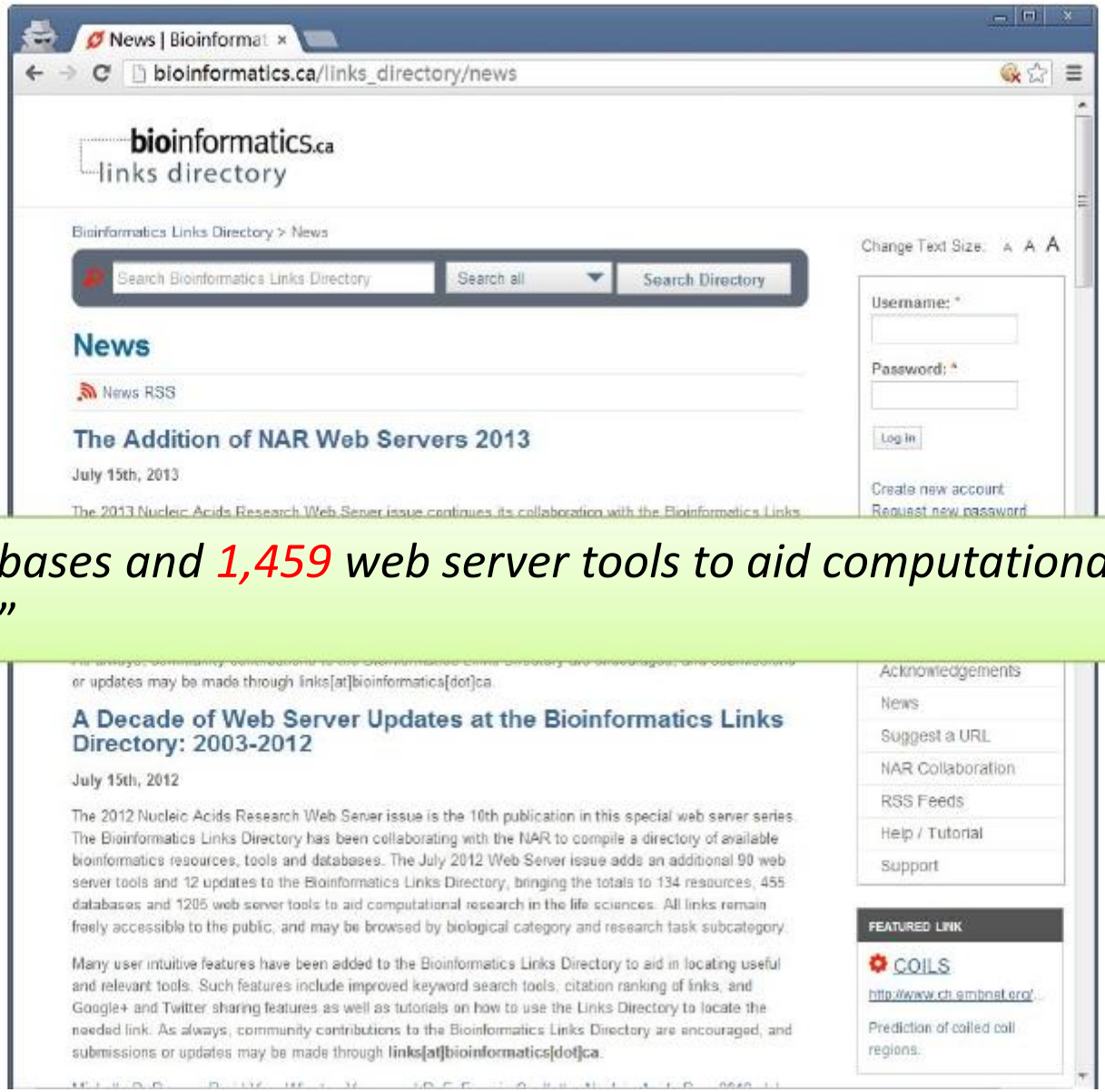
Opportunities and challenges hand-in-hand: the driving forces of bioinformatics

- High-throughput data
  - huge amount
  - explosive growth
  - noisy
  - multi-type
  - multi-scale
  - Heterogeneous



- Requirements for the methods
  - Data needs to be stored in efficient ontology-based database systems
  - The huge amount of data requires efficient methods
  - Exponential growth requires scalable methods
  - The low signal-to-noise ratio requires accurate methods
  - Multiple types of data requires data integrative methods

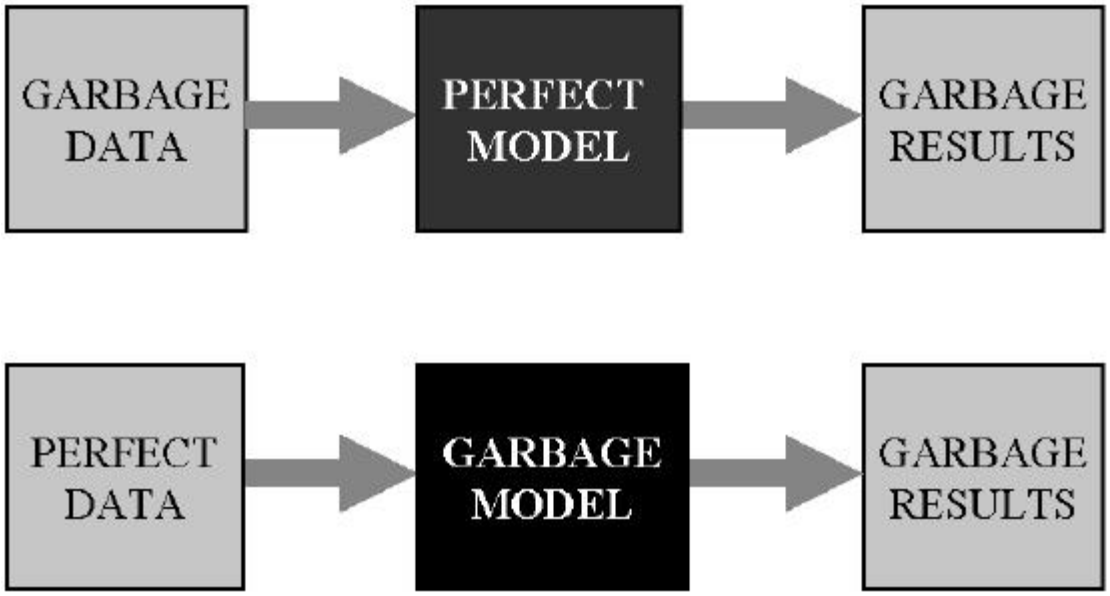




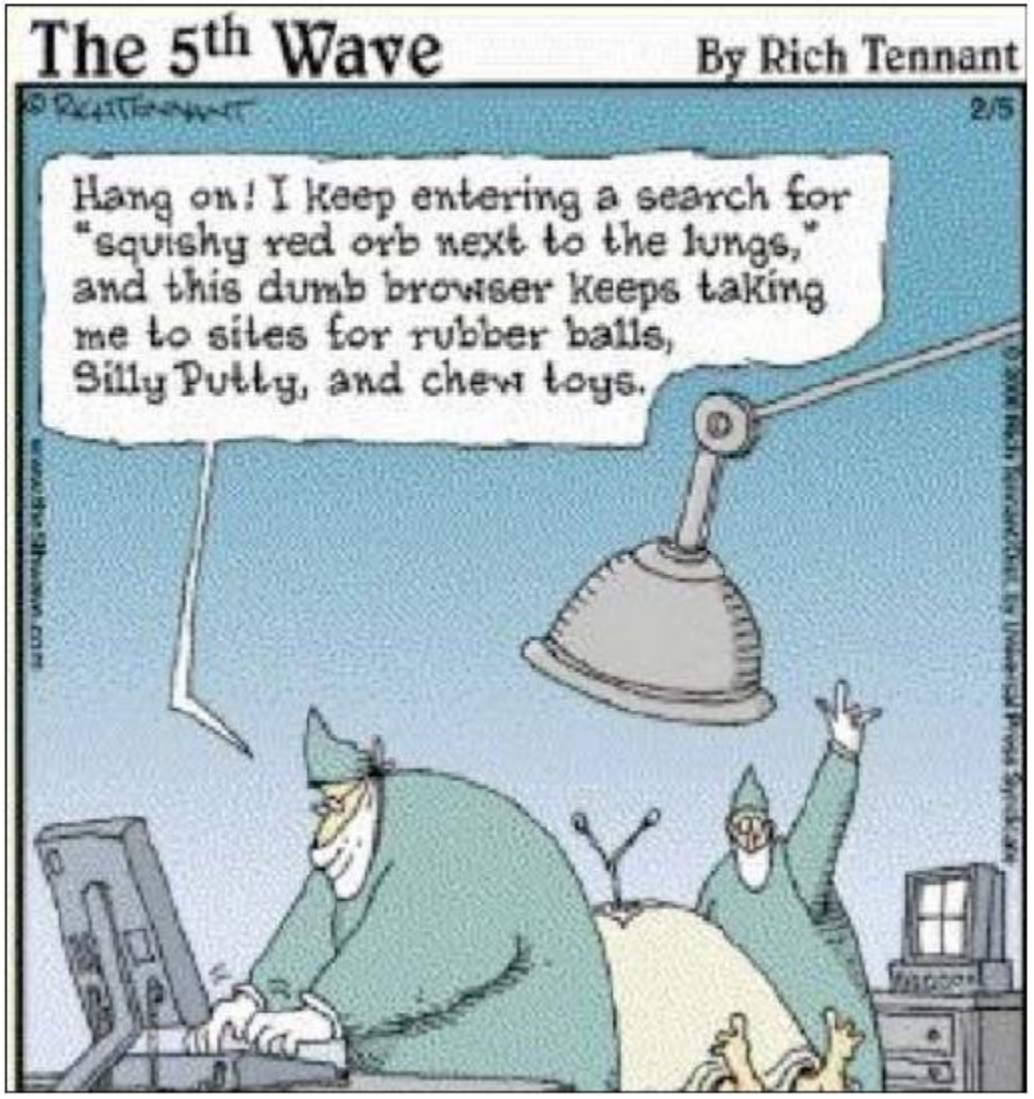
"…620 databases and 1,459 web server tools to aid computational research in the life sciences"

MODEL CALCULATIONS

"Garbage In-garbage Out" Paradigm



(Source: http://blog.potterzot.com/wp-content/uploads/2007/09/garbage_paradigm.gif)



(Source: http://performancemarketingassociation.com/new-working-group-data-feed-standard)

SCIENTIFIC PUBLISHING

A Scientist's Nightmare: Software Problem Leads to Five Retractions

Until recently, Geoffrey Chang's career was o
a trajectory most young scientists only drear
about. In 1999, at the age of 28, the protei
crystallographer landed a faculty position a
the prestigious Scripps Research Institute i
San Diego, California. The next year, in a ce
emony at the White House, Chang received
Presidential Early Career Award
for Scientists and Engineers, the

Source: *Science* 314(5807):1856-1857, 2006

Box 1

The good, the bad and the ugly

The good

In 1995, Fleischman *et al.* [34] were the first to succe
bacterium *Haemophilus influenzae* Rd. The group iden
represent genes. They translated the coding regions i
sequences in a protein database, identifying 1,007 clo
extensive annotation on the function of the entries, all
functions of most of the putative genes.

The bad

In 1997, the discovery of a new plant adenylyl cyclas
plants were not believed to have adenylyl cyclases. Th
for plants. The 'homology' (sequence similarity) they showed was not so weak: there was definitely some similarity, and the homology had a high 'score' (which by itself is not very meaningful) - but when their adenylyl cyclase was aligned to a profile for other known adenylyl cyclases, it was obvious to even first-year graduate students that the characteristics that are common to all other adenylyl cyclases were largely missing.

The ugly

The authors were later forced to retract their paper [36]. What might have saved them from public humiliation was a more careful analysis of their results.

Source: *Genome Biol 2*:reviews2002-review2002.10, 2001

(Source: http://cartoonmela.blogspot.com/2009_11_01_archive.html)

- **B**iology
  - What is the biological question or problem?
- **D**ata
  - What is the input data?
  - What other supportive data can be used?
- **M**odel
  - How is the problem formulated computationally?
  - Or, what's the data model?
- **A**lgorithm
  - What is the computational algorithm?
  - How about its performance/limitation?

Sequence Alignment

Biological Question:

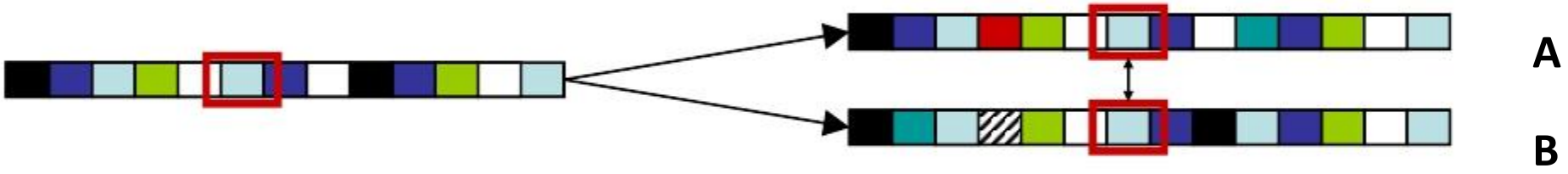
“How can we determine the similarity between two sequences?”

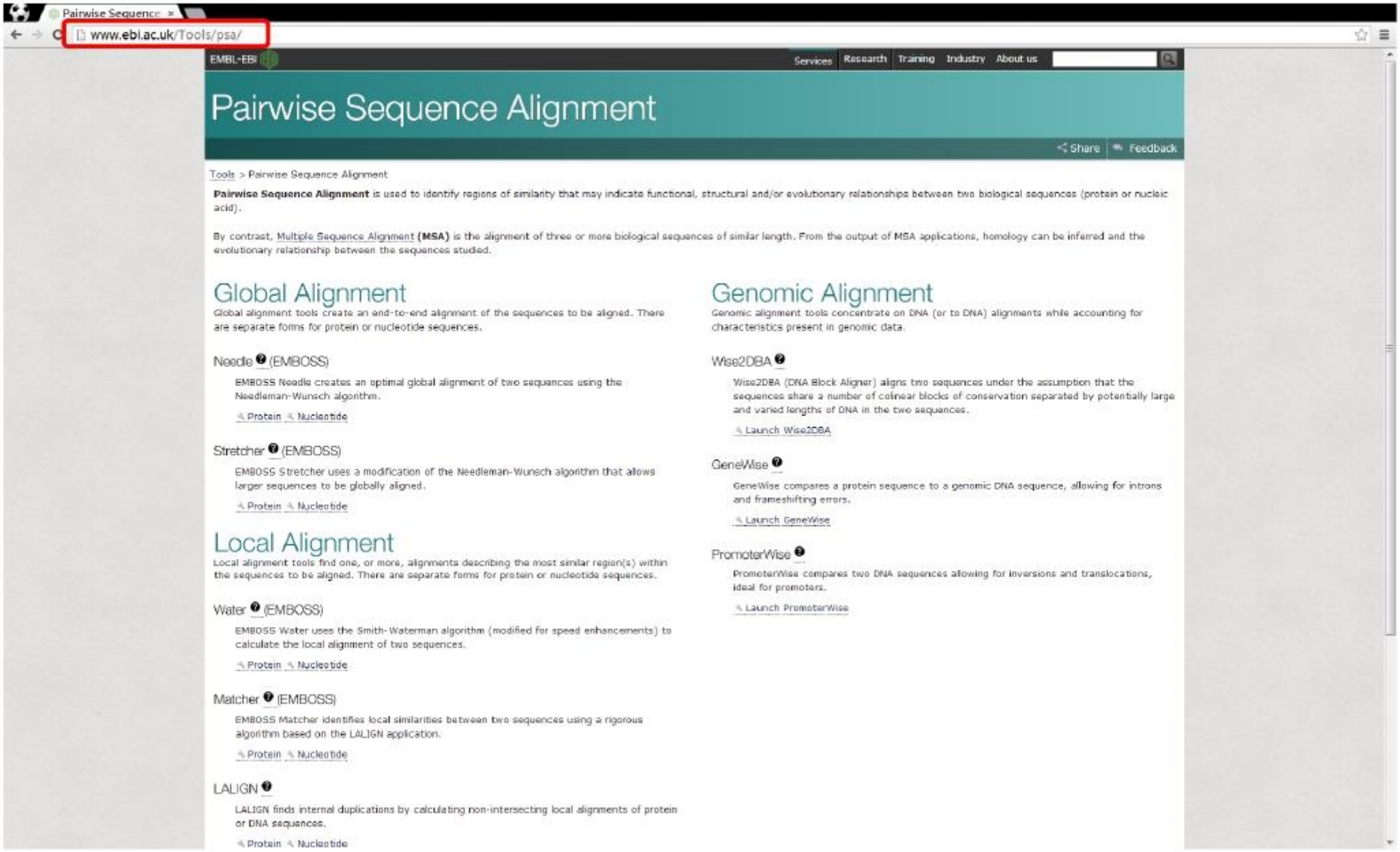
Why is it important?

- Similar sequence ➔ Similar structure ➔ Similar function (The “*Sequence-to-Structure-to-Function Paradigm*”)
- Similar sequence ➔ Common ancestor (“*Homology*”)

Sequence Alignment in Biology

The purpose of a sequence alignment is to line up all residues in the inputted sequence(s) for maximal level of similarity, in the sense of their functional or evolutionary relationship.




Pairwise Sequence
www.ebi.ac.uk/Tools/psa/
EMBL-EBI
Services
Research
Training
Industry
About us
Pairwise Sequence Alignment
Share
Feedback
Tools > Pairwise Sequence Alignment
Pairwise Sequence Alignment is used to identify regions of similarity that may indicate functional, structural and/or evolutionary relationships between two biological sequences (protein or nucleic acid).
By contrast, Multiple Sequence Alignment (MSA) is the alignment of three or more biological sequences of similar length. From the output of MSA applications, homology can be inferred and the evolutionary relationship between the sequences studied.
Global Alignment
Global alignment tools create an end-to-end alignment of the sequences to be aligned. There are separate forms for protein or nucleotide sequences.
Needle (EMBOSS)
EMBOSS Needle creates an optimal global alignment of two sequences using the Needleman-Wunsch algorithm.
Protein Nucleotide
Stretcher (EMBOSS)
EMBOSS Stretcher uses a modification of the Needleman-Wunsch algorithm that allows larger sequences to be globally aligned.
Protein Nucleotide
Local Alignment
Local alignment tools find one, or more, alignments describing the most similar region(s) within the sequences to be aligned. There are separate forms for protein or nucleotide sequences.
Water (EMBOSS)
EMBOSS Water uses the Smith-Waterman algorithm (modified for speed enhancements) to calculate the local alignment of two sequences.
Protein Nucleotide
Matcher (EMBOSS)
EMBOSS Matcher identifies local similarities between two sequences using a rigorous algorithm based on the LALIGN application.
Protein Nucleotide
LALIGN
LALIGN finds internal duplications by calculating non-intersecting local alignments of protein or DNA sequences.
Protein Nucleotide
Genomic Alignment
Genomic alignment tools concentrate on DNA (or to DNA) alignments while accounting for characteristics present in genomic data.
Wise2DBA
Wise2DBA (DNA Block Aligner) aligns two sequences under the assumption that the sequences share a number of colinear blocks of conservation separated by potentially large and varied lengths of DNA in the two sequences.
Launch Wise2DBA
GeneWise
GeneWise compares a protein sequence to a genomic DNA sequence, allowing for introns and frameshifting errors.
Launch GeneWise
PromoterWise
PromoterWise compares two DNA sequences allowing for inversions and translocations, ideal for promoters.
Launch PromoterWise

Pairwise Sequence Alignment (PROTEIN)

EMBOSS Needle reads two input sequences and writes their optimal global sequence alignment to file.

This is the form for protein sequences. Please go to the nucleotide form if you wish to align DNA or RNA sequences.

STEP 1 - Enter your protein sequences

Enter or paste your first protein sequence in any supported format:

Or, upload a file: Choose File No file chosen

AND

Enter or paste your second protein sequence in any supported format:

Or, upload a file: Choose File No file chosen

STEP 2 - Set your pairwise alignment options

The default settings will fulfill the needs of most users and, for that reason, are not visible.

More options... *(Click here, if you want to view or change the default settings.)*

STEP 3 - Submit your job

☐ Be notified by email *(Tick this box if you want to be notified by email when the results are available)*

Submit

Pairwise Sequence Alignment (PROTEIN)

EMBOSS Needle reads two input sequences and writes their optimal global sequence alignment to file.

This is the form for protein sequences. Please go to the nucleotide form if you wish to align DNA or RNA sequences.

STEP 1 - Enter your protein sequences

Enter or paste your first **protein** sequence in any supported format:

```
>sp|P69905|HBA_HUMAN
MVLSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHFDLSHGSAQVK
GHGKKVADALTNAVAHVDDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHL
PAEFTPAVHASLDKFLASVSTVLTSKYR
```

Or, upload a file: Choose File No file chosen

AND

Enter or paste your second **protein** sequence in any supported format:

```
>sp|P68871|HBB_HUMAN
MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMG
NPKVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCV
LAHHFGKEFTPPVQAAYQKVVAGVANALAHKYH
```

Or, upload a file: Choose File No file chosen

STEP 2 - Set your pairwise alignment options

The default settings will fulfil the needs of most users and, for that reason, are not visible.

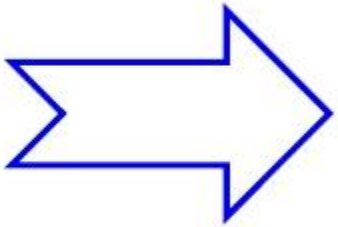
More options *(Click here, if you want to view or change the default settings.)*

STEP 3 - Submit your job

☐ Be notified by email *(Tick this box if you want to be notified by email when the results are available)*

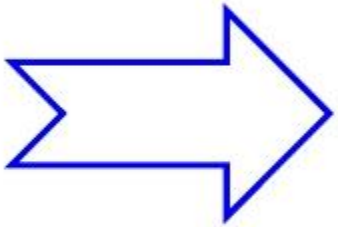
Submit

```
#=======================================
#
# Aligned_sequences: 2
# 1: HBA_HUMAN
# 2: HBB_HUMAN
# Matrix: EBLOSUM62
# Gap_penalty: 10.0
# Extend_penalty: 0.5
#
# Length: 149
# Identity:       65/149 (43.6%)
# Similarity:     90/149 (60.4%)
# Gaps:            9/149 ( 6.0%)
# Score: 292.5
#
#
#=======================================

HBA_HUMAN          1 MV-LSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF-D     48
                     ||  |:|.:|:.|.|.|||| .:..|.|.|||.|:.:.:|.|:.:|..|  |
HBB_HUMAN          1 MVHLTPEEKSAVTALWGKV--NVDEVGGEALGRLLVVYPWTQRFFESFGD     48

HBA_HUMAN         49 LS-----HGSAQVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKLR     93
                     ||     .|:.:||.|||||..|.:::.:||:|::...:.||:||..||.
HBB_HUMAN         49 LSTPDAVMGNPKVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLH     98

HBA_HUMAN         94 VDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSKYR    142
                     |||.||:||.:-|:..||.|...|||.|.|:..|.:|.|:..|..||.
HBB_HUMAN         99 VDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANALAHKYH    147
```



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

```
#=======================================
#
# Aligned_sequences: 2
# 1: HBA_HUMAN
# 2: HBB_HUMAN
# Matrix: EBLOSUM62
# Gap_penalty: 10.0
# Extend_penalty: 0.5
#
# Length: 149
# Identity:       65/149 (43.6%)
# Similarity:     90/149 (60.4%)
# Gaps:            9/149 ( 6.0%)
# Score: 292.5
#
#
#=======================================

HBA_HUMAN         1 MV-LSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF-D     48
                    ||  |:|.:|:.|.|.||||  :..|.|.|||.|:.:.:|.|:.:|..|  |
HBB_HUMAN         1 MVHLTPEEKSAVTALWGKV--NVDEVGGEALGRLLVVYPWTQRFFESFGD     48

HBA_HUMAN        49 LS-----HGSAQVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKLR     93
                    ||     .|:.:||.|||||..|.|:|:.:|||:|::...:.|||:||..||.
HBB_HUMAN        49 LSTPDAVMGNPKVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLH     98

HBA_HUMAN        94 VDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSKYR    142
                    |||.||:||.:.|:..||.|...||||.|.|:..|.:|.|:..|..||.
HBB_HUMAN        99 VDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANALAHKYH    147
```

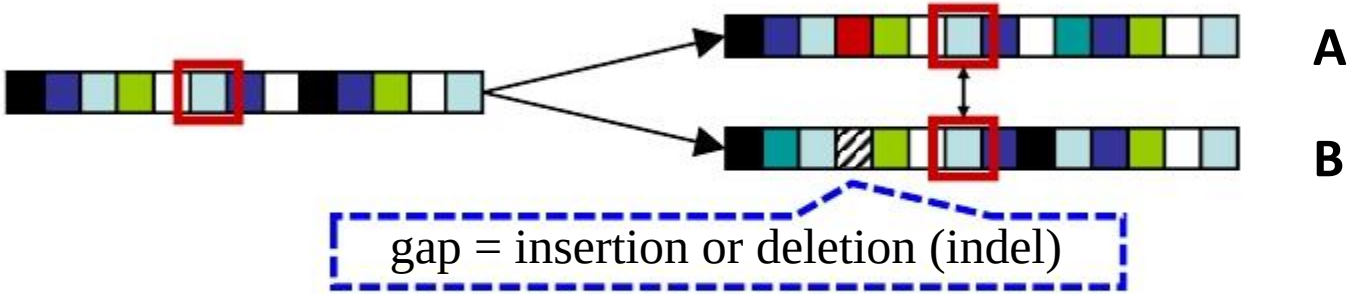


	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	
C	9																				C
S	-1	4																			S
T	-1	1	5																		T
P	-3	-1	-1	7																	P
A	0	1	0	-1	4																A
G	-3	0	-2	-2	0	6															G
N	-3	1	0	-2	-2	0	6														N
D	-3	0	-1	-1	-2	-1	1	6													D
E	-4	0	-1	-1	-1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

1. Symmetry

2. Context-insensitive

```
#=======================================
#
# Aligned_sequences: 2
# 1: HBA_HUMAN
# 2: HBB_HUMAN
# Matrix: EBLOSUM62
# Gap penalty: 10.0
# Extend penalty: 0.5
#
# Length: 149
# Identity:      65/149 (43.6%)
# Similarity:    90/149 (60.4%)
# Gaps:           9/149 ( 6.0%)
# Score: 292.5
#
#
#=======================================

HBA_HUMAN          1 MV-LSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF-D     48
                     ||  |:|.:|:.|.|.||||  :..|.|.|||.|:.:.:|.|:.:|..|  |
HBB_HUMAN          1 MVHLTPEEKSAVTALWGKV--NVDEVGGEALGRLLVVYPWTQRFFESFGD     48

HBA_HUMAN         49 LS-----HGSAQVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKLR     93
                     ||     .|:.:||.|||||..|.:::.:||:|::....:.||:||..||.
HBB_HUMAN         49 LSTPDAVMGNPKVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLH     98

HBA_HUMAN         94 VDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSKYR    142
                     |||.||:||.:-|:..||.|...|||.|.|:..|.:|.|:..|..||.
HBB_HUMAN         99 VDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANALAHKYH    147
```



Affine gap penalty: opening a gap receives a penalty of d; extending a gap receives a penalty of e. So the total Penalty for a gap with length n would be:

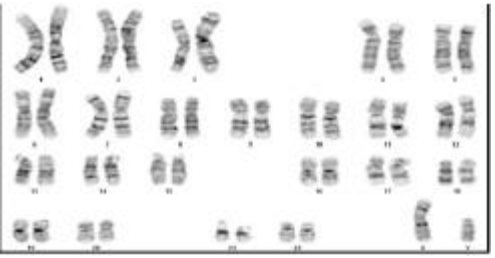
Penalty = d + (n-1)* e

```
#=======================================
#
# Aligned_sequences: 2
# 1: HBA_HUMAN
# 2: HBB_HUMAN
# Matrix: EBLOSUM62
# Gap_penalty: 10.0
# Extend_penalty: 0.5
#
# Length: 149
# Identity:       65/149 (43.6%)
# Similarity:     90/149 (60.4%)
# Gaps:            9/149 ( 6.0%)
# Score: 292.5
#
#
#=======================================

HBA_HUMAN         1 MV-LSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF-D     48
                    || |:|.:|:.|.|.||||  :..|.|.|||.|:.:.:|.|:.:|..| |
HBB_HUMAN         1 MVHLTPEEKSAVTALWGKV--NVDEVGGEALGRLLVVYPWTQRFFESFGD     48

HBA_HUMAN        49 LS-----HGSAQVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKLR     93
                    ||     .|:.:||.|||||..|.:::.:|||:|::....:.||:||..||.
HBB_HUMAN        49 LSTPDAVMGNPKVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLH     98

HBA_HUMAN        94 VDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSKYR    142
                    |||.||:||.:.|:..||.|...|||.|.|.|:..|.:|.|:..|..||.
HBB_HUMAN        99 VDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANALAHKYH    147
```

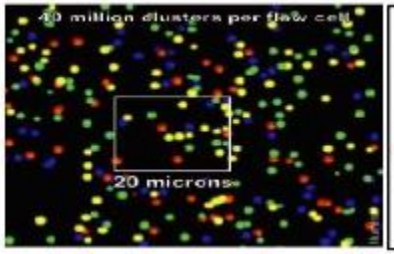
	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	
C	9																				C
S	-1	4																			S
T	-1	1	5																		T
P	-3	-1	-1	7																	P
A	0	1	0	-1	4																A
G	-3	0	-2	-2	0	6															G
N	-3	1	0	-2	-2	0	6														N
D	-3	0	-1	-1	-2	-1	1	6													D
E	-4	0	-1	-1	-1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

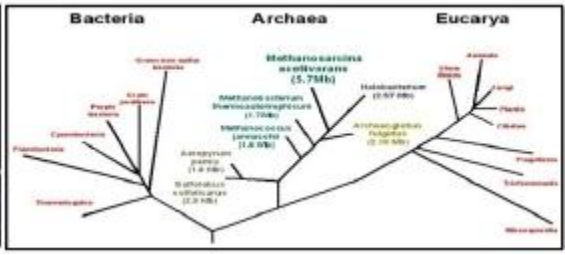
Affine gap penalty: opening a gap receives a penalty of d; extending a gap receives a penalty of e. So the total Penalty for a gap with length n would be:
Penalty = d + (n-1)* e

Final Score = (sum of substitution scores) + (-1) * (sum of Gap Penalty)



TAACCCTAACCCTAACCCTAACCCTAACCCTA
CCTAACCCTAACCCTAACCCTAACCCTAACCC
CCCTAACCCCTAACCCTAACCCTAACCCTAAC
AACCCTAACCCTAACCCTAACCCCTAACCCTA
ACCCTAACCCCAACCCCAACCCCAACCCCAAC
CTACCCTAACCCTAACCCTAACCCTAACCCTA
ACCCTAACCCTAACCCTAACCCCTAACCCTAA



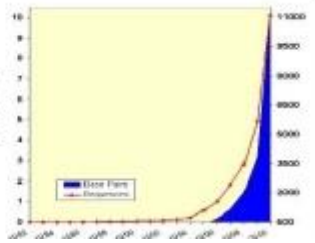


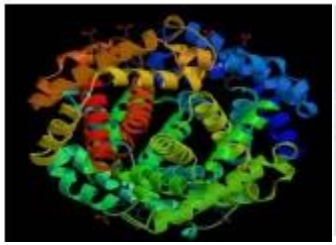
Unit 2: Global Alignment by Dynamic Programming

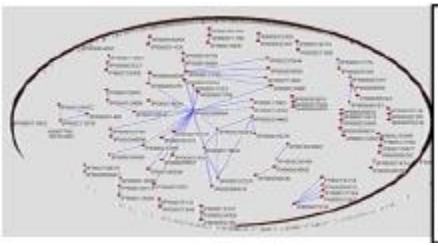
Le Zhang, Ph. D.

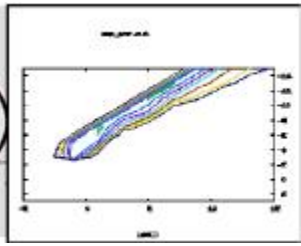
Computer Science Department

Southwest University









Pairwise Sequence Alignment: in Maths

- Input data:
  - Two sequences S1 and S2
- Parameter(s)
  - A scoring function f for
    - Substitutions
    - Gaps

- Output:
  - The optimal alignment of S1 and S2, which has the maximal score.

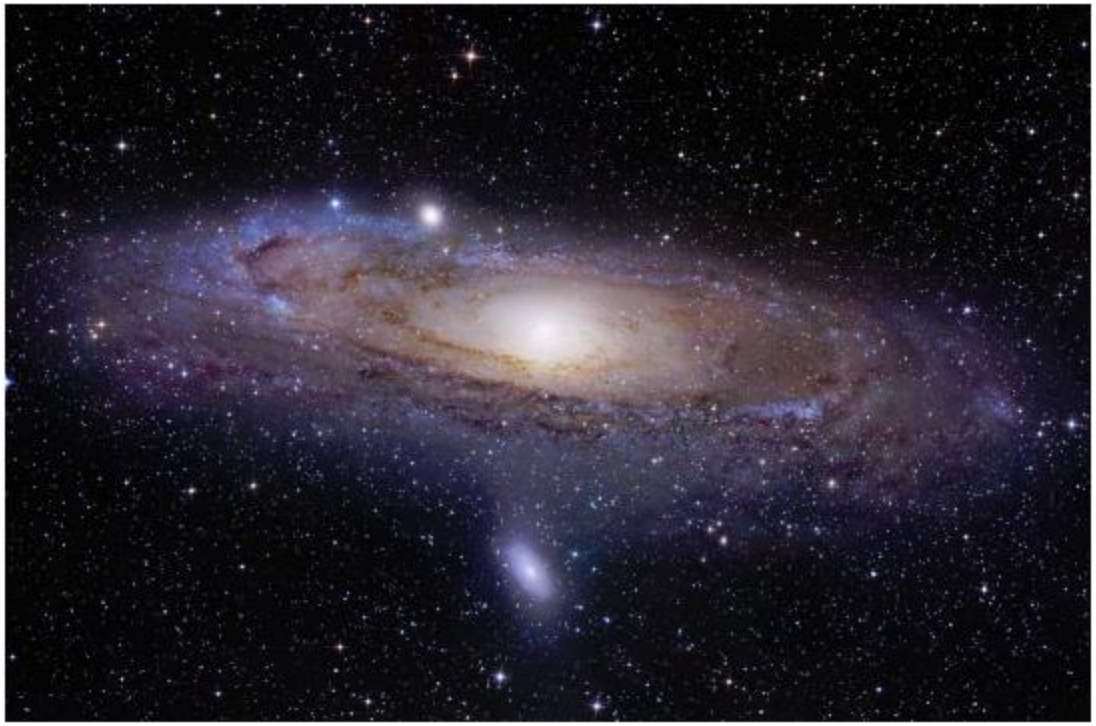
$$\underset{ali}{\text{argmax}}(f(ali(S1,S2))$$

Sequence Alignment: Enumerate?

LSPADK L-SPADK L-SPADK
LTPEEK LTPEEK- LT-PEEK
------LSPADK L-S-P-A-D-K-
LTPEEK------ -L-T-P-E-E-K

$$\left| \binom{2n}{n} \right| = \frac{(2n)!}{(n!)^2}$$

$$\binom{2n}{n} = \frac{(2n)!}{(n!)^2} = \frac{(2*300)!}{(300!)^2} \approx 7\times10^{88}$$



(Source: http://discoverystudentadventures.blogspot.com/2011/11/10-mind-bending-facts-about-universe.html)

The visible universe is estimated to contain 10^{78} ~ 10^{80} atoms (Source: wikianswers) only !

Sequence Alignment:
What is the computational **A**lgorithm?

MV-LSP
MVHLTP

```
HBA_HUMAN      1 MV-LSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF-D     48
HBB_HUMAN      1 MVHLTPEEKSAVTALWGKV--NVDEVGGEALGRLLVVYPWTQRFFESFGD     48

HBA_HUMAN     49 LS-----HGSAQVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKLR     93
HBB_HUMAN     49 LSTPDAVMGNPKVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLH     98

HBA_HUMAN     94 VDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSKYR    142
HBB_HUMAN     99 VDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANALAHKYH    147
```

A residue can either

- Align to other residue, or
- Align to a gap

S	S	-
T	-	T

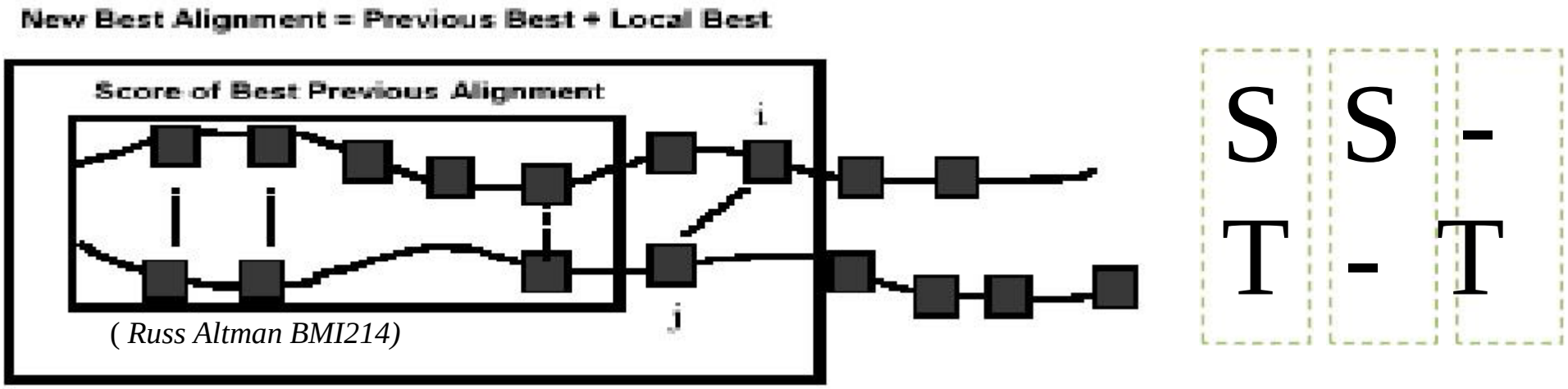
```
#=======================================
#
# Aligned_sequences: 2
# 1: HBA_HUMAN
# 2: HBB_HUMAN
# Matrix: EBLOSUM62
# Gap_penalty: 10.0
# Extend_penalty: 0.5
#
# Length: 149
# Identity:       65/149 (43.6%)
# Similarity:     90/149 (60.4%)
# Gaps:            9/149 ( 6.0%)
# Score: 292.5
#
#
#=======================================

HBA_HUMAN         1 MV-LSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF-D     48
                    || |:|.:|:.|.|.||||  :..|.|.|||.|:.:.:|.|:.:|..| |
HBB_HUMAN         1 MVHLTPEEKSAVTALWGKV--NVDEVGGEALGRLLVVYPWTQRFFESFGD     48

HBA_HUMAN        49 LS-----HGSAQVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKLR     93
                    ||     .|:.:||.|||||..|.:::.:|||:|::....:.||:||..||.
HBB_HUMAN        49 LSTPDAVMGNPKVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLH     98

HBA_HUMAN        94 VDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSKYR    142
                    |||.||:||.:.|:..||.|...|||.|.|.|:..|.:|.|:..|..||.
HBB_HUMAN        99 VDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANALAHKYH    147
```

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

Affine gap penalty: opening a gap receives a penalty of d; extending a gap receives a penalty of e. So the total Penalty for a gap with length n would be:
Penalty = d + (n-1)* e

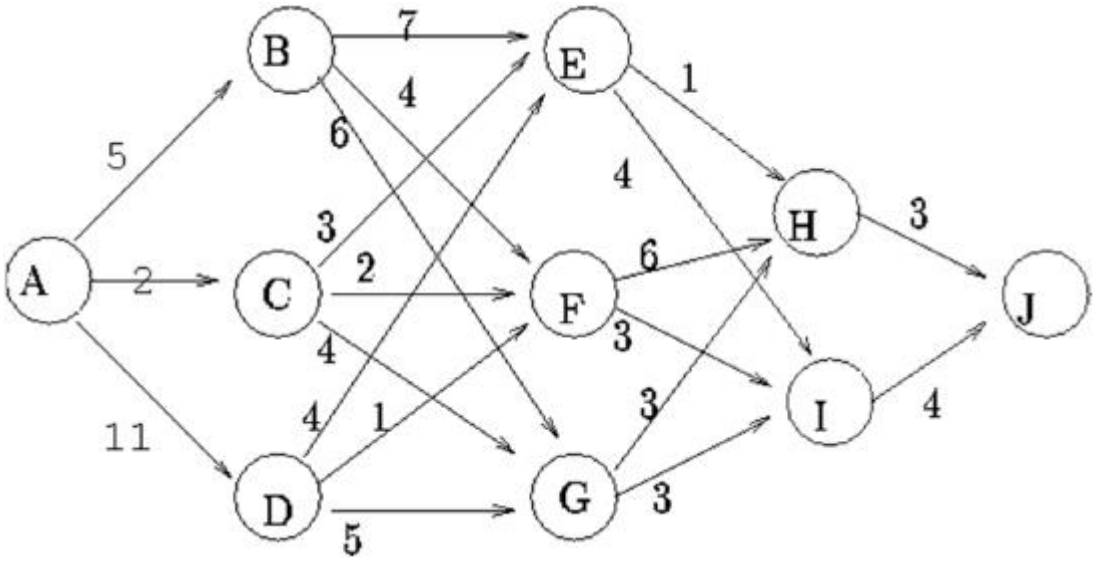
Final Score = (sum of substitution scores) + (-1) * (sum of Gap Penalty)

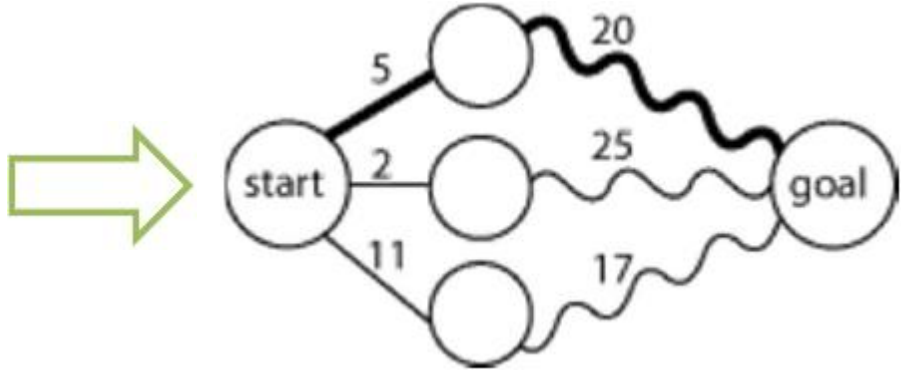
The best alignment that ends at a given pair of symbols is the best alignment of the sequences up to that point, plus the best alignment for the two additional symbols.



Dynamic Programming

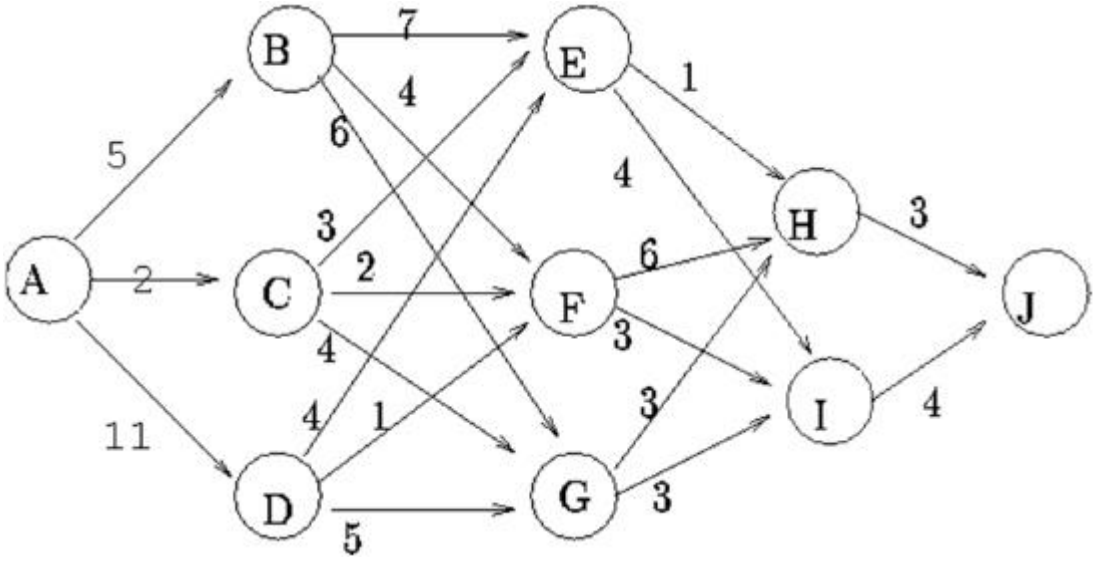
Dynamic Programming solves problems by combining the solutions to sub-problems.

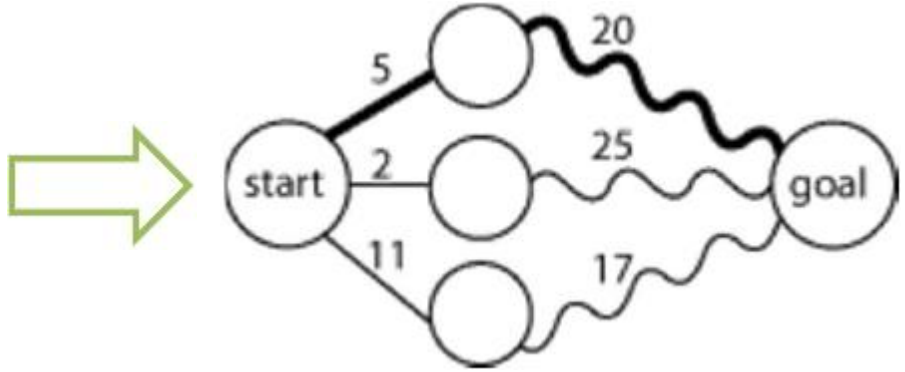




(Modified from Wikipedia)

1. Break the problem into smaller sub-problems.
2. Solve these sub-problems optimally recursively.
3. Use these optimal solutions to construct an optimal solution for the original problem.





(Modified from Wikipedia)

MV-LSP
MVHLTP

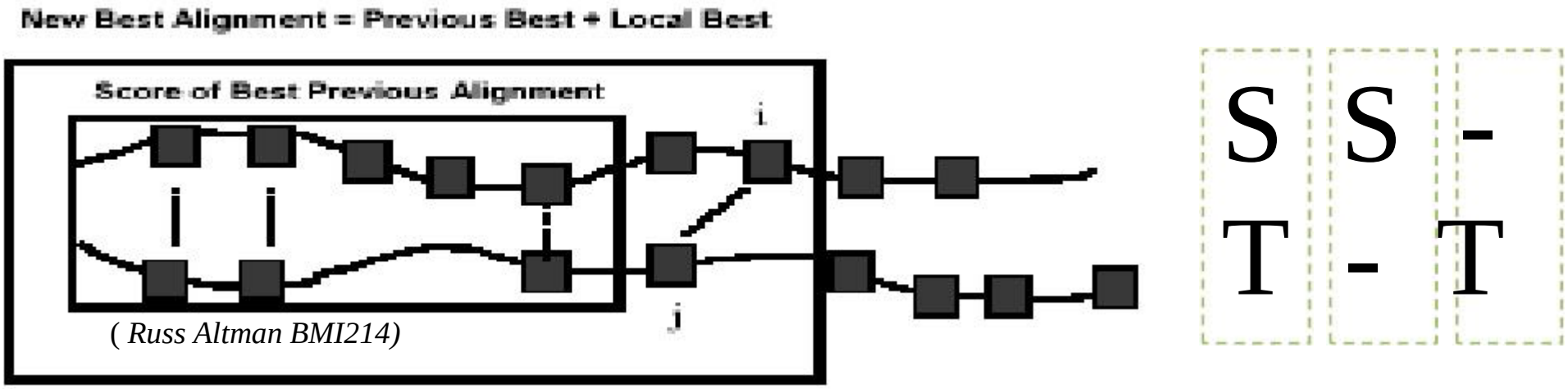
```
HBA_HUMAN      1 MV-LSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF-D     48
HBB_HUMAN      1 MVHLTPEEKSAVTALWGKV--NVDEVGGEALGRLLVVYPWTQRFFESFGD     48

HBA_HUMAN     49 LS-----HGSAQVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKLR     93
HBB_HUMAN     49 LSTPDAVMGNPKVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLH     98

HBA_HUMAN     94 VDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSKYR    142
HBB_HUMAN     99 VDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANALAHKYH    147
```

A residue can either

- Align to other residue, or
- Align to a gap

S	S	-
T	-	T

The best alignment that ends at a given pair of symbols is the best alignment of the sequences up to that point, plus the best alignment for the two additional symbols.



Sequence alignment with Dynamic Programming: the Formula

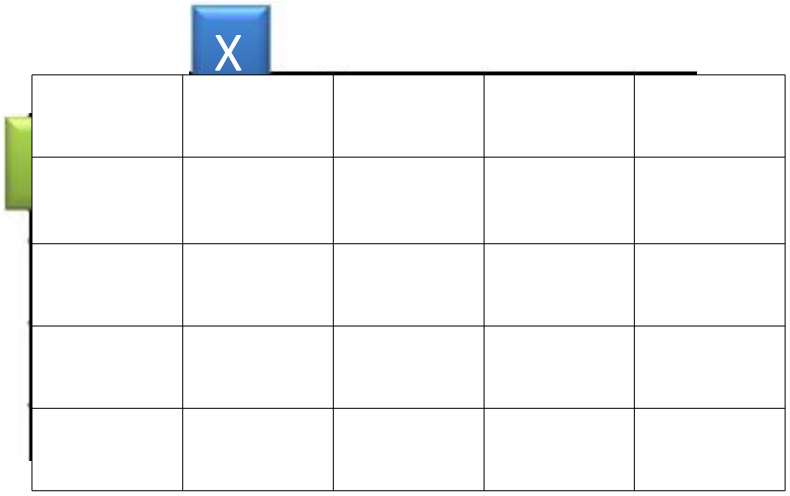
- Align two sequences: x and y
  - F (i,j) is the score of the best alignment between $x_{1\ldots i}$ and $y_{1\ldots j}$
  - s(A,B) is the score for substituting A with B; d is the (linear) gap penalty

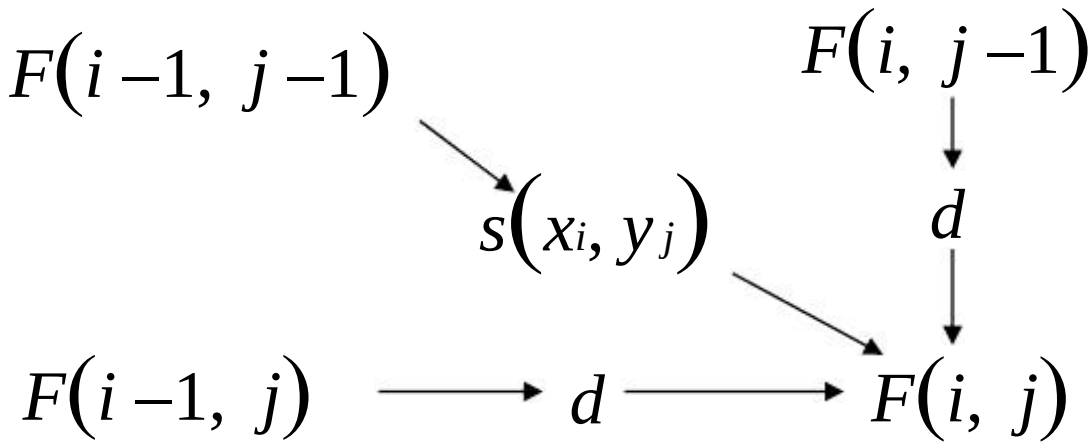
$$F(0,0) = 0$$

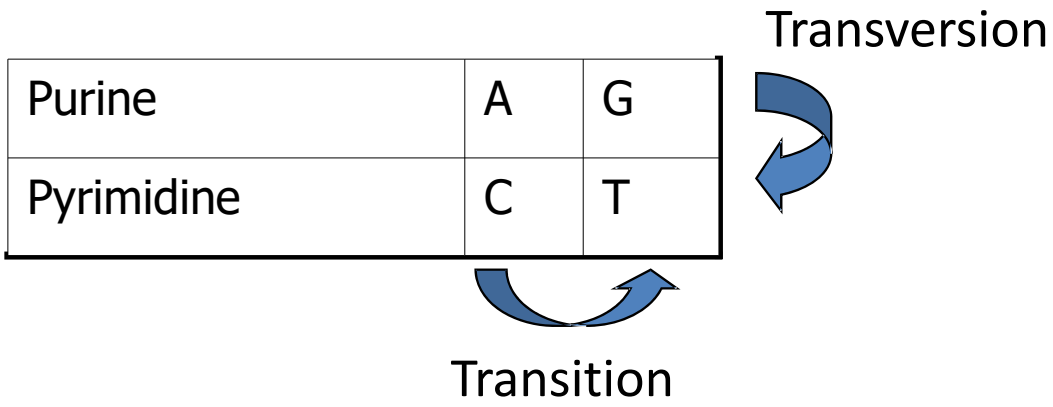
$$F(i,\ j) = \max \begin{cases} F(i-1,\ j-1) + s(x_i, y_j) & \textbf{x_i aligned to y_j} \\ F(i-1,\ j) + d & \textbf{x_i aligned to \textit{a gap}} \\ F(i,\ j-1) + d & \textbf{y_j aligned to \textit{a gap}} \end{cases}$$

$$F(0,0) = 0$$

$$F(i,\ j) = \max \begin{cases} F(i-1,\ j-1) + s(x_i, y_j) & \textbf{x_i aligned to y_j} \\ F(i-1,\ j) + d & \textbf{x_i aligned to \textit{a gap}} \\ F(i,\ j-1) + d & \textbf{y_j aligned to \textit{a gap}} \end{cases}$$







Scoring Nucleotide

A nucleotide substitution matrix:

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

Input Sequence 1: AAG

Input Sequence 2: AGC

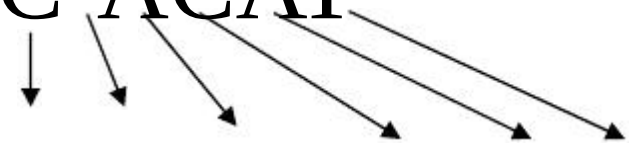
	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

For simplicity, let's set (i.e. linear gap penalty)

gap OPEN (d) = gap EXTEND (e) = -5

GAC-AT

C-ACAT



(-7) + (-5) + (-7) + (-5) + 2 + 2= -20

Dynamic Programming Matrix

		A	A	G
A				
G				
C				

DP for sequence alignment: Example

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

Find the optimal alignment of AAG and AGC.
Use a linear gap penalty of d=-5.

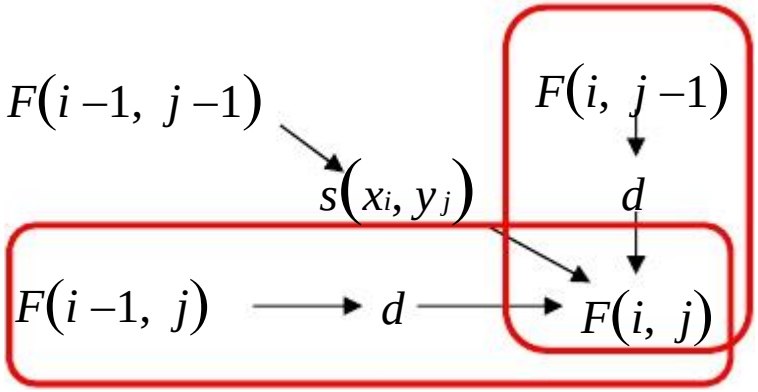
		A	A	G
	0			
A				
G				
C				

$$F(0,0) = 0$$

$$F(i,j) = \max \begin{cases} F(i-1,j-1) + s(x_i, y_j) \\ F(i-1,j) + d \\ F(i,j-1) + d \end{cases}$$

DP for sequence alignment: Example

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

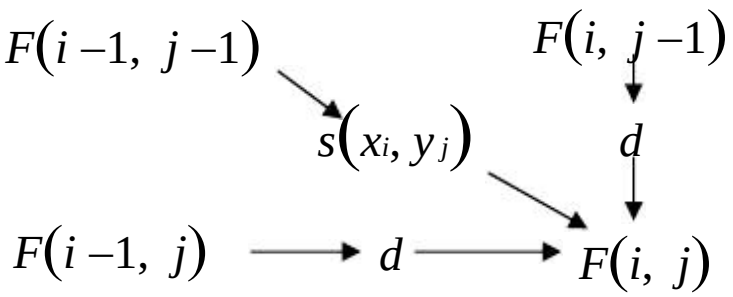


Find the optimal alignment of AAG and AGC.
Use a linear gap penalty of d=-5.

		A	A	G
	0	-5	-10	-15
A	-5			
G	-10			
C	-15			

DP for sequence alignment: Example

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2



Find the optimal alignment of AAG and AGC.
Use a linear gap penalty of d=-5.

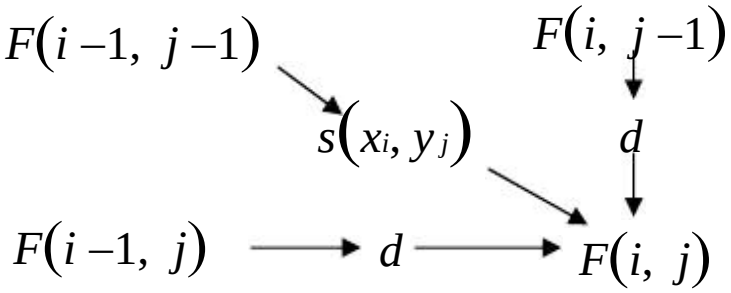
		A	A	G
	0	-5	-10	-15
A	-5	2	-3	-8
G	-10	-3	-3	-1
C	-15	-8	-8	**-6**

DP for sequence alignment: Example

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

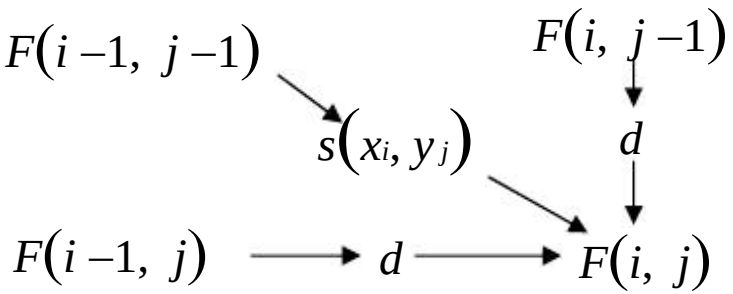
Find the optimal alignment of AAG and AGC.
Use a linear gap penalty of d=-5.

		A
	0	-5
A	-5	2



DP for sequence alignment: Example

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2



Find the optimal alignment of AAG and AGC.
Use a linear gap penalty of d=-5.

		A
	0	-5
A	-5	2

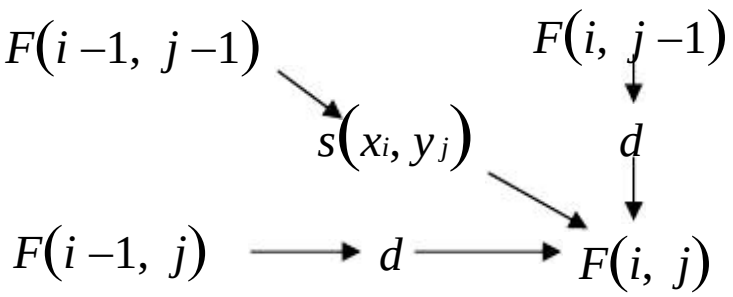
-5 + (-5) = -10

0 + 2 = 2

-5 + (-5) = -10

DP for sequence alignment: Example

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

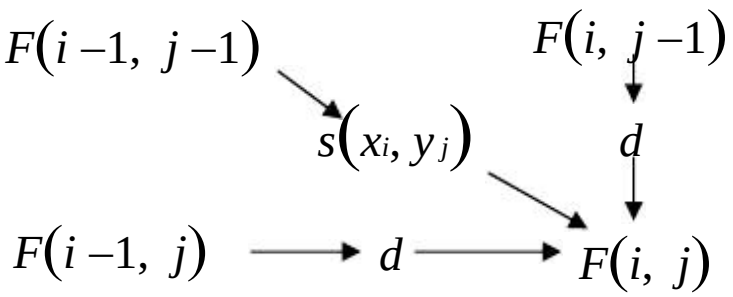


Find the optimal alignment of AAG and AGC.
Use a linear gap penalty of d=-5.

		A	A	G
	0	-5	-10	-15
A	-5	2	-3	-8
G	-10	-3	-3	-1
C	-15	-8	-8	**-6**

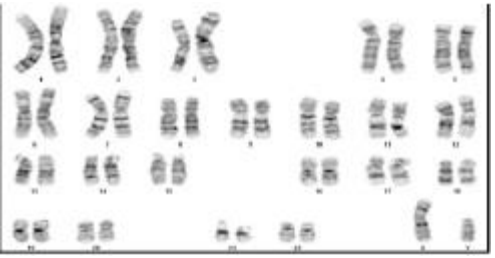
DP for sequence alignment: Example

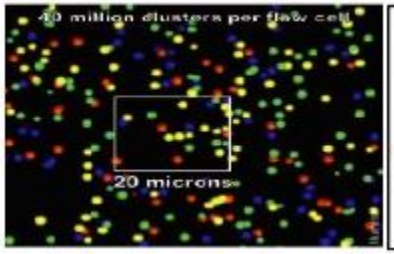
	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

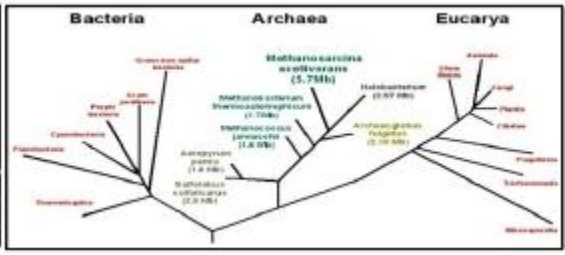


Find the optimal alignment of AAG and AGC.
Use a linear gap penalty of d=-5.

		A	A	G
	0	-5	-10	-15
A	-5	2	-3	-8
G	-10	-3	-3	-1
C	-15	-8	-8	**-6**





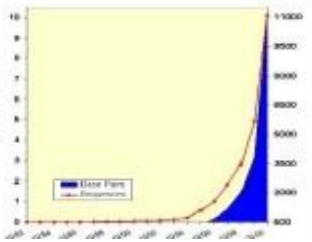


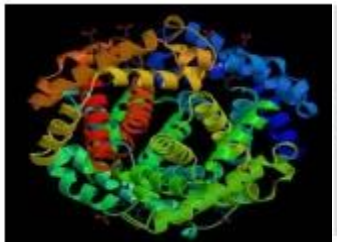
Unit 3:
From Global to Local

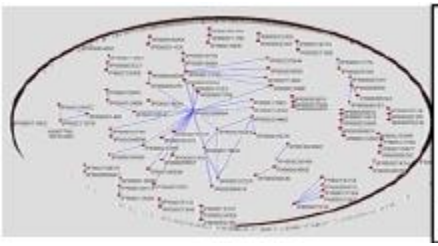
Kun Lu

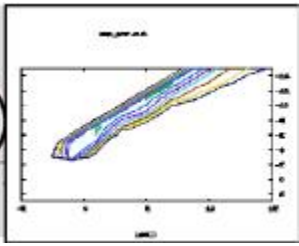
Computer Science Department

Southwest University









A A G -
- A G C

A A G -
A - G C

		A	A	G
	0	-5		
A		2	-3	
G				-1
C				**-6**

End-to-end: Global Alignment

Global Alignment: End-to-end

Needleman–Wunsch algorithm

J. Mol. Biol. (1970) **48**, 443–453

A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins

SAUL B. NEEDLEMAN AND CHRISTIAN D. WUNSCH

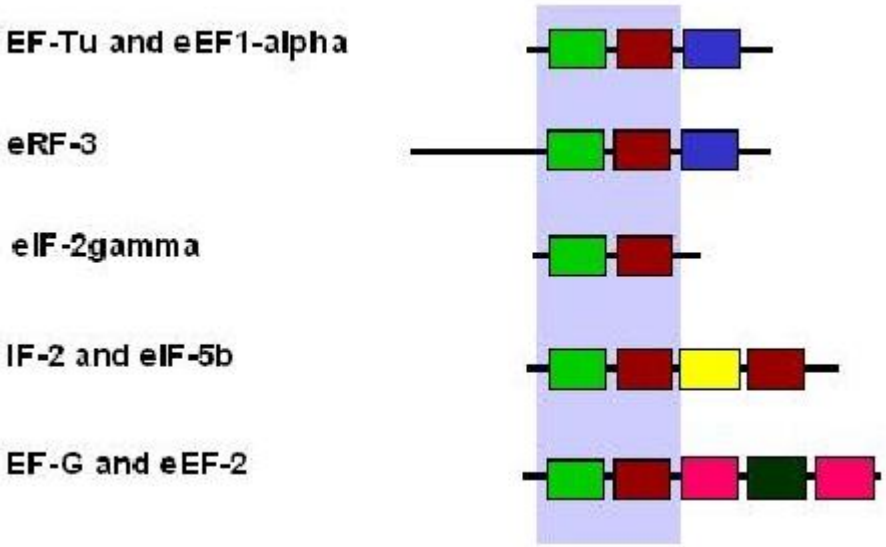
Department of Biochemistry, Northwestern University, and Nuclear Medicine Service, V. A. Research Hospital Chicago, Ill. 60611, U.S.A.

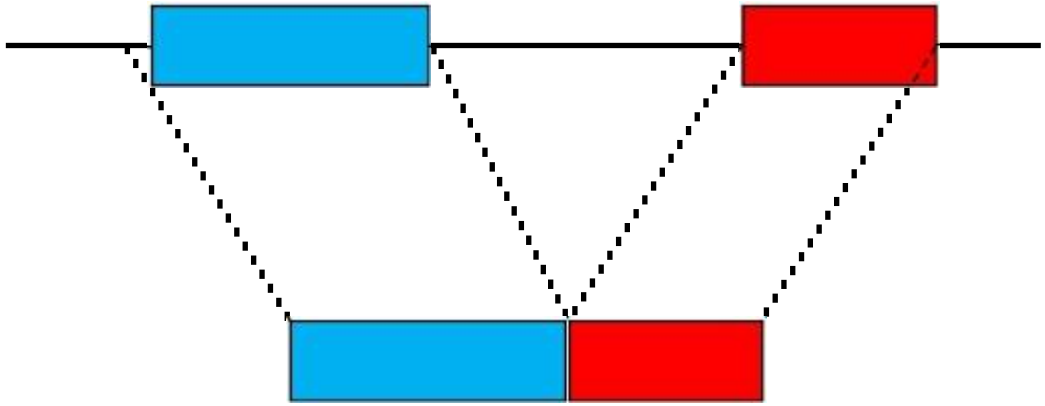
(*Received 21 July 1969*)

A computer adaptable method for finding similarities in the amino acid sequences of two proteins has been developed. From these findings it is possible to determine whether significant homology exists between the proteins. This information is used to trace their possible evolutionary development.

The maximum match is a number dependent upon the similarity of the sequences. One of its definitions is the largest number of amino acids of one protein that can be matched with those of a second protein allowing for all possible interruptions in either of the sequences. While the interruptions give rise to a very large number of comparisons, the method efficiently excludes from consideration those comparisons that cannot contribute to the maximum match.

Comparisons are made from the smallest unit of significance, a pair of amino acids, one from each protein. All possible pairs are represented by a two-dimensional array, and all possible comparisons are represented by pathways through the array. For this maximum match only certain of the possible pathways must be evaluated. A numerical value, one in this case, is assigned to every cell in the array representing like amino acids. The maximum match is the largest number that would result from summing the cell values of every pathway.





Identify similar sub-sequence

Smith and Waterman at Los Alamos, New Mexico

Photo by David Lipman, taken summer of 1980

(http://www.cmb.usc.edu/people/msw/SmithWaterman.html)

J. Mol. Biol. (1981), 147, 195–197

Identification of Common Molecular Subsequences

The identification of maximally homologous subsequences among sets of long sequences is an important problem in molecular sequence analysis. The problem is straightforward only if one restricts consideration to contiguous subsequences (segments) containing no internal deletions or insertions. The more general problem has its solution in an extension of sequence metrics (Sellers 1974; Waterman *et al.*, 1976) developed to measure the minimum number of "events" required to convert one sequence into another.

These developments in the modern sequence analysis began with the heuristic homology algorithm of Needleman & Wunsch (1970) which first introduced an iterative matrix method of calculation. Numerous other heuristic algorithms have been suggested including those of Fitch (1966) and Dayhoff (1969). More mathematically rigorous algorithms were suggested by Sankoff (1972), Reichert *et al.* (1973) and Beyer *et al.* (1979), but these were generally not biologically satisfying or interpretable. Success came with Sellers (1974) development of a true metric measure of the distance between sequences. This metric was later generalized by Waterman *et al.* (1976) to include deletions/insertions of arbitrary length. This metric represents the minimum number of "mutational events" required to convert one sequence into another. It is of interest to note that Smith *et al.* (1980) have recently shown that under some conditions the generalized Sellers metric is equivalent to the

$$F(0,0) = 0$$

$$F(i,j) = \max \begin{cases} F(i-1,\ j-1) + s(x_i, y_j) \\ F(i-1,\ j) + d \\ F(i,\ j-1) + d \end{cases}$$

Global alignment

$$F\ (0,0) = \ 0$$

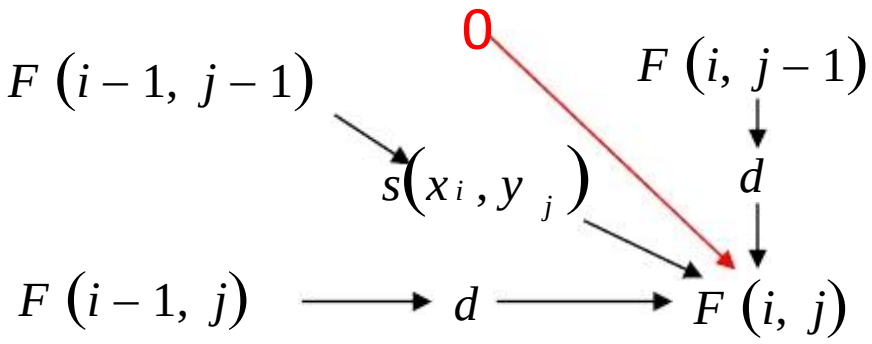
$$F\ (i,\ j) = \ \max \ \begin{cases} F\ (i-1,\ j-1) + \ s(x_i\ ,\ y_{\ \ \ j}\ ) \\ F\ (i-1,\ j) + \ d \\ F\ (i,\ j-1) + \ d \\ 0 \end{cases}$$

Local alignment

DP for Local alignment: Formula

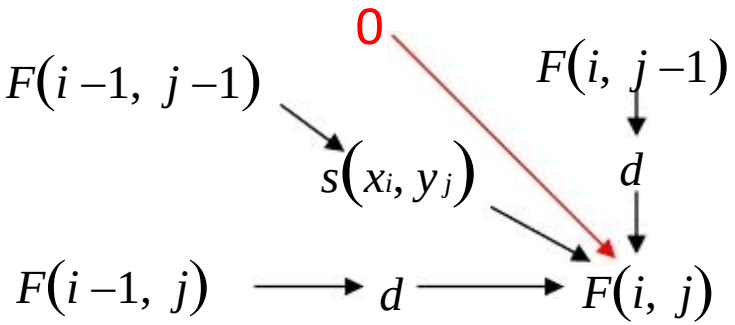
$$F(0,0) = 0$$

$$F(i, j) = \max \begin{cases} F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) + d \\ F(i, j-1) + d \\ 0 \end{cases}$$



DP for Local alignment: Example

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

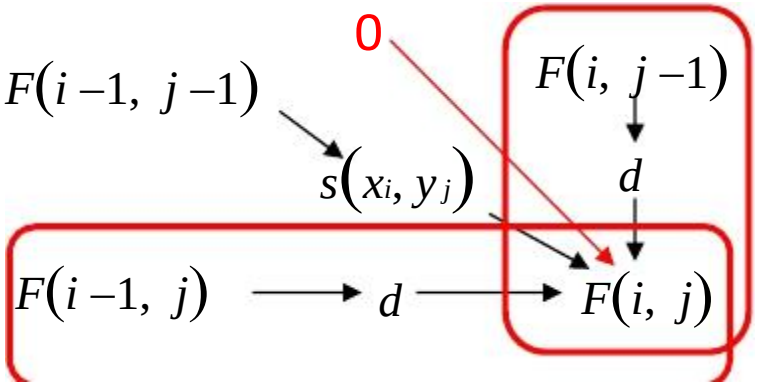


Find the optimal local alignment of AAG and AGC.
Use a linear gap penalty of d = -5.

		A	A	G
A				
G				
C				

DP for Local alignment: Example

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

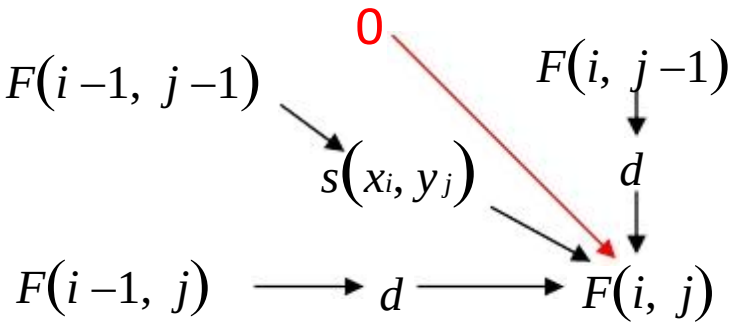


Find the optimal local alignment of AAG and AGC.
Use a linear gap penalty of d = -5.

		A	A	G
	0	0	0	0
A	0			
G	0			
C	0			

DP for Local alignment: Example

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

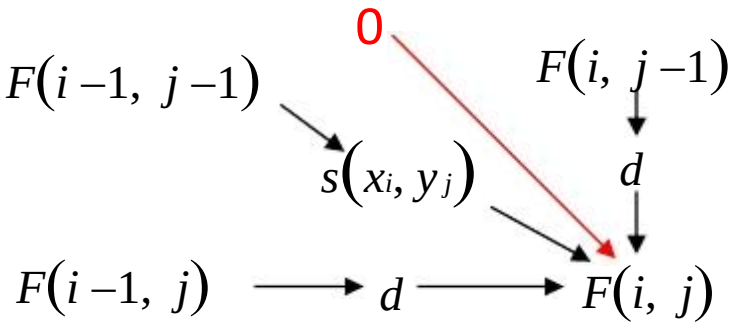


Find the optimal local alignment of AAG and AGC.
Use a linear gap penalty of d = -5.

		A	A	G
	0	0	0	0
A	0	2	2	0
G	0	0	0	4
C	0	0	0	0

DP for Local alignment: Example

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2



Find the optimal local alignment of AAG and AGC.
Use a linear gap penalty of d = -5.

		A	A	G
	0	0	0	0
A	0	2	2	0
G	0	0	0	4
C	0	0	0	0

Traceback: Decode the Local Alignment

- Trace back begins at the highest score in the matrix and continues until you reach 0.

A G
A G

		A	A	G
	0	0	0	0
A	0	2	2	0
G	0	0	0	4
C	0	0	0	0

Traceback: Decode the Local Alignment

- And also the secondary best alignment

A
A

		A	A	G
	0	0	0	0
A	0	2	2	0
G	0	0	0	4
C	0	0	0	0

Global vs. Local

$$F(0,0) = 0$$

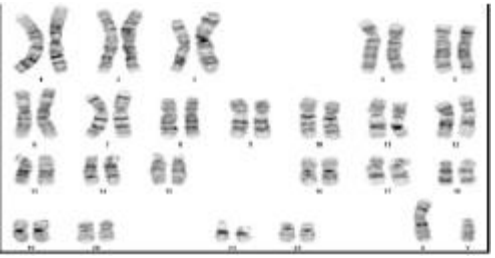
$$F(i,j) = \max \begin{cases} F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) + d \\ F(i, j-1) + d \end{cases}$$

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

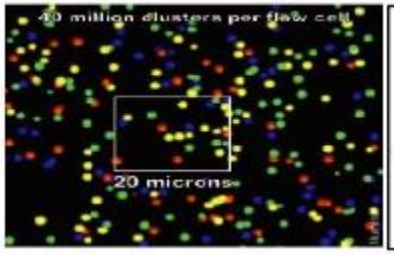
$$F(0,0) = 0$$

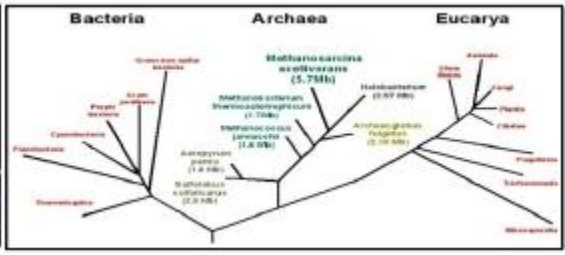
$$F(i,j) = \max \begin{cases} F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) + d \\ F(i, j-1) + d \\ 0 \end{cases}$$

A	G			A
A	G			A



TAACCCTAACCCTAACCCTAACCCTAACCCTA
CCTAACCCTAACCCTAACCCTAACCCTAACCC
CCCTAACCCCTAACCCTAACCCTAACCCTAAC
AACCCTAACCCTAACCCTAACCCCTAACCCTA
ACCCTAACCCCAACCCCAACCCCAACCCCAAC
CTACCCTAACCCTAACCCTAACCCTAACCCTA
ACCCTAACCCTAACCCTAACCCCTAACCCTAA



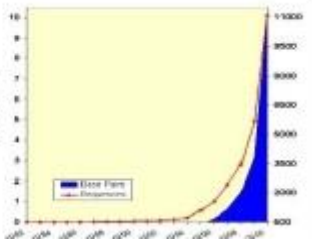


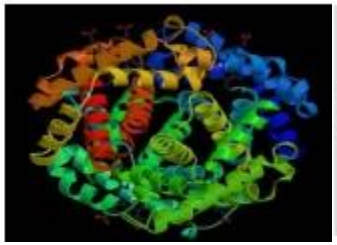
Unit 3:
From Global to Local

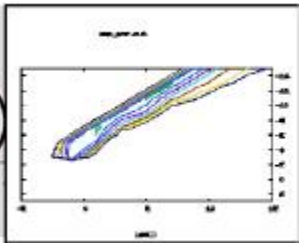
Le Zhang, Ph. D.

Computer Science Department

Southwest University







A A G -
- A G C

A A G -
A - G C

		A	A	G
	0	-5		
A		2	-3	
G				-1
C				**-6**

End-to-end: Global Alignment

Global Alignment: End-to-end

Needleman–Wunsch algorithm

J. Mol. Biol. (1970) **48**, 443–453

A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins

SAUL B. NEEDLEMAN AND CHRISTIAN D. WUNSCH

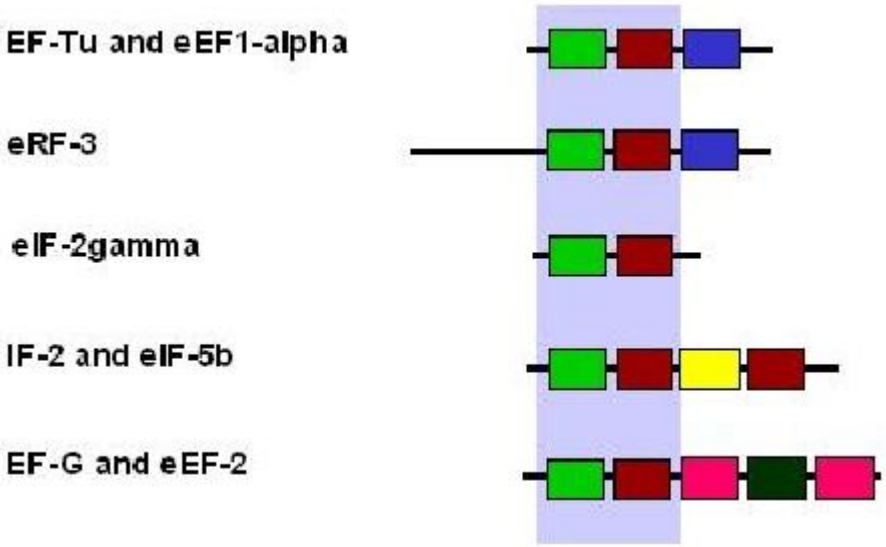
Department of Biochemistry, Northwestern University, and Nuclear Medicine Service, V. A. Research Hospital Chicago, Ill. 60611, U.S.A.

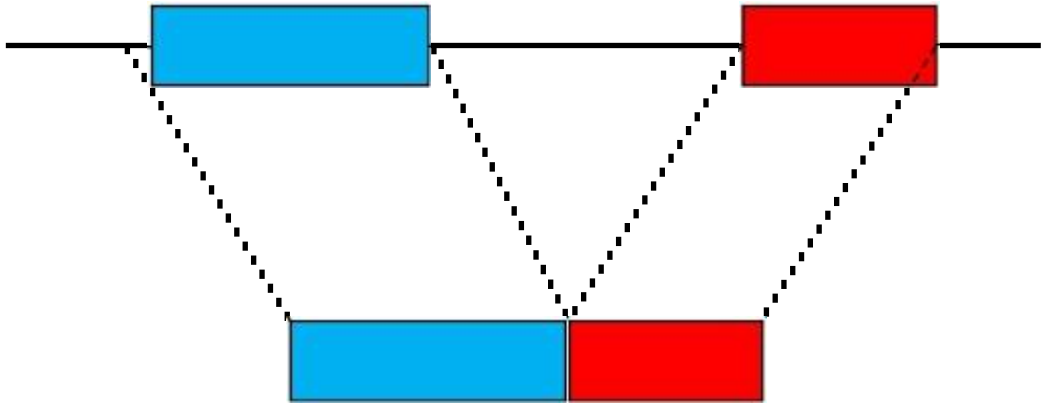
(*Received 21 July 1969*)

A computer adaptable method for finding similarities in the amino acid sequences of two proteins has been developed. From these findings it is possible to determine whether significant homology exists between the proteins. This information is used to trace their possible evolutionary development.

The maximum match is a number dependent upon the similarity of the sequences. One of its definitions is the largest number of amino acids of one protein that can be matched with those of a second protein allowing for all possible interruptions in either of the sequences. While the interruptions give rise to a very large number of comparisons, the method efficiently excludes from consideration those comparisons that cannot contribute to the maximum match.

Comparisons are made from the smallest unit of significance, a pair of amino acids, one from each protein. All possible pairs are represented by a two-dimensional array, and all possible comparisons are represented by pathways through the array. For this maximum match only certain of the possible pathways must be evaluated. A numerical value, one in this case, is assigned to every cell in the array representing like amino acids. The maximum match is the largest number that would result from summing the cell values of every pathway.





Identify similar sub-sequence

Smith and Waterman at Los Alamos, New Mexico

Photo by David Lipman, taken summer of 1980

(http://www.cmb.usc.edu/people/msw/SmithWaterman.html)

J. Mol. Biol. (1981), 147, 195–197

Identification of Common Molecular Subsequences

The identification of maximally homologous subsequences among sets of long sequences is an important problem in molecular sequence analysis. The problem is straightforward only if one restricts consideration to contiguous subsequences (segments) containing no internal deletions or insertions. The more general problem has its solution in an extension of sequence metrics (Sellers 1974; Waterman *et al.*, 1976) developed to measure the minimum number of "events" required to convert one sequence into another.

These developments in the modern sequence analysis began with the heuristic homology algorithm of Needleman & Wunsch (1970) which first introduced an iterative matrix method of calculation. Numerous other heuristic algorithms have been suggested including those of Fitch (1966) and Dayhoff (1969). More mathematically rigorous algorithms were suggested by Sankoff (1972), Reichert *et al.* (1973) and Beyer *et al.* (1979), but these were generally not biologically satisfying or interpretable. Success came with Sellers (1974) development of a true metric measure of the distance between sequences. This metric was later generalized by Waterman *et al.* (1976) to include deletions/insertions of arbitrary length. This metric represents the minimum number of "mutational events" required to convert one sequence into another. It is of interest to note that Smith *et al.* (1980) have recently shown that under some conditions the generalized Sellers metric is equivalent to the

$$F(0,0) = 0$$

$$F(i,j) = \max \begin{cases} F(i-1,\ j-1) + s(x_i, y_j) \\ F(i-1,\ j) + d \\ F(i,\ j-1) + d \end{cases}$$

Global alignment

$$F\ (0,0) = \ 0$$

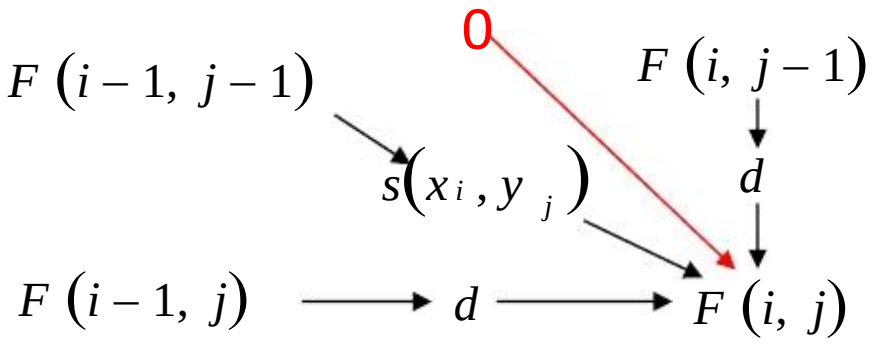
$$F\ (i,\ j) = \ \max \ \begin{cases} F\ (i-1,\ j-1) + \ s(x_i\ ,\ y_{\ \ \ j}\ ) \\ F\ (i-1,\ j) + \ d \\ F\ (i,\ j-1) + \ d \\ 0 \end{cases}$$

Local alignment

DP for Local alignment: Formula

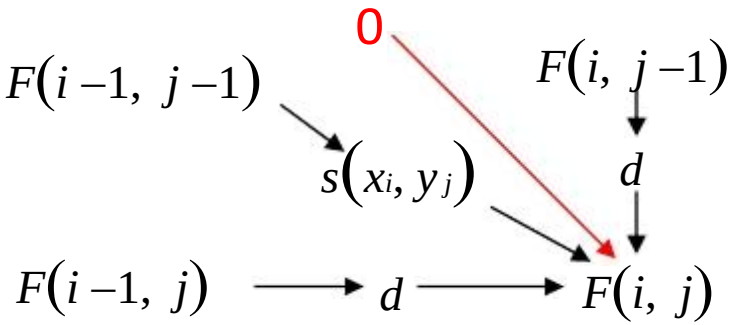
$$F(0,0) = 0$$

$$F(i, j) = \max \begin{cases} F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) + d \\ F(i, j-1) + d \\ 0 \end{cases}$$



DP for Local alignment: Example

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

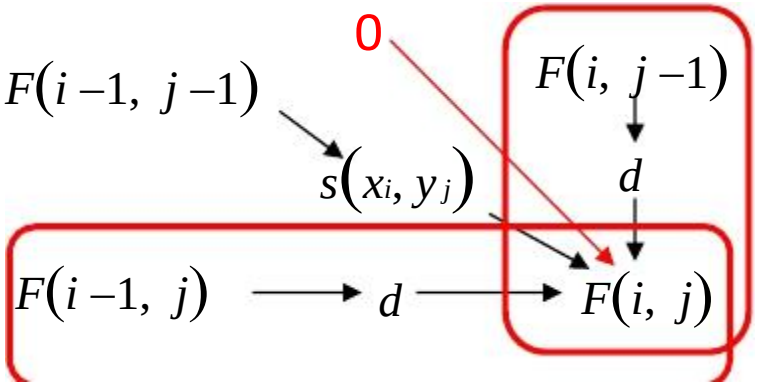


Find the optimal local alignment of AAG and AGC.
Use a linear gap penalty of d = -5.

		A	A	G
A				
G				
C				

DP for Local alignment: Example

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

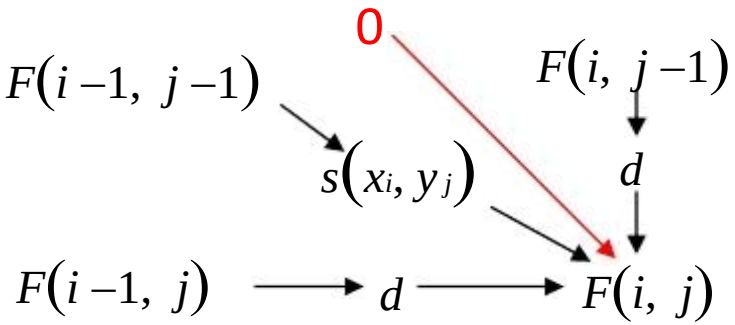


Find the optimal local alignment of AAG and AGC.
Use a linear gap penalty of d = -5.

		A	A	G
	0	0	0	0
A	0			
G	0			
C	0			

DP for Local alignment: Example

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

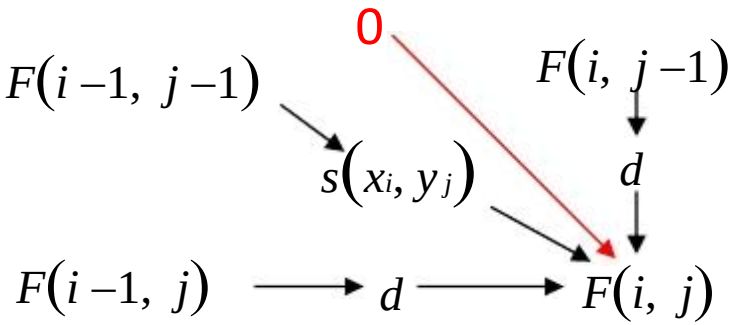


Find the optimal local alignment of AAG and AGC.
Use a linear gap penalty of d = -5.

		A	A	G
	0	0	0	0
A	0	2	2	0
G	0	0	0	4
C	0	0	0	0

DP for Local alignment: Example

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2



Find the optimal local alignment of AAG and AGC.
Use a linear gap penalty of d = -5.

		A	A	G
	0	0	0	0
A	0	2	2	0
G	0	0	0	4
C	0	0	0	0

Traceback: Decode the Local Alignment

- Trace back begins at the highest score in the matrix and continues until you reach 0.

A G
A G

		A	A	G
	0	0	0	0
A	0	2	2	0
G	0	0	0	4
C	0	0	0	0

Traceback: Decode the Local Alignment

- And also the secondary best alignment

A
A

		A	A	G
	0	0	0	0
A	0	2	2	0
G	0	0	0	4
C	0	0	0	0

Global vs. Local

$$F(0,0) = 0$$

$$F(i,j) = \max \begin{cases} F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) + d \\ F(i, j-1) + d \end{cases}$$

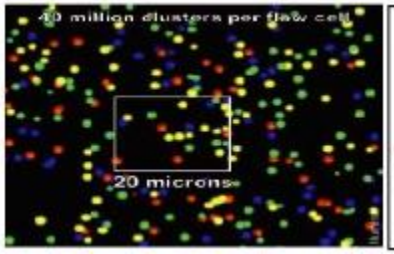
A	A	G	-		A	A	G	-	
-	A	G	C		A	-	G	C	

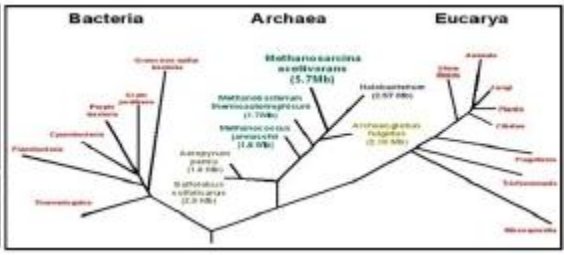
$$F(0,0) = 0$$

$$F(i,j) = \max \begin{cases} F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) + d \\ F(i, j-1) + d \\ 0 \end{cases}$$

A	G			A
A	G			A

TAACCCTAACCCTAACCCTAACCCTAACCCTA
CCTAACCCTAACCCTAACCCTAACCCTAACCC
CCCTAACCCCTAACCCTAACCCTAACCCTAAC
AACCCTAACCCTAACCCTAACCCCTAACCCTA
ACCCTAACCCCAACCCCAACCCCAACCCCAAC
CTACCCTAACCCTAACCCTAACCCTAACCCTA
ACCCTAACCCTAACCCTAACCCCTAACCCTAA





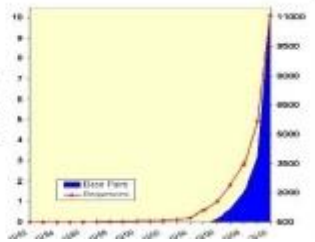
Unit 4:

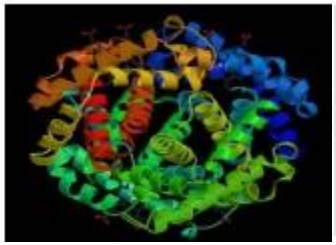
Supplementary Learning Materials

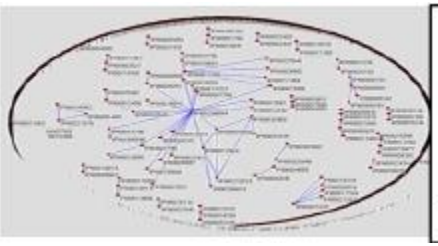
Kun Lu

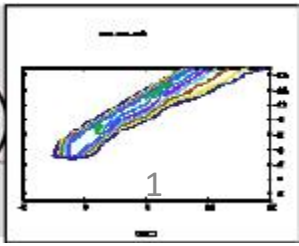
Computer Science Department

Southwest University









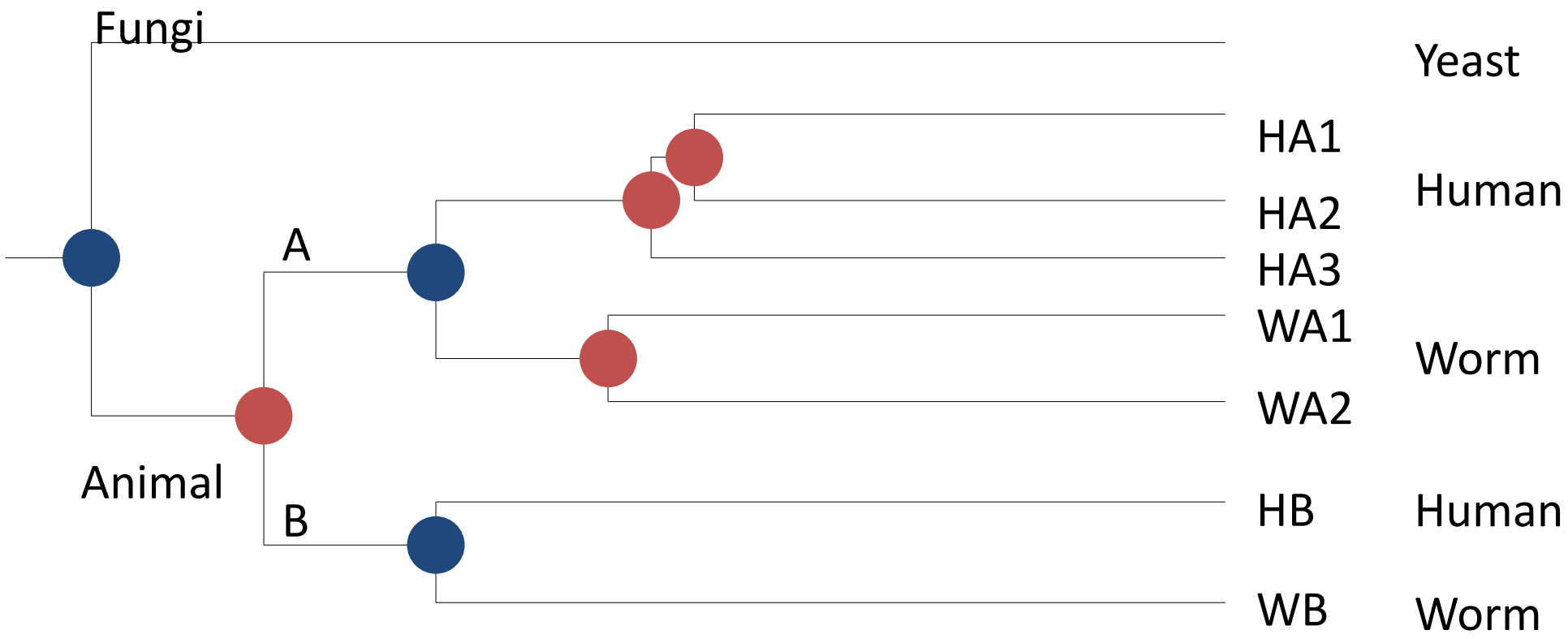
Outline

- Homology & Similarity
- Similarity Matrix
- Dot Matrix

Homology

- Homology
  - derived from a common ancestor
  - ortholog: derived from speciation
  - paralog: derived from duplication

Ortholog vs Paralog

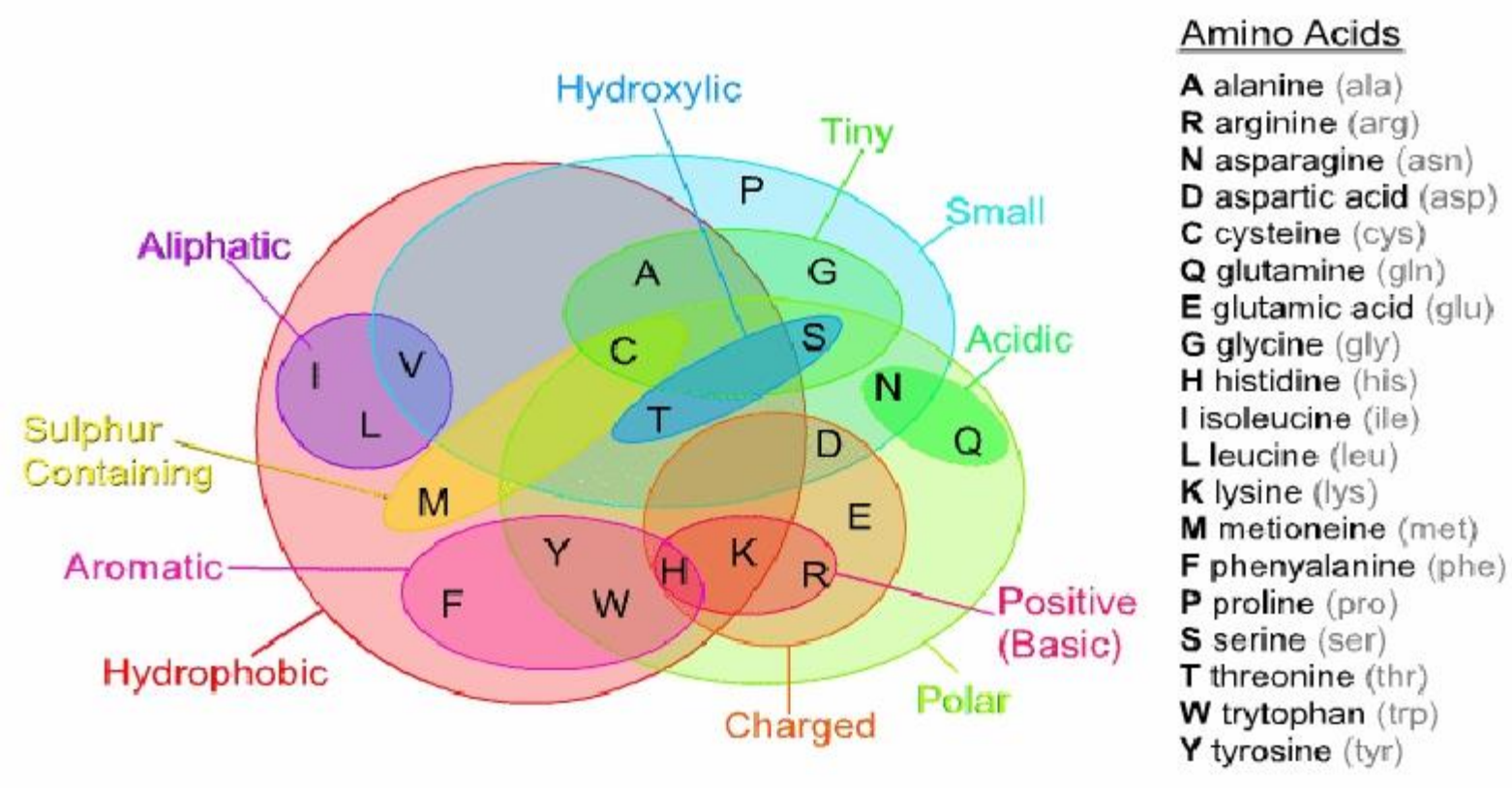


Ortholog comes with speciation Paralog comes with duplication

revised based on Sonnhammer, E.L., and Koonin, E.V. (2002). Orthology, paralogy and proposed classification for paralog subtypes. *TRENDS in Genetics 18*, 619–620.

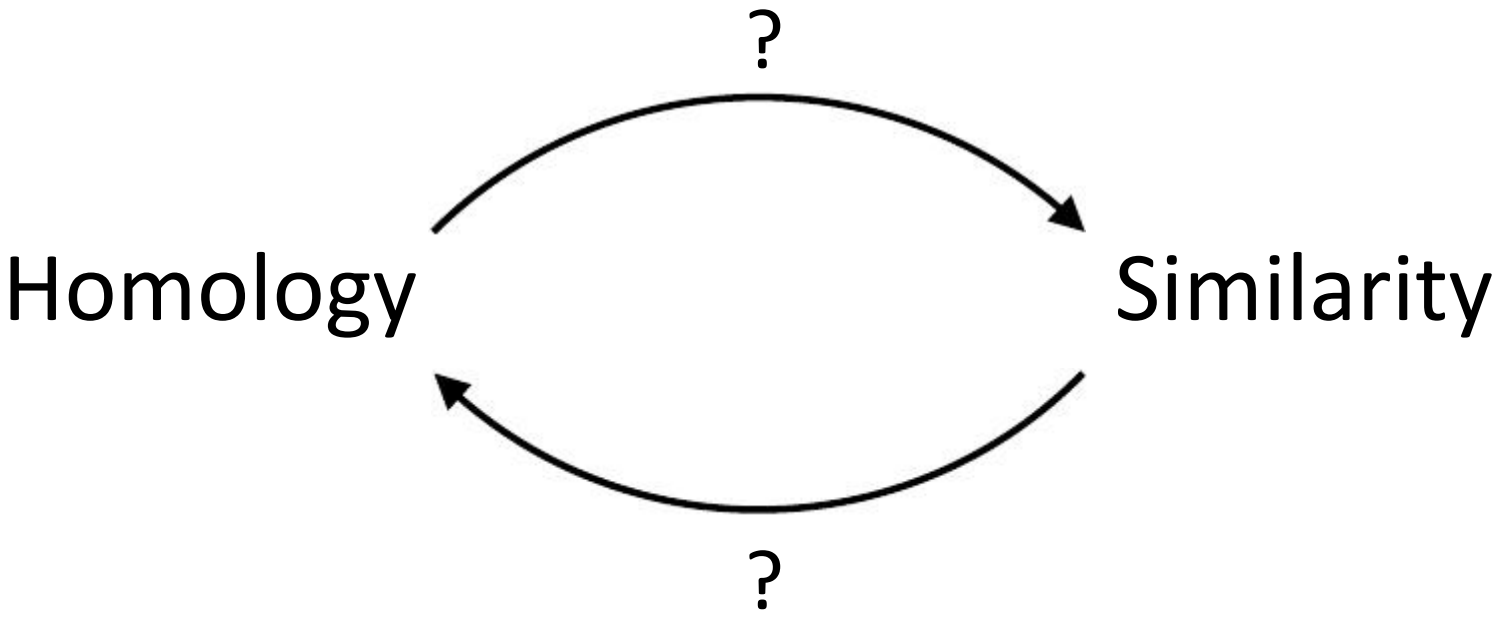
Similarity vs Identity

- Similarity
- Identity



(Adopted from Prof. Jingchu Luo)

Homology vs Similarity



How to let computer do this job?

- How to measure similarity?
  - Similarity matrix

Similarity Matrix

- For nucleotides,
  - usually only distinguish match / mismatch (identity matrix) for sequence alignment
  - but a more complicated substitution model is used for phylogeny reconstruction

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

Similarity Matrix

- For amino acids,
  - PAM (1978, Margaret Dayhoff)
    - Two sequences are 1 PAM apart if they differ in 1 % of the residues.
    - 1 PAM = one step of evolution
  - BLOSUM (1992, Steven Henikoff & Jorja Henikoff)
    - computed by looking at "blocks" of conserved sequences found in multiple protein alignments

PAM

1	A	B	C
A	0.8	0.1	0.1
B	0.05	0.9	0.05
C	0.15	0.05	0.8

- PAM 1

- PAM 2 ?

P(A->?->A) = P(A->A->A) + P(A->B->A) + P(A->C->A)
= P(A->A)P(A->A) + P(A->B)P(B->A) + P(A->C)P(C->A)

P(A->?->B) = P(A->A->B) + P(A->B->B) + P(A->C->B)
= P(A->A)P(A->B) + P(A->B)P(B->B) + P(A->C)P(C->B)

…

PAM

- PAM 1
- PAM 2 = $(\text{PAM 1})_2$

1	A	B	C
A	0.8	0.1	0.1
B	0.05	0.9	0.05
C	0.15	0.05	0.8

$\times$

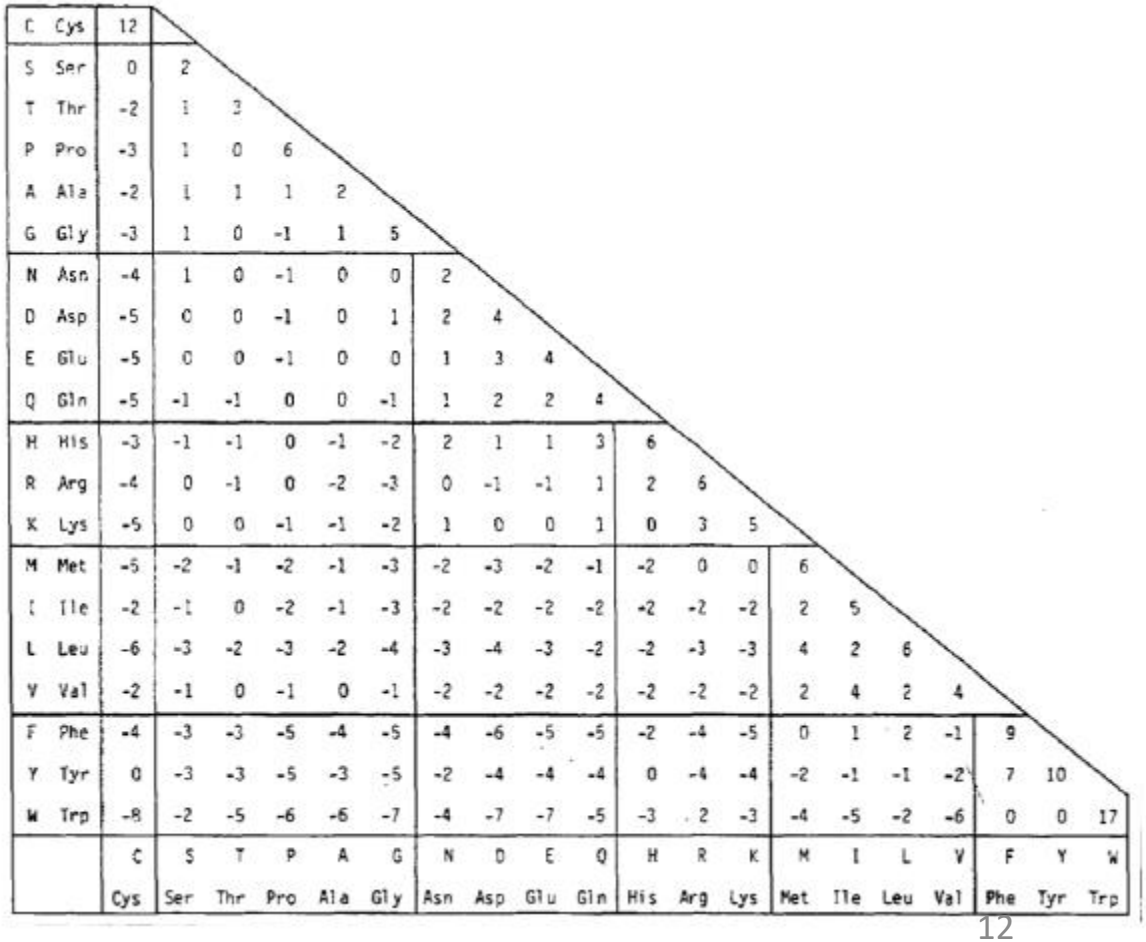
1	A	B	C
A	0.8	0.1	0.1
B	0.05	0.9	0.05
C	0.15	0.05	0.8

=

2	A	B	C
A	0.66	0.175	0.165
B	0.093	0.817	0.09
C	0.243	0.1	0.657

PAM

- PAM 1
- PAM 250
  = $(\text{PAM 1})_{250}$
- Log odds of PAM 250
  log odds = log(p/(1-p))



		C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
		Cys	Ser	Thr	Pro	Ala	Gly	Asn	Asp	Glu	Gln	His	Arg	Lys	Met	Ile	Leu	Val	Phe	Tyr	Trp
C	Cys	12																			
S	Ser	0	2																		
T	Thr	-2	1	3																	
P	Pro	-3	1	0	6																
A	Ala	-2	1	1	1	2															
G	Gly	-3	1	0	-1	1	5														
N	Asn	-4	1	0	-1	0	0	2													
D	Asp	-5	0	0	-1	0	1	2	4												
E	Glu	-5	0	0	-1	0	0	1	3	4											
Q	Gln	-5	-1	-1	0	0	-1	1	2	2	4										
H	His	-3	-1	-1	0	-1	-2	2	1	1	3	6									
R	Arg	-4	0	-1	0	-2	-3	0	-1	-1	1	2	6								
K	Lys	-5	0	0	-1	-1	-2	1	0	0	1	0	3	5							
M	Met	-5	-2	-1	-2	-1	-3	-2	-3	-2	-1	-2	0	0	6						
I	Ile	-2	-1	0	-2	-1	-3	-2	-2	-2	-2	-2	-2	-2	2	5					
L	Leu	-6	-3	-2	-3	-2	-4	-3	-4	-3	-2	-2	-3	-3	4	2	6				
V	Val	-2	-1	0	-1	0	-1	-2	-2	-2	-2	-2	-2	-2	2	4	2	4			
F	Phe	-4	-3	-3	-5	-4	-5	-4	-6	-5	-5	-2	-4	-5	0	1	2	-1	9		
Y	Tyr	0	-3	-3	-5	-3	-5	-2	-4	-4	-4	0	-4	-4	-2	-1	-1	-2	7	10	
W	Trp	-8	-2	-5	-6	-6	-7	-4	-7	-7	-5	-3	2	-3	-4	-5	-2	-6	0	0	17

BLOSUM

Less divergent  More divergent

BLOSUM 80	BLOSUM 62	BLOSUM 45
PAM 1	PAM 120	PAM 250

How to let computer do this job?

- How to measure similarity?
  - Similarity matrix
- How to find out alignment?
  - Dot matrix
  - Dynamic programming
  - BLAST

Dot Matrix

ATAGCTA

ATAGCTA

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

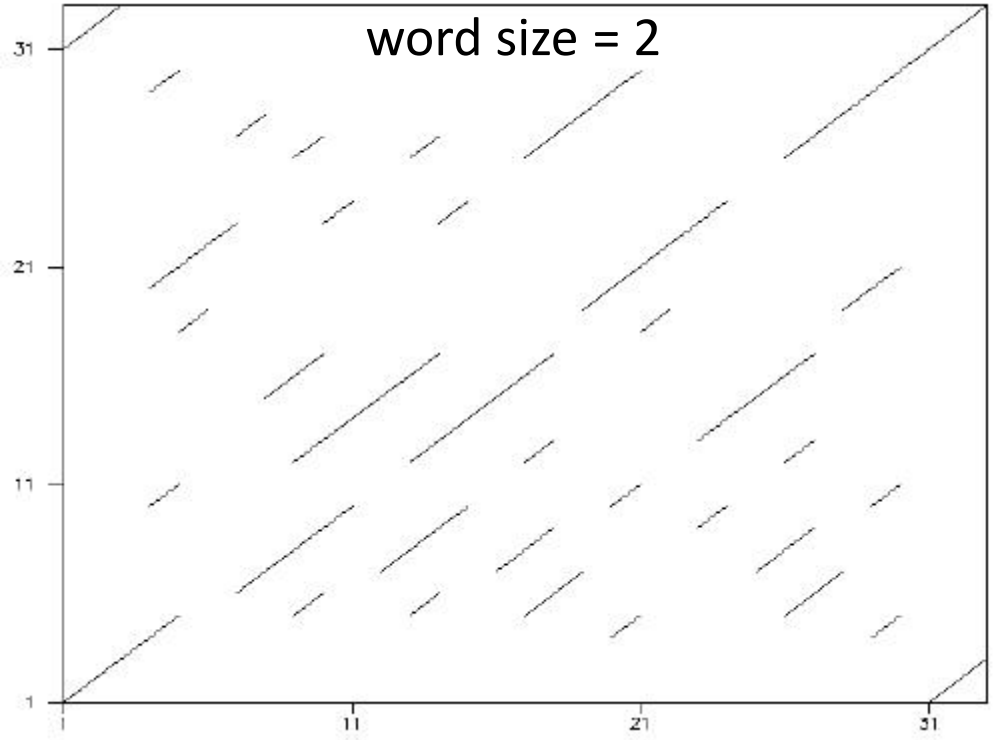
ATAGC-TA

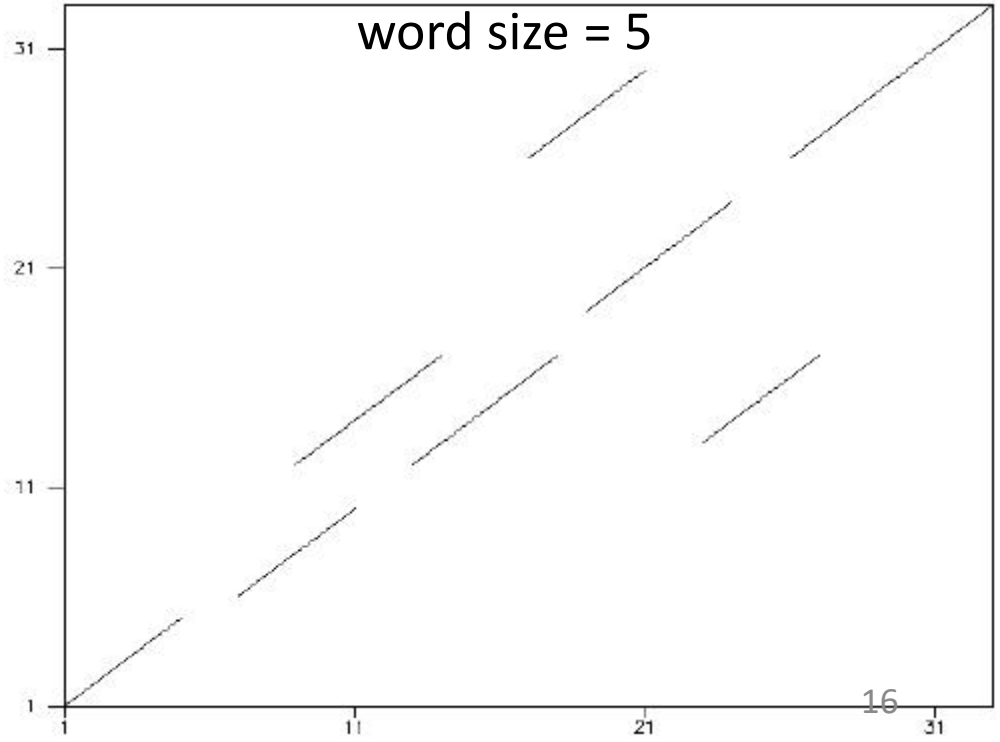
AT-GCCTA

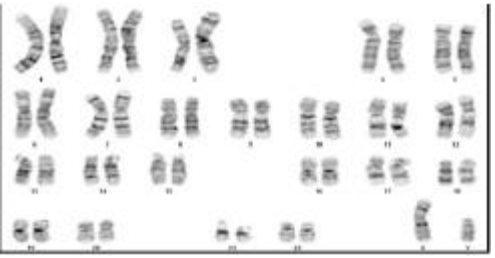
	A	T	A	G	C	T	A
A	1		1				1
T		1				1	
G				1			
C					1		
C					1		
T		1				1	
A	1		1				1

Dot Matrix

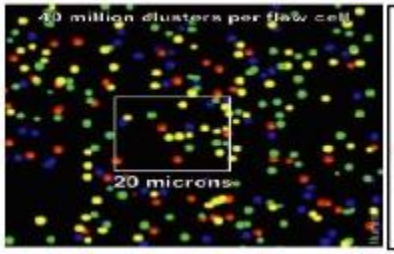
- PQRACGTGCTAGCTAGCT-GACGTAGCTGACPQR
- PQRAC-TGCTACCTAGCTCGACGTATCTGACPQR

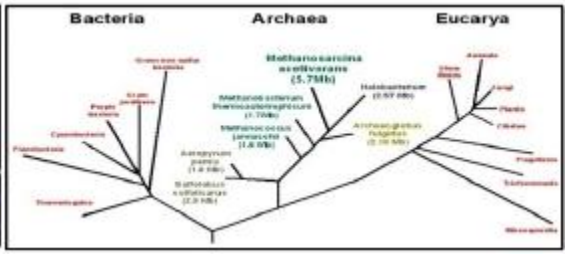






TAACCCTAACCCTAACCCTAACCCTAACCCTA
CCTAACCCTAACCCTAACCCTAACCCTAACCC
CCCTAACCCCTAACCCTAACCCTAACCCTAAC
AACCCTAACCCTAACCCTAACCCCTAACCCTA
ACCCTAACCCCAACCCCAACCCCAACCCCAAC
CTACCCTAACCCTAACCCTAACCCTAACCCTA
ACCCTAACCCTAACCCTAACCCCTAACCCTAA





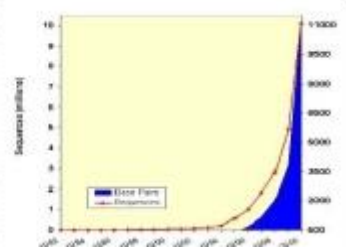
Unit 5:

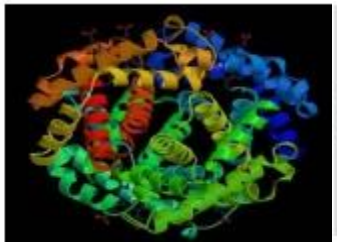
Global alignment vsLocal alignment

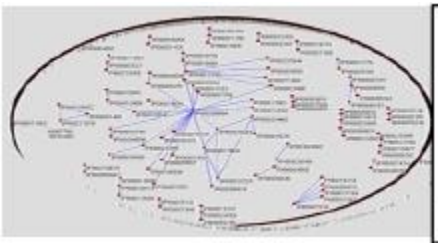
Kun Lu

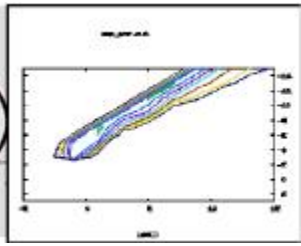
Computer Science Department

Southwest University









Contents

- Background
- Needleman-Wunsch global alignment
- Scoring function and gap penalty
- Alignment significance evaluation
- Smith-Waterman local alignment
- Comparisions between the two algorithms
- References

Background

- Why alignment?

  “Nothing in Biology Makes Sense Except in the Light of Evolution”

  ——T.Dobzhansky,1973

- Why computer arithmetic?

- Why dynamic programming?

Needleman-Wunsch Algorithm

	A	B	C	N	J	R	O	C	L	C	R	P	M
A													
J													
C													
J													
N													
R													
C													
K													
C													
R													
B													
P													

(modified from fig.1 of Needleman&Wunsch,1970)

Needleman-Wunsch Algorithm

	A	B	C	N	J	R	O	C	L	C	R	P	M
A	1	0	0	0	0	0	0	0	0	0	0	0	0
J	0	0	0	0	1	0	0	0	0	0	0	0	0
C	0	0	1	0	0	0	0	1	0	1	0	0	0
J	0	0	0	0	1	0	0	0	0	0	0	0	0
N	0	0	0	1	0	0	0	0	0	0	0	0	0
R	0	0	0	0	0	1	0	0	0	0	1	0	0
C	0	0	1	0	0	0	0	1	0	1	0	0	0
K	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	1	0	0	0	0	1	0	1	0	0	0
R	0	0	0	0	0	1	0	0	0	0	1	0	0
B	0	1	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0	1	0

(modified from fig.1 of Needleman&Wunsch,1970)

Needleman-Wunsch Algorithm

	A	B	C	N	J	R	O	C	L	C	R	P	M
A	1												
J					1								
C			1					1		1			
J					1								
N				1									
R						1					1		
C			1					1		1			
K													
C			1					1		1			
R						1					1		
B		1											
P												1	

(modified from fig.1 of Needleman&Wunsch,1970)

$$F_{ij} = \max_{h<i,k<j} \{F_{h,j-1} + S(A_i, B_j), F_{i-1,k} + S(A_i, B_j)\}$$

(source:http://en.wikipedia.org/wiki/Needleman–Wunsch_algorithm)

Needleman-Wunsch Algorithm

	A	B	C	N	J	R	O	C	L	C	R	P	M
A	1												
J					1								
C			1					1		1			
J					1								
N				1									
R						1					1		
C			1					1		1			
K													
C			1					1		1			
R						1					1		
B		1											
P	0	0	0	0	0	0	0	0	0	0	0	1	0

(modified from fig.1 of Needleman&Wunsch,1970)

$$F_{ij} = \max_{h<i,k<j} \{F_{h,j-1} + S(A_i, B_j), F_{i-1,k} + S(A_i, B_j)\}$$

(source:http://en.wikipedia.org/wiki/Needleman–Wunsch_algorithm)

Needleman-Wunsch Algorithm

	A	B	C	N	J	R	O	C	L	C	R	P	M
A	1												
J					1								
C			1					1		1			
J					1								
N				1									
R						1					1		
C			1					1		1			
K													
C			1					1		1			
R						1					1		
B	1	2	1	1	1	1	1	1	1	1	1	0	0
P	0	0	0	0	0	0	0	0	0	0	0	1	0

(modified from fig.1 of Needleman&Wunsch,1970)

$$F_{ij} = \max_{h<i,k<j} \{F_{h,j-1} + S(A_i, B_j), F_{i-1,k} + S(A_i, B_j)\}$$

(source:http://en.wikipedia.org/wiki/Needleman–Wunsch_algorithm)

Needleman-Wunsch Algorithm

	A	B	C	N	J	R	O	C	L	C	R	P	M
A	1												
J					1								
C			1					1		1			
J					1								
N				1									
R						1	4	3	3	2	2	0	0
C	3	3	4	3	3	3	3	4	3	3	1	0	0
K	3	3	3	3	3	3	3	3	3	2	1	0	0
C	2	2	3	2	2	2	2	3	2	3	1	0	0
R	2	1	1	1	1	2	1	1	1	1	2	0	0
B	1	2	1	1	1	1	1	1	1	1	1	0	0
P	0	0	0	0	0	0	0	0	0	0	0	1	0

(modified from fig.1 of Needleman&Wunsch,1970)

$$F_{ij} = \max_{h<i,k<j} \{F_{h,j-1} + S(A_i, B_j), F_{i-1,k} + S(A_i, B_j)\}$$

(source:http://en.wikipedia.org/wiki/Needleman–Wunsch_algorithm)

Needleman-Wunsch Algorithm

	A	B	C	N	J	R	O	C	L	C	R	P	M
A	1												
J					1								
C			1					1		1			
J					1								
N				1									
R						1	4	3	3	2	2	0	0
C	3	3	4	3	3	3	3	4	3	3	1	0	0
K	3	3	3	3	3	3	3	3	3	2	1	0	0
C	2	2	3	2	2	2	2	3	2	3	1	0	0
R	2	1	1	1	1	2	1	1	1	1	2	0	0
B	1	2	1	1	1	1	1	1	1	1	1	0	0
P	0	0	0	0	0	0	0	0	0	0	0	1	0

(modified from fig.1 of Needleman&Wunsch,1970)

$$F_{ij} = \max_{h<i, k<j} \{F_{h,j-1} + S(A_i, B_j), F_{i-1,k} + S(A_i, B_j)\}$$

(source:http://en.wikipedia.org/wiki/Needleman–Wunsch_algorithm)

Needleman-Wunsch Algorithm

	A	B	C	N	J	R	O	C	L	C	R	P	M
A	1												
J					1								
C			1					1		1			
J					1								
N				1									
R						1	4	3	3	2	2	0	0
C	3	3	4	3	3	3	3	4	3	3	1	0	0
K	3	3	3	3	3	3	3	3	3	2	1	0	0
C	2	2	3	2	2	2	2	3	2	3	1	0	0
R	2	1	1	1	1	2	1	1	1	1	2	0	0
B	1	2	1	1	1	1	1	1	1	1	1	0	0
P	0	0	0	0	0	0	0	0	0	0	0	1	0

(modified from fig.1 of Needleman&Wunsch,1970)

$$F_{ij} = \max_{h<i,k<j} \{F_{h,j-1} + S(A_i, B_j), F_{i-1,k} + S(A_i, B_j)\}$$

(source:http://en.wikipedia.org/wiki/Needleman–Wunsch_algorithm)

Needleman-Wunsch Algorithm

	A	B	C	N	J	R	O	C	L	C	R	P	M
A	1												
J					1								
C			1					1		1			
J					1								
N				1									
R						5	4	3	3	2	2	0	0
C	3	3	4	3	3	3	3	4	3	3	1	0	0
K	3	3	3	3	3	3	3	3	3	2	1	0	0
C	2	2	3	2	2	2	2	3	2	3	1	0	0
R	2	1	1	1	1	2	1	1	1	1	2	0	0
B	1	2	1	1	1	1	1	1	1	1	1	0	0
P	0	0	0	0	0	0	0	0	0	0	0	1	0

(modified from fig.1 of Needleman&Wunsch,1970)

$$F_{ij} = \max_{h<i, k<j} \{F_{h,j-1} + S(A_i, B_j), F_{i-1,k} + S(A_i, B_j)\}$$

(source:http://en.wikipedia.org/wiki/Needleman–Wunsch_algorithm)

Needleman-Wunsch Algorithm

	A	B	C	N	J	R	O	C	L	C	R	P	M
A	8	7	6	6	5	4	4	3	3	2	1	0	0
J	7	7	6	6	6	4	4	3	3	2	1	0	0
C	6	6	7	6	5	4	4	4	3	3	1	0	0
J	6	6	6	5	6	4	4	3	3	2	1	0	0
N	5	5	5	6	5	4	4	3	3	2	1	0	0
R	4	4	4	4	4	5	4	3	3	2	2	0	0
C	3	3	4	3	3	3	3	4	3	3	1	0	0
K	3	3	3	3	3	3	3	3	3	2	1	0	0
C	2	2	3	2	2	2	2	3	2	3	1	0	0
R	2	1	1	1	1	2	1	1	1	1	2	0	0
B	1	2	1	1	1	1	1	1	1	1	1	0	0
P	0	0	0	0	0	0	0	0	0	0	0	1	0

(modified from fig.1 of Needleman&Wunsch,1970)

$$F_{ij} = \max_{h<i,k<j} \{F_{h,j-1} + S(A_i, B_j), F_{i-1,k} + S(A_i, B_j)\}$$

(source:http://en.wikipedia.org/wiki/Needleman–Wunsch_algorithm)

Needleman-Wunsch Algorithm

	A	B	C	N	J	R	O	C	L	C	R	P	M
A	8	7	6	6	5	4	4	3	3	2	1	0	0
J	7	7	6	6	6	4	4	3	3	2	1	0	0
C	6	6	7	6	5	4	4	4	3	3	1	0	0
J	6	6	6	5	6	4	4	3	3	2	1	0	0
N	5	5	5	6	5	4	4	3	3	2	1	0	0
R	4	4	4	4	4	5	4	3	3	2	2	0	0
C	3	3	4	3	3	3	3	4	3	3	1	0	0
K	3	3	3	3	3	3	3	3	3	2	1	0	0
C	2	2	3	2	2	2	2	3	2	3	1	0	0
R	2	1	1	1	1	2	1	1	1	1	2	0	0
B	1	2	1	1	1	1	1	1	1	1	1	0	0
P	0	0	0	0	0	0	0	0	0	0	0	1	0

(modified from fig.1 of Needleman&Wunsch,1970)

$$F_{ij} = \max_{h<i, k<j} \{F_{h,j-1} + S(A_i, B_j), F_{i-1,k} + S(A_i, B_j)\}$$

(source:http://en.wikipedia.org/wiki/Needleman–Wunsch_algorithm)

Needleman-Wunsch Algorithm

	A	B	C	N	J	R	O	C	L	C	R	P	M
A	8	7	6	6	5	4	4	3	3	2	1	0	0
J	7	7	6	6	6	4	4	3	3	2	1	0	0
C	6	6	7	6	5	4	4	4	3	3	1	0	0
J	6	6	6	5	6	4	4	3	3	2	1	0	0
N	5	5	5	6	5	4	4	3	3	2	1	0	0
R	4	4	4	4	4	5	4	3	3	2	2	0	0
C	3	3	4	3	3	3	3	4	3	3	1	0	0
K	3	3	3	3	3	3	3	3	3	2	1	0	0
C	2	2	3	2	2	2	2	3	2	3	1	0	0
R	2	1	1	1	1	2	1	1	1	1	2	0	0
B	1	2	1	1	1	1	1	1	1	1	1	0	0
P	0	0	0	0	0	0	0	0	0	0	0	1	0

(modified from fig.1 of Needleman&Wunsch,1970)

$$F_{ij} = \max_{h<i,k<j} \{F_{h,j-1} + S(A_i, B_j), F_{i-1,k} + S(A_i, B_j)\}$$

(source:http://en.wikipedia.org/wiki/Needleman–Wunsch_algorithm)

Needleman-Wunsch Algorithm

	A	B	C	N	J	R	O	C	L	C	R	P	M
A	8	7	6	6	5	4	4	3	3	2	1	0	0
J	7	7	6	6	6	4	4	3	3	2	1	0	0
C	6	6	7	6	5	4	4	4	3	3	1	0	0
J	6	6	6	5	6	4	4	3	3	2	1	0	0
N	5	5	5	6	5	4	4	3	3	2	1	0	0
R	4	4	4	4	4	5	4	3	3	2	2	0	0
C	3	3	4	3	3	3	3	4	3	3	1	0	0
K	3	3	3	3	3	3	3	3	3	2	1	0	0
C	2	2	3	2	2	2	2	3	2	3	1	0	0
R	2	1	1	1	1	2	1	1	1	1	2	0	0
B	1	2	1	1	1	1	1	1	1	1	1	0	0
P	0	0	0	0	0	0	0	0	0	0	0	1	0

(modified from fig.1 of Needleman&Wunsch,1970)

$$F_{ij} = \max_{h<i,k<j} \{F_{h,j-1} + S(A_i, B_j), F_{i-1,k} + S(A_i, B_j)\}$$

(source:http://en.wikipedia.org/wiki/Needleman–Wunsch_algorithm)

Needleman-Wunsch Algorithm

	A	B	C	N	J	R	O	C	L	C	R	P	M
A	8	7	6	6	5	4	4	3	3	2	1	0	0
J	7	7	6	6	6	4	4	3	3	2	1	0	0
C	6	6	7	6	5	4	4	4	3	3	1	0	0
J	6	6	6	5	6	4	4	3	3	2	1	0	0
N	5	5	5	6	5	4	4	3	3	2	1	0	0
R	4	4	4	4	4	5	4	3	3	2	2	0	0
C	3	3	4	3	3	3	3	4	3	3	1	0	0
K	3	3	3	3	3	3	3	3	3	2	1	0	0
C	2	2	3	2	2	2	2	3	2	3	1	0	0
R	2	1	1	1	1	2	1	1	1	1	2	0	0
B	1	2	1	1	1	1	1	1	1	1	1	0	0
P	0	0	0	0	0	0	0	0	0	0	0	1	0

(modified from fig.1 of Needleman&Wunsch,1970)

$$F_{ij} = \max_{h<i, k<j} \{F_{h,j-1} + S(A_i, B_j), F_{i-1,k} + S(A_i, B_j)\}$$

(source:http://en.wikipedia.org/wiki/Needleman–Wunsch_algorithm)

Needleman-Wunsch Algorithm

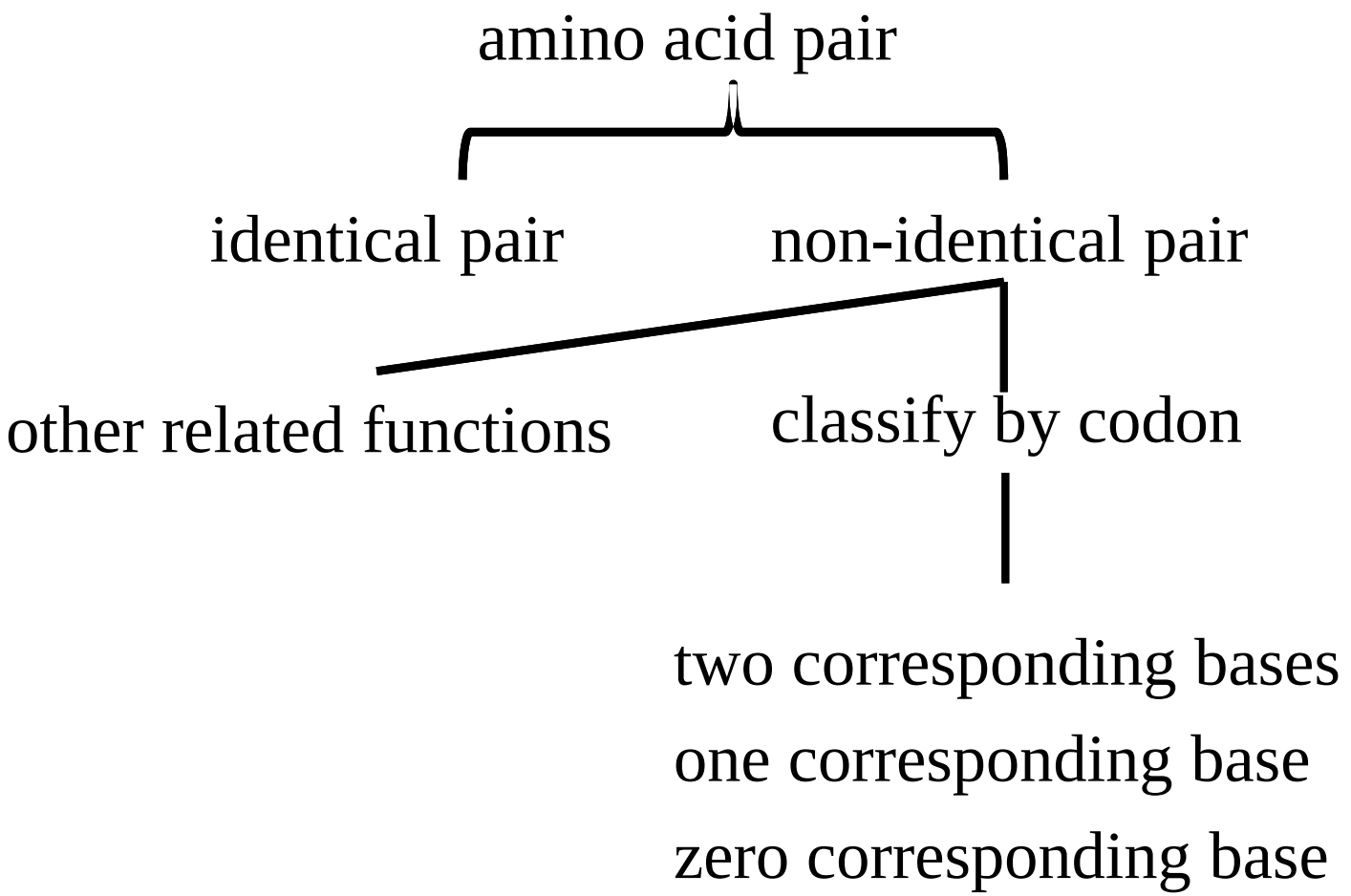
	A	B	C	N	J	R	O	C	L	C	R	P	M
A	8	7	6	6	5	4	4	3	3	2	1	0	0
J	7	7	6	6	6	4	4	3	3	2	1	0	0
C	6	6	7	6	5	4	4	4	3	3	1	0	0
J	6	6	6	5	6	4	4	3	3	2	1	0	0
N	5	5	5	6	5	4	4	3	3	2	1	0	0
R	4	4	4	4	4	5	4	3	3	2	2	0	0
C	3	3	4	3	3	3	3	4	3	3	1	0	0
K	3	3	3	3	3	3	3	3	3	2	1	0	0
C	2	2	3	2	2	2	2	3	2	3	1	0	0
R	2	1	1	1	1	2	1	1	1	1	2	0	0
B	1	2	1	1	1	1	1	1	1	1	1	0	0
P	0	0	0	0	0	0	0	0	0	0	0	1	0

(modified from fig.1 of Needleman&Wunsch,1970)

$$F_{ij} = \max_{h<i,k<j} \{F_{h,j-1} + S(A_i, B_j), F_{i-1,k} + S(A_i, B_j)\}$$

(source:http://en.wikipedia.org/wiki/Needleman–Wunsch_algorithm)

Scoring function and Gap Penalty



Gap penalty: avoid too many gaps

Scoring function and Gap Penalty

- Needleman-Wunsch Algorithm

$$F_{i0} = d * i$$

$$F_{0j} = d * j$$

$$F_{ij} = \max(F_{i-1,j-1} + S(A_i, B_j),\ F_{i,j-1} + d,\ F_{i-1,j} + d)$$

$$F_{ij} = \max_{h<i, k<j} \{F_{h,j-1} + S(A_i, B_j), F_{i-1,k} + S(A_i, B_j)\}$$

(source: http://en.wikipedia.org/wiki/Needleman–Wunsch_algorithm)

Scoring function and Gap Penalty

	C	A	G	C	C	U	C	G	C	U	U	A	G
A													
A													
U													
G													
C													
C													
A													
U													
U													
G													
A													
C													
G													
G													

Scoring function and Gap Penalty

	C	A	G	C	C	U	C	G	C	U	U	A	G
A	8	8	7	7	7	6	6	5	4	3	2	2	0
A	7	8	7	7	7	6	6	5	4	3	2	2	0
U	7	7	6	6	6	7	6	5	4	4	3	1	0
G	6	6	7	6	6	6	5	6	4	3	2	1	1
C	6	5	5	6	6	5	6	5	5	3	2	1	0
C	6	4	4	5	5	4	5	4	5	3	2	1	0
A	4	5	4	4	4	4	4	4	4	3	2	2	0
U	4	4	4	4	4	4	3	3	3	4	3	1	0
U	4	4	3	3	3	4	3	2	2	3	3	1	0
G	4	3	4	3	3	3	2	3	2	2	2	1	1
A	3	4	3	3	3	3	2	2	1	1	1	2	0
C	3	2	2	3	3	2	3	1	2	1	1	1	0
G	1	1	2	1	1	1	1	2	1	1	1	1	1
G	0	0	1	0	0	0	0	1	0	0	0	0	1

Scoring function and Gap Penalty

Pairwise Alignment Result

LENGTH	SCORE	IDENTITY	SIMILARITY	GAPS
19	40.0	8/19 (42.1%)	8/19 (42.1%)	11/19 (57.9%)

a

```
Ned_Seq_2    1 -AAUGCCAUUGACG----G    14
                 |  |||  |  ||     |
Ned_Seq_1    1 CA--GCC--U--CGCUUAG    13
```

Pairwise Alignment Result

LENGTH	SCORE	IDENTITY	SIMILARITY	GAPS
19	40.0	8/19 (42.1%)	8/19 (42.1%)	11/19 (57.9%)

b

```
Ned_Seq_1    1 CA--GCCUCGC-UU-A--G    13
                 |  |    | | || |  |
Ned_Seq_2    1 -AAUG---C-CAUUGACGG    14
```

Needleman algorithm (simplified)

Scoring function and Gap Penalty

Pairwise Alignment Result

LENGTH	SCORE	IDENTITY	SIMILARITY	GAPS
19	34.0	8/19 (42.1%)	8/19 (42.1%)	11/19 (57.9%)

c

```
Ned_Seq_2      1 -AAUGCCAUUGACG----G      14
                   |  |||  |  ||    |
Ned_Seq_1      1 CA--GCC--U--CGCUUAG      13
```

Pairwise Alignment Result

LENGTH	SCORE	IDENTITY	SIMILARITY	GAPS
19	34.0	8/19 (42.1%)	8/19 (42.1%)	11/19 (57.9%)

d

```
Ned_Seq_1      1 CA--GCC--U--CGCUUAG      13
                   |  |||  |  ||    |
Ned_Seq_2      1 -AAUGCCAUUGACG----G      14
```

Needleman algorithm

Alignment significance evaluation

- Question:

Whether a particular result differs significance from a fortuitous match between two random sequences ?

- To answer this question we can do:

  Sequence alignment between two sets of random sequences

  Or:

  Sequence alignment between one set of random sequences and a real sequence

Smith-Waterman Algorithm

$$H_{k0} = H_{0l} = 0 \text{ for } 0 \leq k \leq n \text{ and } 0 \leq l \leq m.$$

$$H_{ij} = \max\{H_{i-1,j-1} + s(a_i, b_j), \max_{k \geq 1}\{H_{i-k,j} - W_k\}, \max_{l \geq 1}\{H_{i,j-l} - W_l\}, 0\}. \quad (1)$$

$1 \leq i \leq n$ and $1 \leq j \leq m$.

$$W_k = 1{\cdot}0 + 1/3{*}k.$$

(source: Smith & Waterman,1981)

	gap	a1	a2	a3	a4
gap	0	1 gap	2 gaps	3 gaps	4 gaps
b1	1 gap	A11	A21	A31	A41
b2	2 gaps	A12	A22	A32	A42
b3	3 gaps	A13	A23	A33	A43

(modified from Fig.3.9 of D.W.Mount Bioinformatics Sequence and Gene Analysis, Second Eition, P74)

Smith-Waterman Algorithm

$$H_{k0} = H_{0l} = 0 \text{ for } 0 \leq k \leq n \text{ and } 0 \leq l \leq m.$$

$$H_{ij} = \max\{H_{i-1,j-1} + s(a_i, b_j), \max_{k \geq 1}\{H_{i-k,j} - W_k\}, \max_{l \geq 1}\{H_{i,j-l} - W_l\}, 0\}. \quad (1)$$

$1 \leq i \leq n$ and $1 \leq j \leq m$.

$$W_k = 1.0 + 1/3 * k.$$

(source: Smith & Waterman,1981)

	gap	a1	a2	a3	a4
gap	0	1 gap	2 gaps	3 gaps	4 gaps
b1	1 gap	A11	A21	A31	A41
b2	2 gaps	A12	A22	A32	A42
b3	3 gaps	A13	A23	A33	A43

(modified from Fig.3.9 of D.W.Mount Bioinformatics Sequence and Gene Analysis, Second Eition, P74)

Smith-Waterman Algorithm

$$H_{k0} = H_{0l} = 0 \text{ for } 0 \leq k \leq n \text{ and } 0 \leq l \leq m.$$

$$H_{ij} = \max\{H_{i-1,j-1} + s(a_i, b_j), \max_{k \geq 1}\{H_{i-k,j} - W_k\}, \max_{l \geq 1}\{H_{i,j-l} - W_l\}, 0\}. \quad (1)$$

$$1 \leq i \leq n \text{ and } 1 \leq j \leq m.$$

$$W_k = 1.0 + 1/3 * k.$$

(source: Smith & Waterman,1981)

	gap	a1	a2	a3	a4
gap	0	1 gap	2 gaps	3 gaps	4 gaps
b1	1 gap	A11	A21	A31	A41
b2	2 gaps	A12	A22	A32	A42
b3	3 gaps	A13	A23	A33	A43

(modified from Fig.3.9 of D.W.Mount Bioinformatics Sequence and Gene Analysis, Second Eition, P74)

Smith-Waterman Algorithm

$$H(i,0) = 0,\ 0 \le i \le m$$
$$H(0,j) = 0,\ 0 \le j \le n$$
if $a_i = b_j$ then $w(a_i, b_j) = w(\text{match})$ or if $a_i \neq b_j$ then $w(a_i, b_j) = w(\text{mismatch})$

$$H(i,j) = \max \left\{ \begin{array}{ll} 0 & \\ H(i-1,j-1) + w(a_i, b_j) & \text{Match/Mismatch} \\ H(i-1,j) + w(a_i, -) & \text{Deletion} \\ H(i,j-1) + w(-, b_j) & \text{Insertion} \end{array} \right\},\ 1 \le i \le m, 1 \le j \le n$$

(source: http://en.wikipedia.org/wiki/Smith-Waterman_algorithm)

	gap	a1	a2	a3	a4
gap	0	1 gap	2 gaps	3 gaps	4 gaps
b1	1 gap	A11	A21	A31	A41
b2	2 gaps	A12	A22	A32	A42
b3	3 gaps	A13	A23	A33	A43

(modified from Fig.3.9 of D.W.Mount Bioinformatics Sequence and Gene Analysis, Second Eition, P74)

Smith-Waterman Algorithm

- Compared with global alignment：
  - Zero could terminate the current local alignment
  - Mismatch must be negative scored
- Other properties：
  - Suitable to identify conserved local sequence (substring)
  - Guaranteed to find the best local alignment
  - Perform poorly when dealing with separated regions within a long sequence

Smith-Waterman Algorithm

Pairwise Alignment Result

LENGTH	SCORE	IDENTITY	SIMILARITY	GAPS
17	34.333	8/17 (47.1%)	8/17 (47.1%)	9/17 (52.9%)

```
Ned_Seq_1        2 A-GCC--U--CGCUUAG      13
                   | |||  |  ||    |
Ned_Seq_2        2 AUGCCAUUGACG----G      14
```

Pairwise Alignment Result

LENGTH	SCORE	IDENTITY	SIMILARITY	GAPS
17	34.333	8/17 (47.1%)	8/17 (47.1%)	9/17 (52.9%)

```
Ned_Seq_2        2 AUGCCAUUGACG----G      14
                   | |||  |  ||    |
Ned_Seq_1        2 A-GCC--U--CGCUUAG      13
```

Comparision

Match	1
Mismatch	0
Gap	0

	A	K	C	A	C	K
C	2	2	3	1	1	0
A	1	1	1	2	0	0
C	0	0	1	0	1	0

Needleman-Wunsch (Simplified)

Match	1
Mismatch	-1
Gap	-1

		A	K	C	A	C	K
	0	-01	-02	-03	-04	-05	-06
C	-01	0	-01	-11	-02	-12	-03
A	-02	10	0	-01	20	-01	-02
C	-03	-01	0	1	0	31	10

Needleman-Wunsch Smith-Waterman

References

1. Needleman, S. B., & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3), 443-453.
2. Sellers, P. H. (1974). On the theory and computation of evolutionary distances.*SIAM Journal on Applied Mathematics*, *26*(4), 787-793..
3. Waterman, M. S., Smith, T. F., & Beyer, W. A. (1976). Some biological sequence metrics. *Advances in Mathematics*, 20(3), 367-387.
4. Dayhoff, M.O., Schwartz, R. and Orcutt, B.C. (1978). A model of Evolutionary Change in Proteins. *Atlas of protein sequence and structure* (volume 5, supplement 3 ed.). *Nat. Biomed. Res. Found*. 345–358.
5. Smith, T. F., & Waterman, M. S. (1981). Comparison of biosequences. *Advances in Applied Mathematics*, *2*(4), 482-489.
6. David, W. M.(2004).Bioinformatics: Sequence and Genome Analysis(Second Edition).*Cold Spring Harbor Laboratory Press*, 69-84.
7. http://en.wikipedia.org/wiki/Needleman–Wunsch_algorithm
8. http://en.wikipedia.org/wiki/Smith-Waterman_algorithm

Bioinformatics: Introduction and Methods

Computer Science Department, Southwest University

Thank you