

# Multiple sequence alignment

Irit Orr  
Shifra Ben-Dor

PAIRWISE ALIGNMENT

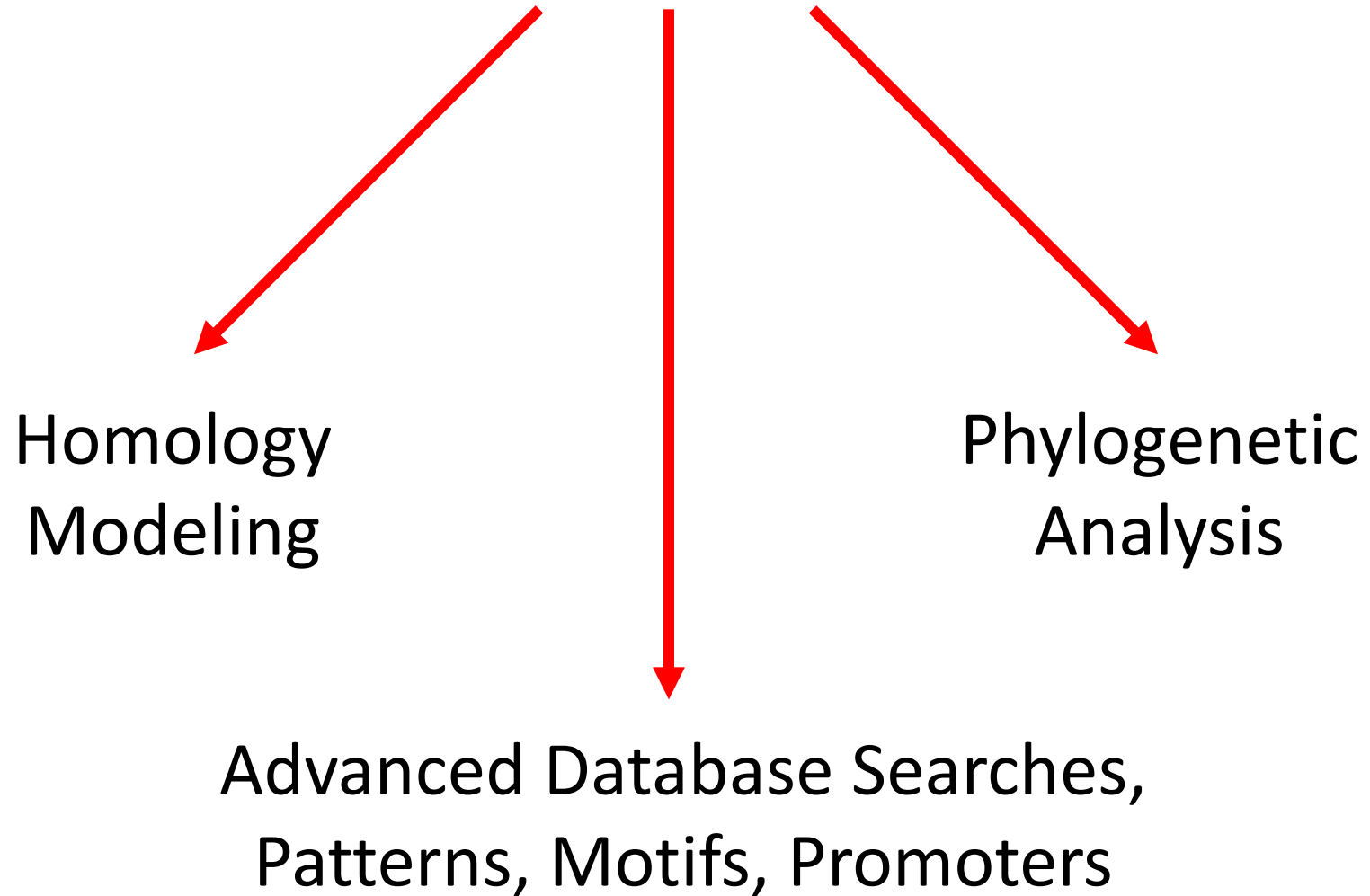


DATABASE SEARCHING



MULTIPLE ALIGNMENT

# MULTIPLE ALIGNMENT



# An example of Multiple Alignment

VTIS**C**TGSSSNIGAG-NHVK**W**YQ**Q**L**P**G  
VTIS**C**TGTSSNIGS--ITVN**W**YQ**Q**L**P**G  
LRLS**C**SSSSGFIFSS--YAMY**W**VR**Q**A**P**G  
LSLT**C**TVSGTSEFDD--YYST**W**VR**Q**P**P**G  
PEVT**C**VVVVDVSHEDPQVKFN**W**YVDG--  
ATLV**C**LISDFYPGA--VTVA**W**KADS--  
AALG**C**LVKDYFPEP--VTVS**W**NSG---  
VSLT**C**LVKGFYPSD--IAVE**W**WSNG--

# Why do we need multiple alignments?



---

Multiple alignment, whether made of DNA or protein sequences, can yield much more information than analysis of a single sequence (or even two).

When dealing with a new protein with unknown function, the presence of several domains similar to domains in other “known” sequences, can imply a similar structure or function.

# Why do we need multiple alignments?



---

In order to reveal the relationship between a group of sequences. (homology)

In order to characterize protein families - to identify conserved regions of a specific family, and locate its variable regions.

In order to retrieve information about domains or active sites. Similar regions may indicate similar functions.



# Why do we need multiple alignments?

---

To plan point mutations based upon highlighted regions of multiple alignments, either very similar or very different.

To build a family profile for use in a more sensitive database scan. Such a search can find new (more distant) members of the family.

Determination of the consensus sequence of several aligned sequences, for further analysis.



# Why do we need multiple alignments?

---

Planning probes in order to fish out distant members of a protein family.

Multiple alignments are used for protein modeling programs.

To help prediction of secondary and tertiary structures of new sequences.

Multiple alignments are input for constructing phylogenetic trees.





# The Computational Challenge of MSA

---

Finding optimal alignment between a group of sequences that include: matches, mismatches and gaps is very difficult.

For Pairwise Alignments, Dynamic Programming methods are used, but they are impractical with multiple alignments (too many calculations, too much CPU time).



# The Computational Challenge of MSA

---

The difficulties with aligning a group of sequences varies with the degree of similarity between the sequences.

A high degree of variation of the compared sequences means many alignments are possible.

Many possibilities – very hard to find “optimal” alignment.



# The Computational Challenge of MSA

---

Approximate methods are used instead of Dynamic programming methods.

Another computational challenge is placement and scoring of gaps in the aligned sequences.



# Approximate Methods

---

## Progressive global alignment:

Starts with the most similar sequences, and builds the alignment by adding the rest of the sequences.

## Iterative methods:

Starts by making initial alignments of small groups of sequences, and then revises the alignment for better results.



# Approximate Methods

---

Consistency based alignments

Alignment based on small conserved domains (or patterns), found in the same order within the aligned sequences.

Alignment based on statistical or probabilistic models of the sequences.

Phylogeny “aware”, Structure “aware” ...

# Various Multiple Alignment algorithms:

ClustalW/ $\Omega$

Muscle

T-Coffee

Mafft

Probcons

PRALINE

MultAlign

DiAlign

HMMER

Sate

PRANK



# Multiple Alignment

---

The most practical and widely used method for multiple alignment is progressive global alignment.

How does it work?

# Steps to create a multiple alignment



---

Pairwise comparisons of all sequences

Perform cluster analysis on the pairwise data to generate a hierarchy for alignment. This may be in the form of a binary tree or simple ordering tree.

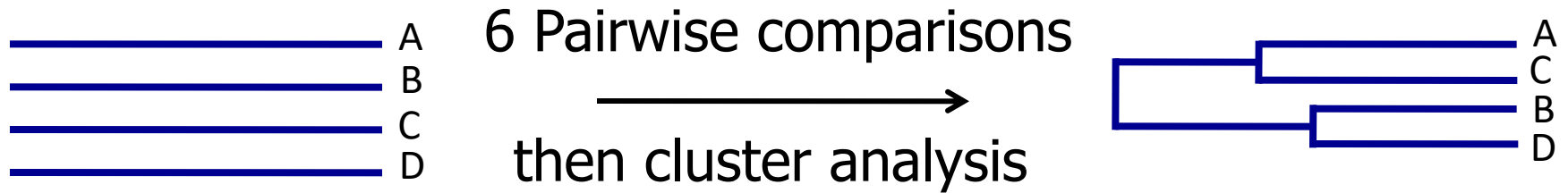
Start with the most related (similar) sequences, then the next most similar pair and so on.

Once an alignment of two sequences has been made, then this is fixed.



# Steps in Progressive Multiple Alignment

## 1) Pairwise Alignment



## 2) Multiple Alignment following the tree



New gap to optimize alignment of BD with AC

# Tips in choosing your sequences

## General considerations

---

Sequences taken directly from the database can contain irrelevant data, (e.g: multiple genes, fragments of different lengths). Check your sequences and use only the relevant parts of them for the alignment.

If you align your own sequences, edit them and remove the unrelated data before alignment.

Try to use sequences with more or less the same length for alignment.

# Tips in choosing your sequences

## General considerations

---

For most uses of multiple alignments:

The more sequences you align the better.

Don't include similar (>80%) sequences.

Sub-groups should be pre-aligned separately, and one member of each subgroup should be included in the final multiple alignment.



# What you need to know about multiple alignment programs

---

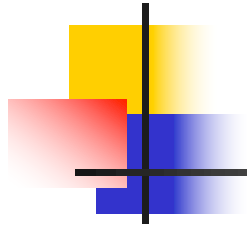
Almost all programs will align whatever sequences the user gives as input.

They will always return an alignment, even if the sequences are completely unrelated. The biology thinking should be done by you.

Most programs will insert gaps. However, if inserted they are there to stay.

You need to check how the program you use treats end gaps.

# ClustalW- for multiple alignment



ClustalW is a global multiple alignment program for DNA or protein.

ClustalW was produced by Julie D. Thompson, Toby Gibson of EMBL, Germany and Desmond Higgins of EBI, Cambridge, UK.

ClustalW is cited: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice. Nucleic Acids Research, 22:4673-4680.



# ClustalW- for multiple alignment

---

ClustalW can create multiple alignments, manipulate existing alignments and create phylogenetic trees.

The initial alignment can be done by 2 methods:

- slow/accurate
- fast/approximate



# ClustalW alignment Method

---

ClustalW alignment algorithm consists of 3 steps: Pairwise Alignments are performed between all sequences in the compared group. Alignment scores are used to build a distance matrix. In calculating the distance matrix, the program takes into account the divergence of the sequences.



# ClustalW alignment Method

---

A guide tree is created from the distance matrix using the Neighbor-Joining method.

This guide tree has branches of different lengths. Their length is proportional to the estimated divergence along each branch.





# ClustalW alignment Method

---

Progressive alignment of the sequences is done, following the branch order of the guide tree.

The sequences are aligned from the tips to the root.

The alignment of the sequences is guided by the phylogenetic relationships indicated by the tree.



# ClustalW alignment Method

---

At each stage of the progressive alignment full dynamic programming is applied, and uses a scoring matrix.

The program calculates sequence weights from the guide tree, and chooses the scoring matrix accordingly (according to the divergence of the compared sequences).

Clustalw calculates the genetic distances as follows:

# mismatches in the alignment



# matches in the alignment

Positions opposite a gap are not scored.



# ClustalW Alignment Method

---

ClustalW weights the sequences according to the distance of each sequence from the root.

ClustalW calculates gaps in a novel way, designed to place them between conserved domains.

ClustalW penalizes for gap opening and extension.



# Running ClustalW

---

The input file for ClustalW is a single file containing all of the sequences for alignment.

It accepts the following formats:

NBRF/PIR, EMBL/SwissProt, Pearson (Fasta), GDE, Clustal, GCG/MSF, RSF.

# Using ClustalW

\*\*\*\*\* MULTIPLE ALIGNMENT MENU \*\*\*\*\*

1. Do complete multiple alignment now (Slow/Accurate)
  2. Produce guide tree file only
  3. Do alignment using old guide tree file
  
  4. Toggle Slow/Fast pairwise alignments = SLOW
  
  5. Pairwise alignment parameters
  6. Multiple alignment parameters
  
  7. Reset gaps between alignments? = OFF
  8. Toggle screen display = ON
  9. Output format options
  
  - S. Execute a system command
  - H. HELP
- or press [RETURN] to go back to main menu

Your choice:

# ClustalW options

Your choice: 5

\*\*\*\*\* PAIRWISE ALIGNMENT PARAMETERS \*\*\*\*\*

Slow/Accurate alignments:

1. Gap Open Penalty :15.00
2. Gap Extension Penalty :6.66
3. Protein weight matrix :BLOSUM30
4. DNA weight matrix :IUB

Fast/Approximate alignments:

5. Gap penalty :5
6. K-tuple (word) size :2
7. No. of top diagonals :4
8. Window size :4

9. Toggle Slow/Fast pairwise alignments = SLOW

H. HELP

Enter number (or [RETURN] to exit):

# ClustalW options

Your choice: 6

\*\*\*\*\* MULTIPLE ALIGNMENT PARAMETERS \*\*\*\*\*

1. Gap Opening Penalty :15.00
2. Gap Extension Penalty :6.66
3. Delay divergent sequences :40 %
  
4. DNA Transitions Weight :0.50
  
5. Protein weight matrix :BLOSUM series
6. DNA weight matrix :IUB
7. Use negative matrix :OFF
  
8. Protein Gap Parameters

H. HELP

Enter number (or [RETURN] to exit):



# CLUSTAL W (2.012) multiple sequence alignment

```
M_MELB      VADYAEFQKNRHDQDATKRKLMEIANYVDKFYRSLNIR----IALVGLVWTHGDKCEVS
M_MELA      VADNREFQROGKDLEKVKQRLIEIANHVDFYRPLNIR----IVLVGVEVWNDIDKCSIS
M_MELG      VVDKERYDMMGRNQTAVREEMIRLANYLDSMYIMLNIR----IVLVGLEIWTDRNPINII
M_ADAM28    VLDNGEFKKYNKNLAEIRKIVLEMANYINMLYNKLDHAH----VALVGVEIWTGDGKIKIT
M_ADAM10    QTDHLFFKYYG-TREAVIAQISSHVKAIDTIYQTTFDFSGIRNISFMVKRIRINTTSDEKD
              *   :.           :   .: : : :*   :           :   ::   .:   .   .

M_MELB      ENPYSTLWSFLSWRR-KLLAQKSHDN---AQLITGRSFQGTIGLAPLMAMCSVY-----
M_MELA      QDPFTRLHEFLDWRKIKLLPRKSHDN---AQLISGVYFQGTIGMAPIMSMCTAE-----
M_MELG      GGAGDVLGNFVQWREKFLITRRRHDS---AQLVLKKGFGG-TAGMAFVGTVCSSRS-----
M_ADAM28    PDANTTLENFSKWRGNDLLKRKHDI---AQLISSTDFSGSTVGLAFMSSMCSPIY-----
M_ADAM10    PTNPFRFPNIGVEKFLELNSEQNHDDYCLAYVFTDRDFDDGVLGLAWVGAPSGSSGGICE
              :   .:   :   *   .: **   *   :.   *   .   .   *:*   :   :   .
```

# Problems with Progressive alignments

In progressive alignment the ultimate multiple alignment is dependent on the initial pairwise alignments.

The first sequences to be aligned are the most similar (closely related on the tree).

If the initial alignments are good, with very few errors, the ultimate multiple alignment will generally be good.

However, if the sequences aligned are distantly related, many more errors can be made, affecting the quality of the final alignment

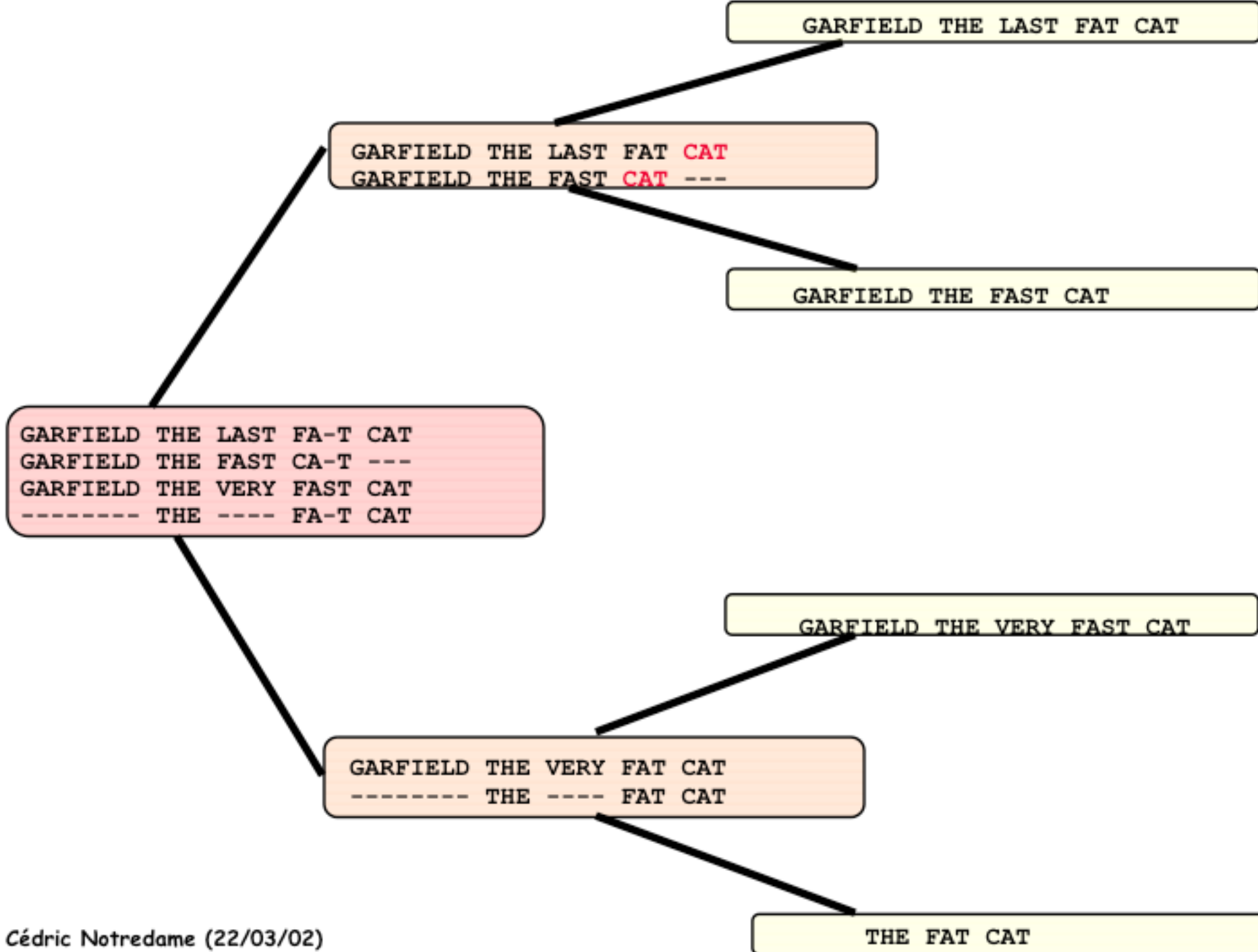
# Problems with progressive alignments

CLUSTALW (Score=20, Gop=-1, Gep=0, M=1)

SeqA	GARFIELD	THE	LAST	FA-T	CAT
SeqB	GARFIELD	THE	FAST	CA-T	---
SeqC	GARFIELD	THE	VERY	FAST	CAT
SeqD	-----	THE	----	FA-T	CAT

CORRECT (Score=24)

SeqA	GARFIELD	THE	LAST	FA-T	CAT
SeqB	GARFIELD	THE	FAST	----	CAT
SeqC	GARFIELD	THE	VERY	FAST	CAT
SeqD	-----	THE	----	FA-T	CAT





# Problems with Progressive alignments

---

Another problem with progressive alignment is that the ultimate multiple alignment is dependent on choosing the correct scoring matrices, and the correct gap penalty



# Muscle – Iterative alignment

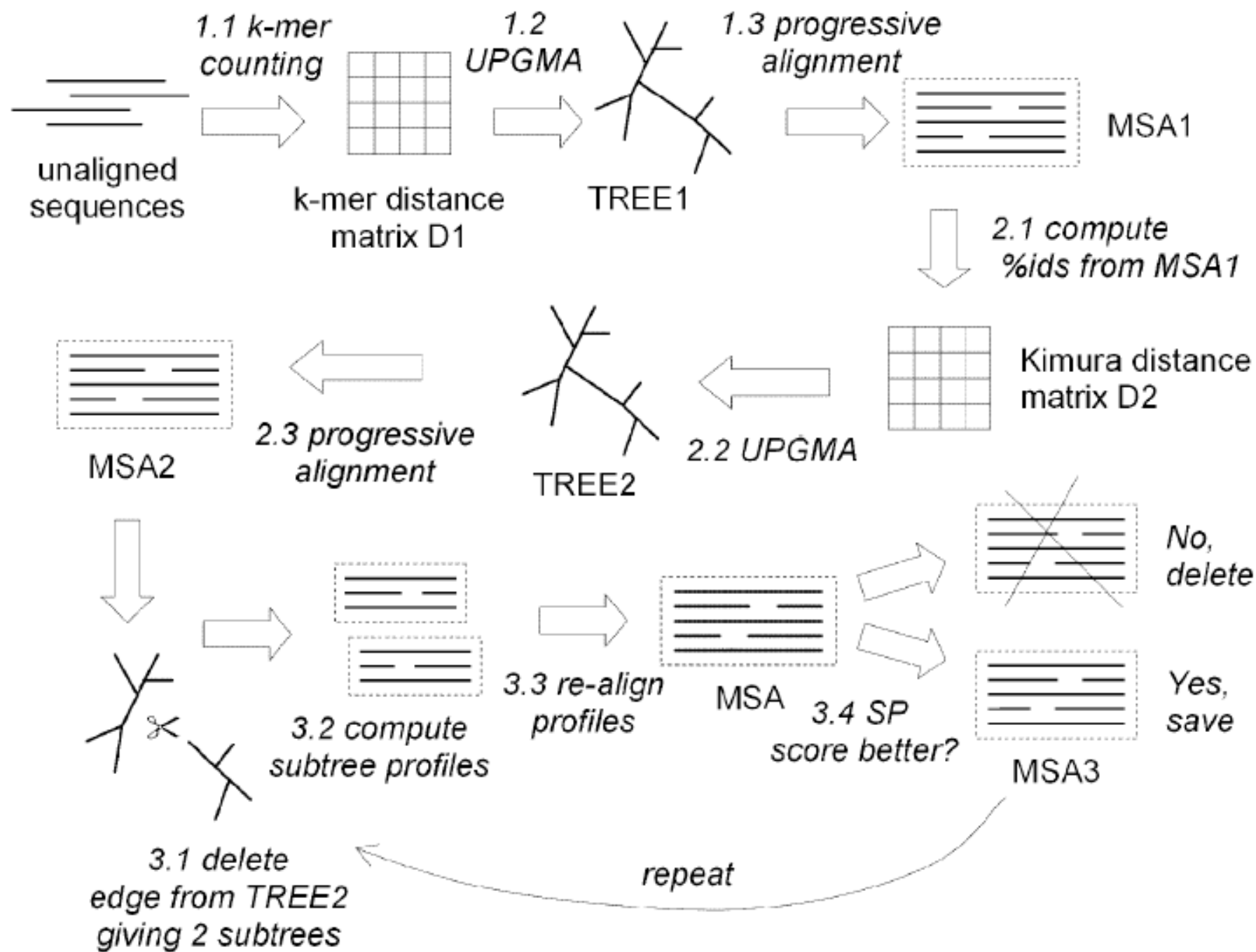
---

Muscle (Multiple Sequence Comparison by log-expectation) is cited:

Edgar, Robert C. (2004), MUSCLE: multiple sequence alignment with high accuracy and high throughput, Nucleic Acids Research 32(5), 1792-97.

Muscle on the WEB

<http://www.ebi.ac.uk/Tools/muscle/>



# Muscle first stage

## Draft Alignment

---

### Building the Guide Tree (“k-mer clustering”):

Calculates number of matching “words”, and calculates distances without doing alignments, builds a distance matrix, and then a tree (UPGMA)

### Progressive alignment

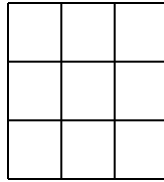
Follows the guide tree 1, from the tips to the root, and at each node aligns either 2 sequences, sequence/profile or profile/profile



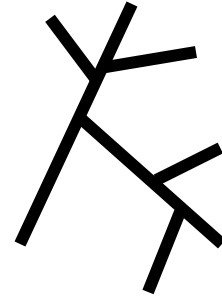
**Unaligned seqs**



**K-mer distance matrix**



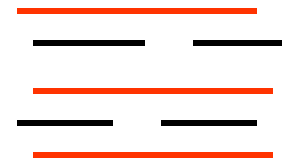
**Tree 1**



**Calculate K-mers**

**UPGMA**

**Progressive Alignment**



**First Multiple Alignment**



# Muscle Second stage Improved alignment

---

**Optimization** (“tree refinement”):

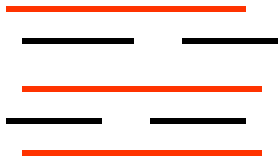
Using the multiple alignment as a base, compute pairwise identities for each of the sequence pairs.

**Build a distance matrix 2** (Kimura distance)

**Build a new tree (UPGMA).**

**Progressive alignment** is done following the guide tree 2, resulting in Multiple Alignment 2.

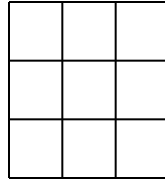
**First Multiple Alignment**



**Compute pairwise %**



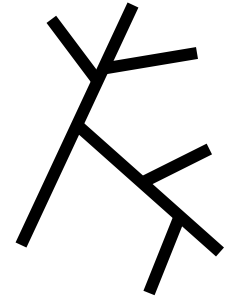
**Kimura distance matrix**



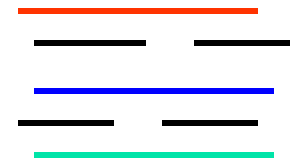
**UPGMA**



**Tree 2**



**Progressive Alignment**



**Second Multiple Alignment**

# Muscle Third stage

## Multiple Alignment Refinement



---

This tree is divided into 2 subtrees. (taking an edge off the tree to create the two groups)

The sequences in the subtree are used to build a multiple alignment and then a profile.

By realigning the 2 profiles a new multiple alignment is built.



# Muscle Third stage

## Multiple Alignment Refinement

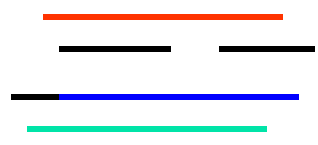
---

If this new alignment improves the score, it is kept. Otherwise it is discarded.

This is done for all the edges in the tree (from the edges to the root.)

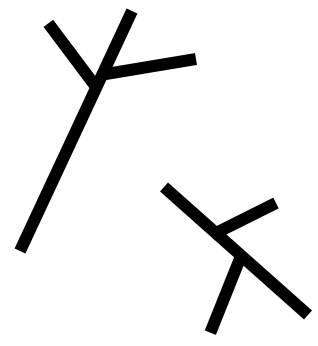
The whole step is iterated until convergence, or a user defined limit

# Second Multiple Alignment



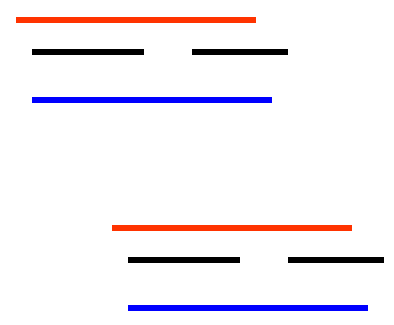
→  
Delete  
an edge

# subtree1



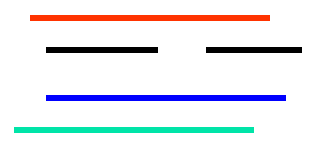
# subtree 2

# Compute subtree profile

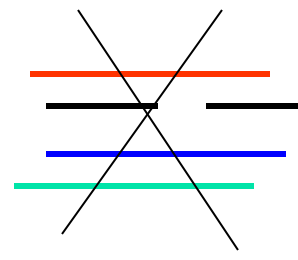


↓  
Realign  
profiles

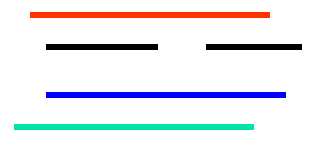
Better SP?  
Save



Not better?  
Delete



# Third Multiple Alignment





# Muscle Summary

---

- Fast
- Works with a large group of sequences
- Sequence length is not important

# T-Coffee - Consistency



---

- T-Coffee: A novel method for fast and accurate multiple sequence alignment. C. Notredame, D. Higgins, J. Heringa, Journal of Molecular Biology, Vol 302, pp205-217, 2000

- T-Coffee in the WEB

<http://www.tcoffee.org/>



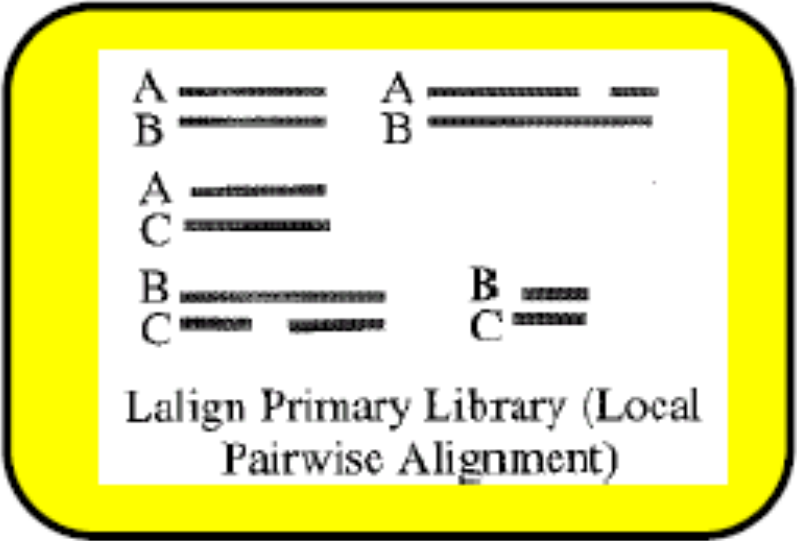
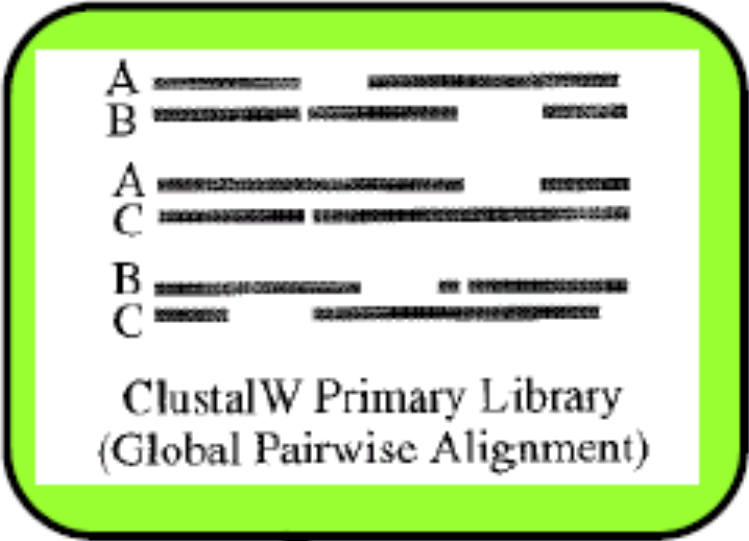


# T-Coffee first step: Creating the primary library

---

Builds a set of all pairwise alignments between all sequences in the dataset

- **Global** alignments of all against all using CLUSTALW
- **Local** alignments of all against all using LALIGN
- In the library – each alignment = a list of pairwise residue matches



Slides taken from  
[http://www.isrec.isb-sib.ch/DEA/module5/Course\\_Cedric/maln3.pdf](http://www.isrec.isb-sib.ch/DEA/module5/Course_Cedric/maln3.pdf)



# T-Coffee second step

---

- After the primary library was created, the program assigns a **WEIGHT** to each pair of aligned residues in the library
- For each set of sequences – 2 primary libraries are computed along with their weight: **Global + Local** alignments
- The library becomes **a list of weighted pairwise aligned scores.**



# T-Coffee third step

---

- Combination of the Global and Local weights to one Primary Library
- Checking the weighted pairs:
  - If the pair of seqs is duplicated (appears) in the 2 libraries, it is merged into a single entry with weight equal to the sum of the 2 libraries weights
  - Otherwise a new entry of this pair is created



# T-Coffee fourth step

---

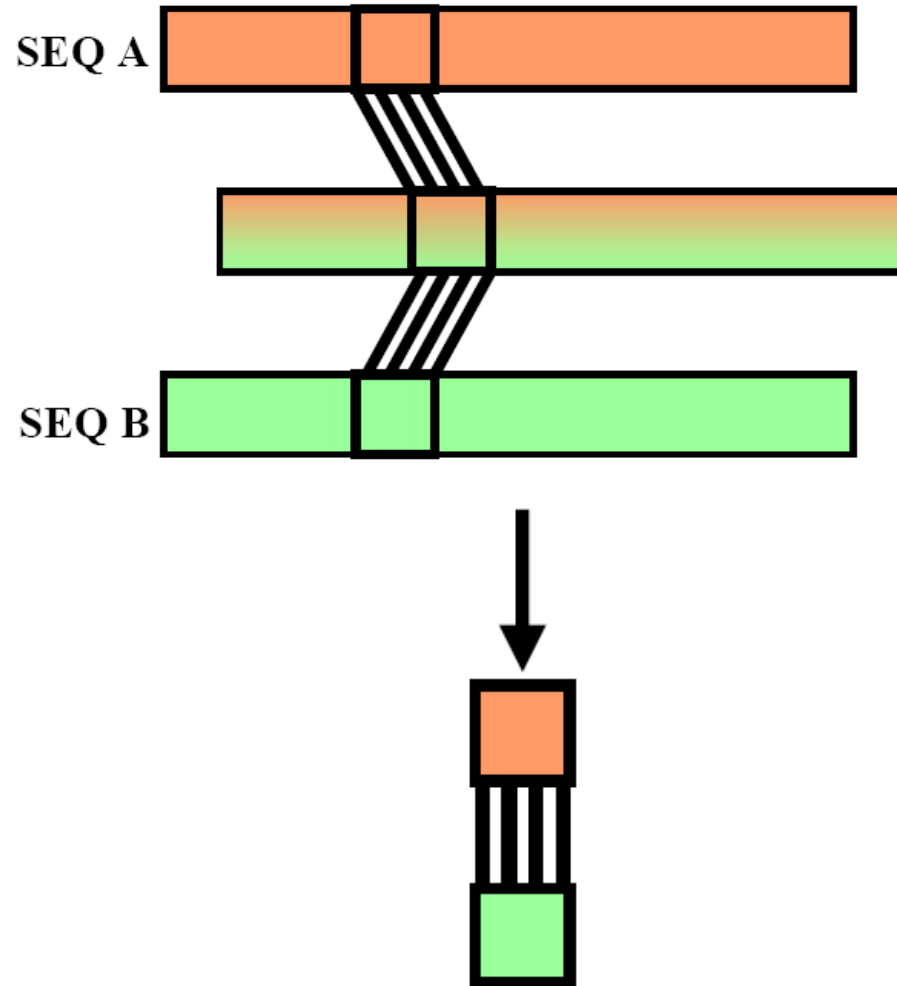
## Library Extension

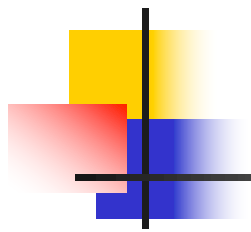
- Is the process where the program assigns a weight for each pair of aligned residues in the Primary Library.
- This weight reflects the degree of a pair consistency in all the seqs in the dataset
- The Extension is done by the Triplet Approach

# The Triplet Approach

Slides taken from

[http://www.isrec.isb-sib.ch/DEA/module5/Course\\_Cedric/main3.pdf](http://www.isrec.isb-sib.ch/DEA/module5/Course_Cedric/main3.pdf)





---

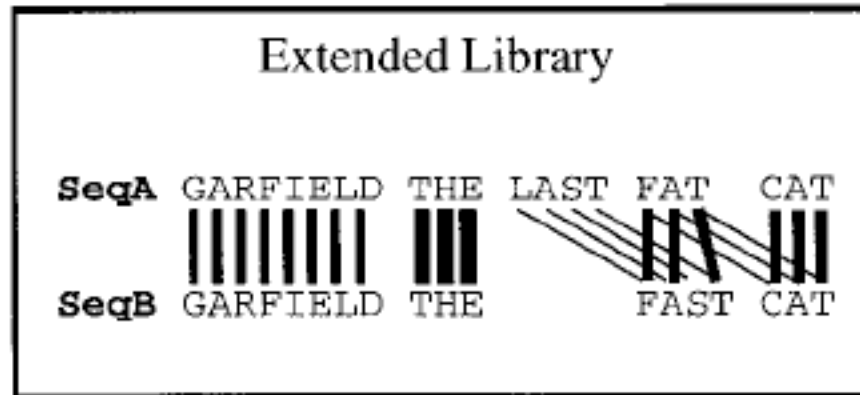
SeqA    GARFIELD    THE    LAST    FAT    CAT  
SeqB    GARFIELD    THE    FAST    CAT  
SeqC    GARFIELD    THE    VERY    FAST    CAT  
SeqD    THE    FAT    CAT

SeqA	GARFIELD	THE	<b>LAST</b>	<b>FAT</b>	CAT	Prim. Weight = 88
SeqB	GARFIELD	THE	<b>FAST</b>	<b>CAT</b>	---	
SeqA	GARFIELD	THE	<b>LAST</b>	FA-T	CAT	Prim. Weight = 77
SeqC	GARFIELD	THE	<b>VERY</b>	FAST	CAT	
SeqA	GARFIELD	THE	LAST	FAT	CAT	Prim. Weight = 100
SeqD	-----	THE	----	FAT	CAT	
SeqB	GARFIELD	THE	----	FAST	CAT	Prim. Weight = 100
SeqC	GARFIELD	THE	VERY	FAST	CAT	
SeqB	GARFIELD	THE	FAST	CAT		Prim. Weight = 100
SeqD	-----	THE	FA-T	CAT		
SeqC	GARFIELD	THE	VERY	FAST	CAT	Prim. Weight = 100
SeqD	-----	THE	----	FA-T	CAT	



c)Extended Library for seq1 and seq2

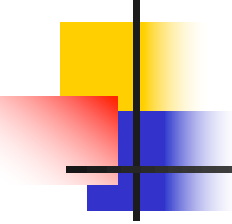
<b>SeqA</b>	GARFIELD	THE	LAST	FAT	CAT	
<b>SeqB</b>	GARFIELD	THE	FAST	CAT		<b>Weight = 88</b>
<b>SeqA</b>	GARFIELD	THE	LAST	FAT	CAT	
				\	\\	
<b>SeqC</b>	GARFIELD	THE	VERY	FAST	CAT	<b>Weight = 77</b>
<b>SeqB</b>	GARFIELD	THE		FAST	CAT	
<b>Seq1</b>	GARFIELD	THE	LAST	FAT	CAT	
<b>SeqD</b>		THE		FAT	CAT	<b>Weight = 100</b>
				\	\\	
<b>SeqB</b>	GARFIELD	THE		FAST	CAT	



Dynamic Programming

<b>SeqA</b>	GARFIELD	THE	LAST	FA-T	CAT
<b>SeqB</b>	GARFIELD	THE	----	FAST	CAT

Figure from JMB Vol 302, pp205-217, 2000

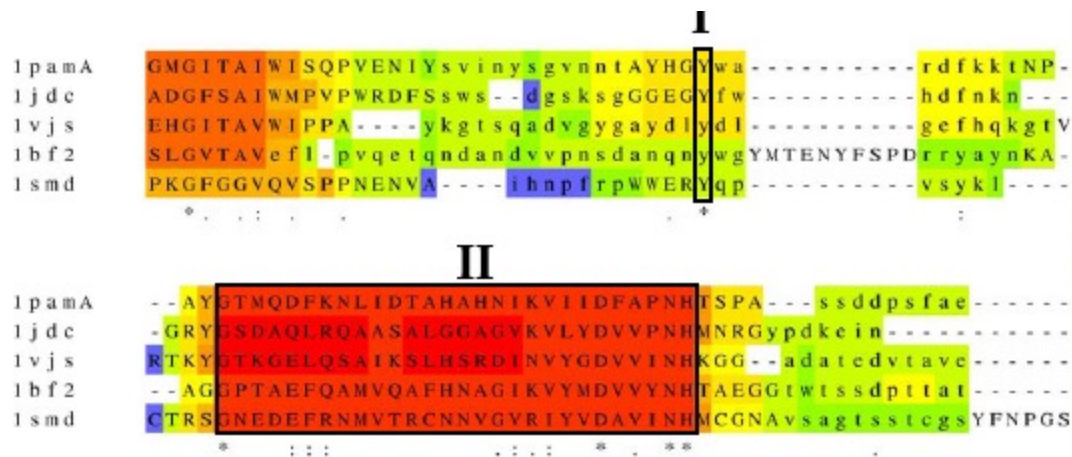
- 
- 
- The complete extension of the Primary Library (check all triplets of the dataset) will assign a weight for each pair of residues that is a sum of all weights gathered for all the triplets that contain the pair.

The more sequences supporting a pair alignment – the higher is its weight

By using **pair weights specific to the dataset** instead of matrix scores the multiple alignment is much more powerful

# T-Coffee fifth step

- **Progressive Alignment** of the extended library set is done by dynamic programming algorithm to achieve the final multiple alignment of the dataset.





# T-Coffee Summary

---

- Good for a limited number of sequences
- Takes long time to run – not good for a large dataset (the newer versions run faster, but the accuracy of large datasets may be questionable)
- Does not deal well (misaligns) sequences which vary a lot in their length



## Bottom Line

---

Speed: Muscle > ClustalW >>T-Coffee

Accuracy (Generally):

Muscle >= T-Coffee > ClustalW

Accuracy depends on the individual sequence family, and for some the order is different...so use more than one algorithm!



# New versions!

---

- Clustal Omega (iterations)
- Muscle 5 (probabilities)
- T-Coffee (regressive algorithm)

All programmed for very large datasets



# Testing accuracy

---

Benchmarking

Running more than one program...





# Visualizing Alignments

---

JalView

<http://www.jalview.org/>

SeaView

<http://doua.prabi.fr/software/seaview>