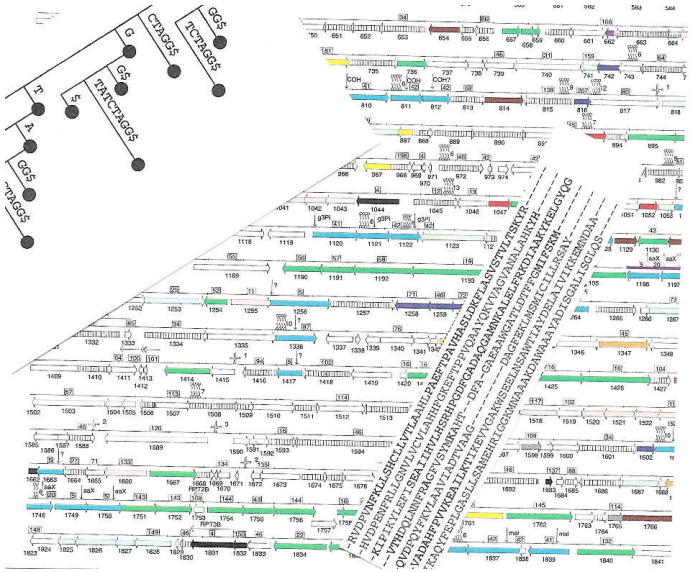
# Lecture 5: Multiple sequence alignment

## Introduction to Computational Biology

Teresa Przytycka, PhD



# Why do we need multiple sequence alignment

Pairwise sequence alignment for more distantly related sequences is not reliable

- it depends on gap penalties, scoring function and other details
- There may be many alignments with the same score – which is right?
- Discovering conserved motifs in a protein family

## pfam00173: Cyt-b5

Cytochrome b5-like Heme/Steroid binding domain. This family includes heme binding domains from a diverse range of proteins. This family also includes proteins that bind to steroids. The family includes progesterone receptors. Many members of this subfamily are membrane anchored by an N-terminal transmembrane alpha helix. This family also includes a domain in some chitin synthases. There is no known ligand for this domain in the chitin synthases.

### Links

### Statistics

### Structure View

### Other Related Conserved Domains

COG4892

### Sequence Alignment

Reformat | Format: Hypertext | Row Display: up to 10 | Color Bits: 2.0 bits | Type Selection: the most diverse members

```
                          10        20        30        40        50        60        70        80
                    ....*....|....*....|....*....|....*....|....*....|....*....|....*....|....*....|
1LTD_A                5 KISPAEVAKHN--KPDDCWVVINGYVYDLTR-FLPNHPGG-------------QDVIKFNAGKDVTAIF---------- 57
gi 75050696          62 DFTPAELRRFDgvQDPRILMAINGKVFDVTKgRKFYGPEG---------------PYGVFAGRDASRGLatfcldkeal 125
gi 75024827          64 DMTVEELRKYDgvKNEHILFGLNGTIYDVTR-GKGFYGPG---------------KAYGTLAGHDATRALgtmdqnavss 127
gi 91206848        1290 YVRRADMENLL--LDGSRCIILAGYVCDLSG-YNCESETL-------------RSVLDSGLGKDLTAEMs---------- 1343
gi 74739702        1209 LIRKADLENHN--KDGGFWTVIDGKVYDIKD-FQTQSLTG-------------NSILAQFAGEDPVVAL---------- 1261
gi 91206849        1210 LIRKADLENHN--KDGGFWTVIDGKVYDIKD-FQTQSLTG-------------NSILAQFAGEDPVVAL---------- 1262
gi 74582634         303 YYNWTDI--HE---PGTSLMVFNGNVLDLSR-LRYLTPNIplpiq-----iaqiVGPGSAFIGRDATYWLs--------- 362
gi 5921760          407 YFTWADIRNNS-----RNLFVYSGNVLDLDL-LFWFNRDQvniprrfeelrdknNAANRAIRGRDATRTF---------- 470
gi 44889038         372 YFTWDDIKNSS-----RNLVVYSGHVLDLDL-LHWFNDTQvtyparfkelrdknTAGNQAIRGRDITHAF---------- 435
gi 122065155        402 QVSLQWNNVTD---PARNLAVYRGSVLDLNR-LNNLTTGLsypel-----ydtlKRRNDSWAGRDVTSAV---------- 462


                          90       100       110
                    ....*....|....*....|....*....|....*..
```

# Multiple alignment as generalization of pairwise alignment

$S^1, S^2, \ldots, S^k$ a set of sequences over the same alphabet

As for the pair-wise alignment, the goal is to find alignment that maximizes some scoring function:

M Q P I L LP
M L R – L- P
M P V I L KP

How to score such multiple alignment?

# Sum of pairs (SP) score

Example consider all pairs of letters in each column and add the scores:

$$\text{SP-score}\begin{pmatrix} A \\ V \\ V \\ - \end{pmatrix} =$$

score(A,V)+score(V,V)+score(V,-)+score(A,-)+score(A,V)

k sequences gives k(k-1)/2 addends

Remark:

Score(-,-) = 0

# Sum of pairs is not prefect scoring system

No theoretical justification for the score.

- In the example below identical pairs are scored 1 and different 0.

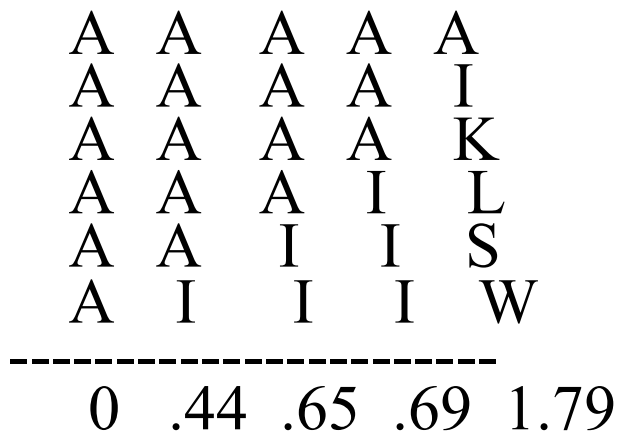
| | | | |
|---|---|---|---|
| A | A | A | A |
| A | A | A | A |
| A | A | A | A |
| A | A | A | I |
| A | A | I | I |
| A | I | I | I |
| 15 | 10 | 7 | 6 |

# Entropy based score (minimum)

$$-\sum_{j}(c_j/C)\log(c_j/C)$$

$c_j$- number of occurrence of amino-acid j in the column

C – number of symbols in the column




(in the example natural ln)

# Dynamic programming solution for multiple alignment

Recall recurrence for multiple alignment:

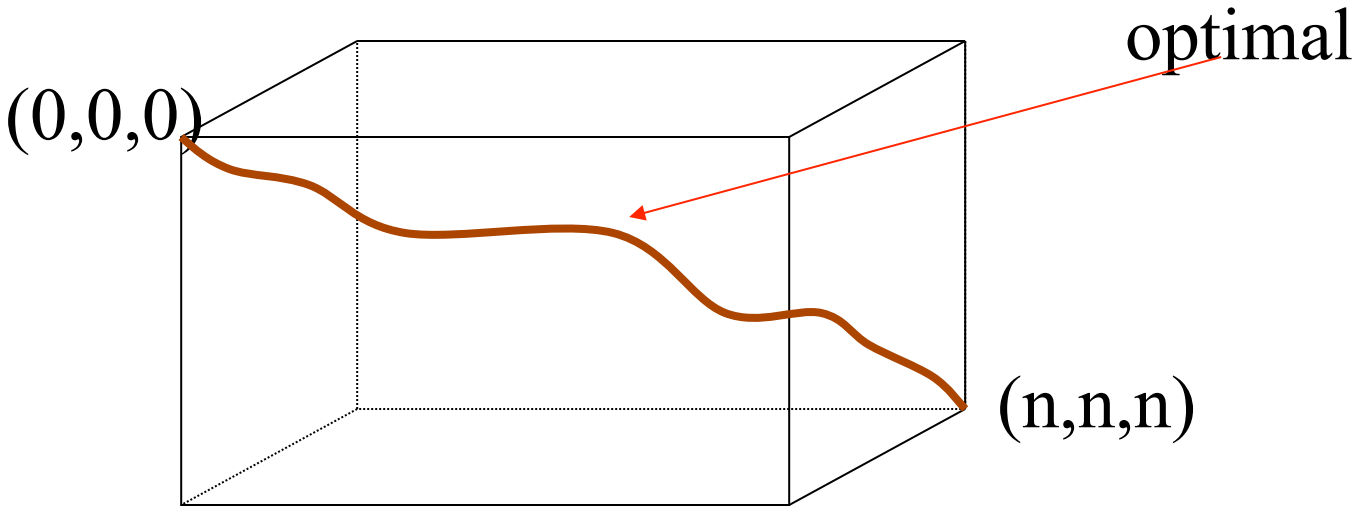
$$\text{Align}(S^1_i, S^2_j) = \max \begin{cases} \text{Align}(S^1_{i-1}, S^2_{j-1}) + s(a_i, a_j) \\ \text{Align}(S^1_{i-1}, S^2_j) - g \\ \text{Align}(S^1_i, S^2_{j-1}) - g \end{cases}$$

For multiple alignment, under max we have all possible combinations of matches and gaps on the last position

For k sequences dynamic programming table will have size $n^k$

# Recurrence for 3 sequences

$$\text{Align}(S^1_i, S^2_j, S^3_k) = \max \begin{cases} \text{Align}(S^1_{i-1}, S^2_{j-1}, S^3_{k-1}) + s(a_i, a_j, a_k) \\ \text{Align}(S^1_{i-1}, S^2_j, S^3_{k-1}) + s(a_i, -, a_k) \\ \text{Align}(S^1_i, S^2_{j-1}, S^3_{k-1}) + s(-, a_j, a_k) \\ \text{Align}(S^1_{i-1}, S^2_{j-1}, S^3_k) + s(a_i, a_j, -) \\ \text{Align}(S^1_i, S^2_j, S^3_{k-1}) + s(a_i, -, -) \\ \text{Align}(S^1_i, S^2_{j-1}, S^3_k) + s(-, a_j, -) \\ \text{Align}(S^1_{i-1}, S^2_j, S^3_k) + s(-, -, a_k) \end{cases}$$



## In dynamic programming approach running time grows elementally with the number of sequences

- Two sequences $O(n^2)$
- Three sequences $O(n^3)$
- k sequences $O(n^k)$

Some approaches to accelerate computation:

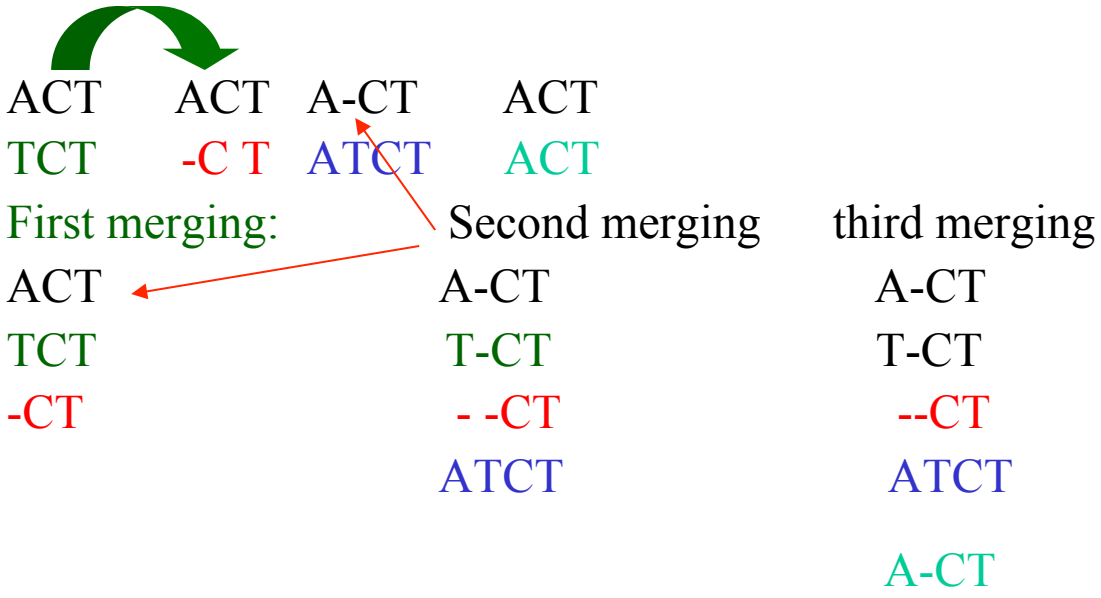
- Use only part of the dynamic programming table centered along the diagonal.
- Use programming technique known as branch and bound
- Use heuristic solutions

# Heuristic approaches to multiple sequence alignment

- Heuristic methods:
  - Star alignment
  - Progressive alignment methods
    - CLUSTALW
    - T-Cofee
    - MUSCLE
  - Heuristic variants of Dynamic Programming Approach
  - Genetic algorithms
  - Gibbs sampler
  - Branch and bound

# Star alignment - using pairwise alignment for heuristic multiple alignment

- Choose one sequence to be the center
- Align all pair-wise sequences with the center
- Merge the alignments: use the center as reference.
- Rule "once a gap always a gap"



# Merging the sequences in stair alignment :

- Use the center as the “guide” sequence
- Add iteratively each pair-wise alignment to the multiple alignment
- Go column by column:
  - If there is no gap neither in the guide sequence in the multiple alignment nor in the merged alignment (or both have gaps) simply put the letter paired with the guide sequence into the appropriate column (all steps of the first merge are of this type.
  - If pair-wise alignment produced a gap in the guide sequence, force the gap on the whole column of already aligned sequences (compare second merge)
  - If there us a gap in added sequence but not in the guide sequences, keep the gap in the added sequence

# Larger example

```
ATTGCCATT
ATGGCCATT
```

```
ATTGCCATT--
ATC-CAATTTT
```

```
ATTGCCATT
ATCTTC-TT
```

```
ATTGCCATT
ACTGACC
```

```
ATTGCCATT--
ATGGCCATT--
ATC-CAATTTT
ATCTTC-TT--
ACTGACC----
```

# Two ways of choosing the center

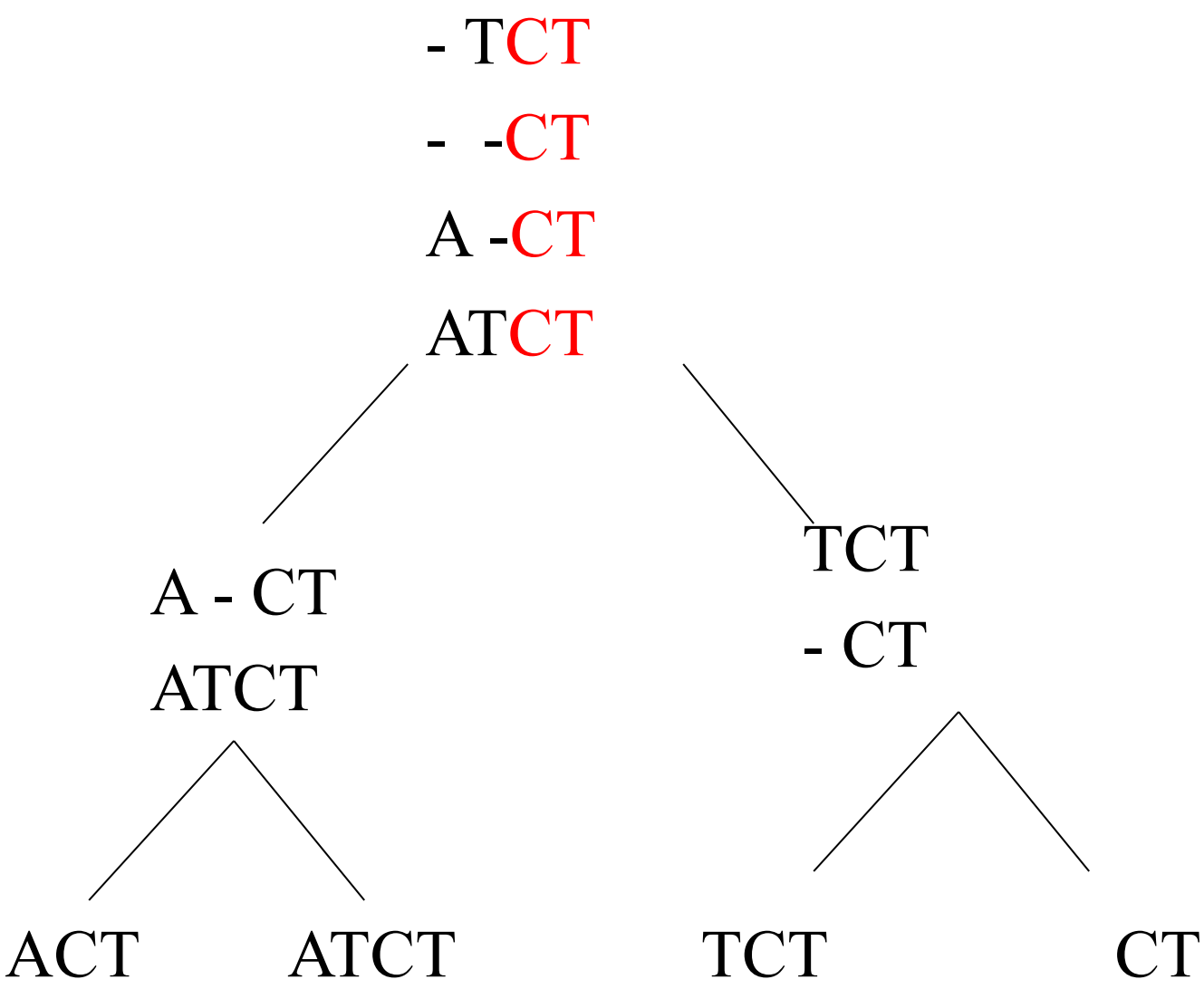
1. Try all possibilities and choose the resulting alignment that gives highest score; or
2. Take sequence $S_c$ that maximizes

$$\Sigma_{\text{ i different than c}} \text{ pairwise-score}(S_c, S_i)$$

(need to compute all pairwise alignments)

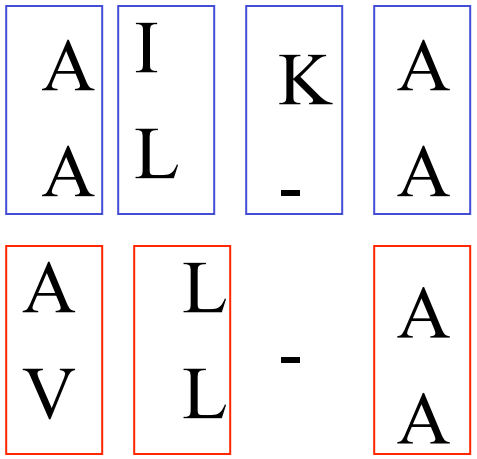
# Progressive alignment

- Idea:
  - First align pair(s) of most closely related sequences
  - Then interactively align the alignments to obtain an alignment for larger number of sequences

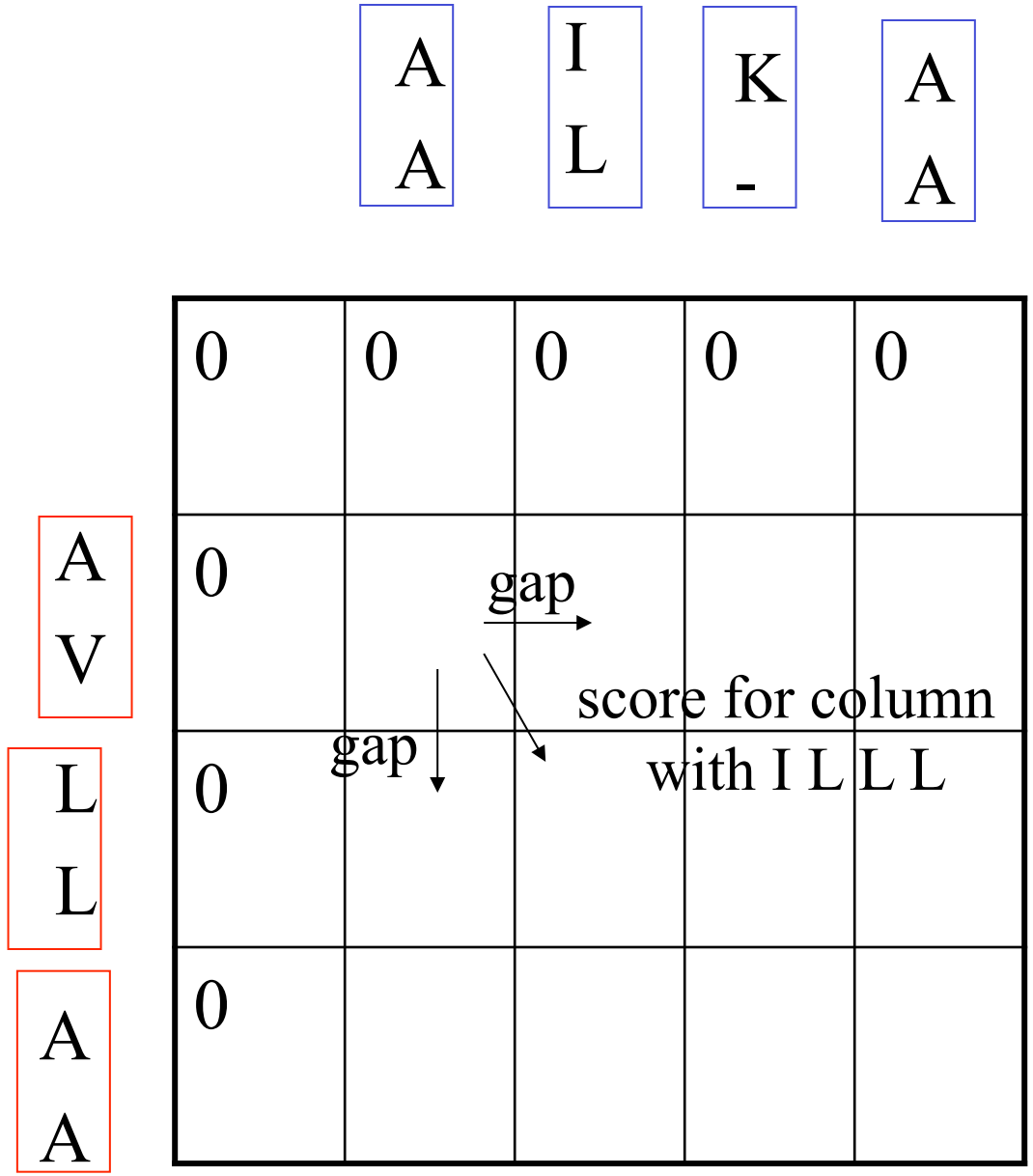

- TCT
- -CT
A -CT
ATCT
A - CT
ATCT
TCT
- CT
ACT
ATCT
TCT
CT

# Aligning alignments

Dynamic programming where a column in each alignment is treated as sequence element




Score of a match – score for the composite column




Gaps:

as for sequences

Match for position (i,j):

Alignment score for the column composed from colum i in the first sequence and column j in the second sequence

# Deciding on the order to merge the alignment

- You want to make most similar sequences first – you are less likely to miss-align them.
- After you align more sequences the alignment works like a profile and you know which columns are to be conserved in a given family – this helps in correct alignment of more distant family members

# CLASTALW

"CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice" Julie D. Thompson, Desmond G. Higgins and Toby J. Gibson*Nucleic Acids Research, 1994, Vol. 22, No. 22 4673-4680

1. Perform all pair pairwise alignments
2. Use the alignment score to produce distance based phylogenic tree *(phylogenic tree constructed methods will be presented later in class)*
3. Align sequences in the order defined by the tree: from the leaves towards the root.

(Initially this involves alignment of sequences and later alignment of alignments.)

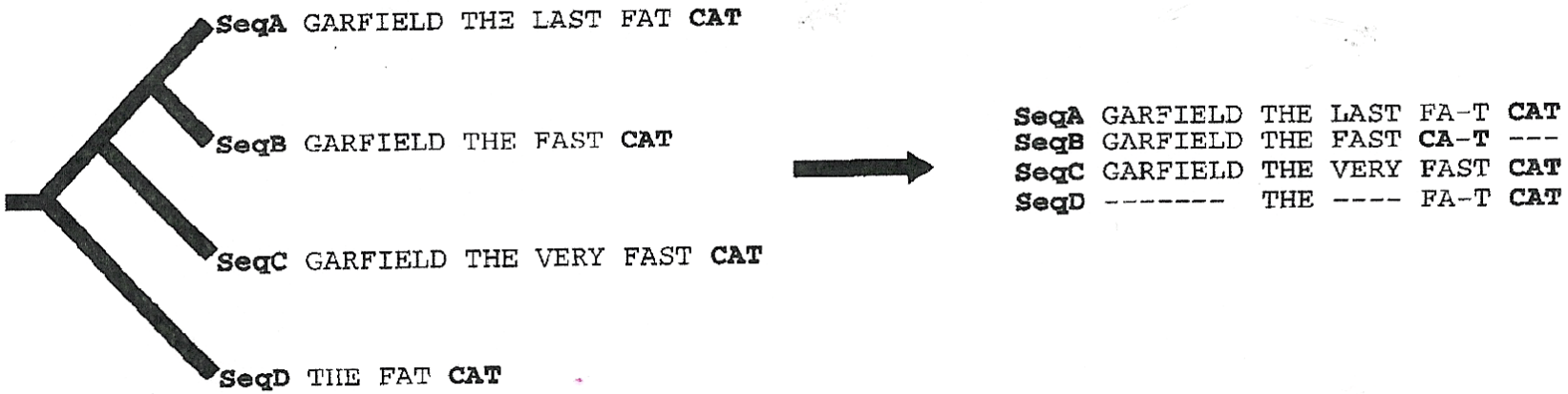
# Problems with CLUSTAL W and other "progressive alignments"

- Dependence of the initial pair-wise sequence alignment.
- Propagating errors form initial alignments.

# Example

*This and next figures examples are from T-coffee paper:*
*Noterdame, Higgins, Heringa, JMB 2000, 302 205-217*

a)Regular Progressive Alignment Strategy



# T-Coffee (Tree-Based Consistency Objective Function for alignment Evaluation)

*Noterdame, Higgins, Heringa, JMB 2000, 302 205-217*

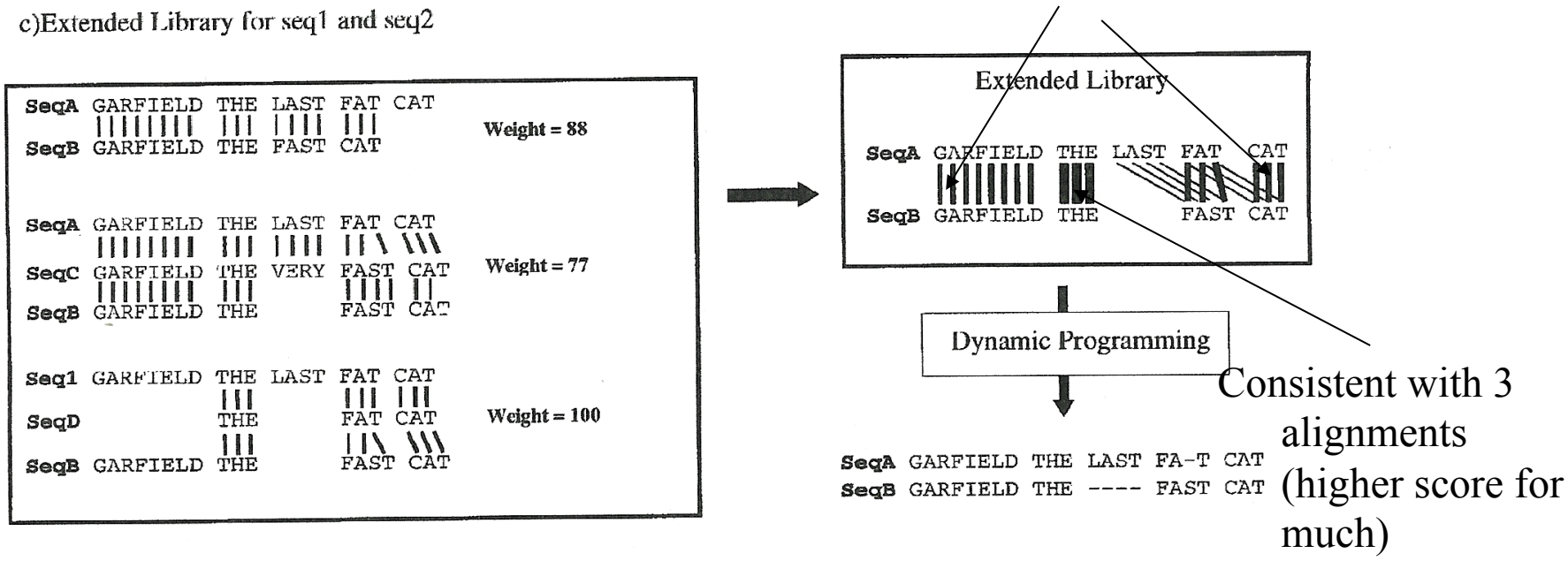
- Construct a library of pair-wise alignments
  - In library each alignment is represented as a list of pair-wise residue matches (e.g.res.x sequence A is aligned with res. y of sequence B)
  - The weight of each alignment corresponds to percent identity (per aligned residua)

b)Primary Library

```
SeqA GARFIELD THE LAST FAT CAT   Prim. Weight = 88
SeqB GARFIELD THE FAST CAT ---

SeqA GARFIELD THE LAST FA-T CAT  Prim. Weight = 77
SeqC GARFIELD THE VERY 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
```

# T-coffee continued

- Consistency alignment: for every pair-wise alignments (A,B) consider alignment with third sequence C. What would be the alignment "through" third sequence A-C-B
- Sum-up the weights over all possible choices if C to get "



# Last step of T-coffee

- Do progressive alignment using the tree but using the weights from extended library for scoring the alignment.

(e.g. “A” in FAST will have higher score with “A” in FAT and lower with “A” in LAST.)

# T-coffee summary

- More accurate than CLUSTALW
- Slower (significantly) the CLUSTALW but much faster than MSA and can handle more sequences.

# A newer consistency based approach

Resource

## ProbCons: Probabilistic consistency-based multiple sequence alignment

Chuong B. Do,[1] Mahathi S.P. Mahabhashyam,[1] Michael Brudno,[1] and Serafim Batzoglou[1,2]

[1]*Department of Computer Science, Stanford University, Stanford, California 94305, USA*

To study gene evolution across a wide range of organisms, biologists need accurate tools for multiple sequence alignment of protein families. Obtaining accurate alignments, however, is a difficult computational problem because of not only the high computational cost but also the lack of proper objective functions for measuring alignment quality. In this paper, we introduce *probabilistic consistency*, a novel scoring function for multiple sequence comparisons. We present ProbCons, a practical tool for progressive protein multiple sequence alignment based on probabilistic consistency, and evaluate its performance on several standard alignment benchmark data sets. On the BAliBASE, SABmark, and PREFAB benchmark alignment databases, ProbCons achieves statistically significant improvement over other leading methods while maintaining practical speed. ProbCons is publicly available as a Web resource.

[Supplemental material is available online at www.genome.org. Source code and executables are available as public domain software at http://probcons.stanford.edu.]

Genome research 2005

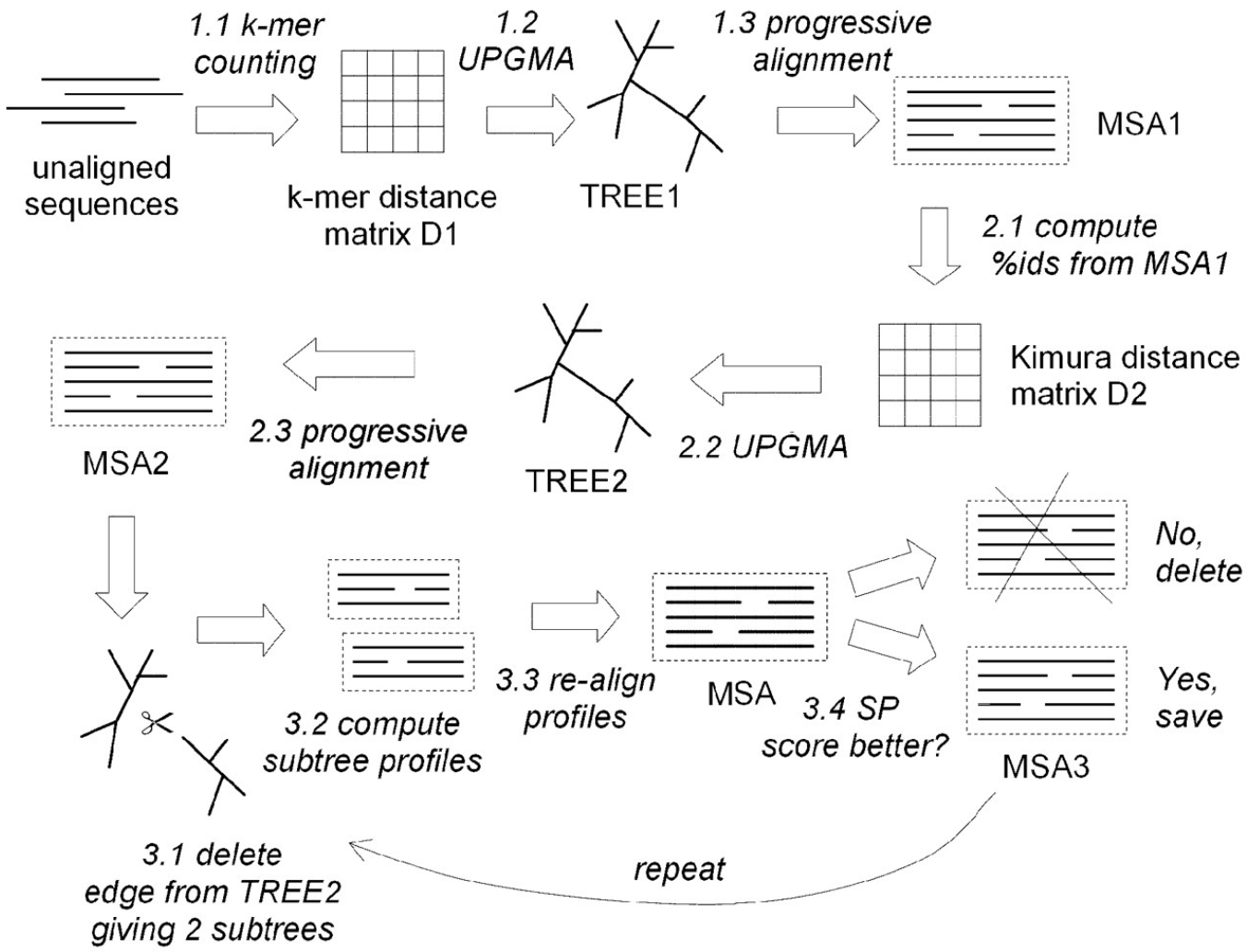
# MUSCLE

**Robert C. Edgar*** Nucleic Acids Research, 2004, Vol. 32, No. 5 **1792-1797**

**MUSCLE: multiple sequence alignment with high accuracy and high throughput**

# MUSCLE idea

- Build quick approximate sequence similarity tree – without pair-wise alignment but compute distances by computing the number of short "hits" (short gapless matching) between any pair of sequences.
- Compute MSA using the tree.
- Compute pair-wise  distances from MSA and new tree
- Re-compute MSA using new tree
- Refine the alignment by iteratively partitioning the sequence into two groups and merging the aligning multiple alignment from the two groups




**Edgar, R. C. Nucl. Acids Res. 2004 32:1792-1797; doi:10.1093/nar/gkh340**

Nucleic Acids Research

# Where the speed-up comes from

- Finding all short hits is fast due because we can use methods like hashing
- ClustalW computed n(n-1)/2 pairwise alignments while given a tree one needs to do only n-1 alignments

# Refining multiple sequence alignment

- Given – multiple alignment of sequences
- Goal improve the alignment
- One of several methods:
  - Choose a random sentence
  - Remove from the alignment (n-1 sequences left)
  - Align the removed sequence to the n-1 remaining sequences.
  - Repeat
- Alternatively – (MUSCLE approach) the alignment set can be subdivided into two subsets, the alignment of the subsets recomputed and alignment aligned

# Evaluating MSA

- Based on alignment of structures
  (e.g. BaliBase test set)
- Simulation: simulate random evolutionary changes
- Testing for correct alignment of annotated functional residues