BIOS 477/877
Bioinformatics and Molecular Evolution

Instructor: Etsuko Moriyama
(School of Biological Sciences)

Spring 2026 **Lecture 15**

1

Today's topics

- Multiple Sequence Alignment
  - Progressive alignment (ClustalW)
  - T-Coffee

2

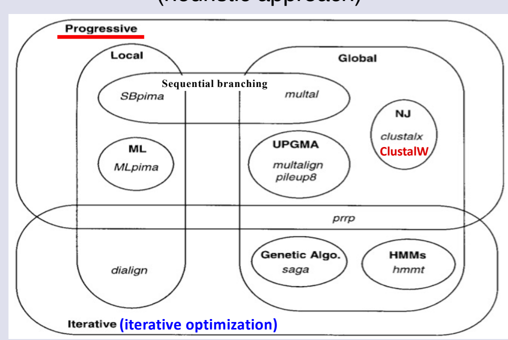
Multiple alignment as an extension of pairwise alignment

- Dynamic programming algorithm
  - → Guarantees to find the optimal alignment
  - → Optimal alignments are searched based on alignment score
    - How can we score multiple alignment?
      - Sum of pairs score: $S(A) = \Sigma_{i,j} S(A_{ij})$
        - → No statistical justification
  - → Not very efficient for many sequences

We need good/efficient heuristic methods!

3

Multiple alignment methods
(heuristic approach)



Thompson *et al.* (1999)

4

ClustalW

Thompson *et al.* (1994)

1. Pairwise alignment
(fast approximation or full dynamic programming)

2. Generate a distance matrix
(% identities converted to distances)

3. Construct a guide tree
(neighbor-joining phylogenetic tree)

4. Progressive alignment following the guide tree
(scoring matrix, sequence weight, gap penalties, etc.)

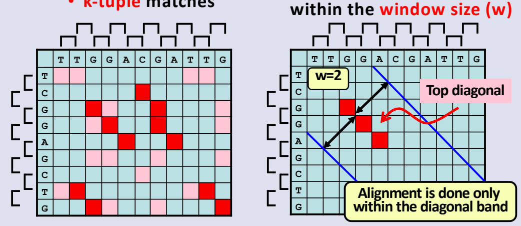
Updated to ClustalΩ and should not be used; to try, see Course Web 'Links' page

5

ClustalW: 1. Pairwise alignment

- Fast approximation using a diagonal band

- k-tuple matches
- Choose diagonals within the window size (w)

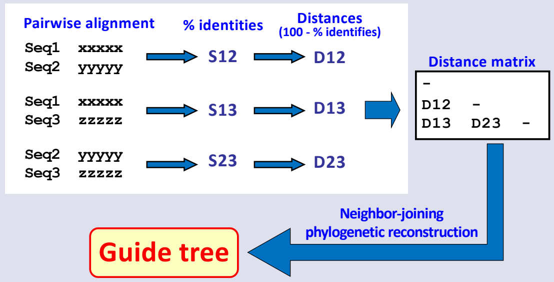


The default is to use full dynamic programming: better

6

ClustalW: 2 & 3. Guide tree generation

- A guide tree is generated from a distance matrix



BIOS477/877 L15- 7

7

ClustalW: steps

Thompson *et al.* (1994)

1. Pairwise alignment
(fast approximation or full dynamic programming)

2. Generate a distance matrix
(% identities converted to distances)

3. Construct a guide tree
(neighbor-joining phylogenetic tree)

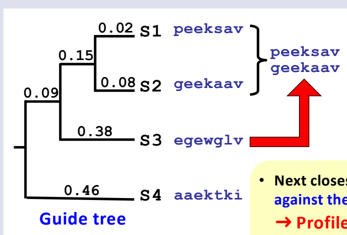
4. Progressive alignment following the guide tree
(scoring matrix, sequence weight, gap penalties, etc.)

BIOS477/877 L15 - 8

8

ClustalW: 4. progressive alignment

- Alignment is done following the guide tree



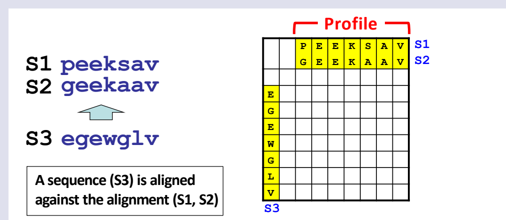
- Closest sequences are aligned first
- Pairwise alignment can be done by using simply the dynamic programming algorithm
- Next closest sequence is aligned against the existing alignment
  → Profile alignment
  (can be alignment between alignments)

BIOS477/877 L15 - 9

9

ClustalW: profile alignment

- Alignment against existing alignment



A sequence (S3) is aligned against the alignment (S1, S2)

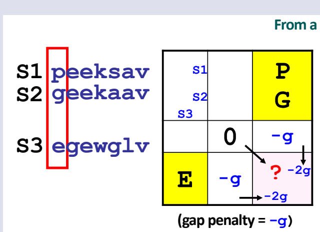
Existing alignment = profile
→ Each position of alignment shows the configuration of possible amino acids

BIOS477/877 L15 - 10

10

ClustalW: profile alignment

- How cell scores can be calculated



S1 vs. S3 = S(P,E)
S2 vs. S3 = S(G,E)

S1/S2 vs. S3
= {S(P,E)+S(G,E)}/2

Simple average from all pairwise scores

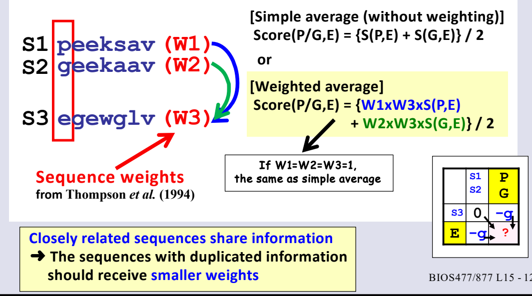
Existing alignment = profile
→ Each position of alignment shows the configuration of possible amino acids

BIOS477/877 L15 - 11

11

ClustalW: profile alignment

- Cell scores using simple vs. weighted average



[Simple average (without weighting)]
Score(P/G,E) = {S(P,E) + S(G,E)} / 2

or

[Weighted average]
Score(P/G,E) = {W1xW3xS(P,E)
+ W2xW3xS(G,E)} / 2

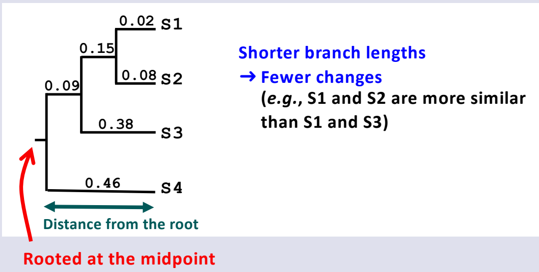
Closely related sequences share information
→ The sequences with duplicated information should receive smaller weights

BIOS477/877 L15 - 12

12

ClustalW: profile alignment

➢ **Sequence weights**: based on branch lengths

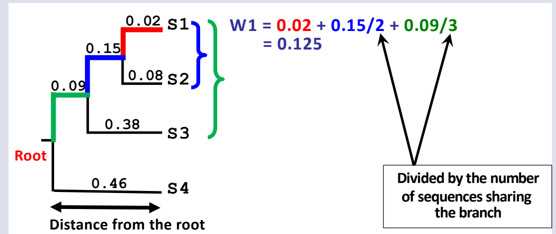


BIOS477/877 L15 - 13

13

ClustalW: profile alignment

➢ **Sequence weights**: based on branch lengths



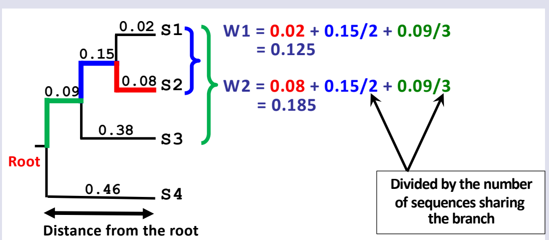
Closely related sequences share information
➜ shorter branch lengths

BIOS477/877 L15 - 14

14

ClustalW: profile alignment

➢ **Sequence weights**: based on branch lengths



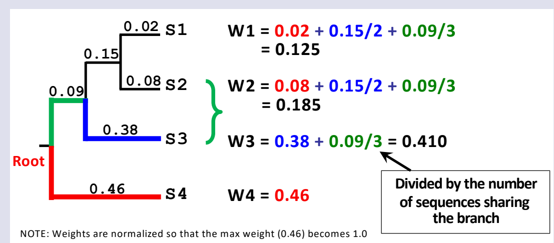
Closely related sequences share information
➜ shorter branch lengths

BIOS477/877 L15 - 15

15

ClustalW: profile alignment

➢ **Sequence weights**: based on branch lengths



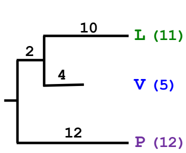
Closely related sequences share information
➜ The sequences with duplicated information should receive smaller weights

BIOS477/877 L15 - 16

16

ClustalW: profile alignment

➢ **How sequence weighting works: example 1**



Aligning the 3rd sequence (P) to the first 2 sequences (L, V) previously aligned

[Simple average]
Score(L/V,P)
= {S(L,P) + S(V,P)} / 2
= 0.5 S(L,P) + 0.5 S(V,P)

[Weighted average]
Score(L/V,P)
= {11x12xS(L,P) +
5x12xS(V,P)} / 2
= 66 S(L,P) + 30 S(V,P)

S(L,P) >> S(V,P)

S(L,P) is weighted more

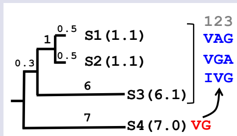
Closely related sequences share information
➜ The sequences with duplicated information should receive smaller weights

BIOS477/877 L15 - 17

17

ClustalW: profile alignment

➢ **How sequence weighting works: example 2**



How should 'VG' in S4 be aligned against the alignment of S1/S2/S3?

(S1, S2, and S3 are previously aligned)

	(a) 123		(b) 123		(c) 123
S1	VAG		VAG		VAG
S2	VGA	or	VGA	or	VGA
S3	IVG		IVG		IVG
S4	VG-		V-G		-VG

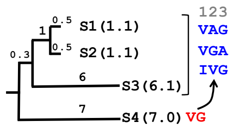
Which is the best alignment, (a) or (b) or (c)?

BIOS477/877 L15 - 18

18

ClustalW: profile alignment

➢ How sequence weighting works: example 2



	(a) 123		(b) 123		(c) 123
S1	VAG		VAG		VAG
S2	VGA	or	VGA	or	VGA
S3	IVG		IVG		IVG
S4	VG-		V-G		-VG

[Simple average] (match=2, mismatch=-1)

Alignment (a) [Score = 1 + gap penalty]
1: Score = {S(V,V)+S(V,V)+S(I,V)}/3 = (2+2-1)/3 = 1
2: Score = {S(A,G)+S(G,G)+S(V,G)}/3 = (-1+2-1)/3 = 0
3: Score = (gap penalty x 3)/3

Alignment (b) [Score = 2 + gap penalty]
1: Score = {S(V,V)+S(V,V)+S(I,V)}/3 = (2+2-1)/3 = 1
2: Score = (gap penalty x 3)/3
3: Score = {S(G,G)+S(A,G)+S(G,G)}/3 = (2-1+2)/3 = 1

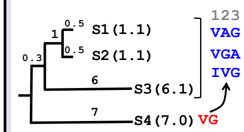
Alignment (c) [Score = 1 + gap penalty]
1: Score = (gap penalty x 3)/3
2: Score ={S(A,V)+S(G,V)+S(V,V)}/3 = (-1-1+2)/3 = 0
3: Score = {S(G,G)+S(A,G)+S(G,G)}/3 = (2-1+2)/3 = 1

BIOS477/877 L15 - 19

19

ClustalW: profile alignment

➢ How sequence weighting works: example 2



	(a) 123		(b) 123		(c) 123
S1	VAG		VAG		VAG
S2	VGA	or	VGA	or	VGA
S3	IVG		IVG		IVG
S4	VG-		V-G		-VG

[Weighted average] (match=2, mismatch=-1)

Alignment (a)

1: Score =
{1.1x7xS(V,V)+1.1x7xS(V,V)+6.1x7xS(I,V)}/3
= {7.7x2+7.7x2+42.7x(-1)}/3 = -3.97

2: Score =
{1.1x7xS(A,G)+1.1x7xS(G,G)+6.1x7xS(V,G)}/3
= {7.7x(-1)+7.7x2+42.7x(-1)}/3 = -11.67

3: Score = (gap penalty x 3)/3

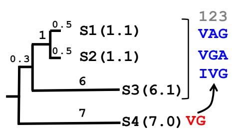
Alignment Score = -15.64 + gap penalty

BIOS477/877 L15 - 20

20

ClustalW: profile alignment

➢ How sequence weighting works: example 2



	(a) 123		(b) 123		(c) 123
S1	VAG		VAG		VAG
S2	VGA	or	VGA	or	VGA
S3	IVG		IVG		IVG
S4	VG-		V-G		-VG

[Weighted average] (match=2, mismatch=-1)

Alignment (b)

1: Score =
{1.1x7xS(V,V)+1.1x7xS(V,V)+6.1x7xS(I,V)}/3
= {7.7x2+7.7x2+42.7x(-1)}/3 = -3.97

2: Score = (gap penalty x 3)/3

3: Score =
{1.1x7xS(G,G)+1.1x7xS(A,G)+6.1x7xS(G,G)}/3
= {7.7x2+7.7x(-1)+42.7x(2)}/3 = 31.03

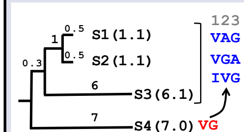
Alignment Score = 27.06 + gap penalty

BIOS477/877 L15 - 21

21

ClustalW: profile alignment

➢ How sequence weighting works: example 2



	(a) 123		(b) 123		(c) 123
S1	VAG		VAG		VAG
S2	VGA	or	VGA	or	VGA
S3	IVG		IVG		IVG
S4	VG-		V-G		-VG

[Weighted average] (match=2, mismatch=-1)

Alignment (c)

1: Score = (gap penalty x 3)/3

2: Score =
{1.1x7xS(A,V)+1.1x7xS(G,V)+6.1x7xS(V,V)}/3
= {7.7x(-1)+7.7x(-1)+42.7x(2)}/3 = 23.33

3: Score =
{1.1x7xS(G,G)+1.1x7xS(A,G)+6.1x7xS(G,G)}/3
= {7.7x2+7.7x(-1)+42.7x(2)}/3 = 31.03

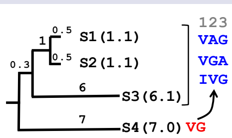
Alignment Score = 54.36 + gap penalty

BIOS477/877 L15 - 22

22

ClustalW: profile alignment

➢ How sequence weighting works: example 2



	(a) 123		(b) 123		(c) 123
S1	VAG		VAG		VAG
S2	VGA	or	VGA	or	VGA
S3	IVG		IVG		IVG
S4	VG-		V-G		-VG

[Simple average]

(a) Alignment score = 1 + gap penalty
(b) Alignment score = 2 + gap penalty
(c) Alignment score = 1 + gap penalty

[Weighted average]

(a) Alignment score = -15.64 + gap penalty
(b) Alignment score = 27.06 + gap penalty
(c) Alignment score = 54.36 + gap penalty

BIOS477/877 L15 - 23

23

ClustalW: parameter selection

➢ How scoring matrix is chosen
- Users choose only a scoring matrix series (BLOSUM, PAM, *etc.*)
- Specific matrix (BLOSUM80, *etc.*) is determined based on distance between sequences
  - 80 - 100% identity → BLOSUM80
  - 60 - 80% identity → BLOSUM62
  - 30 - 60% identity → BLOSUM45
  - 0 - 30% identity → BLOSUM30

BIOS477/877 L15 - 24

24

ClustalW: parameter selection

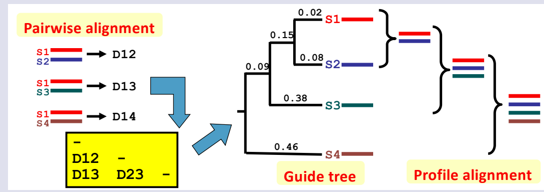
- How **gap penalties** are determined
  - Initial gap penalties: **GOP** (gap opening) and **GEP** (gap extension) are set by the user
  - **Position- or residue-specific** gap penalties
    - Weight (scoring) matrix dependent
    - Similarity level dependent
    - Sequence length dependent
    - Position-specific
      - → if gaps already exist
      - → Residue-specific (*e.g.*, hydrophilic stretches)

See Thompson *et al.* (1994) for more details

BIOS477/877 L15 - 25

25

ClustalW: summary & limitation



- Progressive alignment
  - → **Greedy** (finds local optima, but no guarantee for global optima)
  - → **Guide tree** is built only once at the beginning (cannot be fixed)
  - → **Errors** in the early alignments (incorrect gap positions) cannot be rectified later
  - → ClustalW does not rely on one set of alignment parameters
    - Position- and various feature-specific scoring matrix and gap penalties
- **Global alignment** only (local similarity may be missed)

BIOS477/877 L15 - 26

26

How to solve progressive-alignment problems

- Incorporate more information to reduce **early errors**
  - **Structural alignment** (*e.g.*, Expresso, PROMALS3D, TM-Coffee, PRALINE, MAFFT-DASH)
  - **Profile/profile-HMM alignment** (*e.g.*, PRALINE, PSI-Coffee, PROMALS3D, ProbCons/CONTRAlign, ClustalΩ, MUSCLE5)
- Avoid the **greedy-algorithm** problem
  - **Iterative refinement** to search the global maxima
    - → A good objective function is required (*e.g.*, MUSCLE/MUSCLE5, MAFFT, ProbCons/CONTRAlign)
- **Global (or local) only** alignment problem
  - **Combine both methods** (*e.g.*, T-Coffee)
- More accurate **insertion/deletion placement**
  - **Phylogeny aware gap-placement** (*e.g.*, PRANK, ProPIP, Bali-Phy, SATé)

BIOS477/877 L15 - 27

27

T-Coffee

Notredame, Higgins & Heringa (2000); Taly *et al.* (2011)

- T-Coffee: **Tree-based Consistency Objective Function for alignment Evaluation**
- Progressive alignment
  - → Uses a guide tree, fast
- Tries to avoid the **greedy** nature of the progressive algorithm
  - → Using **alignment libraries** derived from a mixture of alignment methods (global, local, *etc.*)

[T-Coffee on the Web]

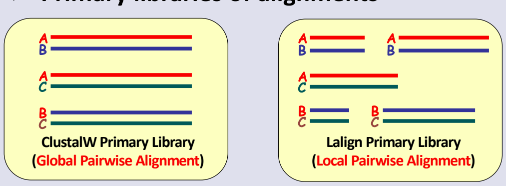
- T-Coffee home page
- T-Coffee site from Notredame Lab
- T-Coffee @ EBI Tools (only T-Coffee, without other associated programs)

BIOS477/877 L15 - 28

28

T-Coffee: primary libraries

- Primary libraries of alignments

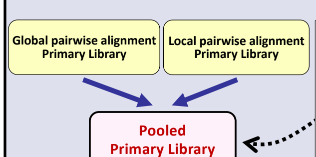


- Global and local, or any combination of pairwise alignment methods
- Each pairwise alignment is given a **weight** based on % identity ignoring gap sites ATTCGG / ATAGCG ⇨ 3/6 = 50% ⇨ W=50

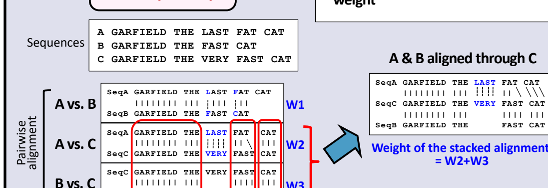
BIOS477/877 L15 - 29

29

T-Coffee: pooled primary library

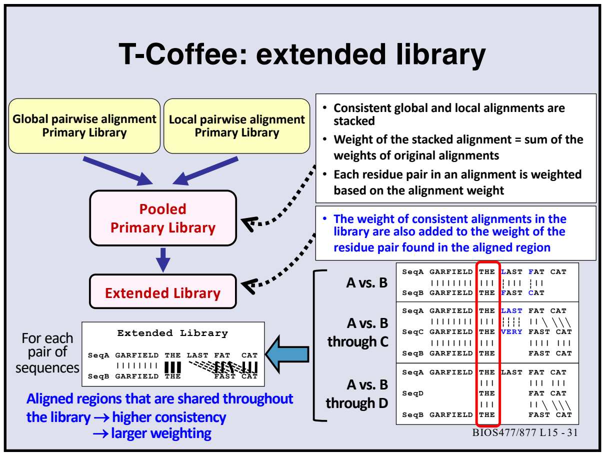


- Consistent global and local alignments are stacked
- Weight of the stacked alignment = sum of the weights of original alignments
- Each residue pair in an alignment is weighted based on the alignment weight

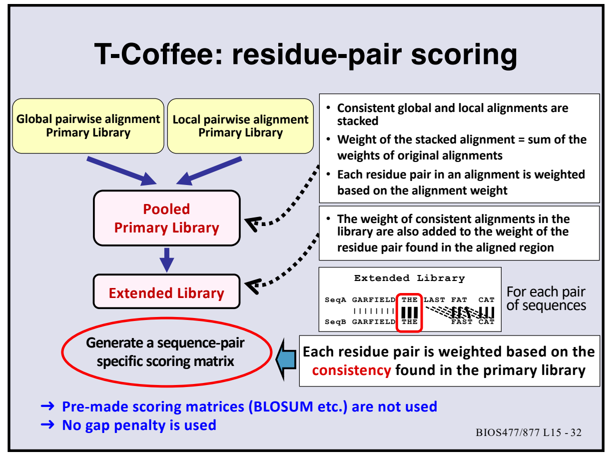


BIOS477/877 L15 - 30

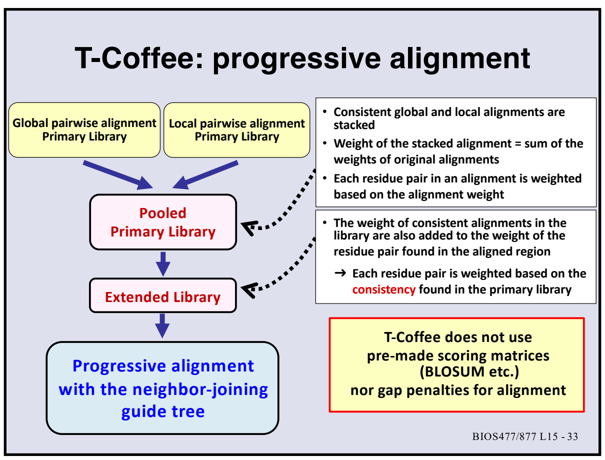
30



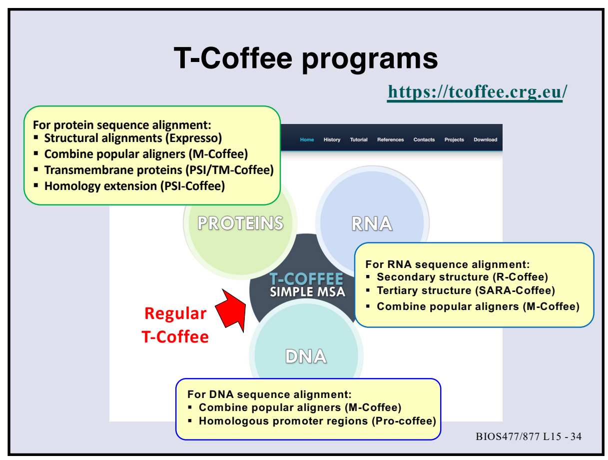
31



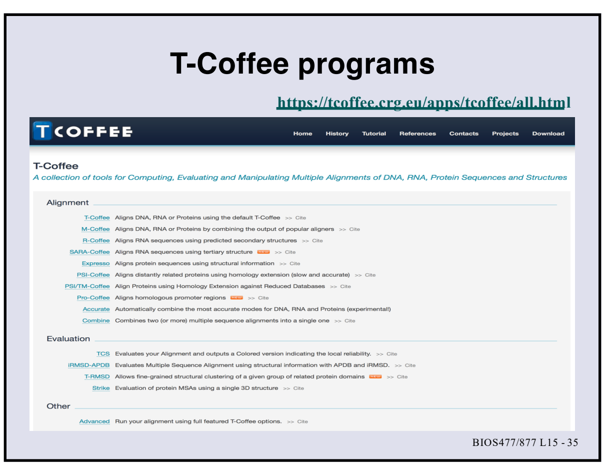
32



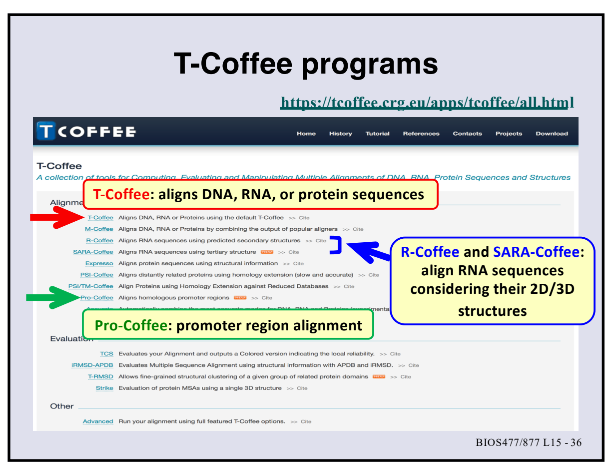
33



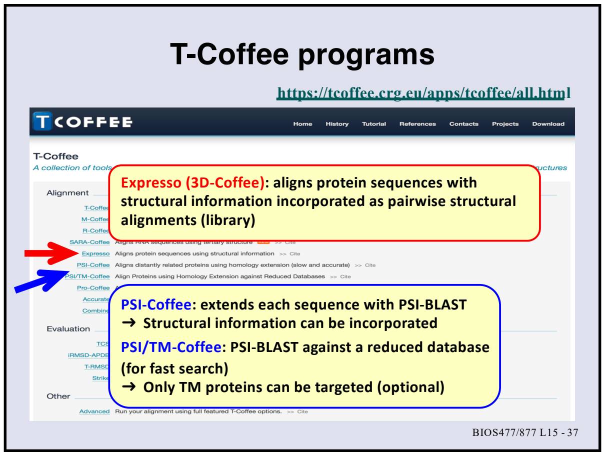
34



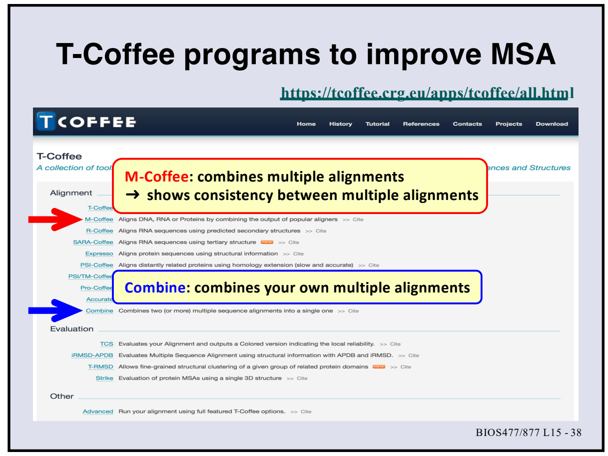
35



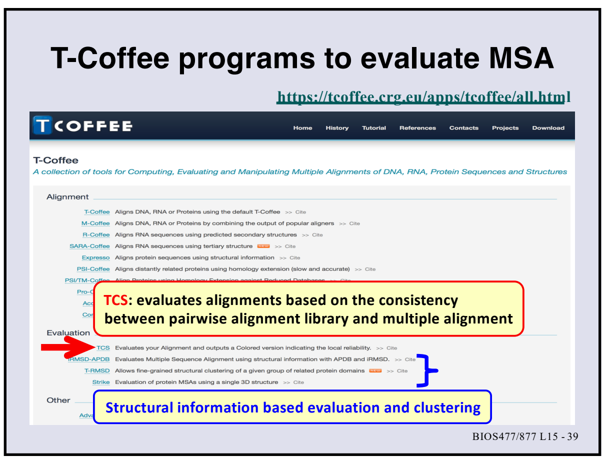
36


T-Coffee programs
https://tcoffee.crg.eu/apps/tcoffee/all.html
Expresso (3D-Coffee): aligns protein sequences with structural information incorporated as pairwise structural alignments (library)
PSI-Coffee: extends each sequence with PSI-BLAST
→ Structural information can be incorporated
PSI/TM-Coffee: PSI-BLAST against a reduced database (for fast search)
→ Only TM proteins can be targeted (optional)
BIOS477/877 L15 - 37

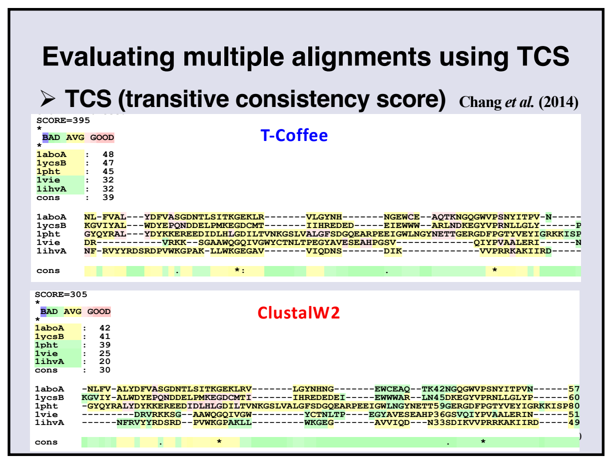
37


T-Coffee programs to improve MSA
https://tcoffee.crg.eu/apps/tcoffee/all.html
M-Coffee: combines multiple alignments
→ shows consistency between multiple alignments
Combine: combines your own multiple alignments
BIOS477/877 L15 - 38

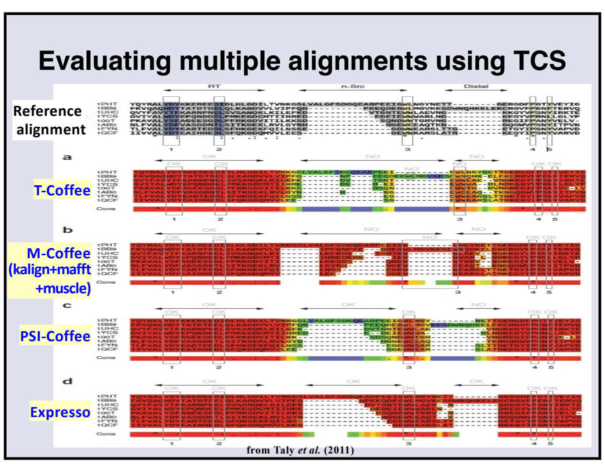
38


T-Coffee programs to evaluate MSA
https://tcoffee.crg.eu/apps/tcoffee/all.html
TCS: evaluates alignments based on the consistency between pairwise alignment library and multiple alignment
Structural information based evaluation and clustering
BIOS477/877 L15 - 39

39


Evaluating multiple alignments using TCS
➤ TCS (transitive consistency score) Chang et al. (2014)
SCORE=395
BAD AVG GOOD
T-Coffee
SCORE=305
BAD AVG GOOD
ClustalW2

40


Evaluating multiple alignments using TCS
Reference alignment
T-Coffee
M-Coffee (kalign+mafft +muscle)
PSI-Coffee
Expresso
from Taly et al. (2011)

41