Multiple Sequence Alignment

Bartek Wilczyński

March 24^{th}, 2020

How sequences evolve?

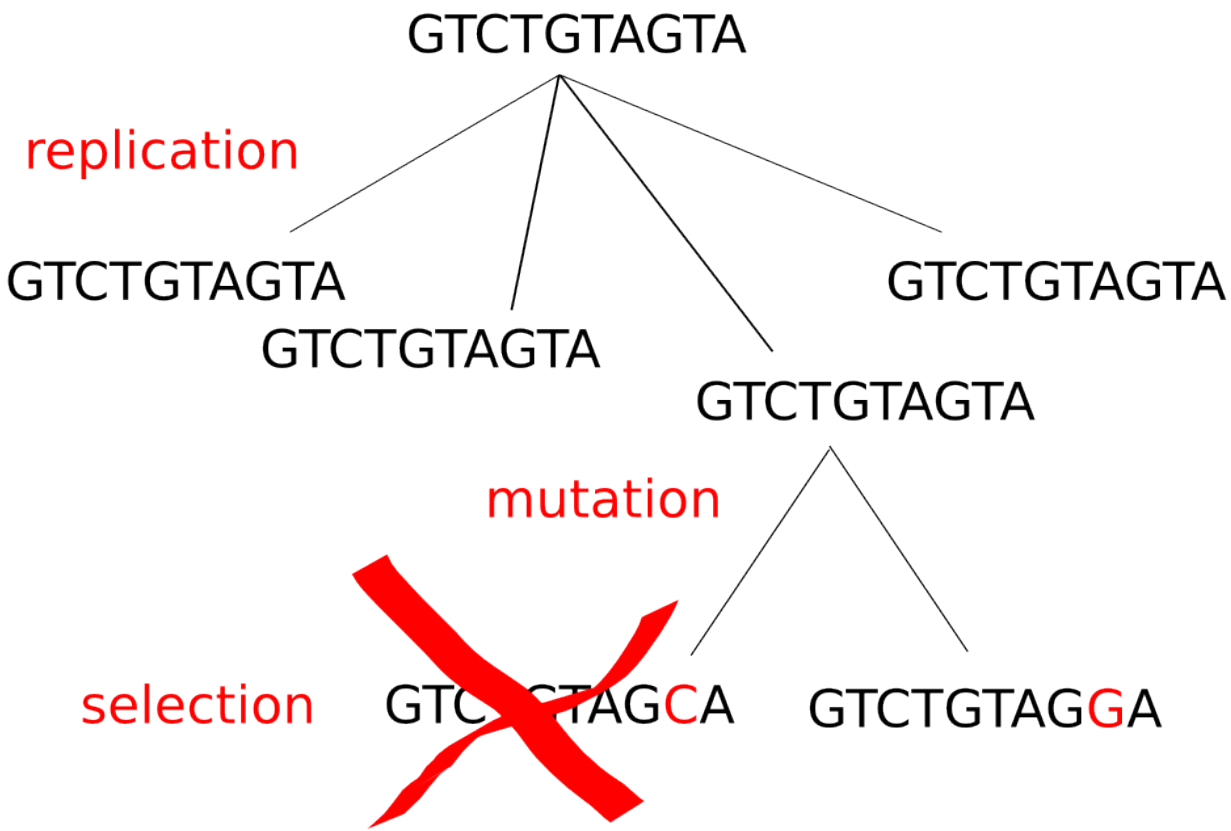


image (c)

BW

Reconstructing alignments

		H	E	A	G	A	W	G	H	E	E
	0	**–8**	**–16**	–24	–32	–40	–48	–56	–64	–72	–80
P	–8	–2	–9	**–17**	**–25**	–33	–41	–49	–57	–65	–73
A	–16	–10	–3	–4	–12	**–20**	–28	–36	–44	–52	–60
W	–24	–18	–11	–6	–7	–15	**–5**	**–13**	–21	–29	–37
H	–32	–14	–18	–13	–8	–9	–13	–7	**–3**	–11	–19
E	–40	–22	–8	–16	–16	–9	–12	–15	–7	**3**	–5
A	–48	–30	–16	–3	–11	–11	–12	–12	–15	**–5**	2
E	–56	–38	–24	–11	–6	–12	–14	–15	–12	–9	**1**

```
HEAGAWGHE-E
--P-AW-HEAE
```

image (c) Durbin et al.

Greedy tree inference – example

	A	B	C	D
A	0	17	21	27
B		0	12	18
C			0	14
D				0

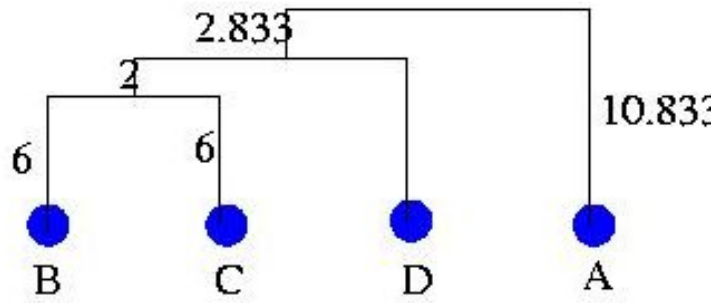


image (c) P. Winter

Inconsistencies with pairwise alignments

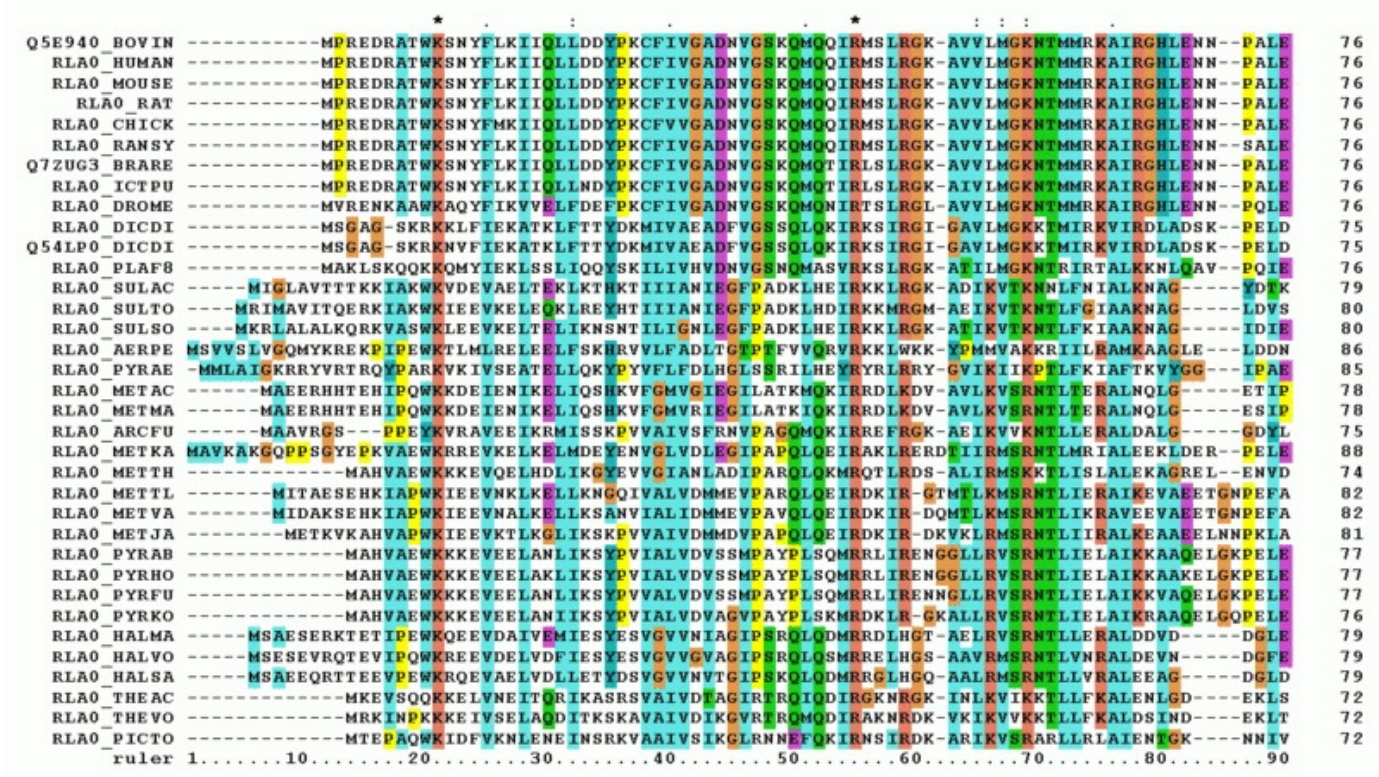
Pairwise alignments used to calculate distances (and reconstruct a tree) may lead to inconsistent picture. For example consider alignment of all pairs of 3 sequences: `CAAC`, `AACA`, `ACAA`

- `CAAC- AACA- ACAA-`
- `-AACA -ACAA -CAAC`

Which `C` in `CAAC` was in the ancestral sequence?

What is the solution to those inconsistencies?

Can we make a generalization of the pairwise alignment idea?



Note: the correspondence with edit distance is lost

image (c) P. Winter

How to score multiple alignments?

What is the natural way to score multiple alignment "quality"?

- Assume column independence (as usual)
- Sum of pairs (SP) score

$$S(m_i) = \Sigma_{k<l} s(m_i^k, m_i^l)$$

- Does it work well (for counting parsimonous mutations)? *hint: Consider a column with all characters different.*
- How much does it overestimate the number of necessary mutations?

Can we use dynamic programming?

$$\alpha_{i_1,i_2,\ldots,i_N} = \max \begin{cases} \alpha_{i_1-1,i_2-1,\ldots,i_N-1} + S(x^1_{i_1}, x^2_{i_2}, \ldots, x^N_{i_N}), \\ \alpha_{i_1,i_2-1,\ldots,i_N-1} + S(-, x^2_{i_2}, \ldots, x^N_{i_N}), \\ \alpha_{i_1-1,i_2,i_3-1,\ldots,i_N-1} + S(x^1_{i_1}, -, \ldots, x^N_{i_N}), \\ \vdots \\ \alpha_{i_1-1,i_2-1,\ldots,i_N} + S(x^1_{i_1}, x^2_{i_2}, \ldots, -), \\ \alpha_{i_1,i_2,i_3-1,\ldots,i_N-1} + S(-, -, \ldots, x^N_{i_N}), \\ \vdots \\ \alpha_{i_1,i_2-1,\ldots,i_{N-1}-1,i_N} + S(-, x^2_{i_2}, \ldots, -), \\ \vdots \end{cases}$$

image (c) Durbin et al

Sum of Pairs is NP-complete...

- Dynamic algorithm has the cost of $\mathcal{O}(n^k)$
- In general, the problem is NP-Complete
- We can still try to slightly improve its performance by looking at the lower bound of alignment score (Carillo-lipman)

$$\sigma(a) \leq S(a^{kl}) - S(\hat{a}^{kl}) + \sum_{k'<l'} S(\hat{a}^{k'l'})$$

$$S(a^{kl}) \geq \beta^{kl}$$

$$\text{where } \beta^{kl} = \sigma(a) + S(\hat{a}^{kl}) - \sum_{k'<l'} S(\hat{a}^{k'l'}).$$

image (c) Durbin et al

Carillo-Lippman lower bound method

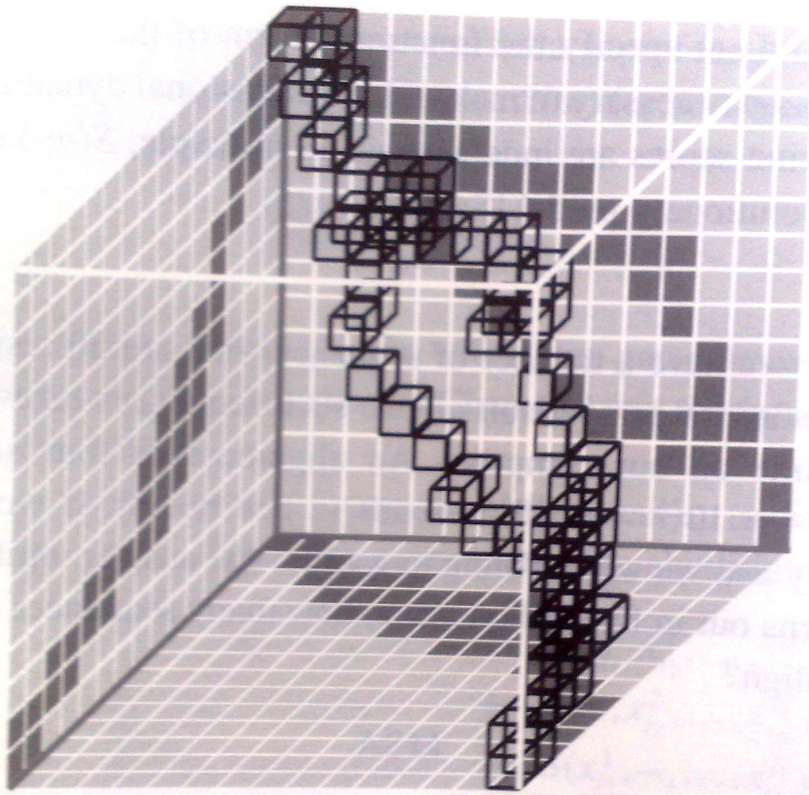


Figure 6.3 *Carrillo & Lipman's algorithm allows the search for optimal alignments to be restricted to a subset of the multidimensional programming matrix, shown here as three-dimensional. The sets B^{kl} are shown in dark grey, and the cells in the matrix to which the search can be confined are outlined in black.*

image (c) Durbin et al

Feng-Doolittle ('87) greedy approach

- We can use the greedy approach similar to UPGMA
- In each step we choose the nearest pair (using pairwise sequence distances)
- And we can merge alignments based on the pairwise alignment between the sequences
- Uses the principle of "once a gap, always a gap".
- *We need to "elegantly" align alignments of more than one sequence*

How to align profiles?

In the process of incremental alignment we need to align profiles (alignments)

$$\sum_i S(m_i) = \sum_i \sum_{k<l\le N} s(m_i^k, m_i^l)$$

$$= \sum_i \sum_{k<l\le n} s(m_i^k, m_i^l) + \sum_i \sum_{n<k<l\le N} s(m_i^k, m_i^l) + \sum_i \sum_{k\le n, n<l\le N} s(m_i^k, m_i^l).$$

image (c) Durbin et al

ClustalW

Algorithm: CLUSTALW **progressive alignment**

(i) Construct a distance matrix of all $N(N-1)/2$ pairs by pairwise dynamic programming alignment followed by approximate conversion of similarity scores to evolutionary distances using the model of Kimura [1983].

(ii) Construct a guide tree by a neighbour-joining clustering algorithm by Saitou & Nei [1987].

(iii) Progressively align at nodes in order of decreasing similarity, using sequence–sequence, sequence–profile, and profile–profile alignment. ◁

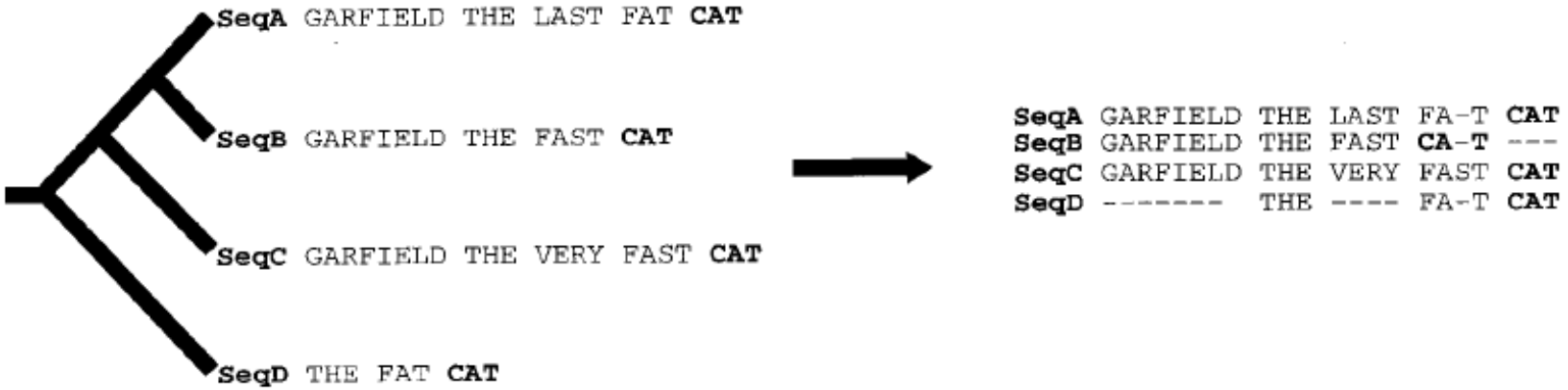
image (c) Durbin et al

ClustalW improvements (1994)

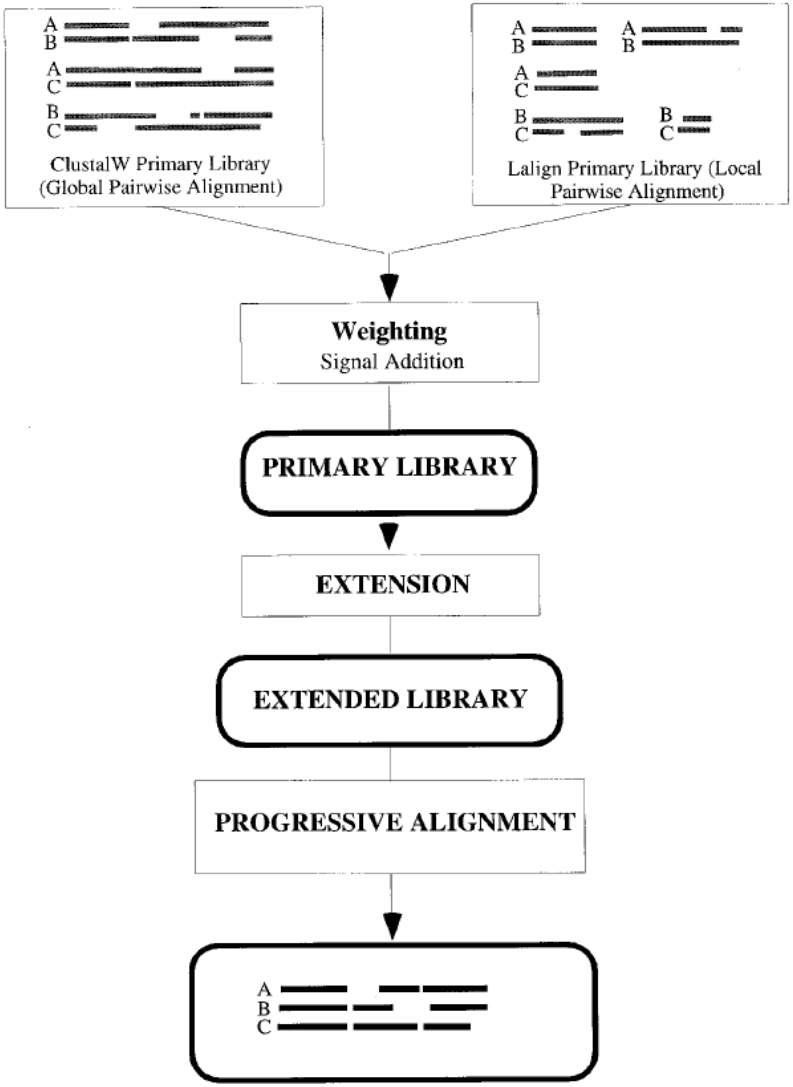
- Sequences might be weighted in the profile alignments
- Different substitution matrices might be used at different levels of merging
- Gap scores are now dependent on the AA removed i.e. $s(-, x) \neq s(-, y)$

Incremental alignment problems

a)Regular Progressive Alignment Strategy



T-Coffee algorithm for the rescue (Notredame, 2000)

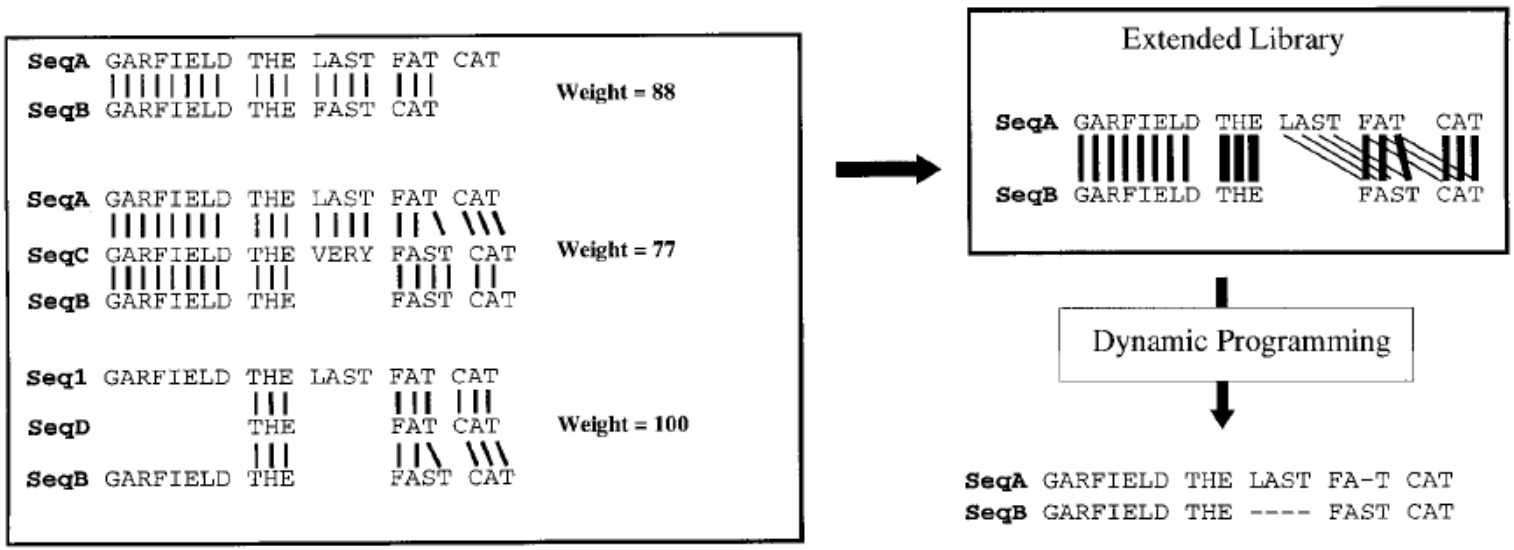


T-Coffee in more detail

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
```

c)Extended Library for seq1 and seq2



Muscle algorithm (Edgar 2004)

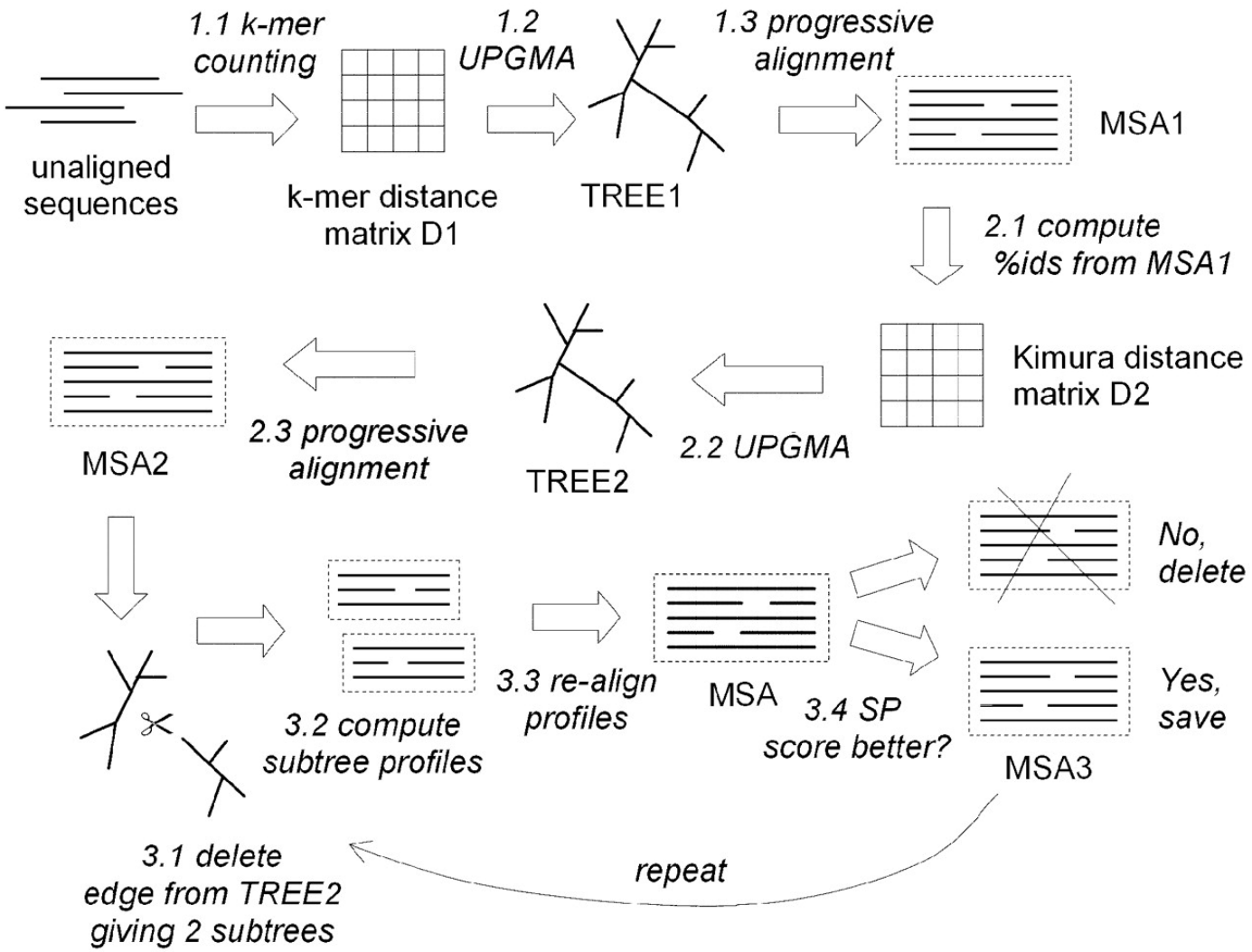


image (c) RC. Edgar, NAR 2004