Algorithmen der Bioinformatik I
WS 2017/2018

Burkhard Morgenstern
Peter Meinicke

Dept. Bioinformatics
Institute of Microbiology and Genetics (IMG)
University of Göttingen

November 20, 2017

CLUSTAL W

Most important implementation: CLUSTAL W

Nucleic Acids Research, 1994, Vol. 22, No. 22 **4673–4680**

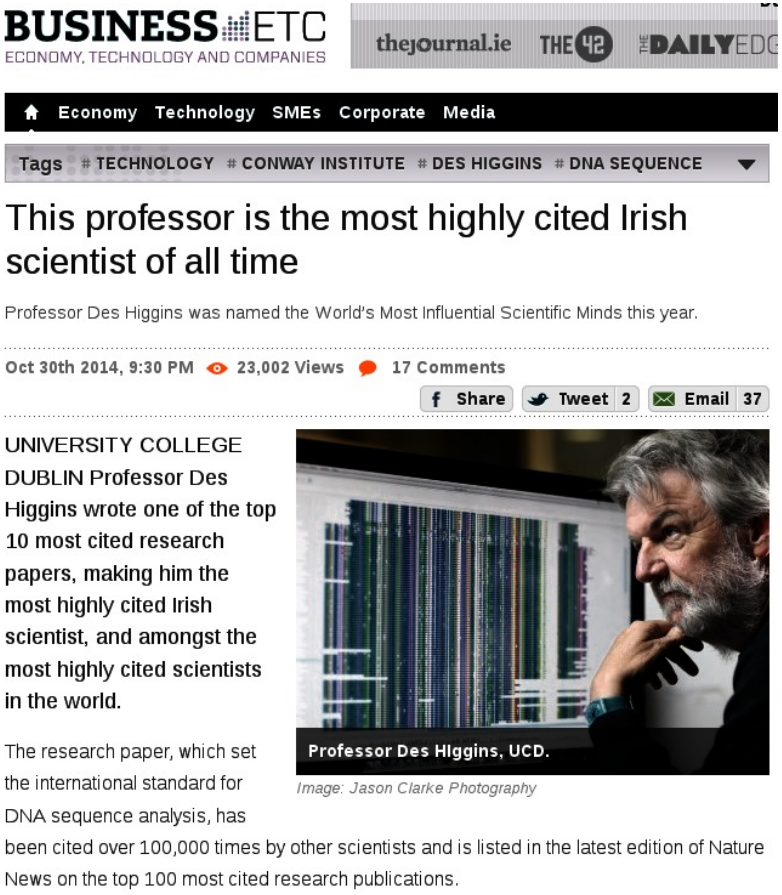
CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice

Julie D.Thompson, Desmond G.Higgins+ and Toby J.Gibson*
European Molecular Biology Laboratory, Postfach 102209, Meyerhofstrasse 1, D-69012 Heidelberg, Germany

Received July 12, 1994; Revised and Accepted September 23, 1994

45.933 citations in the scientific literature (Web of Science)

CLUSTAL W



BUSINESS ETC
ECONOMY, TECHNOLOGY AND COMPANIES

thejournal.ie THE 42 THE DAILY EDG

Economy Technology SMEs Corporate Media

Tags # TECHNOLOGY # CONWAY INSTITUTE # DES HIGGINS # DNA SEQUENCE

This professor is the most highly cited Irish scientist of all time

Professor Des Higgins was named the World's Most Influential Scientific Minds this year.

Oct 30th 2014, 9:30 PM 23,002 Views 17 Comments

Share Tweet 2 Email 37

UNIVERSITY COLLEGE DUBLIN Professor Des Higgins wrote one of the top 10 most cited research papers, making him the most highly cited Irish scientist, and amongst the most highly cited scientists in the world.

Professor Des Higgins, UCD.

Image: Jason Clarke Photography

The research paper, which set the international standard for DNA sequence analysis, has been cited over 100,000 times by other scientists and is listed in the latest edition of Nature News on the top 100 most cited research publications.

Figure: Des Higgins, Dublin (`http://businessetc.thejournal.ie/`)

CLUSTAL W

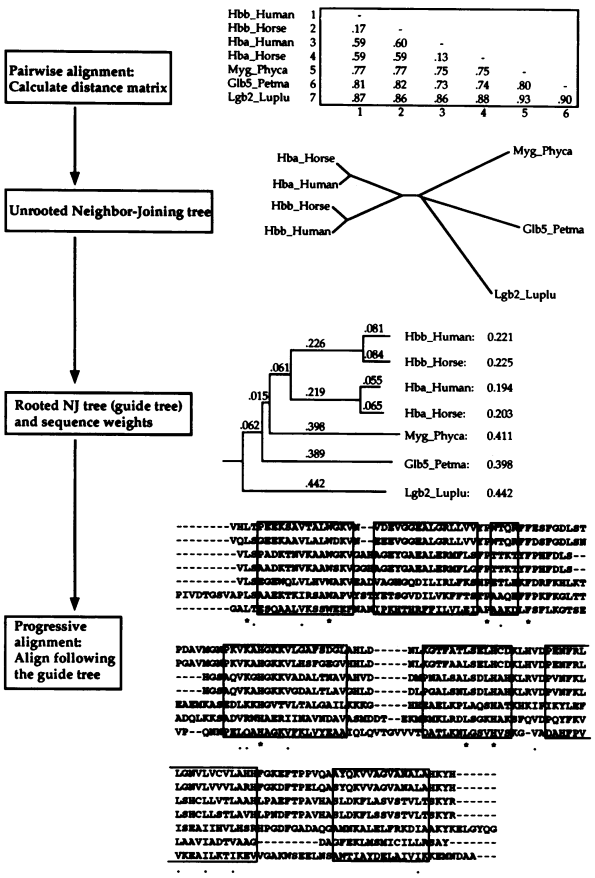


Figure: Progressive Alignment in CLUSTAL W (Thompson *et al*, 1994)

CLUSTAL W

```
1   peeksavtal
2   geekaavlal
3   padktnvkaa
4   aadktnvkaa

5   egewqlvlhv
6   aaektkirsa
```

Without sequence Weights:

$$
\begin{aligned}
\text{Score} = \; & M(t,v) \\
+ \; & M(t,i) \\
+ \; & M(l,v) \\
+ \; & M(l,i) \\
+ \; & M(k,v) \\
+ \; & M(k,i) \\
+ \; & M(k,v) \\
+ \; & M(k,i)/8
\end{aligned}
$$

With sequence Weights W_i:

$$
\begin{aligned}
\text{Score} = \; & M(t,v)*W_1*W_5 \\
+ \; & M(t,i)*W_1*W_6 \\
+ \; & M(l,v)*W_2*W_5 \\
+ \; & M(l,i)*W_2*W_6 \\
+ \; & M(k,v)*W_3*W_5 \\
+ \; & M(k,i)*W_3*W_6 \\
+ \; & M(k,v)*W_4*W_5 \\
+ \; & M(k,i)*W_4*W_6/8
\end{aligned}
$$

Figure: Sequence weighting in profile alignment (Thompson *et al*, 1994)

CLUSTAL W

Gap penalties in *CLUSTAL W*

- Use affine-linear gap penalties with *gap opening penalty (GOP)* and *gap extension penalty (GEP)*
- Initial values of *GOP* and *GEP* specified by user
- During progressive alignment, *GOP* and *GEP* modified depending on
  - Substitution matrix
  - Similarity between sequences
  - Length of sequences
  - Differences in sequence lengths
  - Local sequence composition (hydrophilic or hydrophobic amino acid residues)
  - Existing gaps
  - Position in sequence: end gaps get lower penalty

CLUSTAL W

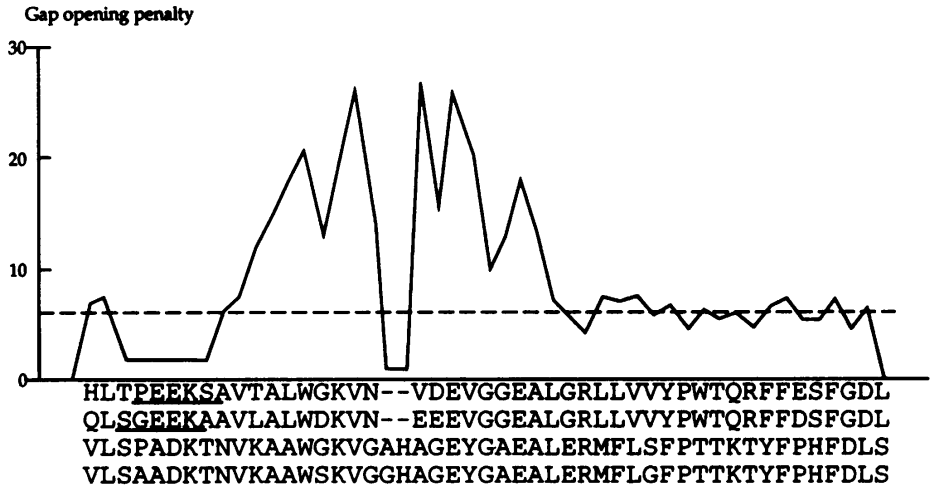
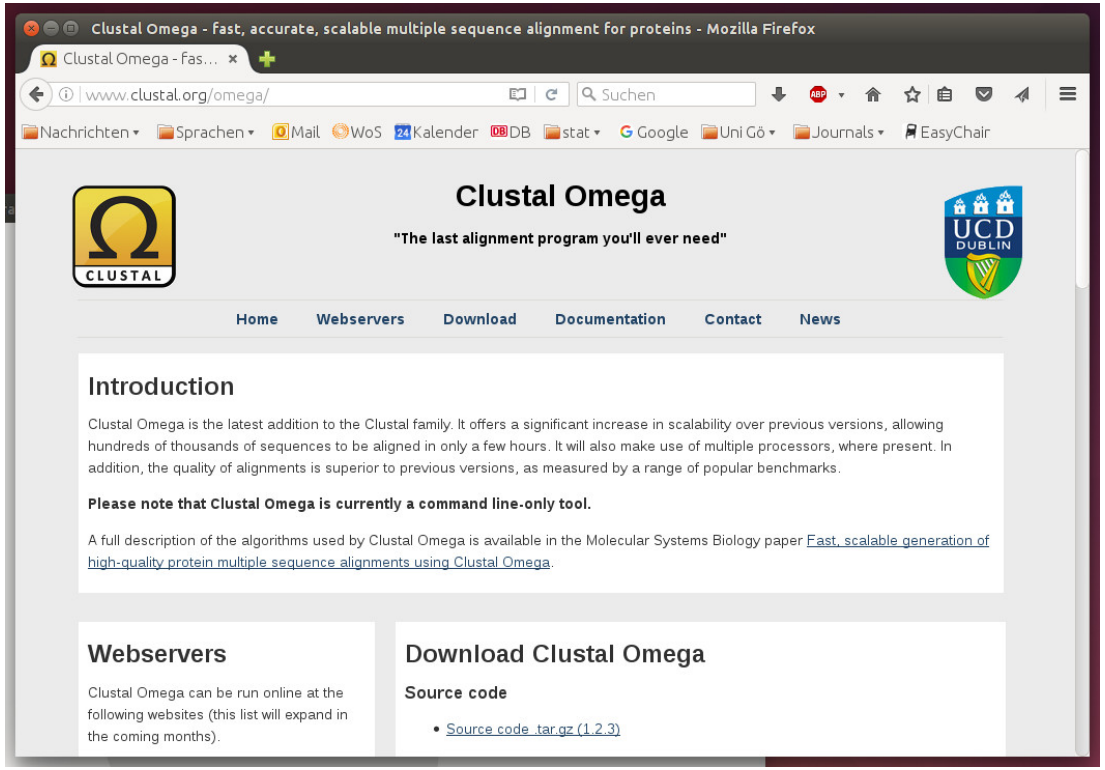


Figure: Variable gap penatlies depending on sequence composition, existing gaps etc. (Thompson *et al*, 1994)

CLUSTAL Ω



Latest version of *CLUSTAL*: align up to 100.000 sequences.
Based on fast clustering algorithm

Multiple Alignment using *MUSCLE*

1792–1797 *Nucleic Acids Research, 2004, Vol. 32, No. 5*
DOI: 10.1093/nar/gkh340

MUSCLE: multiple sequence alignment with high accuracy and high throughput

Robert C. Edgar*

195 Roque Moraes Drive, Mill Valley, CA 94941, USA

Received January 19, 2004; Revised January 30, 2004; Accepted February 24, 2004

ABSTRACT

We describe MUSCLE, a new computer program for creating multiple alignments of protein sequences. Elements of the algorithm include fast distance estimation using *k*mer counting, progressive alignment using a new profile function we call the log-expectation score, and refinement using tree-dependent restricted partitioning. The speed and

variant on this strategy is used by T-Coffee (5), which aligns profiles by optimizing a score derived from local and global alignments of all pairs of input sequences. Misalignments by progressive methods are sometimes readily apparent (Fig. 1), motivating further processing (refinement). For a recent review of multiple alignment methods, see Notredame (6). Here we describe MUSCLE (multiple sequence comparison by log-expectation), a new computer program for multiple protein sequence alignment.

Multiple Alignment using *MUSCLE*

Most time-consuming step in progressive alignment: Calculation of *guide tree*.

To calculate distances for N sequences of length ℓ:

$O(N^2)$ pairwise alignments calculated, total complexity

$$O(N^2 \cdot \ell^2)$$

Calculating guide tree with *Neighbour-Joining* takes

$$O(N^3)$$

time

Multiple Alignment using *MUSCLE*

Strategy of *MUSCLE* (1):

- Calculate pairwise similarities of sequences using k-mer occurrences (k-mer = word of length k)
- Turn similarity values into distance values $d_{i,j}$
- Calculate guide tree using *UPGMA*
- Progressive alignment

Multiple Alignment using *MUSCLE*

To calculate distances $d_{i,j}$ for sequences X_i, X_j of length ℓ_i, ℓ_j using k-mer frequencies:

Define for k-mer τ:

- $n_i(\tau)$ = frequency of τ in X_i
- $n_j(\tau)$ = frequency of τ in X_j

and 'k-mer similarity' as

$$F_{i,j} = \frac{\sum_\tau \min\left\{n_i(\tau), n_j(\tau)\right\}}{\min\left\{\ell_i, \ell_j\right\} - k + 1}$$

Distance between X_i and X_j defined as

$$d_{i,j} = 1 - F_{i,j}$$

Multiple Alignment using *MUSCLE*

In general: *NJ* produces phylogenetically more accurate trees than *UPGMA* since it can deal with different mutation rates

But: test runs showed that *UPGMA* may be superior to *NJ* to construct guide trees.

Possible reason: Alignment most accurate if sequences closely related

Multiple Alignment using *MUSCLE*

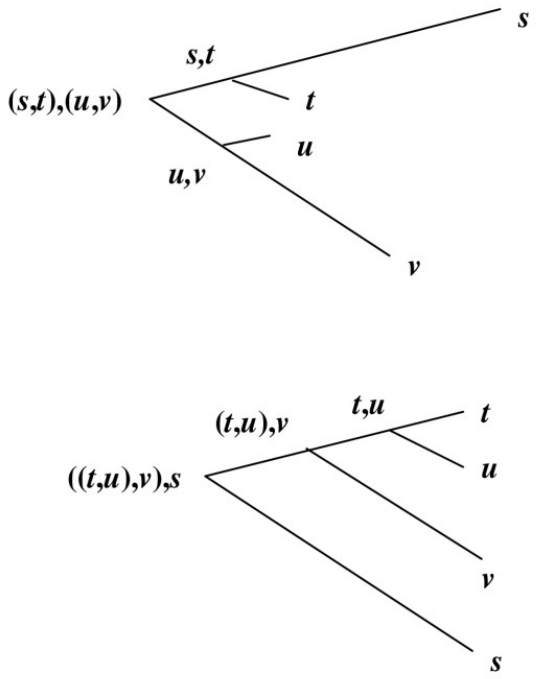


Figure: Wrong branching order in UPGMA tree (Edgar, 2004)

Multiple Alignment using *MUSCLE*

Strategy of *MUSCLE* (2):

- Iterative improvement of alignment:
  - Use Kimura distances to calculate new guide tree (Kimura distance estimates number of substitutions based on *observed* mismatches in alignment)
  - For nodes that are different in new tree, re-calculate progressive alignment

Multiple Alignment using *MUSCLE*

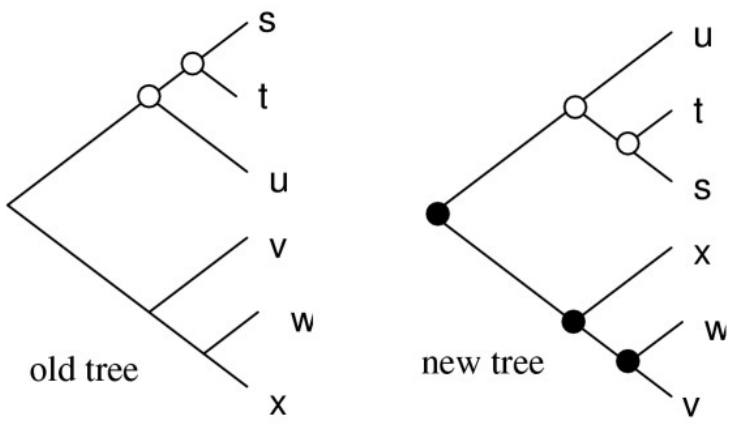


Figure: Progressive alignment newly calculated only if branching order is different: black nodes in new tree (Edgar, 2004)

Multiple Alignment using *MUSCLE*

Strategy of *MUSCLE* (3):

- Refinement of MSA
  - Partition sequences into two groups
  - Re-align profiles from the two groups
  - Accept new alignment, if score is improved

Multiple Alignment using *MUSCLE*

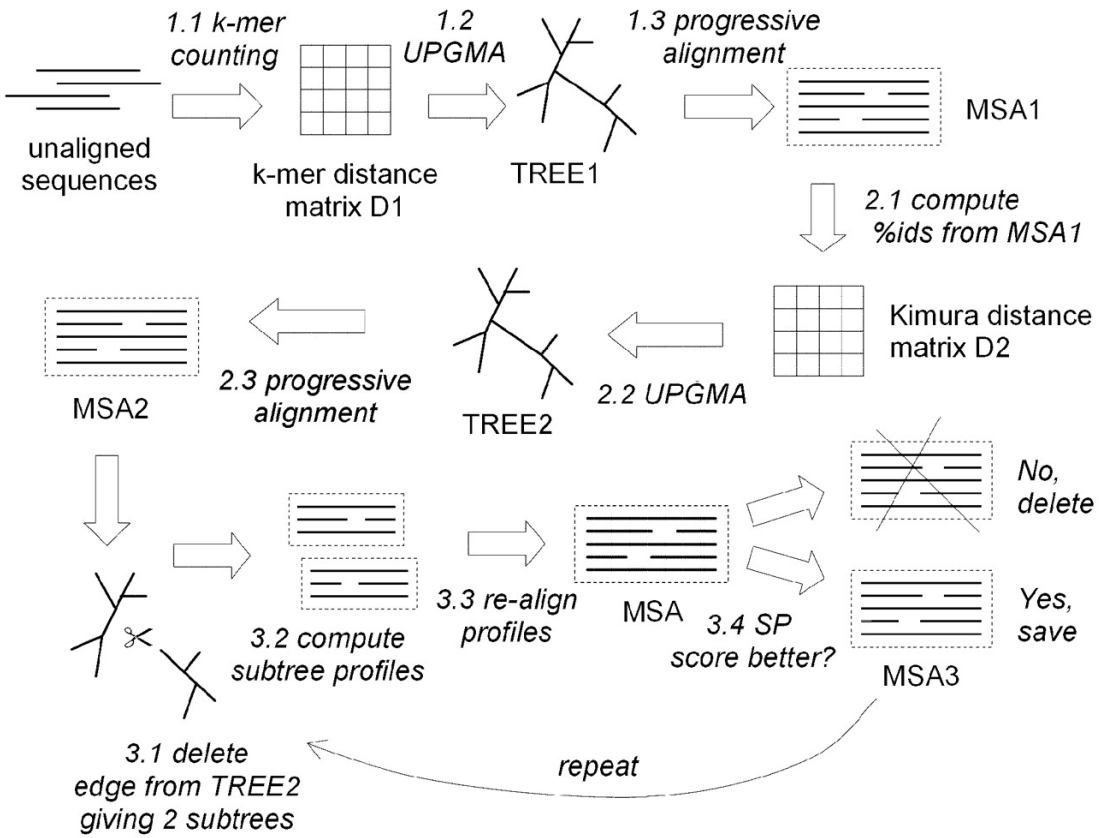


Figure: Multiple alignment with *MUSCLE* (Edgar, 2004)

Multiple alignment with DIALIGN

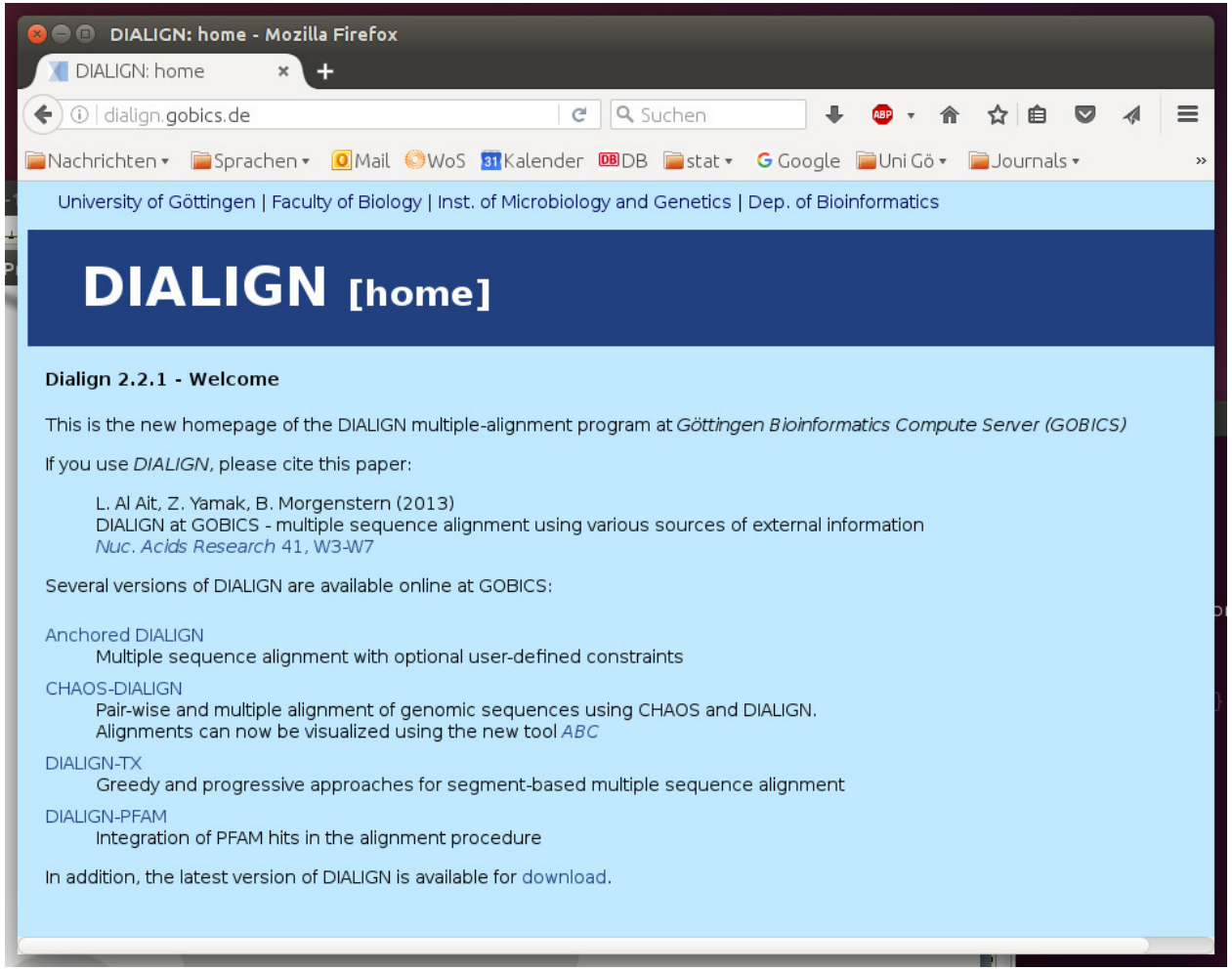


Figure: DIALIGN at GOBICS

Multiple alignment with DIALIGN

Idea: combine global and local alignment.

- Search for local pairwise similarities
- Compose alignments as *consistent* set of local pairwise alignments
- Ignore non-related parts of sequences
- No gap penalty

Local pairwise gap-free alignment called *fragment*

Multiple alignment with DIALIGN

S_1	Y	I	A	V	L	F	A	E	D	
S_2	L	A	C	V	I	F	G	S		
S_3	P	W	D	D	V	T	F	D	A	E

Figure: MSA composed of *fragments*, *i.e.* gap-free pairwise alignments

Morgenstern *et al.* (1999), *Bioinformatics* 15, 211-218

Multiple alignment with DIALIGN

S_1	Y	I	A	V	L	F	A	E	D	
S_2	L	A	C	V	I	F	G	S		
S_3	P	W	D	D	V	T	F	D	A	E

Figure: MSA composed of *fragments*, *i.e.* gap-free pairwise alignments

Morgenstern *et al.* (1999), *Bioinformatics* 15, 211-218

Multiple alignment with DIALIGN

S_1	Y	I	A	V	L	F	A	E	D	
S_2	L	A	C	V	I	F	G	S		
S_3	P	W	D	D	V	T	F	D	A	E

Figure: MSA composed of *fragments*, *i.e.* gap-free pairwise alignments

Morgenstern *et al.* (1999), *Bioinformatics* 15, 211-218

Multiple alignment with DIALIGN

S_1	Y	I	A	V	L	F	A	E	D	
S_2	L	A	C	V	I	F	G	S		
S_3	P	W	D	D	V	T	F	D	A	E

Figure: MSA composed of *fragments*, *i.e.* gap-free pairwise alignments

Morgenstern *et al.* (1999), *Bioinformatics* 15, 211-218

Multiple alignment with DIALIGN

S_1	Y	I	A	V	L	F	A	E	D	
S_2	L	A	C	V	I	F	G	S		
S_3	P	W	D	D	V	T	F	D	A	E

Figure: MSA composed of *fragments*, *i.e.* gap-free pairwise alignments

Morgenstern *et al.* (1999), *Bioinformatics* 15, 211-218

Multiple alignment with DIALIGN

S_1	Y	I	A	-	V	L	F	-	A	E	D
S_2	-	L	A	C	V	I	F	-	G	S	-
S_3	P	W	D	D	V	T	F	D	A	E	-

Figure: MSA as *consistent* set of fragments

Morgenstern *et al.* (1999), *Bioinformatics* 15, 211-218

Multiple alignment with DIALIGN

S_1	y	I	A	-	V	L	F	-	A	E	d
S_2	-	L	A	c	V	I	F	-	G	s	-
S_3	p	w	d	d	V	T	F	d	A	E	-

Figure: Resulting MSA: non-aligned positions shown in lower case

Morgenstern *et al.* (1999), *Bioinformatics* 15, 211-218

Multiple alignment with DIALIGN

- Weight score of fragment f:

$$w(f) = -\log Pr(f)$$

  $Pr(f) =$ probability of occurrence of fragment f in random sequences of same length

- Score of alignment: sum of weight scores of fragments in alignment - no gap penalty!

- Optimization problem: Find *consistent* set of fragments with max total weight, i.e. set of fragments that fits into one single multiple alignment.

Multiple alignment with DIALIGN

- Algorithm:
  - Calculate all pairwise optimal alignments (= chains of fragments)

    $$\mathcal{M}_1 := \text{ set of fragments from pairwise optimal alignments}$$

  - Calculate *overlap weights* for fragments from $\mathcal{M}_1$ depending on weights of overlapping fragments.
  - Sort set $\mathcal{M}_1$ of fragments according to 'overlap weights'
  - *Greedily* select consistent subset $\mathcal{M}_2$ of $\mathcal{M}_1$
  - Repeat iteratively, given consistency constraints imposed by $\mathcal{M}_2$

Multiple alignment with DIALIGN

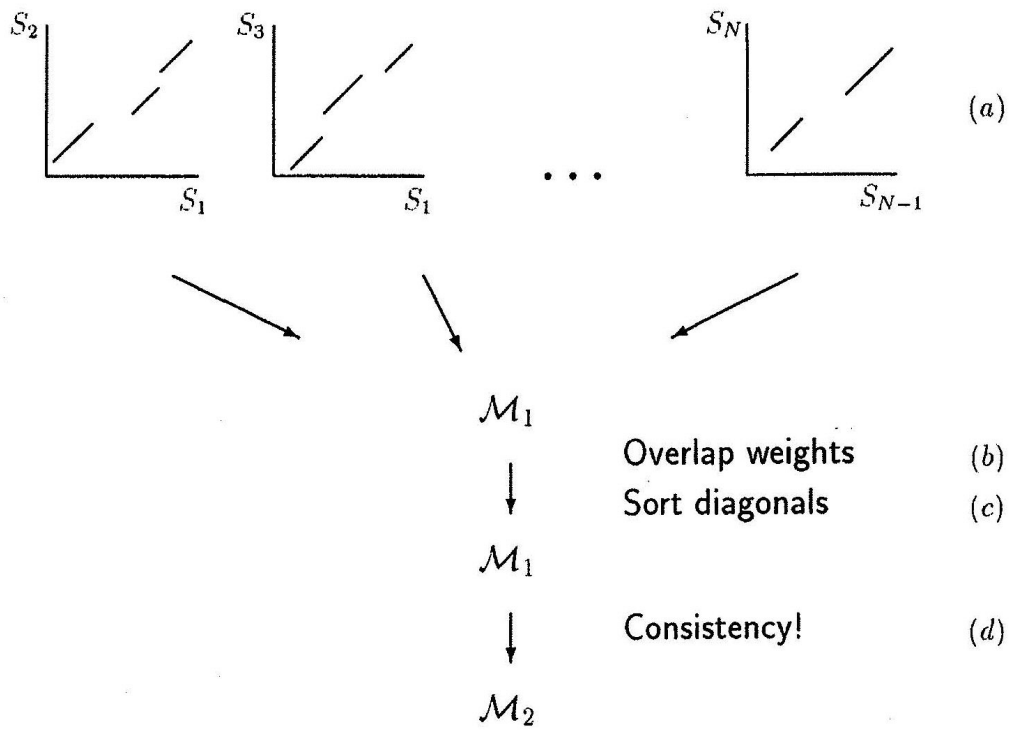


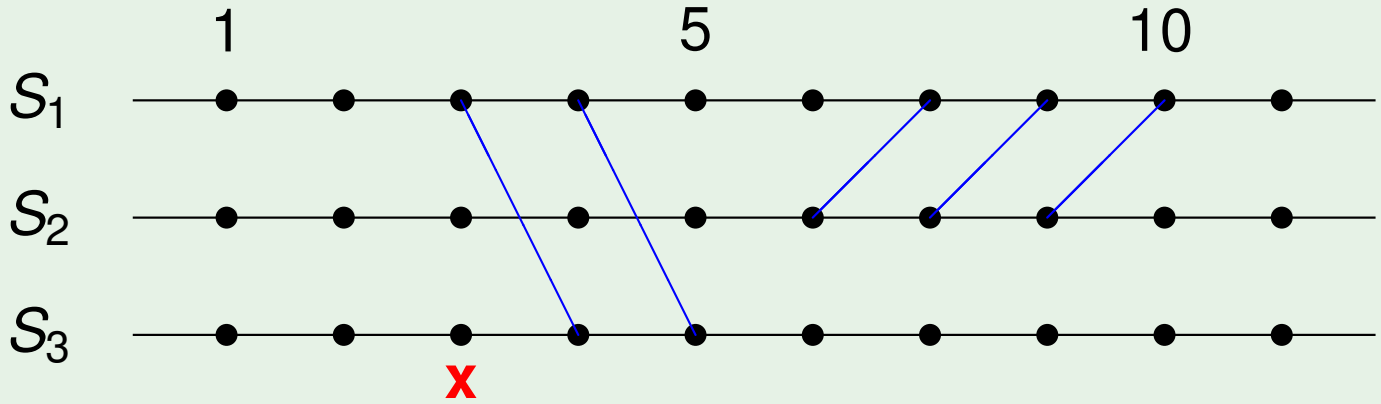
Figure: Greedy algorithm to calculate MSA, iteratively applied

Morgenstern *et al.* (1999), *Bioinformatics* 15, 211-218

Multiple alignment with DIALIGN

To decide if new fragment is consistent:
use *consistency bounds* $\underline{b}(x, i)$ and $\overline{b}(x, i)$

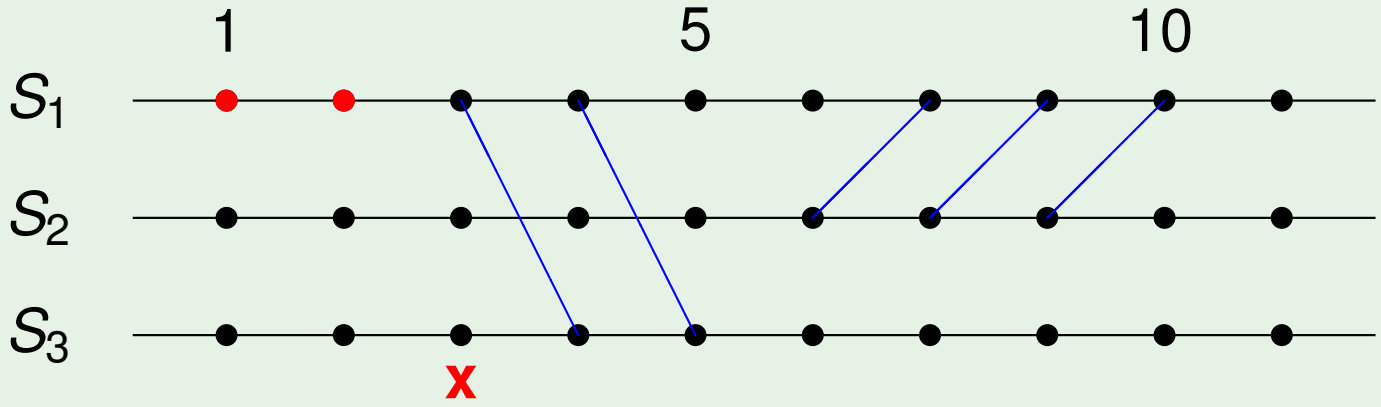
Example



Multiple alignment with DIALIGN

To decide if new fragment is consistent:
use *consistency bounds* $\underline{b}(x, i)$ and $\overline{b}(x, i)$

Example



$\underline{b}(x, 1) = 1 \qquad \overline{b}(x, 1) = 2$

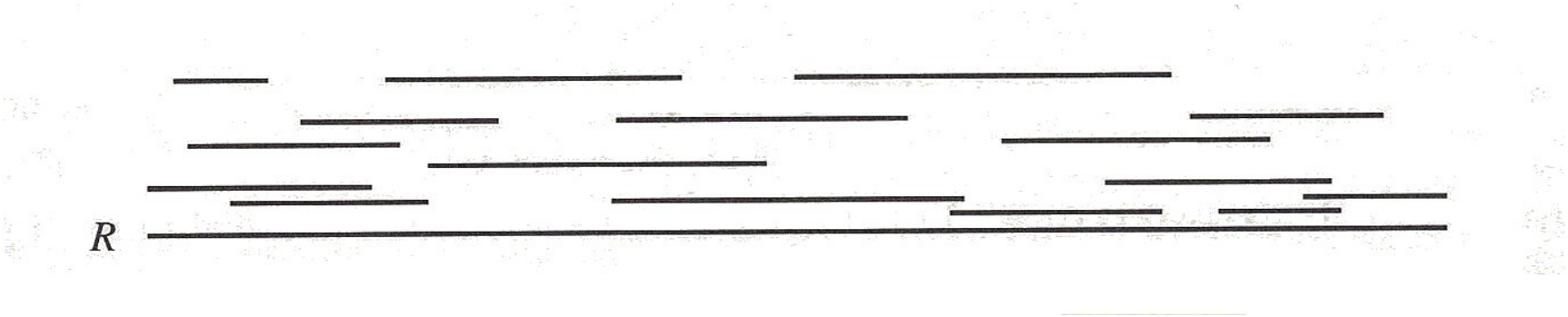
Multiple alignment with DIALIGN

Results for DIALIGN:

- Better than other methods for *locally* related sequences
- Inferior on *globally* related sequences

Fragment chaining

One-dimensional chaining problem:

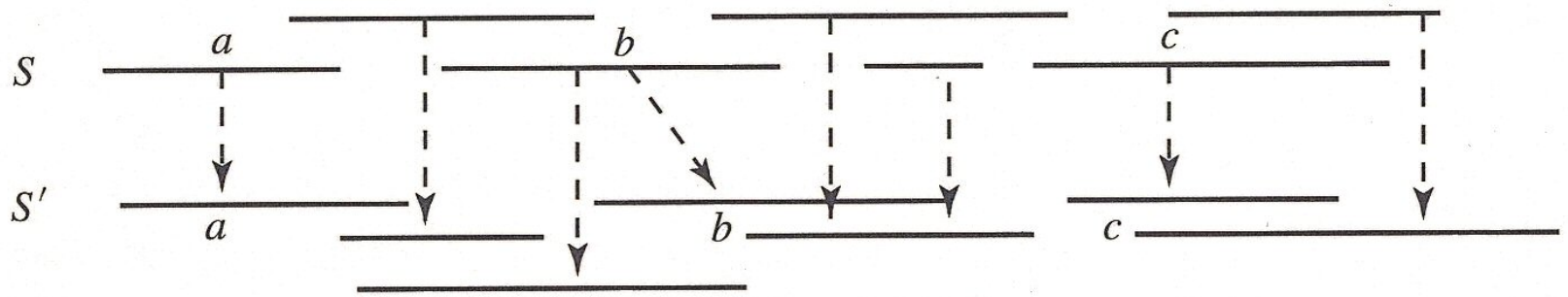


Find a best chain of weighted intervals ('fragments')

For input set of n fragments, best chain can be found in $O(n \cdot \log n)$ time by *dynamic programming*

Fragment chaining

Two-dimensional chaining problem:



Find best chain of two-dimensional fragments

Also solvable in $O(n \cdot \log n)$ time.

Fragment chaining

In sequence alignment, number n of fragments can be large.
Space-efficient algorithm possible:

- Go *column-wise* through *DP* matrix
- For i-th column, calculate arrays $S[i]$ and $L[i]$ with *score* and *last* fragment in longest chain up to (i, j) for each j.
- For new column i, calculate weight $w(f)$ for each fragment f *starting* in i. Calculate total weight $W(f)$ of optimal chain ending in f and its 'predecessor' $P(f)$ using arrays $S[i-1]$ and $L[i-1]$:

$$\begin{aligned} W(f) &= w(f) + W[i-1, j-1] \\ P(f) &= P[i-1, j-1] \end{aligned}$$

Fragment chaining

- For new fragment ending in column i', update list $E[i']$ of fragments *ending* in column i'
- After fragments starting in column i have been processed: calculate $S[i]$ and $L[i]$ from $S[i-1]$, $L[i-1]$ and $E[i]$. Delete $S[i-1]$, $L[i-1]$.
- Maintain fragment f^* in which best chain so far ends
- Finally: start *trace back* at f^*

Evaluation of Multiple Protein Alignment Software

```
1aboA     1    .NLFVALYDfvasgdntlsitkGEKLRVLgynhn..............gE
1ycsB     1    kGVIYALWDyepqnddelpmkeGDCMTIIhrede............deiE
1pht      1    gYQYRALYDykkereedidlhlGDILTVNkgslvalgfsdgqearpeeiG
1ihvA     1    .NFRVYYRDsrd......pvwkGPAKLLWkg..................eG
1vie      1    .drvrkksga.........awqGQIVGWYctnlt.............peG

1aboA    36    WCEAQt..kngqGWVPSNYITPVN......
1ycsB    39    WWWARl..ndkeGYVPRNLLGLYP......
1pht     51    WLNGYnettgerGDFPGTYVEYIGrkkisp
1ihvA    27    AVVIQd..nsdiKVVPRRKAKIIRd.....
1vie     28    YAVESeahpgsvQIYPVAALERIN......
```

Key

alpha helix RED
beta strand GREEN
core blocks UNDERSCORE

BAliBASE
Reference alignments

Figure: Reference alignment with *core blocks* from *BAliBASE*

Thompson *et al.* (1999), *Bioinformatics* 15, 87–88

Evaluation of Multiple Protein Alignment Software

Benchmark alignments contain reliable *core blocks*

For Evaluation:

- Sum-of-Pairs (SP) score: ration of correctly aligned pairs of positions in core blocks
- Total-column (TC) score: ration of correctly aligned columns in core blocks

Evaluation of Multiple Protein Alignment Software

Results:

- DIALIGN best method for local MSA
- Outperformed by other methods on weakly but globally related sequences