Algorithmen der Bioinformatik I
WS 2017/2018

Burkhard Morgenstern
Peter Meinicke

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

November 21, 2017

Multiple alignment with DIALIGN

Compose alignments from local gap-free pairwise alignments (‘fragments’)

S_1	Y	I	A	V	L	F	A	W	E	D	I	R	
S_2	L	A	C	V	I	F	G	S	D	V	R	A	V

Figure: Pairwise alignment as *chain of ‘fragments’*

Multiple alignment with DIALIGN

Compose alignments from local gap-free pairwise alignments ('fragments')

S_1	Y	I	A	V	L	F	A	W	E	D	I	R	
S_2	L	A	C	V	I	F	G	S	D	V	R	A	V

Figure: Pairwise alignment as *chain of 'fragments'*

Multiple alignment with DIALIGN

Compose alignments from local gap-free pairwise alignments (‘fragments’)

S_1	Y	I	A	V	L	F	A	W	E	D	I	R	
S_2	L	A	C	V	I	F	G	S	D	V	R	A	V

Figure: Pairwise alignment as *chain of ‘fragments’*

Multiple alignment with DIALIGN

Compose alignments from local gap-free pairwise alignments (‘fragments’)

S_1	Y	I	A	V	L	F	A	W	E	D	I	R	
S_2	L	A	C	V	I	F	G	S	D	V	R	A	V

Figure: Pairwise alignment as *chain of ‘fragments’*

Multiple alignment with DIALIGN

Compose alignments from local gap-free pairwise alignments (‘fragments’)

S_1	Y	I	A	–	V	L	F	A	W	E	D	I	R	–	–
S_2	–	L	A	C	V	I	F	G	S	–	D	V	R	A	V

Figure: Pairwise alignment as *chain of ‘fragments’*

Multiple alignment with DIALIGN

Compose alignments from local gap-free pairwise alignments (‘fragments’)

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: Multiple alignment as *‘consistent’ set of fragments*

Multiple alignment with DIALIGN

Compose alignments from local gap-free pairwise alignments (‘fragments’)

```
S1    Y  I  A  V  L  F  A  E  D
S2    L  A  C  V  I  F  G  S
S3    P  W  D  D  V  T  F  D  A  E
```

Figure: Multiple alignment as *‘consistent’ set of fragments*

Multiple alignment with DIALIGN

Compose alignments from local gap-free pairwise alignments ('fragments')

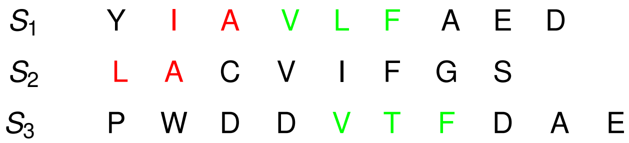
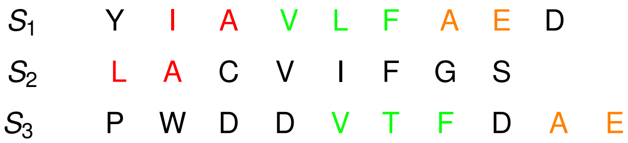


Figure: Multiple alignment as *'consistent' set of fragments*

Multiple alignment with DIALIGN

Compose alignments from local gap-free pairwise alignments (‘fragments’)



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: Multiple alignment as *‘consistent’ set of fragments*

Multiple alignment with DIALIGN

Compose alignments from local gap-free pairwise alignments (‘fragments’)

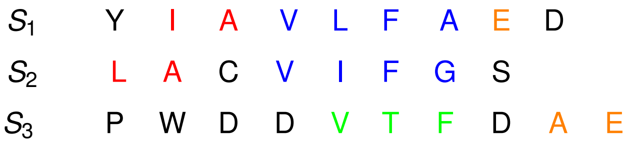


Figure: Multiple alignment as *‘consistent’ set of fragments*

Multiple alignment with DIALIGN

Compose alignments from local gap-free pairwise alignments (‘fragments’)

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: Multiple alignment as *‘consistent’ set of fragments*

Multiple alignment with DIALIGN

Compose alignments from local gap-free pairwise alignments (‘fragments’)

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: Multiple alignment as *‘consistent’ set of fragments*

Multiple alignment with DIALIGN

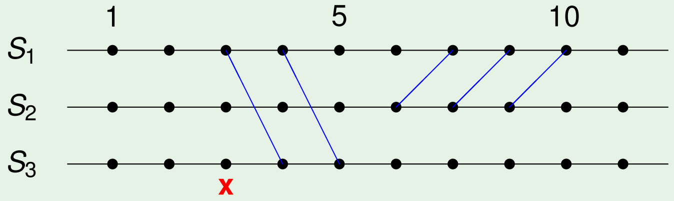
- Define ‘weight’ for each possible ‘fragment’
- Goal: find ‘consistent’ set of ‘fragments’ with maximum total weight

For multiple alignment: use *greedy* heuristic: include fragments from optimal pairwise alignments one-by-one.

Multiple alignment with DIALIGN

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

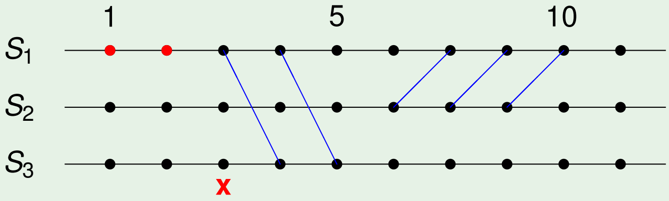
Example



Multiple alignment with DIALIGN

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

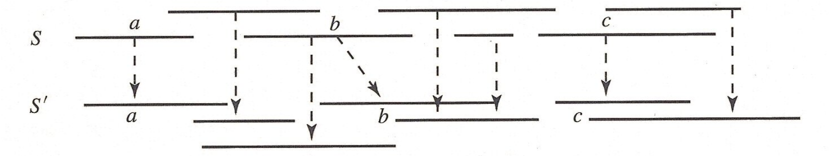
Example



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

Fragment chaining

Two-dimensional chaining problem:



Calculate weights of all possible fragments; find best fragment chain in $O(n \cdot \log n)$ time.

But: number n of fragments can be large. Therefore, use space-efficient algorithm

Fragment chaining

- Go *column-wise* through *DP* matrix
- Calculate arrays $W(i,j)$ and $L(i,j)$ as *score* and *last* fragment of optimal chain up to (i,j).
- For each fragment f *starting* in (i,j):
    - Calculate weight $w(f)$
    - Calculate total weight $W(f)$ of optimal chain ending in f and its ‘predecessor’ $P(f)$

Fragment chaining

Recursion

$$\begin{aligned} W(f) &= w(f) + W(i-1, j-1) \\ P(f) &= P(i-1, j-1) \\ W(i,j) &= \max \left\{ \begin{array}{l} W(i-1, j) \\ W(i, j-1) \\ \max_{f' \text{ ending in } (i,j)} W(f') \end{array} \right\} \end{aligned}$$

Fragment chaining

- For new fragment ending in column i', update list of fragments *ending* in column i'
- After fragments starting in column i have been processed: values $W(i,j)$ and $P(i,j)$ can be deleted
- Maintain fragment f^* in which best chain so far ends
- Finally: start *trace back* at f^*

Anchored alignment

Idea: use user-defined anchor points for constrained alignment. Program *forced* to align anchor points - provided they are consistent with each other.

- User-defined anchor points as *fragments*, i.e. ungapped local alignments
- Anchor points sorted according to user-defined *weights*
- Greedy consistency algorithm used to select consistent set of anchor points and to define

  $$\underline{b}(x, i) \text{ and } \overline{b}(x, i)$$

  before main alignment procedure starts ('0-th iteration')

Anchored alignment

Anchor 1:	1	2	72	80	4	4.5
Anchor 2:	1	5	140	115	3	3.8
⋮	2	3	84	80	5	5.3
	2	4	130	114	12	12.1
	3	6	93	89	10	10.9
	4	5	119	103	6	6.0
	1	2	90	5	4	4.2
	1	2	124	38	4	4.7

Figure: Anchor points defined by 6 coordinates: sequences, beginning positions, length, score

Anchored alignment

Main applications:

- Use expert knowledge for improved alignment quality
- Speed-up alignment for alignment of genomic sequences
- Test new versions of the program: *e.g.* define new weight scores to modify ordering of fragments in greedy algorithm