Multiple Sequence Alignment

Burkhard Morgenstern
Universität Göttingen
Institut für Mikrobiologie und Genetik
Göttingen, March 2005

Introduction

Multiple Sequence Alignment (MSA) most important tool for sequence analysis in Molecular Biology and Genomics:

Introduction

Multiple Sequence Alignment (MSA) most important tool for sequence analysis in Molecular Biology and Genomics:

- Evolution of genes and species
- Structure of proteins and RNA
- Detection of functional sites in sequences
- Analysis of genomic sequences
- Database searching

Introduction

Seq 1	NLFVALYDFVASGDNTLSITKGEKLRVLGYNHN
Seq 2	KGVIYALWDYEPQNDDELPMKEGDCMTIIHREDE
Seq 3	GYQYRALYDYKKEREEDIDLHLGDILTVNKGSLVALGFS
Seq 4	NFRVYYRDSRDPVWKGPAKLLWKG
Seq 5	DRVRKKSGAAWQGQIVGWYCTNLT

Introduction

```
Seq 1   NLFVALYDFVASGDNTLSITKGEKLRVLGYNHN
Seq 2   KGVIYALWDYEPQNDDELPMKEGDCMTIIHREDE
Seq 3   GYQYRALYDYKKEREEDIDLHLGDILTVNKGSLVALGFS
Seq 4   NFRVYYRDSRDPVWKGPAKLLWKG
Seq 5   DRVRKKSGAAWQGQIVGWYCTNLT
```

Input: set of sequence data

Introduction

```
Seq 1   NLFVALYDFVASGDNTLSITKGEKLRVLGYNHN
Seq 2   KGVIYALWDYEPQNDDELPMKEGDCMTIIHREDE
Seq 3   GYQYRALYDYKKEREEDIDLHLGDILTVNKGSLVALGFS
Seq 4   NFRVYYRDSRDPVWKGPAKLLWKG
Seq 5   DRVRKKSGAAWQGQIVGWYCTNLT
```

Input: set of sequence data

Goal: align *biologically* related residues!
= residues related by structure, function, evolution

Introduction

```
Seq 1   -NLFVALYDfvasgdntlsitkGEKLRVLgynhn-----
Seq 2   kGVIYALWDyepqnddelpmkeGDCMTIIhrede-----

Seq 3   gYQYRALYDykkereedidlhlGDILTVNkgslvalgfs
Seq 4   -NFRVYYRDsrd------pvwkGPAKLLWkg--------


Seq 5   -drvrkksga---------awqGQIVGWYctnlt-----
```

Input: set of sequence data

Goal: align *biologically* related residues!
 = residues related by structure, function, evolution

Introduction

```
Seq 1    -NLFVALYDfvasgdntlsitkGEKLRVLgynhn-----
Seq 2    kGVIYALWDyepqnddelpmkeGDCMTIIhrede-----

Seq 3    gYQYRALYDykkereedidlhlGDILTVNkgslvalgfs
Seq 4    -NFRVYYRDsrd------pvwkGPAKLLWkg--------

Seq 5    -drvrkksga---------awqGQIVGWYctnlt-----
```

Input: set of sequence data

Goal: align *biologically* related residues!

= residues related by structure, function, evolution

Introduction

Most multi-alignment approaches *automated,* i.e. based on algorithmic rules.

Introduction

Most multi-alignment approaches *automated,* i.e. based on algorithmic rules.

Two components:

Introduction

Most multi-alignment approaches *automated,* i.e. based on algorithmic rules.

Two components:

- *Objective function:* assess alignment quality

Introduction

Most multi-alignment approaches *automated,* i.e. based on algorithmic rules.

Two components:

- *Objective function:* assess alignment quality
- *Optimization algorithm:* find optimal or near-optimal alignment

Introduction

Objective functions *far* more important than optimization algorithms!

Introduction

Fully automated alignment programs necessary

Introduction

Fully automated alignment programs necessary

- If no expert knowledge available

Introduction

Fully automated alignment programs necessary

- If no expert knowledge available
- If large amounts of data to be analyzed

Tools for multiple sequence alignment

First question:

What is a good alignment?

(f) biologically?

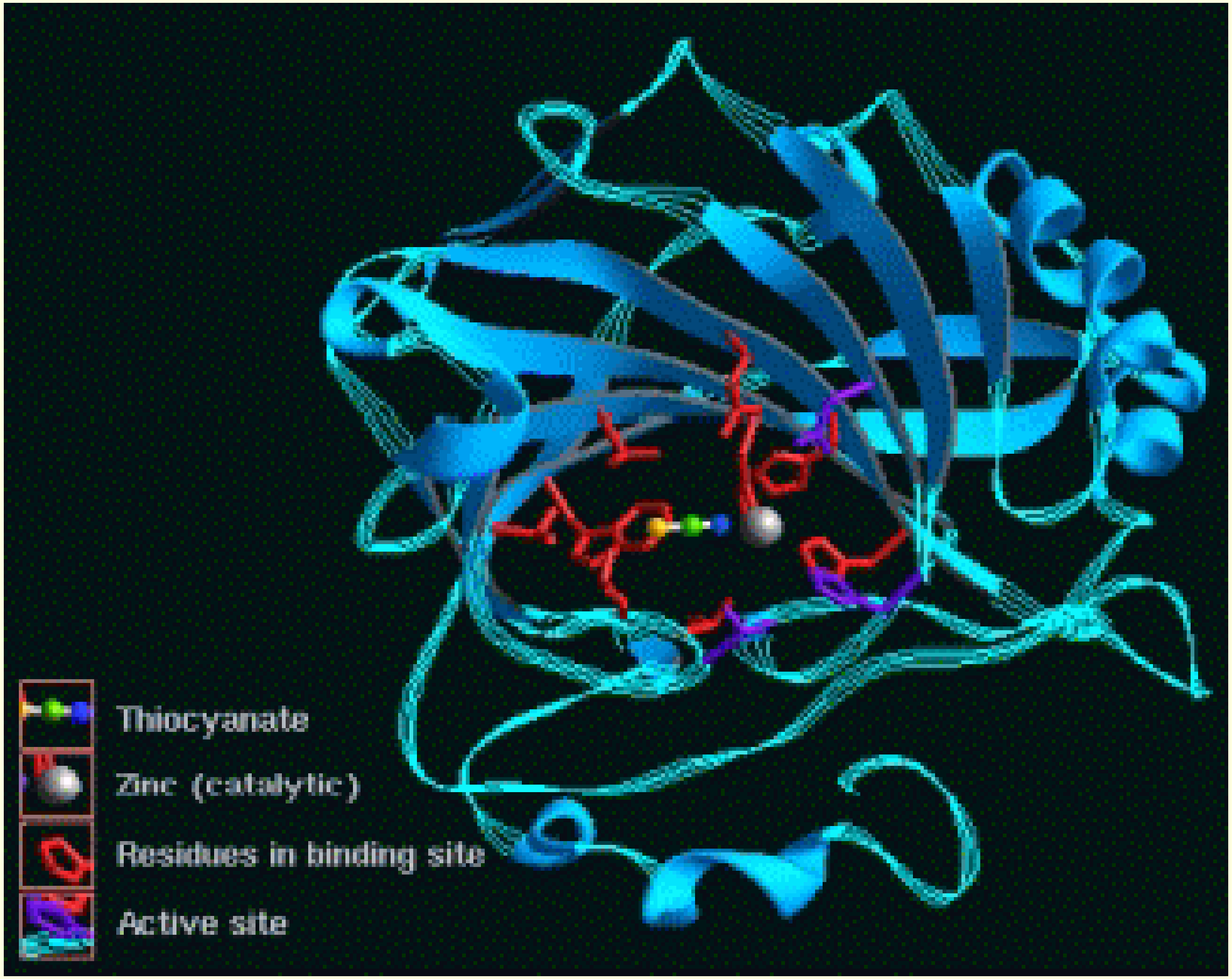
(g) how can this be translated to an objective function?

Tools for multiple sequence alignment

Criteria for alignment quality:

3. 3D-Structure: align residues at corresponding positions in 3D structure of protein!

Tools for multiple sequence alignment



Tools for multiple sequence alignment

Criteria for alignment quality:

3. 3D-Structure: align residues at corresponding positions in 3D structure of protein!

Tools for multiple sequence alignment

Criteria for alignment quality:

3. 3D Structure: align residues at corresponding positions in 3D structure of protein!

5. Evolution: align residues with common ancestors!

Both criteria related: 3D structures conserved in evolution!

Tools for multiple sequence alignment

```
Seq 1   T Y I - M R E A Q Y E
Seq 2   T C I V M R E A - Y E
Seq 3   - Y I - M Q E V Q Q E
Seq 4   - Y I A M R E - Q Y E
```

Alignment hypothesis about sequence evolution

Search for most plausible hypothesis!

Tools for multiple sequence alignment

```
Seq 1   T Y I - M R E A Q Y E
Seq 2   T C I V M R E A - Y E
Seq 3   Y - I - M Q E V Q Q E
Seq 4   - Y I A M R E - Q Y E
```

Alignment hypothesis about sequence evolution

Search for most plausible hypothesis!

Tools for multiple sequence alignment

```
Seq 1   T Y I - M R E A Q Y E
Seq 2   T C I V M R E A - Y E
Seq 3   - Y I - M Q E V Q Q E
Seq 4   - Y I A M R E - Q Y E
```

Alignment hypothesis about sequence evolution

Search for most plausible hypothesis!

Tools for multiple sequence alignment

```
Seq 1   T Y I - M R E A Q Y E
Seq 2   T C I V M R E A - Y E
Seq 3   - Y I - M Q E V Q Q E
Seq 4   - Y I A M R E - Q Y E
```

Assumption: Evolutionary events (insertions, deletions, substitutions) *independent* of each other.

Tools for multiple sequence alignment

Compute

- Probability $p_{a,b}$ of substitution

  $$a \rightarrow b \quad (\text{or } b \rightarrow a),$$

- Frequency q_a of a

Define

$$S(a,b) = log\ (p_{a,b} / \ q_a\ q_b)$$

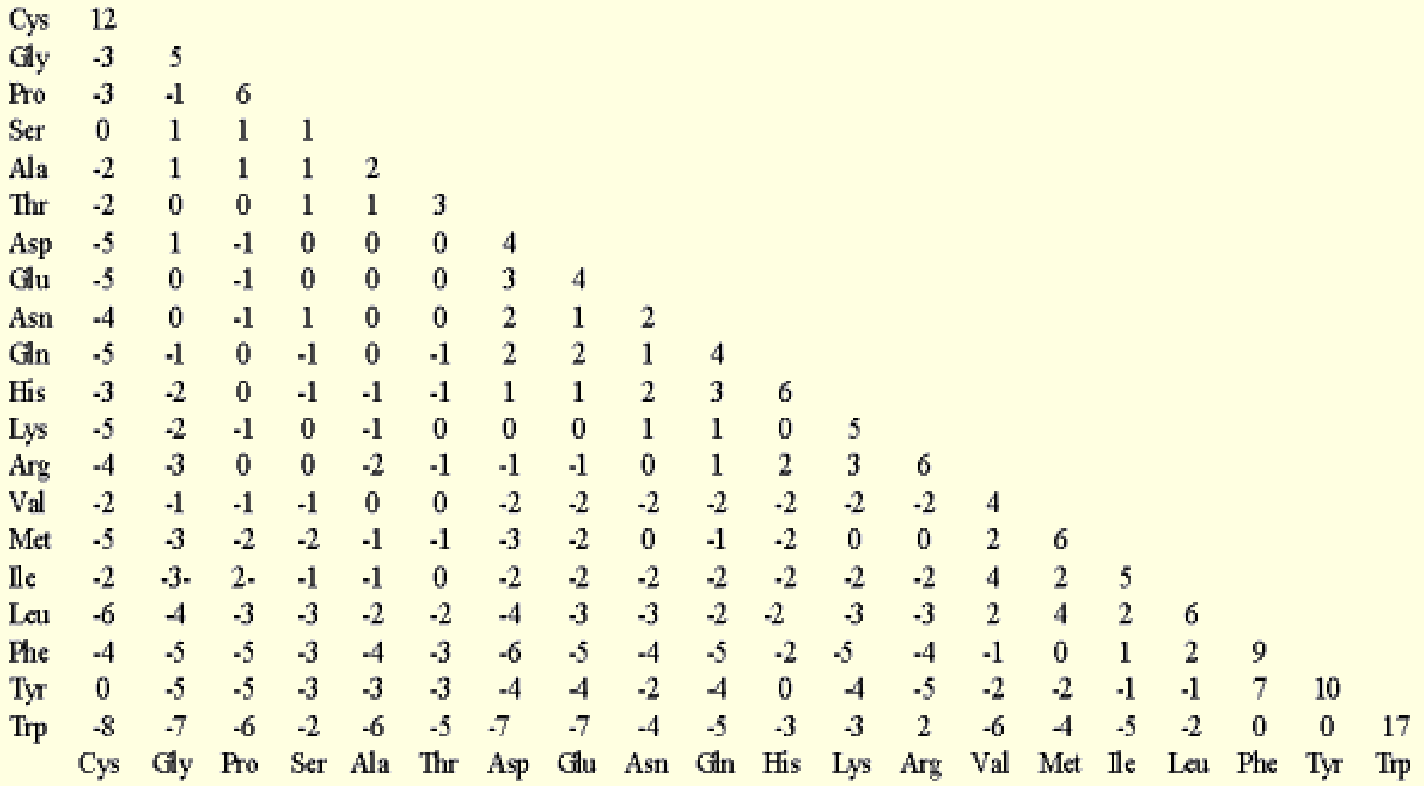
Tools for multiple sequence alignment

Probabilities $p_{a,b}$ calculated based on alignments of closely related protein families

(M. Dayhoff *et al*.)

Calculate substitution matrices (PAM matrices)

Tools for multiple sequence alignment



Cys	12																			
Gly	-3	5																		
Pro	-3	-1	6																	
Ser	0	1	1	1																
Ala	-2	1	1	1	2															
Thr	-2	0	0	1	1	3														
Asp	-5	1	-1	0	0	0	4													
Glu	-5	0	-1	0	0	0	3	4												
Asn	-4	0	-1	1	0	0	2	1	2											
Gln	-5	-1	0	-1	0	-1	2	2	1	4										
His	-3	-2	0	-1	-1	-1	1	1	2	3	6									
Lys	-5	-2	-1	0	-1	0	0	0	1	1	0	5								
Arg	-4	-3	0	0	-2	-1	-1	-1	0	1	2	3	6							
Val	-2	-1	-1	-1	0	0	-2	-2	-2	-2	-2	-2	-2	4						
Met	-5	-3	-2	-2	-1	-1	-3	-2	0	-1	-2	0	0	2	6					
Ile	-2	-3-	2-	-1	-1	0	-2	-2	-2	-2	-2	-2	-2	4	2	5				
Leu	-6	-4	-3	-3	-2	-2	-4	-3	-3	-2	-2	-3	-3	2	4	2	6			
Phe	-4	-5	-5	-3	-4	-3	-6	-5	-4	-5	-2	-5	-4	-1	0	1	2	9		
Tyr	0	-5	-5	-3	-3	-3	-4	-4	-2	-4	0	-4	-5	-2	-2	-1	-1	7	10	
Trp	-8	-7	-6	-2	-6	-5	-7	-7	-4	-5	-3	-3	2	-6	-4	-5	-2	0	0	17
	Cys	Gly	Pro	Ser	Ala	Thr	Asp	Glu	Asn	Gln	His	Lys	Arg	Val	Met	Ile	Leu	Phe	Tyr	Trp

Tools for multiple sequence alignment

Traditional Objective functions:

Define *Score* of pairwise alignment as

- Sum of individual similarity scores *S(a,b)*
- Minus gap penalties

Tools for multiple sequence alignment

Optimal alignment of two sequences of length l_1, l_2 can be calculated in O($l_1 * l_2$) time and space by dynamic programming (Needleman and Wunsch, 1970)

First step in sequence comparison: alignment

- *global* alignment (Needleman and Wunsch, 1970; Clustal W)

```
atctaatagttaatactcgtccaagtat

atctgtattactaaacaactggtgctacta
```

First step in sequence comparison: alignment

- *global* alignment (Needleman and Wunsch, 1970; Clustal W)

```
atc--taatagttaat--actcgtccaagtat
|||  || || | ||   ||| || | | ||
atctgtattact-aaacaactggtgctacta-
```

First step in sequence comparison: alignment

- *global* alignment (Needleman and Wunsch, 1970; Clustal W)

```
atc--taatagttaat--actcgtccaagtat
|||  || || | ||   ||| || | | ||
atctgtattact-aaacaactggtgctacta-
```

- *local* alignment (Smith and Waterman, 1983)

```
atctaatagttaatactcgtccaagtat

gcgtgtattactaaacggttcaatctaacat
```

First step in sequence comparison: alignment

- *global* alignment (Needleman and Wunsch, 1970; Clustal W)

```
atc--taatagttaat--actcgtccaagtat
|||  || || | ||   ||| || | | ||
atctgtattact-aaacaactggtgctacta-
```

- *local* alignment (Smith and Waterman, 1983)

atc**taatagttaa**tactcgtccaagtat

gcgtg**tattactaa**acggttcaatctaacat

First step in sequence comparison: alignment

- *global* alignment (Needleman and Wunsch, 1970; Clustal W)

```
atc--taatagttaat--actcgtccaagtat
|||  || || | ||   ||| || | | ||
atctgtattact-aaacaactggtgctacta-
```

- *local* alignment (Smith and Waterman, 1983)

```
atc--taatagttaatactcgtccaagtat
     || || | ||
gcgtgtattact-aaacggttcaatctaacat
```

Tools for multiple sequence alignment

Traditional Objective functions

Can be generalized to *multiple* alignment
(e.g. sum-of-pair score, tree alignment)

Efficient heuristics for multiple alignment:

- Progressive methods

`Progressive´ Alignment

```
WCEAQTKNGQGWVPSNYITPVN

WWRLNDKEGYVPRNLLGLYP

AVVIQDNSDIKVVPKAKIIRD

YAVESEAHPGSFQPVAALERIN

WLNYNETTGERGDFPGTYVEYIGRKKISP
```

`Progressive´ Alignment

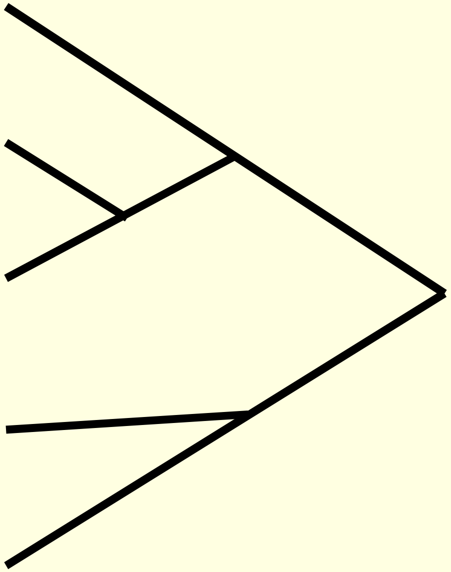
WCEAQTKNGQGWVPSNYITPVN

WWRLNDKEGYVPRNLLGLYP

AVVIQDNSDIKVVPKAKIIRD

YAVESEAHPGSFQPVAALERIN

WLNYNETTGERGDFPGTYVEYIGRKKISP



Guide tree

`Progressive´ Alignment

```
WCEAQTKNGQGWVPSNYITPVN

WW--RLNDKEGYVPRNLLGLYP-
AVVIQDNSDIKVVP--KAKIIRD

YAVESEASFQPVAALERIN

WLNYNEERGDFPGTYVEYIGRKKISP
```

Profile alignment, “once a gap - always a gap”

`Progressive´ Alignment

```
WCEAQTKNGQGWVPSNYITPVN

WW--RLNDKEGYVPRNLLGLYP-
AVVIQDNSDIKVVP--KAKIIRD

YAVESEASVQ--PVAALERIN------
WLN-YNEERGDFPGTYVEYIGRKKISP
```

Profile alignment, “once a gap - always a gap”

`Progressive´ Alignment

```
WCEAQTKNGQGWVPSNYITPVN-
WW--RLNDKEGYVPRNLLGLYP-
AVVIQDNSDIKVVP--KAKIIRD

YAVESEASVQ--PVAALERIN------
WLN-YNEERGDFPGTYVEYIGRKKISP
```

Profile alignment, “once a gap - always a gap”

`Progressive´ Alignment

```
WCEAQTKNGQGWVPSNYITPVN--------
WW--RLNDKEGYVPRNLLGLYP--------
AVVIQDNSDIKVVP--KAKIIRD-------
YAVESEA---SVQ--PVAALERIN------
WLN-YNE---ERGDFPGTYVEYIGRKKISP
```

Profile alignment, “once a gap - always a gap”

Tools for multiple sequence alignment

Problems with traditional approach:

Tools for multiple sequence alignment

Problems with traditional approach:

- Results depend on gap penalty

Tools for multiple sequence alignment

Problems with traditional approach:

- Results depend on gap penalty
- Heuristic guide tree determines alignment; alignment used for phylogeny reconstruction

Tools for multiple sequence alignment

Problems with traditional approach:

- Results depend on gap penalty
- Heuristic guide tree determines alignment; alignment used for phylogeny reconstruction
- Algorithm produces *global* alignments. Many sequence families share only *local* similarities

New question: sequence families with multiple local similarities

The DIALIGN approach to *multiple* alignment of nucleic acid and protein sequences:

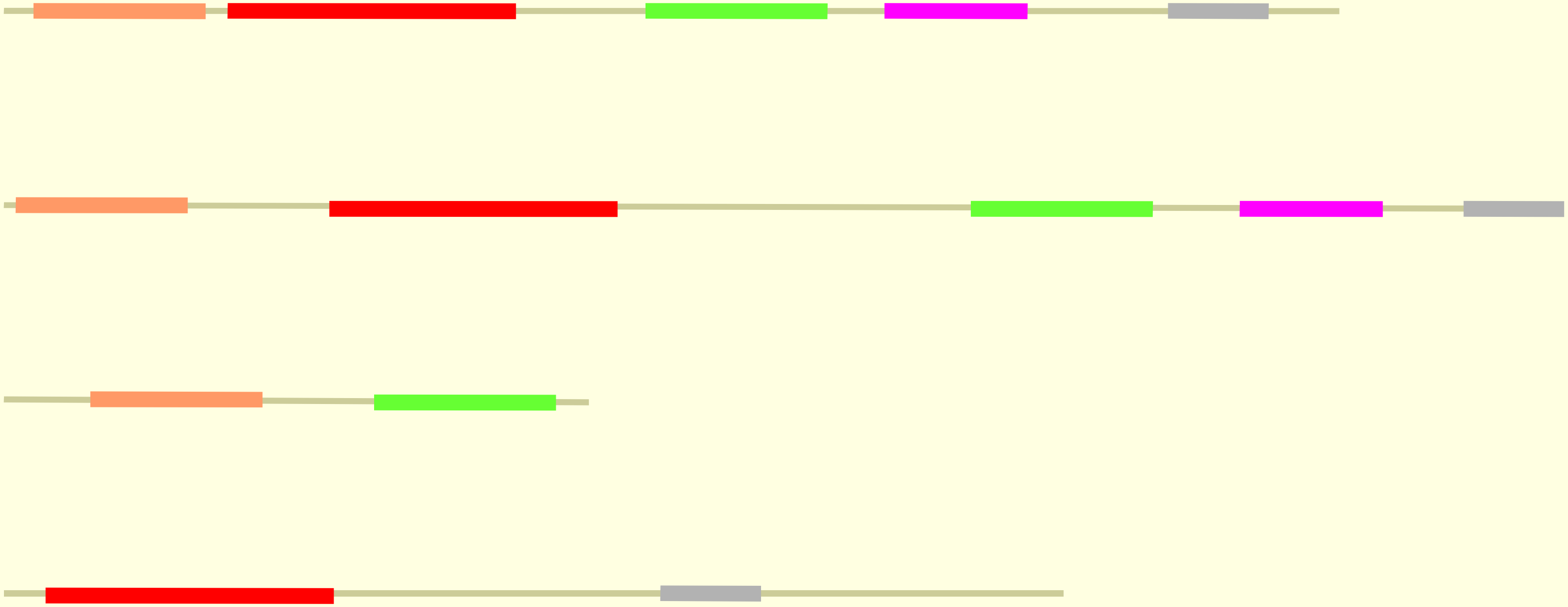
Combination of local and global approaches

New question:
sequence families with multiple local similarities

The DIALIGN approach to *multiple* alignment of nucleic acid and protein sequences:

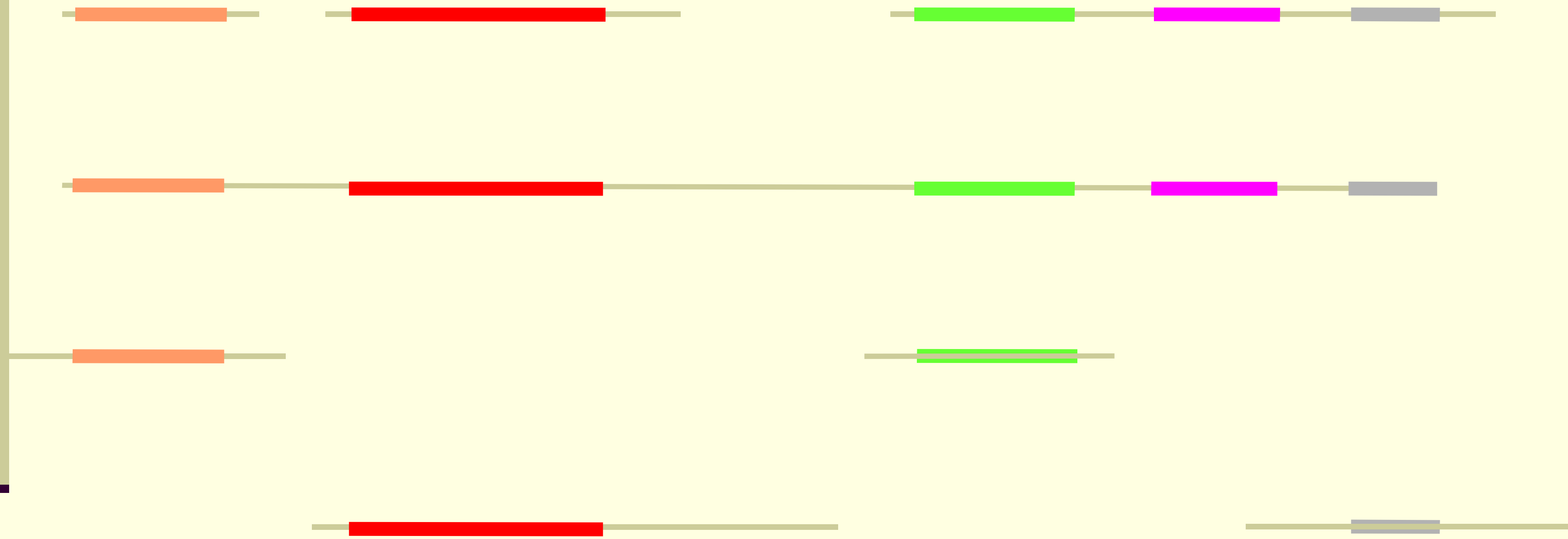
Combination of local and global approaches

New question:
sequence families with multiple local similarities



Neither local nor global methods applicable

New question:
sequence families with multiple local similarities



Alignment possible if order conserved

The *DIALIGN* approach

- *Combination* of global and local methods

- Assemble multiple alignment from *gap-free local pair-wise* alignments („fragments“)

Morgenstern, Dress, Werner (1996),
PNAS 93, 12098-12103

The *DIALIGN* approach

```
atctaatagttaaactcccccgtgcttag

cagtgcgtgtattactaacggttcaatcgcg

caaagagtatcacccctgaattgaataa
```

The *DIALIGN* approach

atctaatagttaaactccccgtgcttag

cagtgc**gtgtatta**ctaacggttcaatcgcg

caaa**gagtatca**cccctgaattgaataa

The *DIALIGN* approach

atctaatagttaaactccccgtgcttag

cagtgc**gtgtatta**ctaacggttcaatcgcg

caaa**gagtatca**cccctgaattgaataa

fragment = gap-free pair-wise alignment

The *DIALIGN* approach

atctaatagttaaactccccgtgcttag

cagtgc**gtgtatta**ctaacggttcaatcgcg

caaa**gagtatca**cccctgaattgaataa

fragment = gap-free pair-wise alignment

= pair of equal-length segments

The *DIALIGN* approach

atctaatagttaaactccccgtgcttag

cagtgc**gtgtatta**ctaacggttcaatcgcg

caaa**gagtatca**cccctgaattgaataa

The *DIALIGN* approach

atc**taatagtta**aactccccgtgcttag

cagtgc**gtgtattactaa**cggttcaatcgcg

caaa**gagtatca**cccctgaattgaataa

The *DIALIGN* approach

atc**taatagtta**aactcccccgtgcttag

cagtgc**gtgtattactaa**cggttcaatcgcg

caaa**gagtatca**cccctgaattgaataa

overlap possible if different sequence pairs involved!

The *DIALIGN* approach

atc**taatagtta**aactccccgtgcttag

cagtgc**gtgtattactaa**cggttcaatcgcg

caaa**gagtatca**cccctgaattgaataa

The *DIALIGN* approach

atc**taatagtta**aactcccccgtgcttag

cagtgc**gtgtattactaa**cgg**ttcaat**cgcg

caaa**gagtatca**cccctgaa**ttgaat**aa

The *DIALIGN* approach

atc**taatagtta**aactcccc**cgtg**cttag

cagtgc**gtgtattactaa**cgg**ttcaat**cgcg

caaa**gagtatca**cc**cctg**aa**ttgaat**aa

The *DIALIGN* approach

atc**taatagtta**aactcccc**cgtgctt**ag

cagtgc**gtgtattactaa**c**ggttcaat**cgcg

caaa**gagtatca**cc**cctg**aa**ttgaat**aa

The *DIALIGN* approach

atc------**taatagtta**aactcccc**cgtgctt**ag

cagtgc**gtgtattactaa**c**ggttcaat**cgcg

caaa**gagtatca**cc**cctg**aa**ttgaat**aa

The *DIALIGN* approach

```
atc------taatagttaaactccccgtgcttag

cagtgcgtgtattactaac----------ggttcaatcgcg

caaagagtatcacc------------cctgaattgaataa
```

The *DIALIGN* approach

```
atc------taatagttaaactccccgtgcttag

cagtgcgtgtattactaac----------ggttcaatcgcg

caaa--gagtatcacc----------cctgaattgaataa
```

The *DIALIGN* approach

```
atc------taatagttaaactccccgtgc-ttag

cagtgcgtgtattactaac----------gg-ttcaatcgcg

caaa--gagtatcacc----------cctgaattgaataa
```

The *DIALIGN* approach

Consistency!

```
atc------taatagttaaactccccgtgc-ttag

cagtgcgtgtattactaac----------gg-ttcaatcgcg

caaa--gagtatcacc----------cctgaattgaataa
```

The *DIALIGN* approach

```
atc------TAATAGTTAaactccccCGTGC-TTag

cagtgcGTGTATTACTAAc----------GG-TTCAATcgcg

caaa--GAGTATCAcc----------CCTGaaTTGAATaa
```

The *DIALIGN* approach

```
atc------TAATAGTTAaactccccCGTGC-TTag

cagtgcGTGTATTACTAAc----------GG-TTCAATcgcg

caaa--GAGTATCAcc----------CCTGaaTTGAATaa
```

Program output: fragments not visible

The *DIALIGN* approach

```
atc------TAATAGTTAaactccccCGTGC-TTag

cagtgcGTGTATTACTAAc----------GG-TTCAATcgcg

caaa--GAGTATCAcc----------CCTGaaTTGAATaa
```

Lower-case residues not part of fragments

The *DIALIGN* approach

How to find good fragment-based alignments ??

Evaluation of multi-alignment methods

Two main questions in sequence alignment:

3. Scoring scheme (= objective function): How good is a given alignment?

5. Optimization algorithm: Find alignment with best score!

Evaluation of multi-alignment methods

Objective function for DIALIGN:

- Weight score for every possible fragment based on *P-value*
- Find consistent collection of fragments with maximum total weight score; no gap penalty!

The *DIALIGN* approach

```
atctaatagttaaaccccctcgtgcttagagatccaaac
cagtgcgtgtattactaacggttcaatcgcgcacatccgc
```

Pair-wise alignment:

The *DIALIGN* approach

atc**taatagtta**aaccccctcgt**gctt**ag**agatcc**aaac

cagtgcgtg**tattactaa**c**ggtt**caatcgcgc**acatcc**gc

Pair-wise alignment:

- recursive algorithm finds optimal chain of fragments.

The *DIALIGN* approach

```
------atctaatagttaaacccccctcgtgcttag-------agatccaaac
cagtgcgtgtattactaac-----------ggttcaatcgcgcacatccgc--
```

Pair-wise alignment:

- recursive algorithm finds optimal chain of fragments.

The *DIALIGN* approach

Multiple alignment:

```
atctaatagttaaactcccccgtgcttag

cagtgcgtgtattactaacggttcaatcgcg

caaagagtatcacccctgaattgaataa
```

The *DIALIGN* approach

Multiple alignment:

atc**taatagtta**aactcccccgt**gctt**ag

cagtgcgtg**tattactaa**c**ggtt**caatcgcg

caaccctgaattgaagagtatcacataa

(1) Calculate all optimal pair-wise alignments

The *DIALIGN* approach

Multiple alignment:

atc**taatagtt**aaactcccc**cgtg**ct**tag**

cagtgcgtgtattactaacggttcaatcgcg

caaa**gagtatca**cc**cctg**aattgaa**taa**

(1) Calculate all optimal pair-wise alignments

The *DIALIGN* approach

Multiple alignment:

atctaatagttaaactccccgtgcttag

cagtgc**gtgtatta**c**ta**acgg**ttcaat**cgcg

caaa**gagtatca**cccc**tg**aa**ttgaat**aa

(1) Calculate all optimal pair-wise alignments

The *DIALIGN* approach

Fragments from optimal pair-wise alignments might be *inconsistent*

The *DIALIGN* approach

atctaatagttaaactccccgtgcttag

cagtgc**gtgtatta**ctaacggttcaatcgcg

caaa**gagtatca**cccctgaattgaataa

The *DIALIGN* approach

atc**taatagtta**aactccccgtgcttag

cagtgc**gtgtattactaa**cggttcaatcgcg

caaa**gagtatca**cccctgaattgaataa

The *DIALIGN* approach

atc**taatagtta**aa**ctc**ccccgtgcttag

cagtgc**gtgtattactaa**cggttcaatcgcg

caaa**gagtatca**cccctgaattgaataa

The *DIALIGN* approach

atc**taatagtta**aa**ctc**ccccgtgcttag

cagtgc**gtgtattactaa**cggttcaatcgcg

caaa--**gagtatca**cccctgaattgaataa

The *DIALIGN* approach

atc------**taatagtta**aa**ctc**ccccgtgcttag

cagtgc**gtgtattactaa**cggttcaatcgcg

caaa--**gagtatca**cccctgaattgaataa

The *DIALIGN* approach

atc**taatagtta**aactccccgtgcttag

cagtgc**gtgtattactaa**cggttcaatcgcg

caaa**gagtatca**cccctgaattgaataa

The *DIALIGN* approach

Fragments from optimal pair-wise alignments might be *inconsistent*

(2) Sort fragments according to *scores*

The *DIALIGN* approach

Fragments from optimal pair-wise alignments might be *inconsistent*

(2) Sort fragments according to *scores*

(3) Include them one-by-one into growing multiple alignment – as long as they are *consistent*

The *DIALIGN* approach

Fragments from optimal pair-wise alignments might be *inconsistent*

(2) Sort fragments according to *scores*

(3) Include them one-by-one into growing multiple
    alignment – as long as they are *consistent*

(*greedy* algorithm)

The *DIALIGN* approach

Advantages of segment-based approach:

- Program can produce global *and* local alignments!
- Applicable to sequence families that cannot be aligned with standard methods

Evaluation of multi-alignment methods

Lassmann und Sonnhammer (2002),
FEBS Letters 529, 126-130

- Comparison with known 3D structure (BAliBASE)

Evaluation of multi-alignment methods

```
1aboA      1    .NLFVALYDfvasgdntlsitkGEKLRVLgynhn..............g
1ycsB      1    kGVIYALWDyepqnddelpmkeGDCMTIIhrede............dei
1pht       1    gYQYRALYDykkereedidlhlGDILTVNkgslvalgfsdgqearpeei
1ihvA      1    .NFRVYYRDsrd......pvwkGPAKLLWkg.................e
1vie       1    .drvrkksga.........awqGQIVGWYctnlt............pe

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:
> 100 Reference alignments

Evaluation of multi-alignment methods

Lassmann und Sonnhammer (2002),
FEBS Letters 529, 126-130

- Comparison with known 3D structure (BAliBASE)

Evaluation of multi-alignment methods

Lassmann und Sonnhammer (2002),
FEBS Letters 529, 126-130

- Comparison with known 3D structure (BAliBASE)
- Artificial sequences with simulated molecular evolution (ROSE)

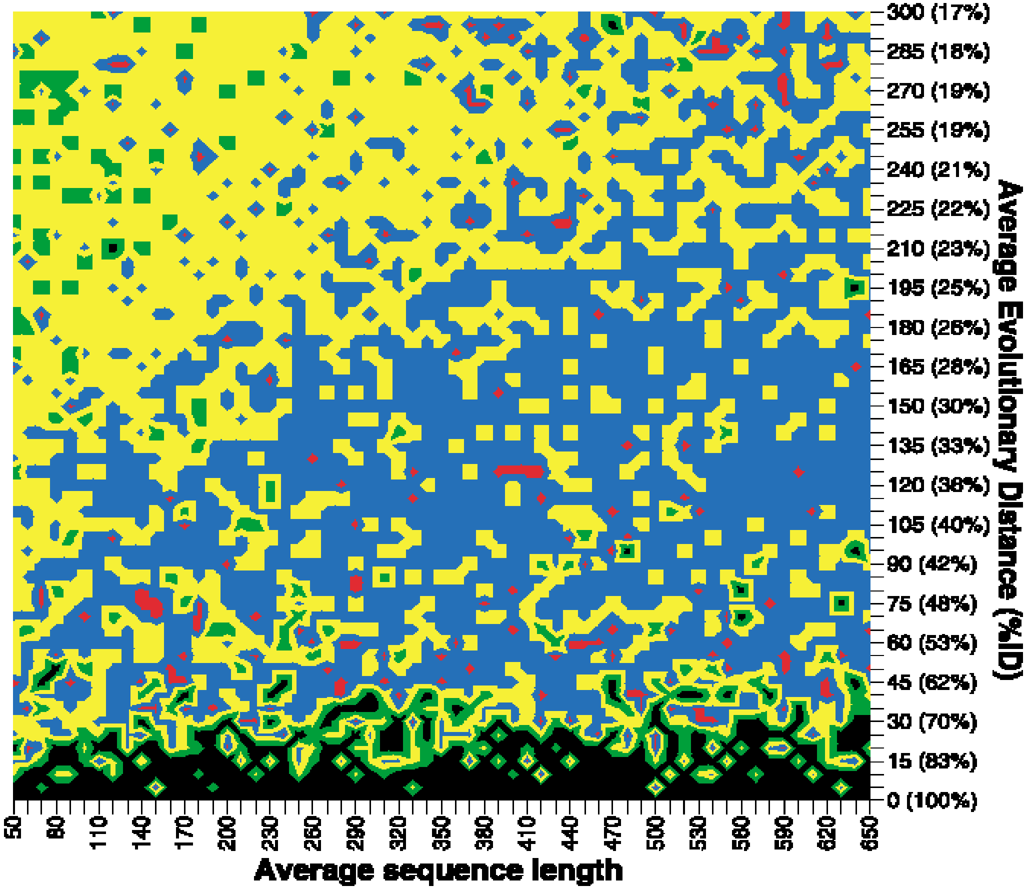


Fig. 1. Color coded matrix showing which method performed best for each pair-combination of conditions: average sequence length (x-axis) and average evolutionary distance (y-axis). The methods are Poa (green), Dialign (yellow), T-Coffee (blue) and ClustalW (red).

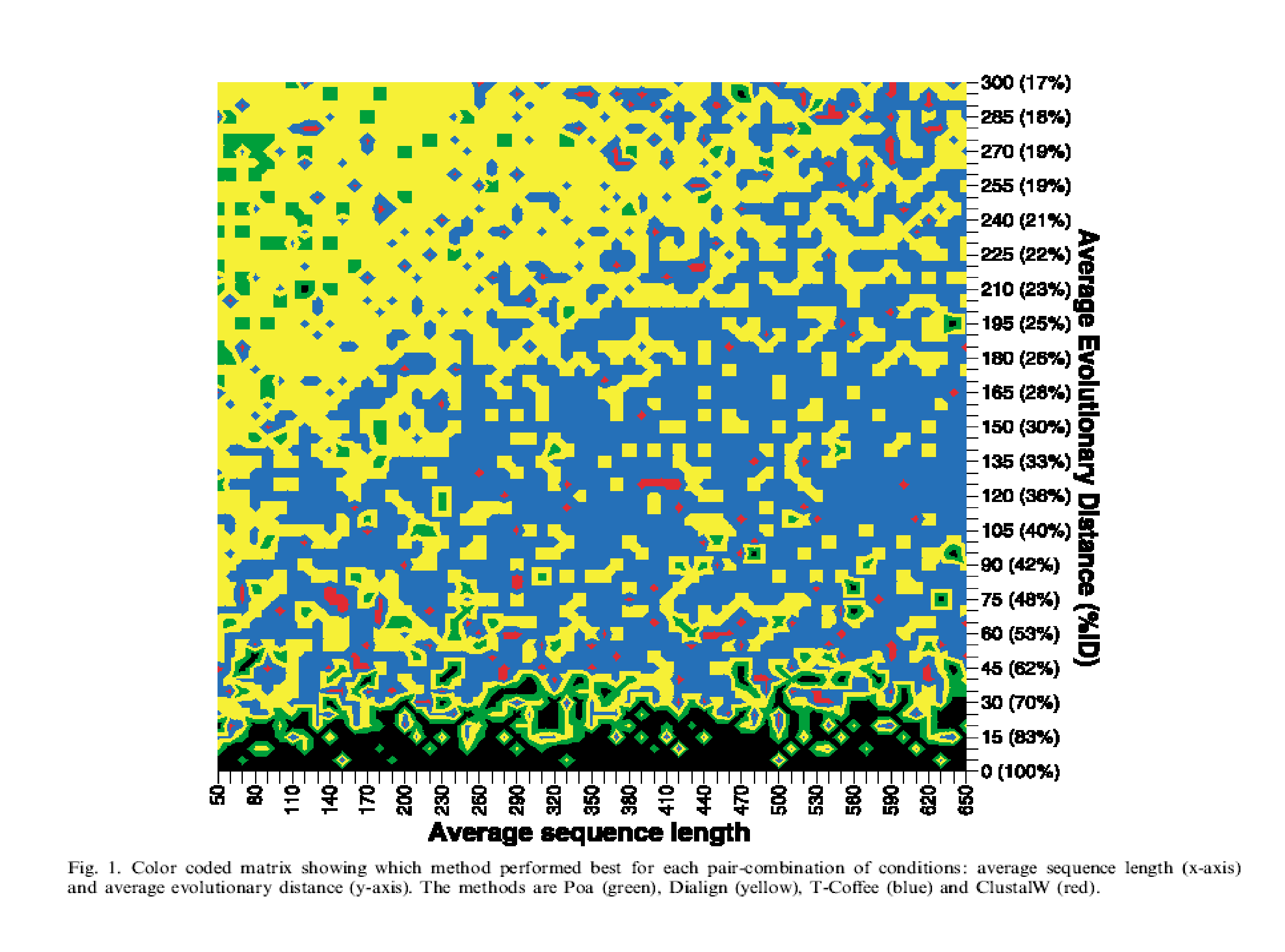


Fig. 1. Color coded matrix showing which method performed best for each pair-combination of conditions: average sequence length (x-axis) and average evolutionary distance (y-axis). The methods are Poa (green), Dialign (yellow), T-Coffee (blue) and ClustalW (red).

Result: DIALIGN best method for *distantly* related sequences

Alignment of large genomic sequences

Fragment-based alignment approach useful for alignment of genomic sequences.

Possible applications:

- Detection of regulatory elements
- Identification of pathogenic microorganisms
- Gene prediction

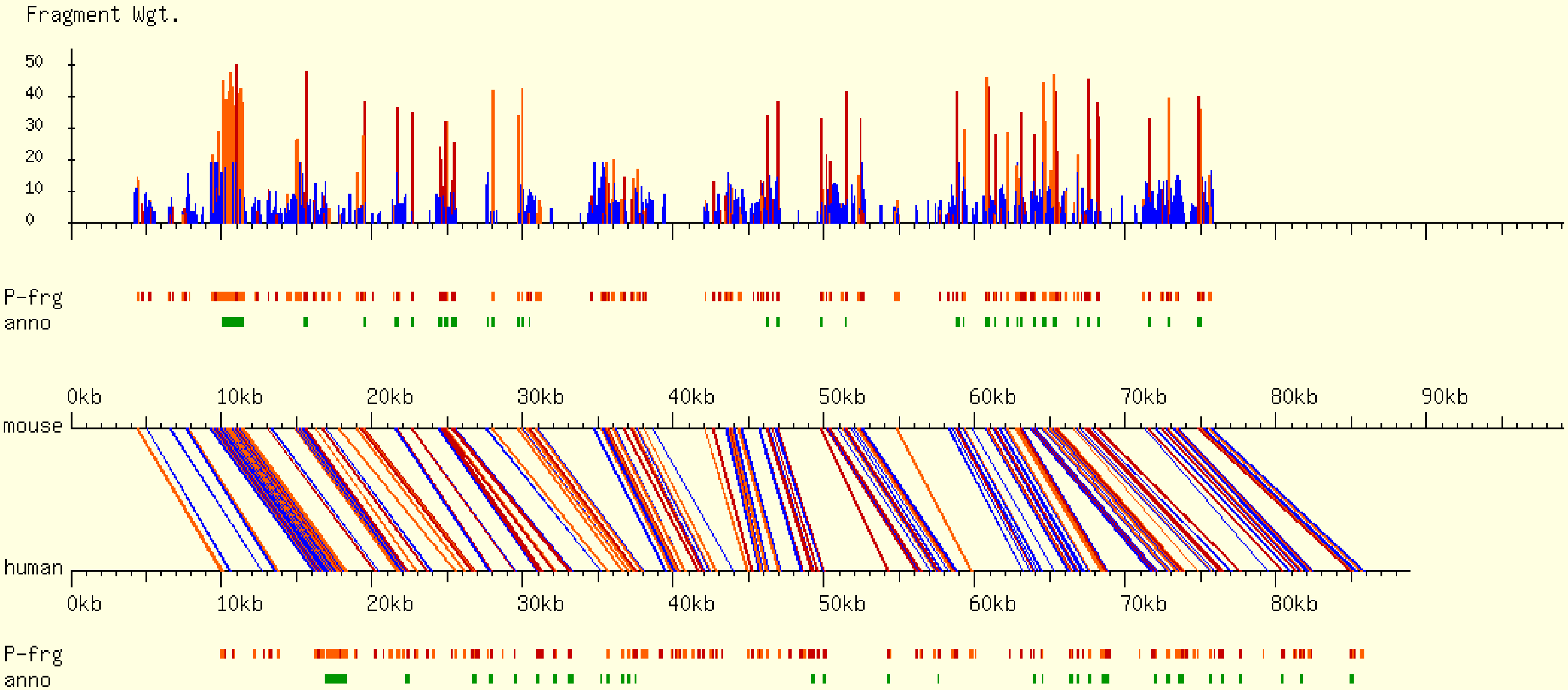
Alignment of large genomic sequences

Alignment of large genomic sequences to identify functional elements (*phylogenetic footprinting*)

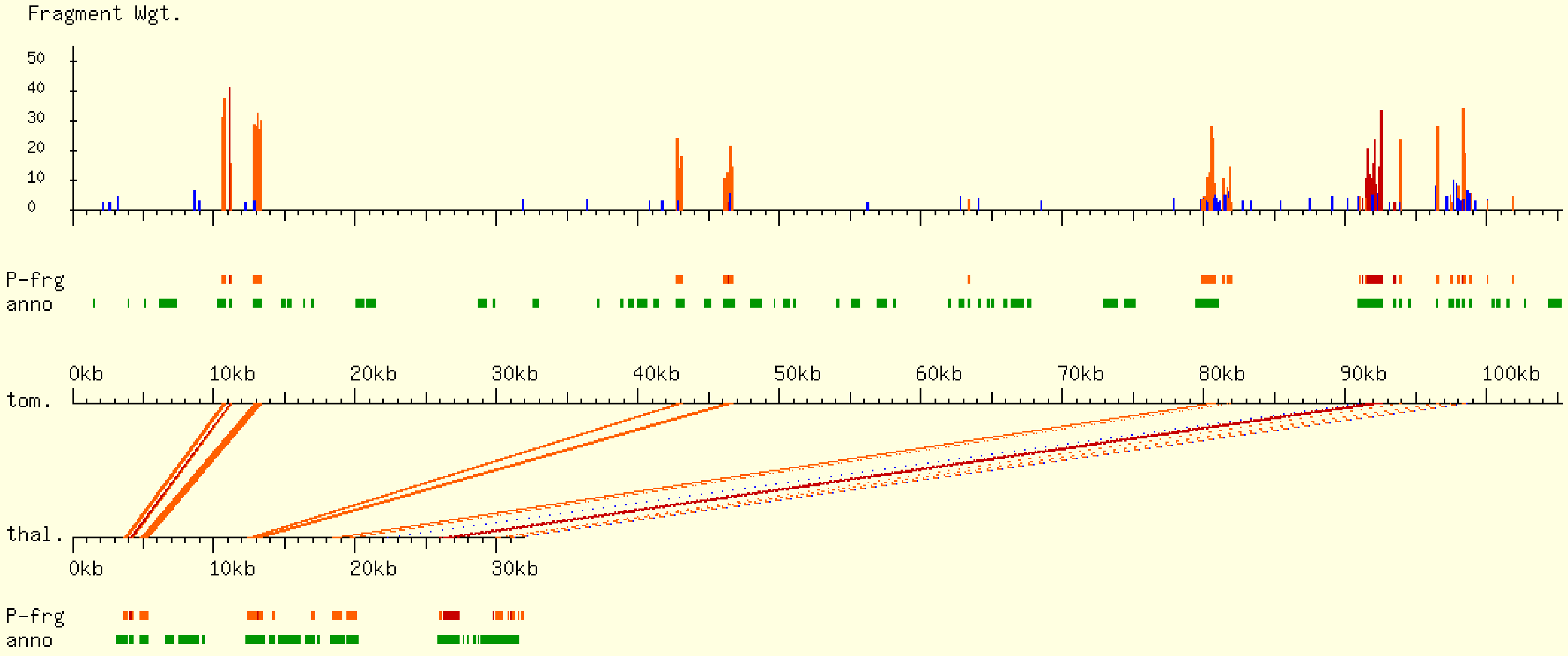
- Göttgens *et al*., 2000, 2001, 2002, …
- Pollard *et al*., 2004

DIALIGN, MGA, PipMaker, LAGAN, AVID, Mummer, WABA, …

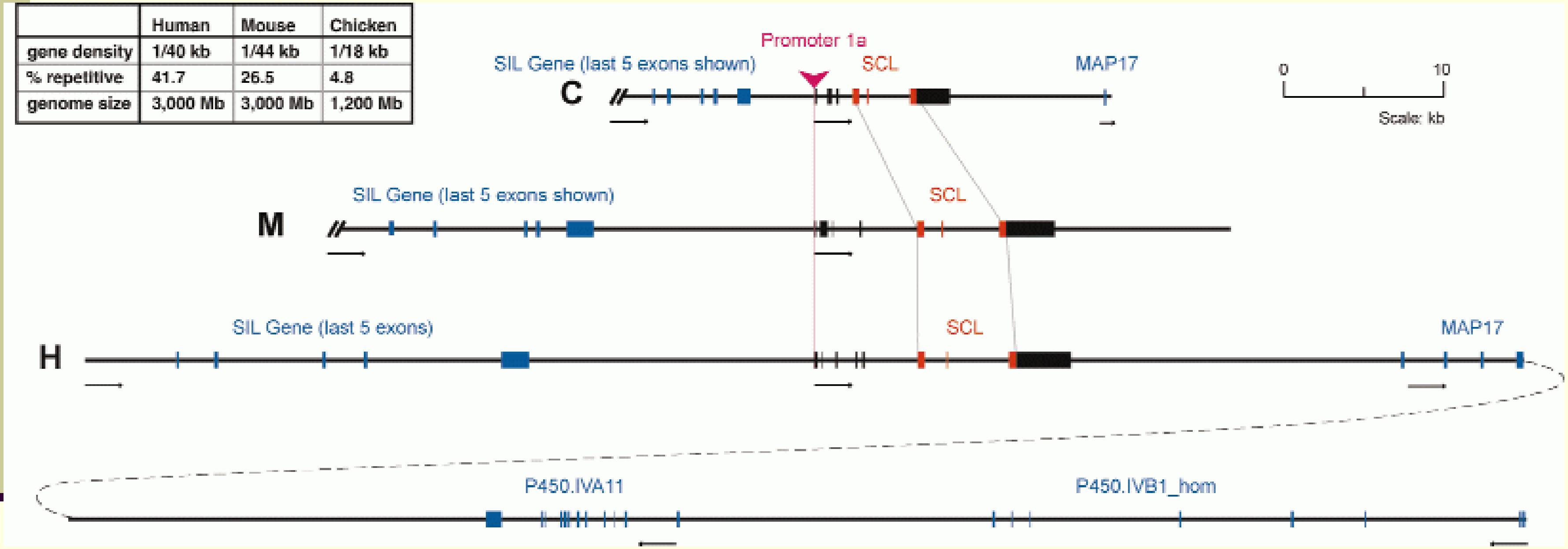
DIALIGN alignment of human and murine genomic sequences



DIALIGN alignment of tomato and *Thaliana* genomic sequences



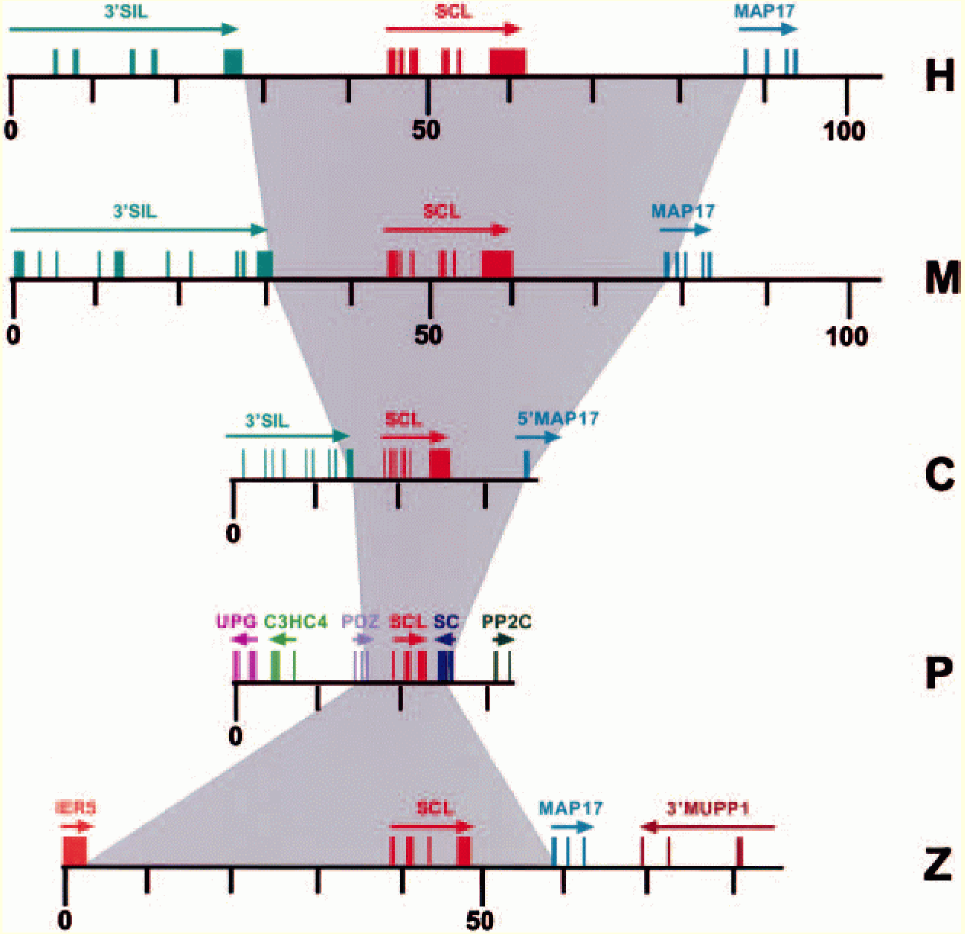
Alignment of large genomic sequences



	Human	Mouse	Chicken
gene density	1/40 kb	1/44 kb	1/18 kb
% repetitive	41.7	26.5	4.8
genome size	3,000 Mb	3,000 Mb	1,200 Mb

Gene-regulatory sites identified by mulitple sequence alignment (*phylogenetic footprinting*)

Alignment of large genomic sequences



Alignment of large genomic sequences

DIALIGN used by **`tracker`** for phylogenetic footprinting (Prohaska *et al*., 2004)

Alignment of large genomic sequences

DIALIGN used by **`tracker`** for phylogenetic footprinting (Prohaska *et al*., 2004)

Alignment of *Hox* gene cluster:

Alignment of large genomic sequences

DIALIGN used by `tracker` for phylogenetic footprinting (Prohaska *et al*., 2004)

Alignment of *Hox* gene cluster:

- DIALIGN able to identify small regulatory elements, but

Alignment of large genomic sequences

DIALIGN used by `tracker` for phylogenetic footprinting (Prohaska *et al.*, 2004)

Alignment of *Hox* gene cluster:

- DIALIGN able to identify small regulatory elements, but
- Entire genes *totally* mis-aligned

Alignment of large genomic sequences

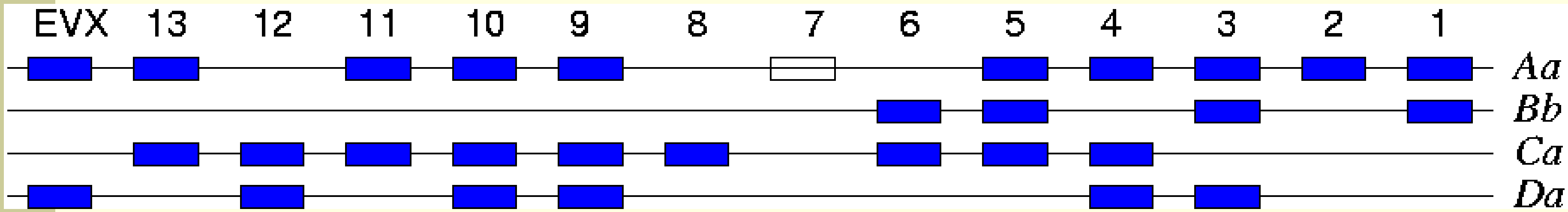
DIALIGN used by `tracker` for phylogenetic footprinting (Prohaska *et al.*, 2004)

Alignment of *Hox* gene cluster:

- DIALIGN able to identify small regulatory elements, but
- Entire genes *totally* mis-aligned
- Reason for mis-alignment: *duplications* !

Alignment of large genomic sequences

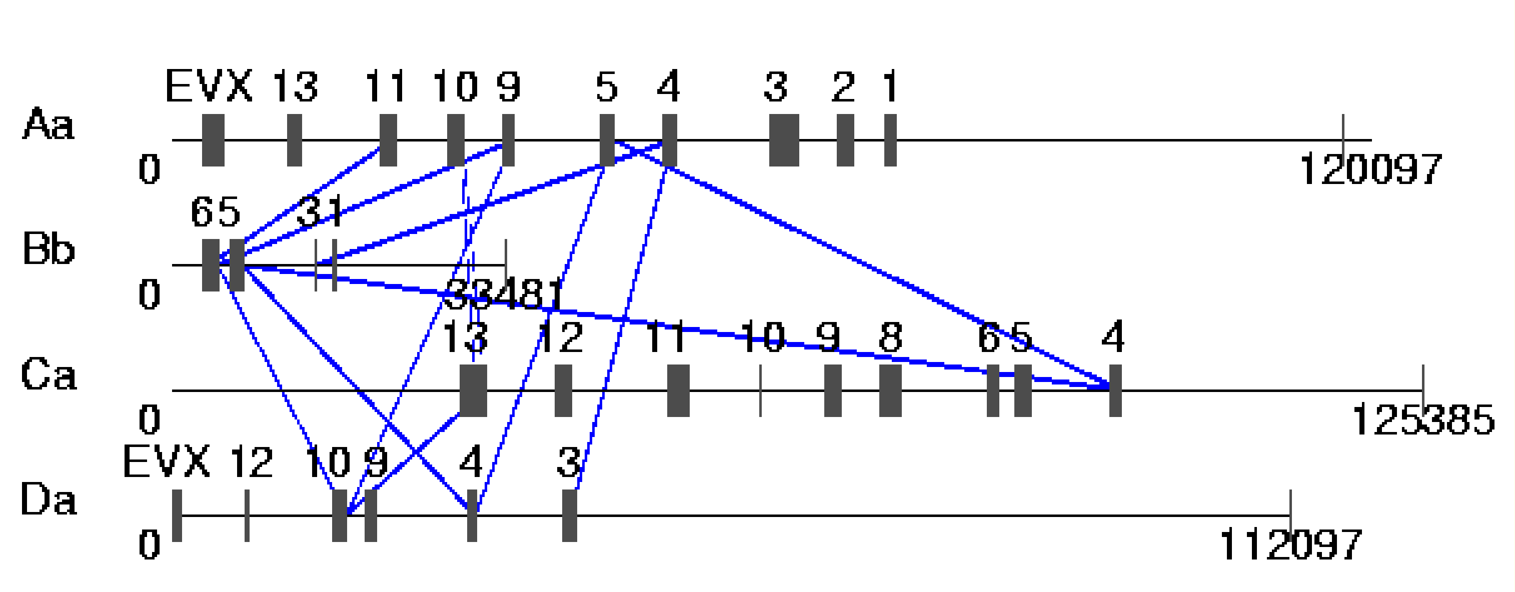
The *Hox* gene cluster:



4 *Hox* gene clusters in pufferfish. 14 genes, different genes in different clusters!

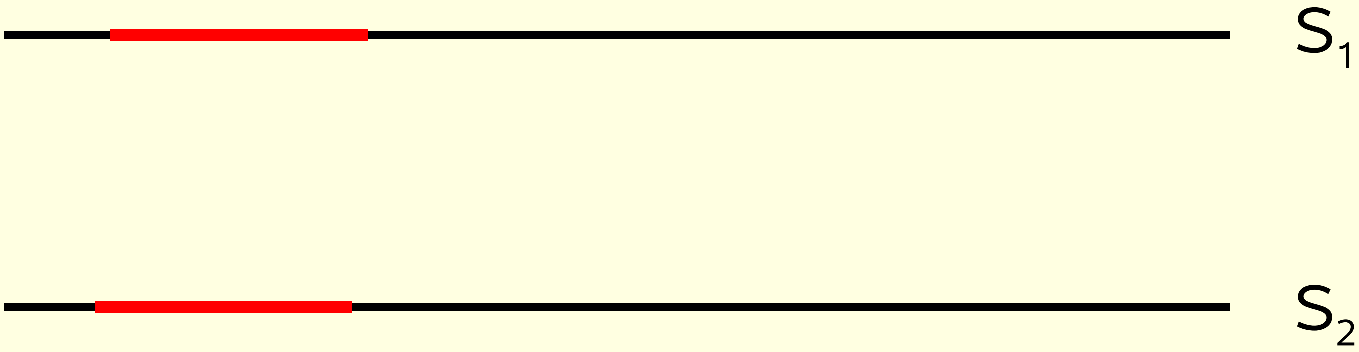
Alignment of large genomic sequences

The *Hox* gene cluster:

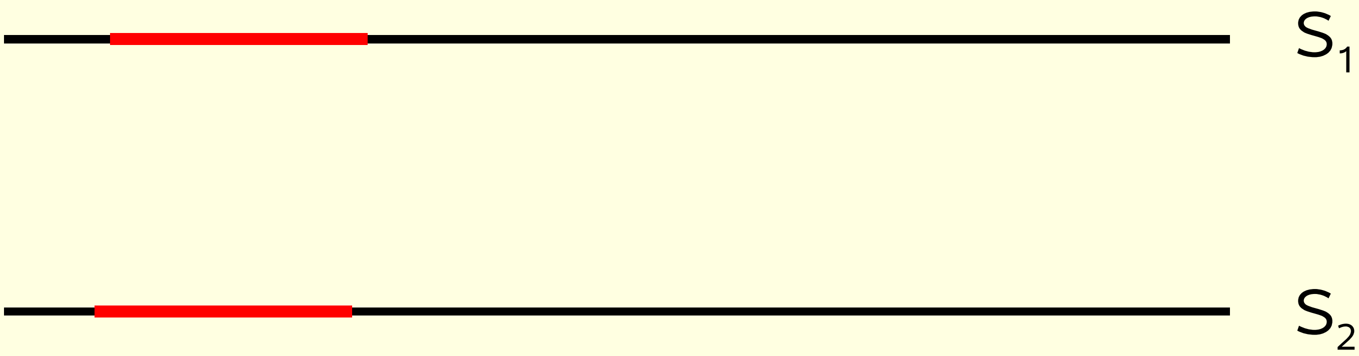


Complete mis-alignment of entire genes!

Alignment of sequence duplications

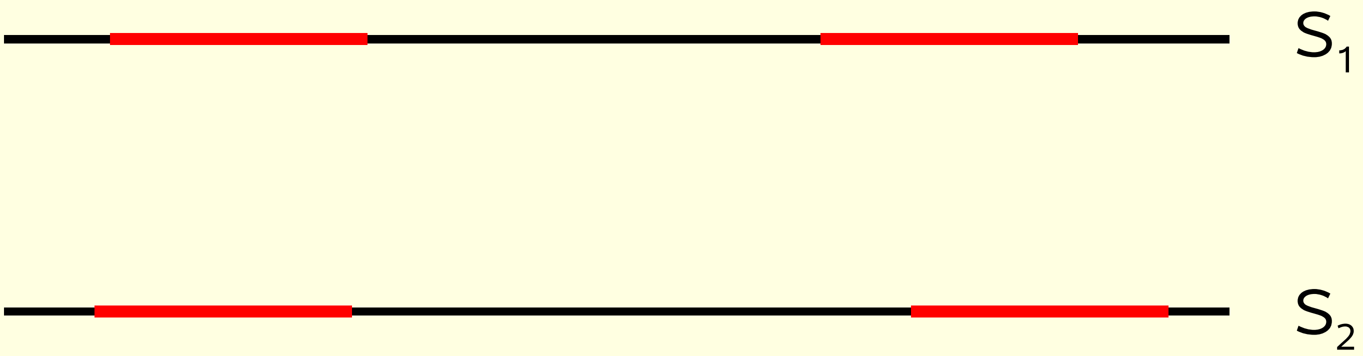


Alignment of sequence duplications



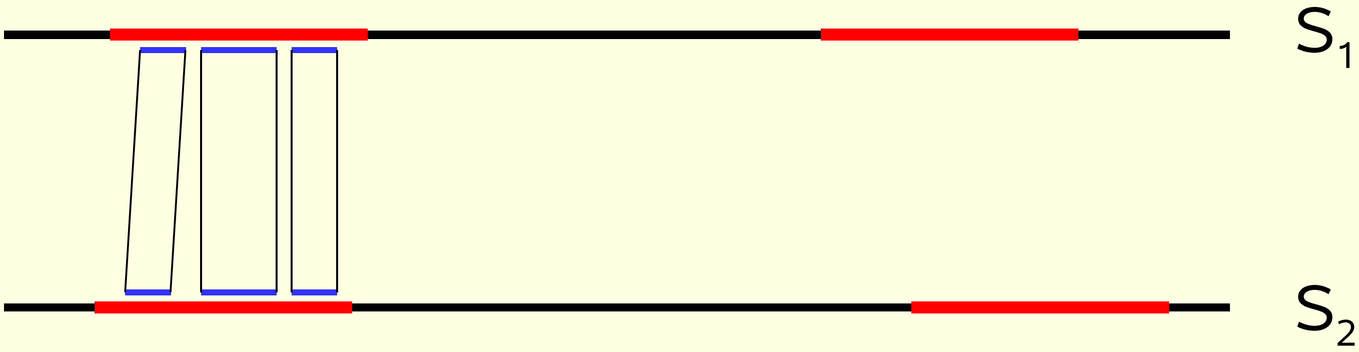
Conserved motivs; no similarity outside motifs

Alignment of sequence duplications



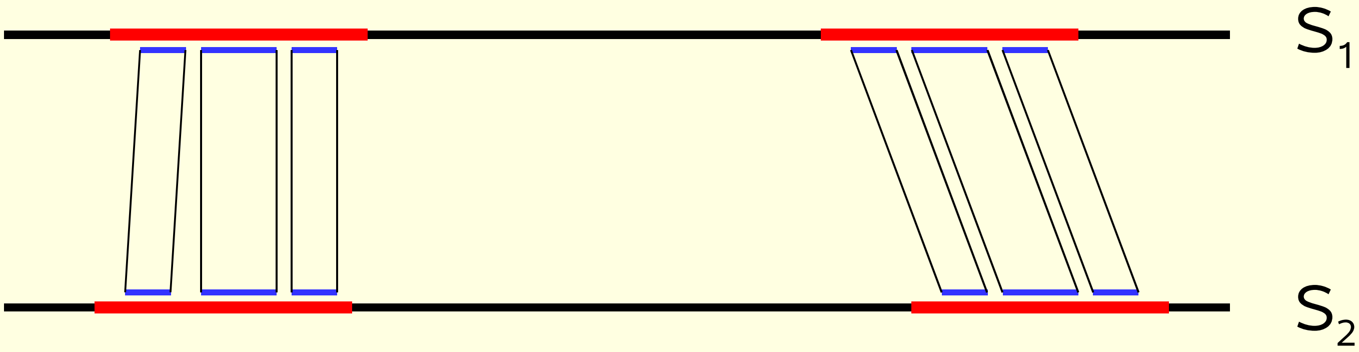
Duplication in *two* sequences

Alignment of sequence duplications



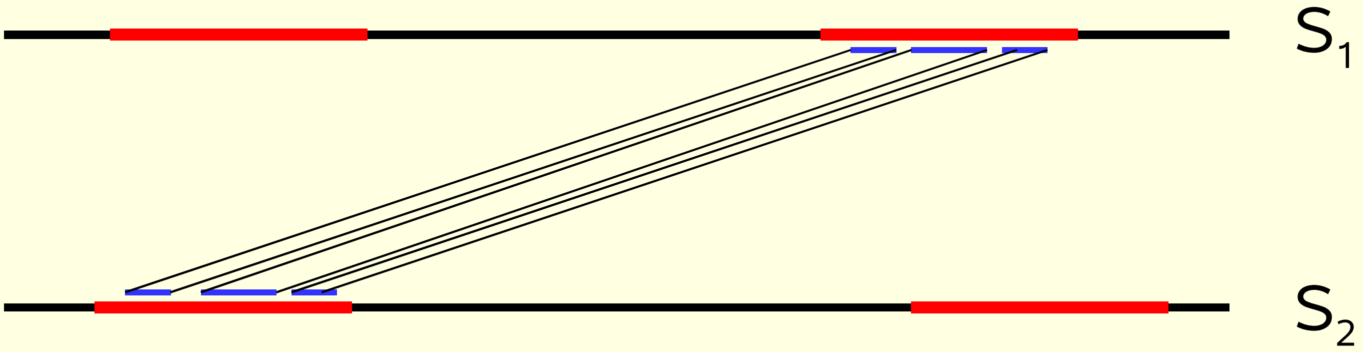
Duplication in *two* sequences

Alignment of sequence duplications



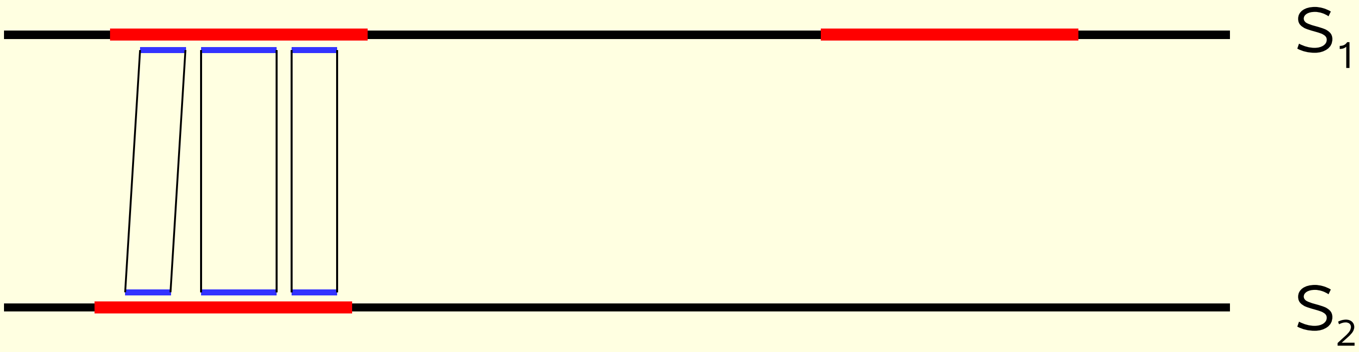
Duplication in *two* sequences

Alignment of sequence duplications



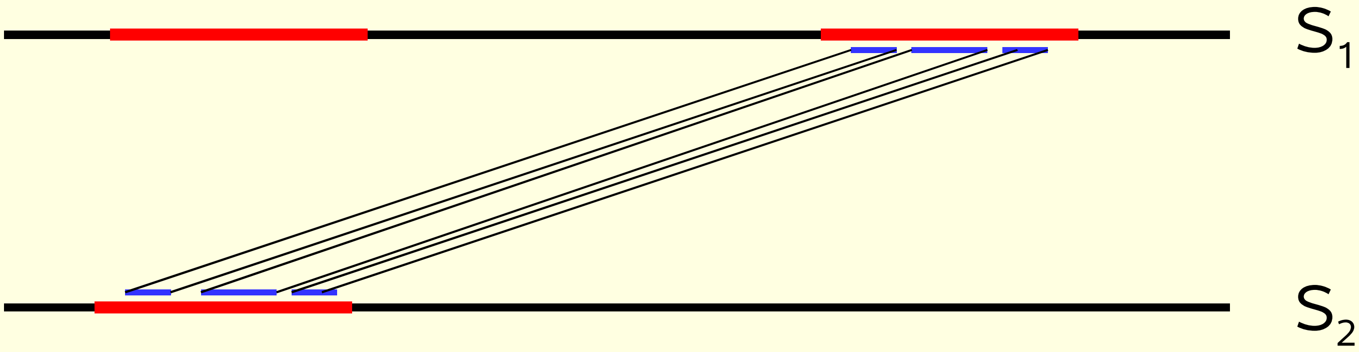
Mis-alignment would have lower score!

Alignment of sequence duplications



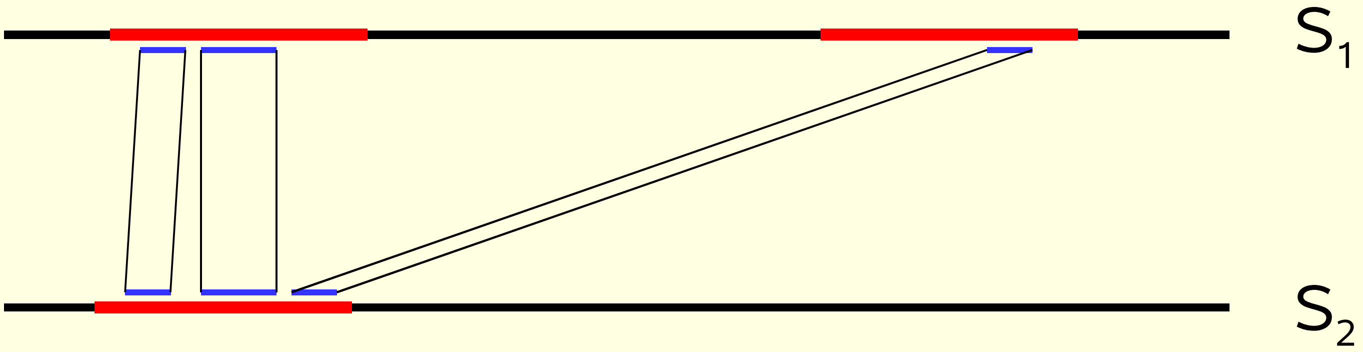
Duplication in *one* sequence

Alignment of sequence duplications



Duplication in *one* sequence

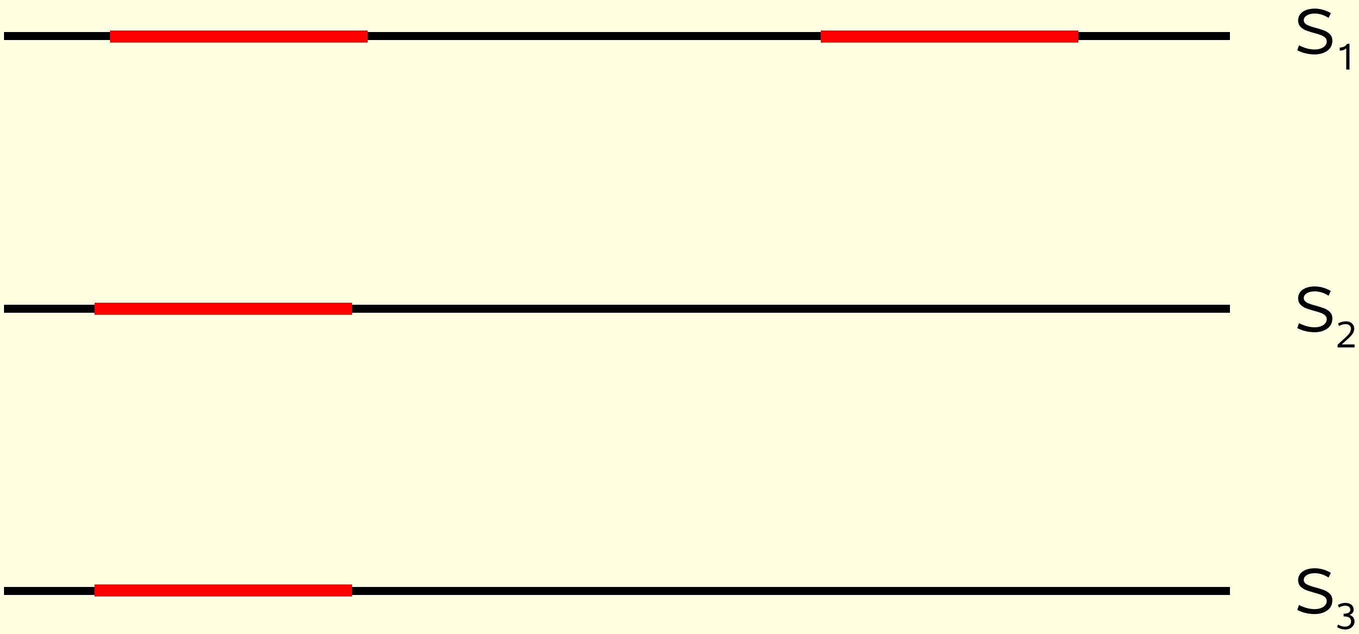
Alignment of sequence duplications



Duplication in *one* sequence

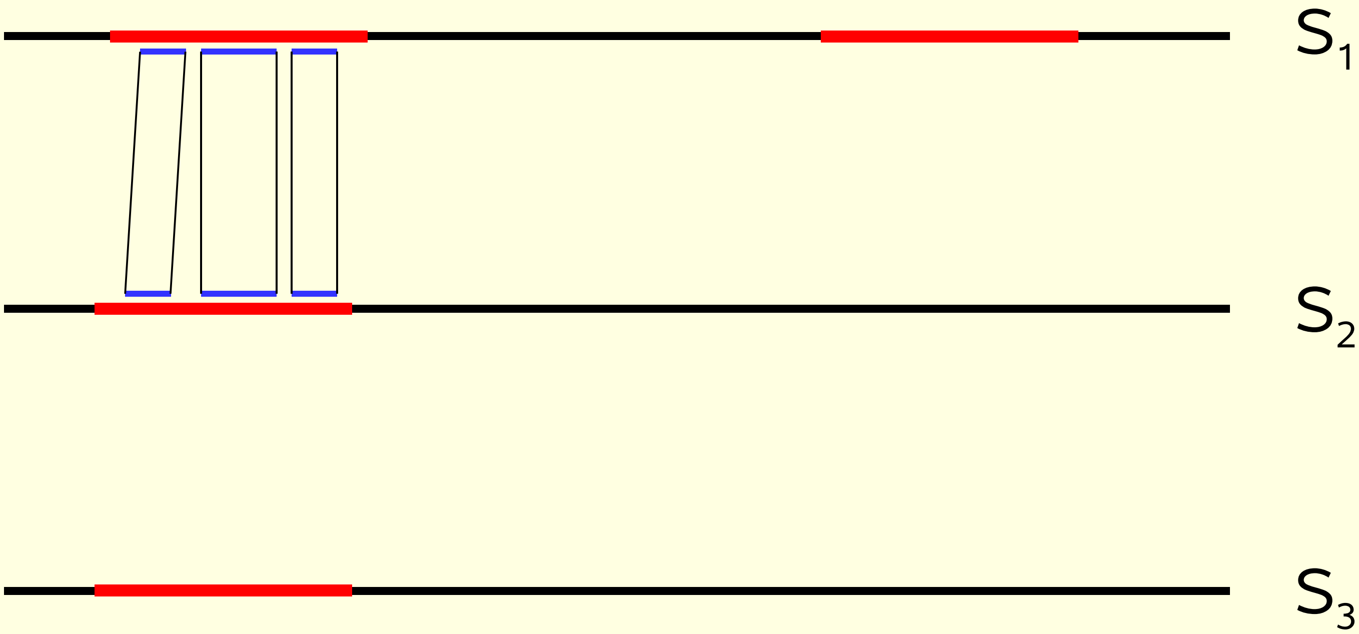
Possible mis-alignment

Alignment of sequence duplications



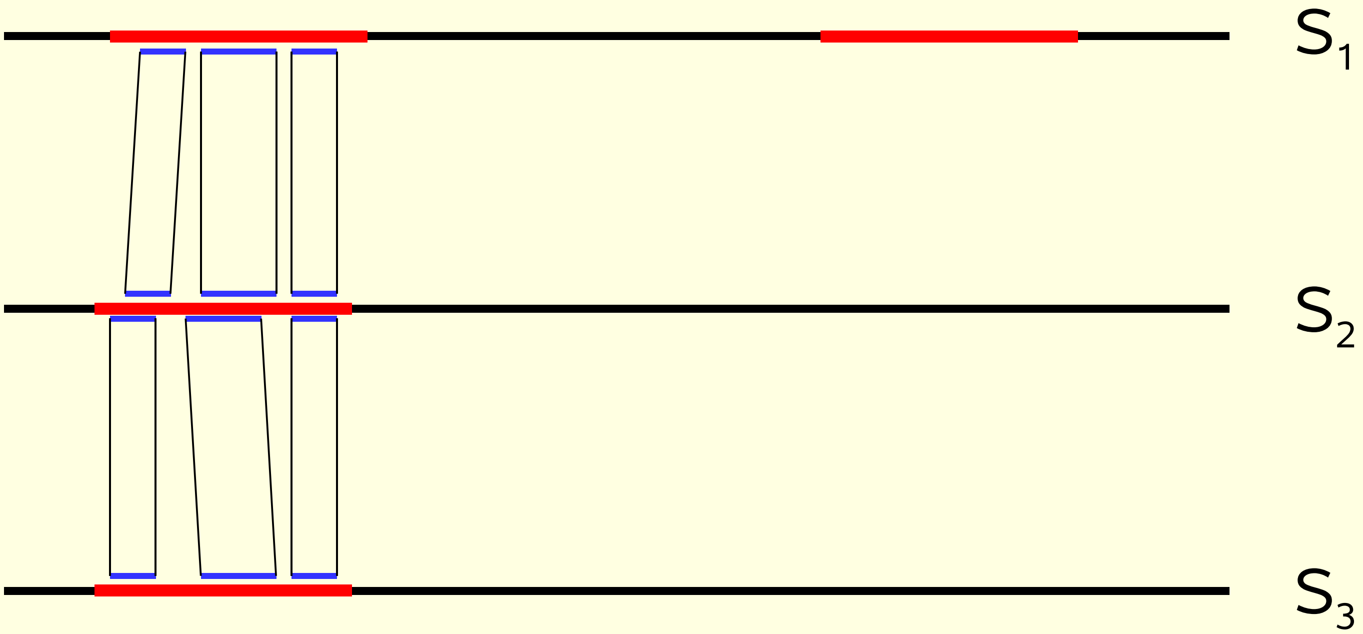
Duplication in *one* sequence

Alignment of sequence duplications



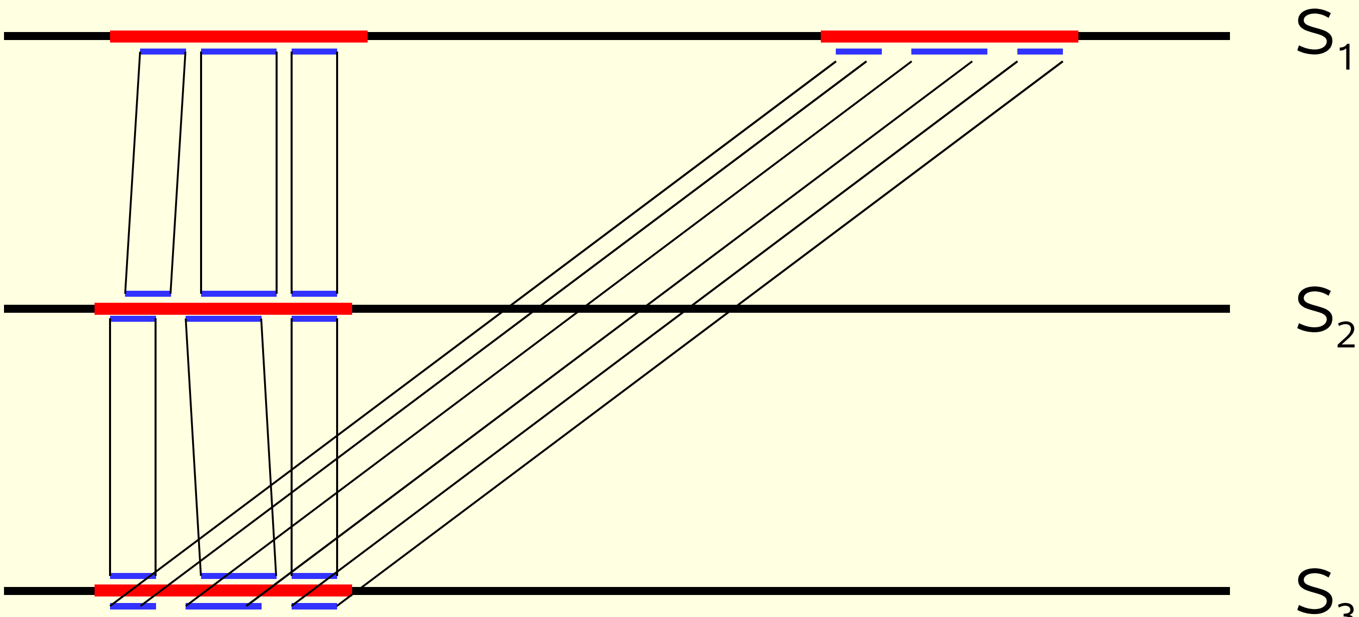
Duplication in *one* sequence

Alignment of sequence duplications



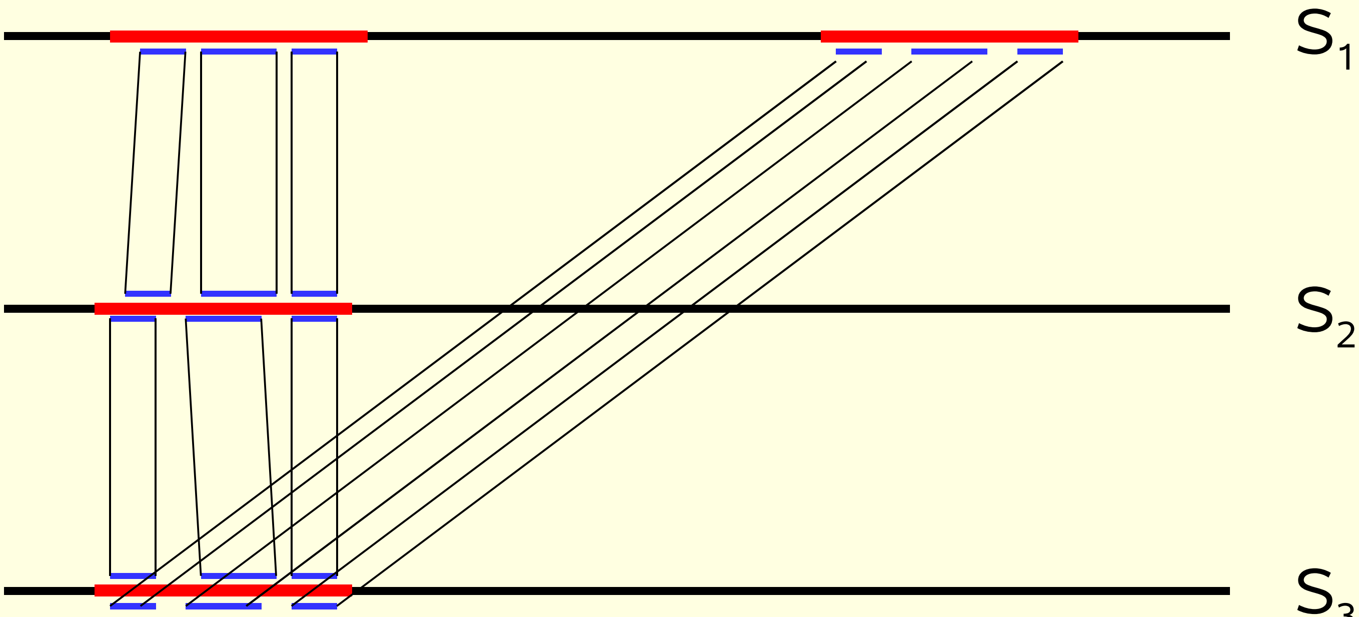
Duplication in *one* sequence

Alignment of sequence duplications



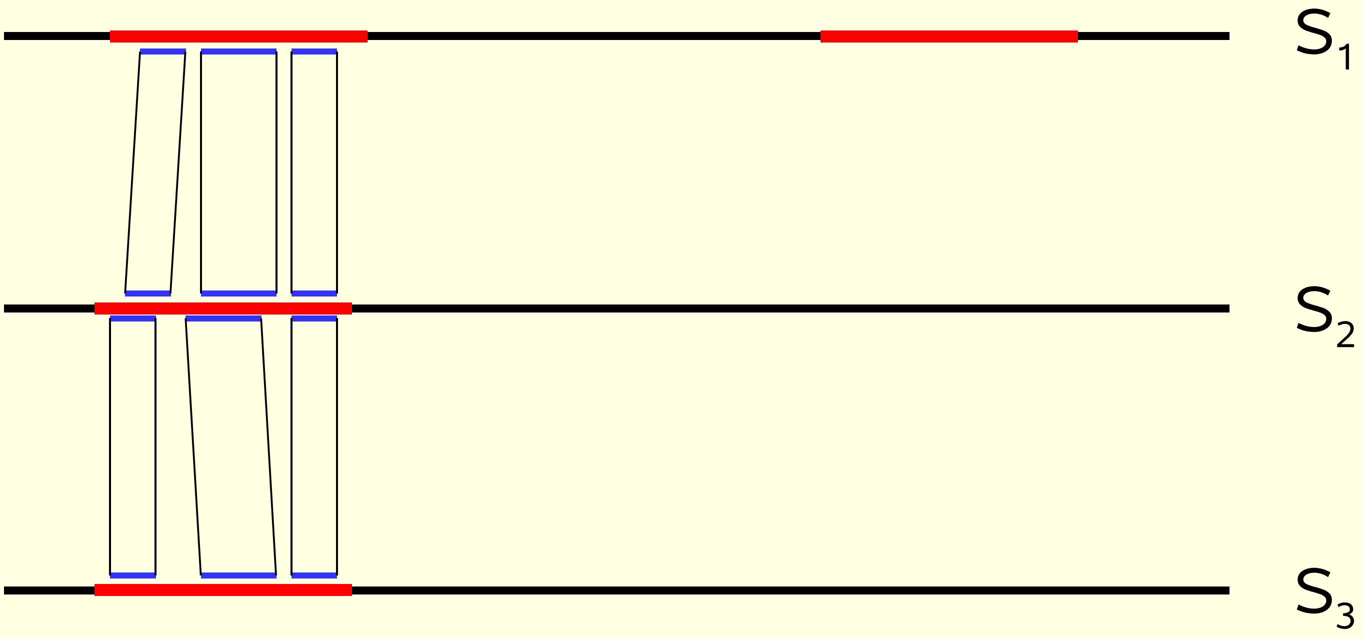
Duplication in *one* sequence

Alignment of sequence duplications



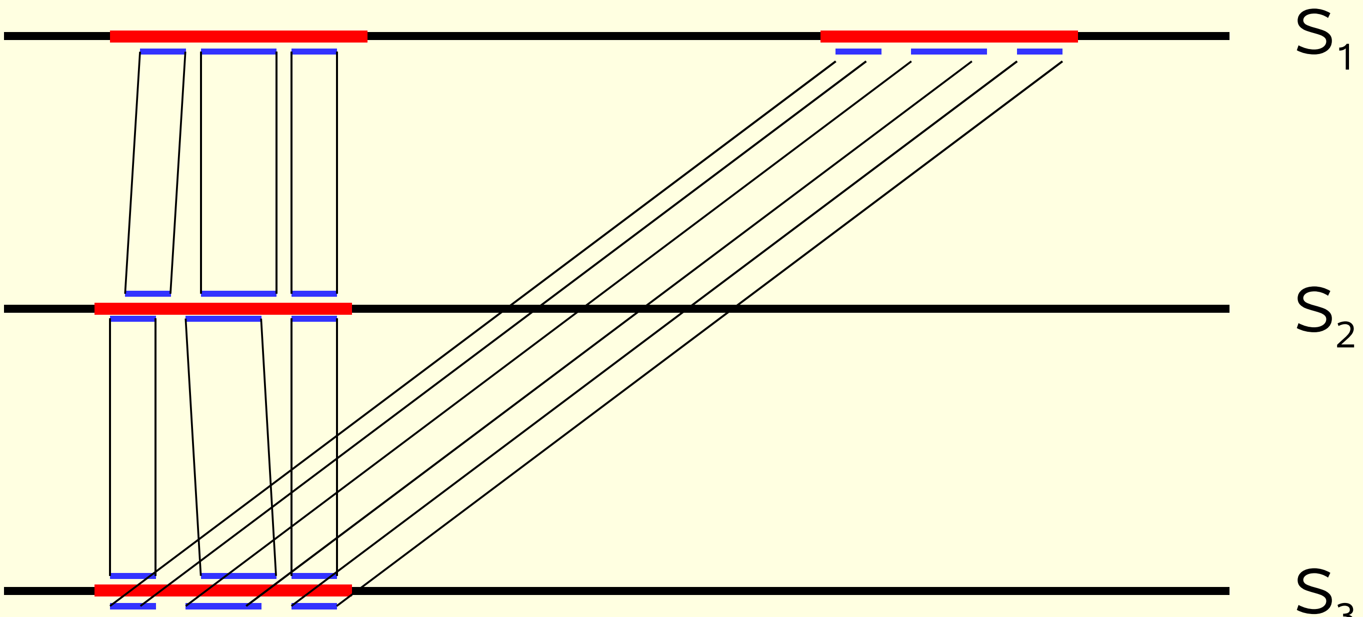
Consistency problem

Alignment of sequence duplications



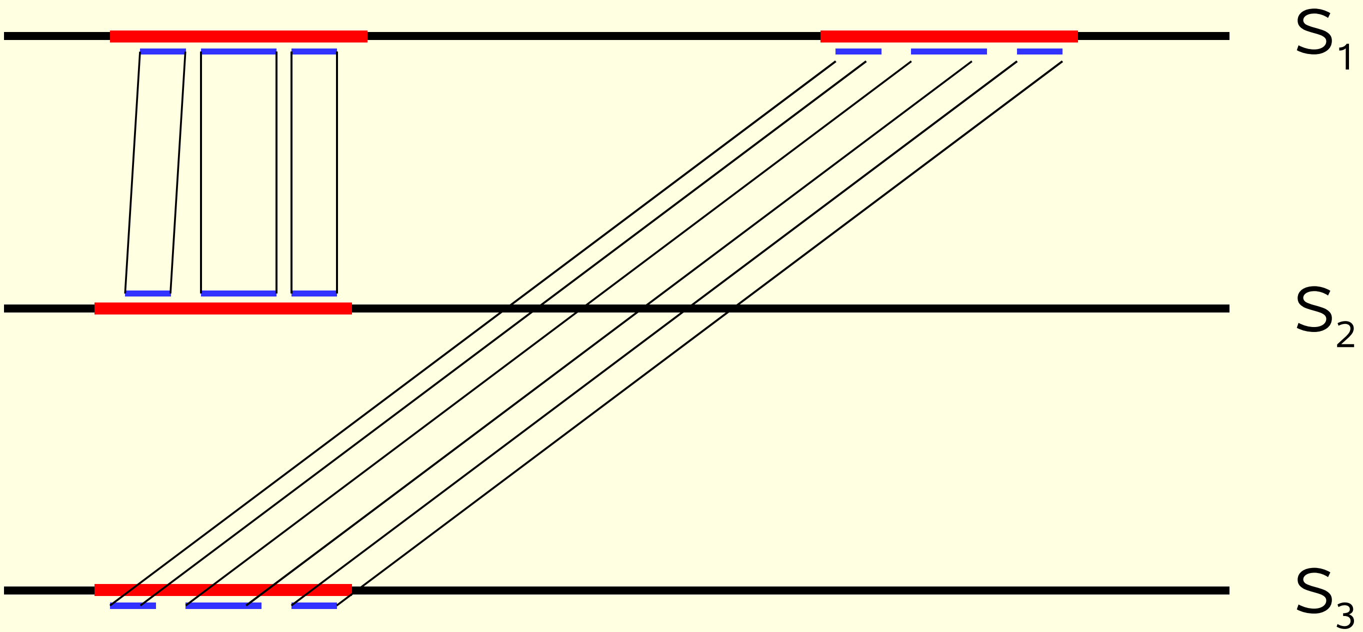
More plausible alignment – and higher score:

Alignment of sequence duplications



Consistency problem

Alignment of sequence duplications



Alternative alignment; probably biologically wrong; lower numerical score!

Anchored sequence alignment

Biologically meaningful alignment often not *possible* by automated approaches.

Anchored sequence alignment

Biologically meaningful alignment not *possible* by automated approaches.

Idea: use expert knowledge to guide alignment procedure

Anchored sequence alignment

Biologically meaningful alignment not *possible* by automated approaches.

Idea: use expert knowledge to guide alignment procedure

User defines a set anchor points that are to be „respected“ by the alignment procedure

Anchored sequence alignment

```
NLFVALYDFVASGDNTLSITKGEKLRVLGYNHN
IIHREDKGVIYALWDYEPQNDDELPMKEGDCMT
```

Anchored sequence alignment

NLFVALYDFVASGDNTLSITKGEKLRVLGYNHN

IIHREDKGVIYALWDYEPQNDDELPMKEGDCMT

Anchored sequence alignment

```
NLFVALYDFVASGDNTLSITKGEKLRVLGYNHN
IIHREDKGVIYALWDYEPQNDDELPMKEGDCMT
```

Use known homology as *anchor point*

Anchored sequence alignment

```
NLFV       ALYDFVASGDNTLSITKGEKLRVLGYNHN
IIHREDKGVIYALWDYEPQNDDELPMKEGDCMT
```

Use known homology as *anchor point*

Anchored sequence alignment

```
NLFV       ALYDFVASGDNTLSITKGEKLRVLGYNHN
IIHREDKGVIYALWDYEPQNDDELPMKEGDCMT
```

Use known homology as *anchor point*

Anchor point = anchored *fragment* (gap-free pair of segments)

Anchored sequence alignment

```
NLFV       ALYDFVASGDNTLSITKGEKLRVLGYNHN
IIHREDKGVIYALWDYEPQNDDELPMKEGDCMT
```

Use known homology as *anchor point*

Anchor point = anchored *fragment* (gap-free pair of segments)

Remainder of sequences aligned automatically

Anchored sequence alignment

```
NLFV       ALYDFVASGDNTLSITKGEKLRVLGYNHN
IIHREDKGVIYALWDYEPQNDDELPMKEGDCMT
```

Alignment of anchored positions *a* and *b* **not** enforced – *a* and *b* may be un-aligned –, but:

Anchored sequence alignment

```
NLFV       ALYDFVASGDNTLSITKGEKLRVLGYNHN
IIHREDKGVIYALWDYEPQNDDELPMKEGDCMT
```

Alignment of anchored positions *a* and *b* **not** enforced – *a* and *b* may be un-aligned –, but:

- *a* is only residue that can be aligned to *b*

Anchored sequence alignment

```
NLFV       ALYDFVASGDNTLSITKGEKLRVLGYNHN
IIHREDKGVIYALWDYEPQNDDELPMKEGDCMT
```

Alignment of anchored positions *a* and *b* **not** enforced – *a* and *b* may be un-aligned –, but:

- *a* is only residue that can be aligned to *b*
- Residues left of *a* aligned with residues left of *b*

Anchored sequence alignment

```
-------NLF VALYDFVASG DNTLSITKGE klrvlgynhn
iihredkGVI YALWDYEPQN DDELPMKEGD cmt-------
```

Anchored alignment

Anchored sequence alignment

```
NLFVALYDFVASGDNTLSITKGEKLRVLGYNHN
IIHREDKGVIYALWDYEPQNDDELPMKEGDCMT
GYQYRALYDYKKEREEDIDLHLGDILTVNKGSLVALGFS
```

Anchor points in multiple alignment

Anchored sequence alignment

```
NLFV       ALYDFVASGDNTLSITKGEKLRVLGYNHN
IIHREDKGVIYALWDYEPQND  DELPMKEGDCMT
GYQYRALYDYKKEREEDIDLHLGDILTVNKGSLVALGFS
```

Anchor points in multiple alignment

Anchored sequence alignment

```
-------NLF V-ALYDFVAS GD-------- NTLSITKGEk lrvLGYNhn
iihredkGVI Y-ALWDYEPQ ND-------- DELPMKEGDC MT-------
-------GYQ YrALYDYKKE REedidlhlg DILTVNKGSL VA-LGFS--
```

Anchored multiple alignment

Algorithmic questions

Goal:

- Find optimal alignment (=consistent set of fragments) under costraints given by user-specified anchor points!

Algorithmic questions

Additional input file with anchor points:

```
1   3    215    231     5    4.5
2   3     34     78    23    1.23
1   4    317    402     8    8.5
```

Algorithmic questions

```
NLFVALYDFVASGDNTLSITKGEKLRVLGYNHN
IIHREDKGVIYALWDYEPQNDDELPMKEGDCMT
GYQYRALYDYKKEREEDIDLHLGDILTVNKGSLVALGFS
```

Algorithmic questions

Additional input file with anchor points:

```
1   3    215    231      5    4.5
2   3     34     78     23    1.23
1   4    317    402      8    8.5
```

Algorithmic questions

Additional input file with anchor points:

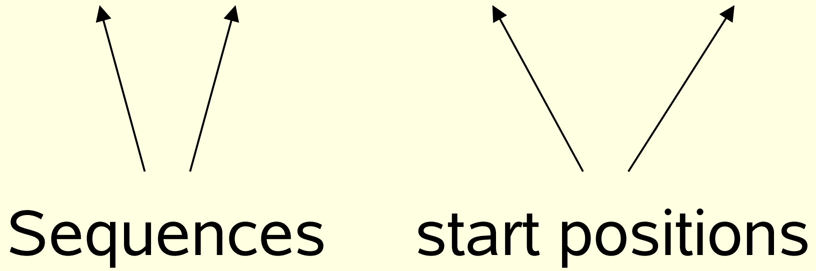
```
1   3    215    231     5    4.5
2   3     34     78    23    1.23
1   4    317    402     8    8.5
```

Sequences

Algorithmic questions

Additional input file with anchor points:

```
1   3   215   231    5   4.5
2   3    34    78   23   1.23
1   4   317   402    8   8.5
```

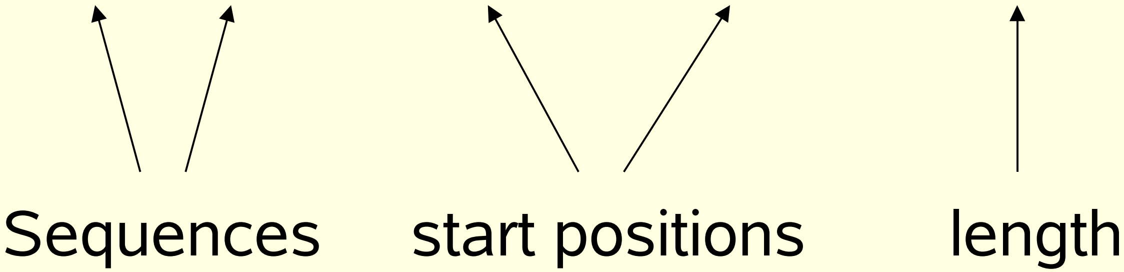


Sequences start positions

Algorithmic questions

Additional input file with anchor points:

```
1   3   215   231    5   4.5
2   3    34    78   23   1.23
1   4   317   402    8   8.5
```

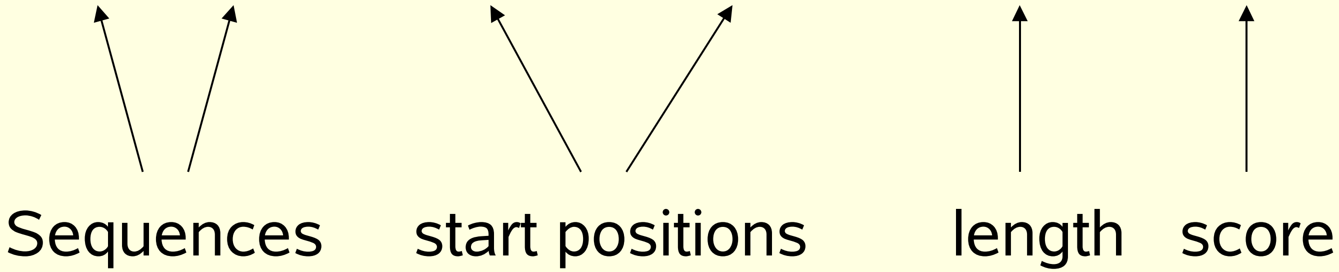


Sequences start positions length

Algorithmic questions

Additional input file with anchor points:

```
1   3   215   231    5   4.5
2   3    34    78   23   1.23
1   4   317   402    8   8.5
```



Sequences   start positions   length   score

Algorithmic questions

Requirements:

- Anchor points need to be *consistent*! – if necessary: select consistent subset from user-specified anchor points

Algorithmic questions

atctaatagttaaactcccccgtgcttag

cagtgcgtgtattactaacggttcaatcgcg

caaagagtatcacccctgaattgaataa

Algorithmic questions

atctaatagttaaactccccgtgcttag

cagtgcgtgtattactaacggttcaatcgcg

caaagagtatcacccctgaattgaataa

Algorithmic questions

```
atctaatagttaaactccccgtgcttag

cagtgcgtgtattactaacggttcaatcgcg

caaagagtatcacccctgaattgaataa
```

Inconsistent anchor points!

Algorithmic questions

```
atctaat---agttaaactcccccgtgcttag

Cagtgcgtgtattac-taacggttcaatcgcg

caaagagtatcacccctgaattgaataa
```

Inconsistent anchor points!

Algorithmic questions

Requirements:

- Anchor points need to be *consistent*! – if necessary: select consistent subset from user-specified anchor points

Algorithmic questions

Requirements:

- Anchor points need to be *consistent*! – if necessary: select consistent subset from user-specified anchor points
- Find alignment under constraints given by anchor points!

Algorithmic questions

Use data structures from multiple alignment

Algorithmic questions

atctaatagttaaactcccccgtgcttag

cagtgc**gtgtatta**ctaacggttcaatcgcg

caaa**gagtatca**cccctgaattgaataa

Algorithmic questions

atctaatagttaaactccccgtgcttag

cagtgc**gtgtatta**ctaacggttcaatcgcg

caaa**gagtatca**cccctgaattgaataa

Greedy procedure for multiple alignment

Algorithmic questions

atc**taatagtta**aactcccccgtgcttag

cagtgc**gtgtattactaa**cggttcaatcgcg

caaa**gagtatca**cccctgaattgaataa

Greedy procedure for multiple alignment

Algorithmic questions

atc**taatagtta**aactcccccgtgcttag

cagtgc**gtgtattactaa**cggttcaatcgcg

caaa**gagtatca**ccccctgaattgaataa

Question: which positions are still alignable ?

Algorithmic questions

atc**taatagtta**aactccccgtgcttag S_i

cagtgc**gtgtattactaa**cggttcaatcgcg

caaa**gagtatca**cccctga**a**ttgaataa

x

For each position *x* and each sequence S_i exist an upper bound *ub(x,i)* and a lower bound *lb(x,i)* for residues *y* in S_i that are *alignable* with *x*

Algorithmic questions

atc**taata|gtta**aactcccccgtgcttag| S_i

cagtgc**gtgtattactaa**cggttcaatcgcg

caaa**gagtatca**cccctga**a**ttgaataa

x

For each position *x* and each sequence S_i exist an upper bound *ub(x,i)* and a lower bound *lb(x,i)* for residues *y* in S_i that are *alignable* with *x*

Algorithmic questions

atc**taata|gtta**aactcccccgtgcttag| S_i

cagtgc**gtgtattactaa**cggttcaatcgcg

caaa**gagtatca**cccctga**a**ttgaataa

x

ub(x,i) and *lb(x,i)* updated during greedy procedure

Algorithmic questions

|atctaatagttaaactccccgtgcttag| S_i

cagtgcgtgtattactaacggttcaatcgcg

caaagagtatcacccctga**a**ttgaataa

x

Initial values of *lb(x,i)*, *ub(x,i)*

Algorithmic questions

|atctaatagttaaactcccccgtgcttag| S_i

cagtgc**gtgtatta**ctaacggttcaatcgcg

caaa**gagtatca**cccctga**a**ttgaataa

x

ub(x,i) and *lb(x,i)* updated during greedy procedure

Algorithmic questions

atc**taata**|**gtta**aactccccgtgcttag| S_i

cagtgc**gtgtattactaa**cggttcaatcgcg

caaa**gagtatca**cccctga**a**ttgaataa

x

ub(x,i) and *lb(x,i)* updated during greedy procedure

Algorithmic questions

Anchor points treated like *fragments* in greedy algorithm:

Algorithmic questions

Anchor points treated like *fragments* in greedy algorithm:

- Sorted according to user-defined *scores*

Algorithmic questions

Anchor points treated like *fragments* in greedy algorithm:

- Sorted according to user-defined *scores*
- Accepted if *consistent* with previously accepted anchors

Algorithmic questions

Anchor points treated like *fragments* in greedy algorithm:

- Sorted according to user-defined *scores*
- Accepted if *consistent* with previously accepted anchors
- *ub(x,i)* and *lb(x,i)* updated during greedy procedure

Algorithmic questions

Anchor points treated like *fragments* in greedy algorithm:

- Sorted according to user-defined *scores*
- Accepted if *consistent* with previously accepted anchors
- *ub(x,i)* and *lb(x,i)* updated during greedy procedure

Resulting values of *ub(x,i)* and *lb(x,i)* used as *initial* values for alignment procedure

Algorithmic questions

|atctaatagttaaactcccccgtgcttag| S_i

cagtgcgtgtattactaacggttcaatcgcg

caaagagtatcacccctga**a**ttgaataa
x

Initial values of *lb(x,i)*, *ub(x,i)*

Algorithmic questions

atctaa|tagttaaact|cccccgtgcttag S_i

cagtgcgtgtattactaacggttcaatcgcg

caaagagtatcacccctga**a**ttgaataa

x

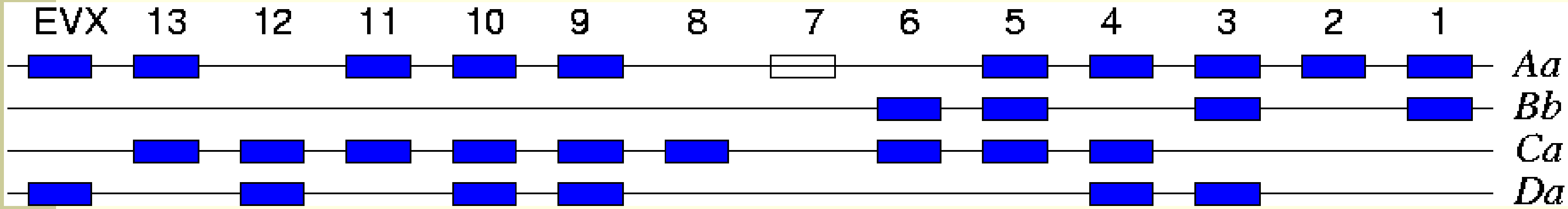
Initial values of *lb(x,i)*, *ub(x,i)* calculated using anchor points

Algorithmic questions

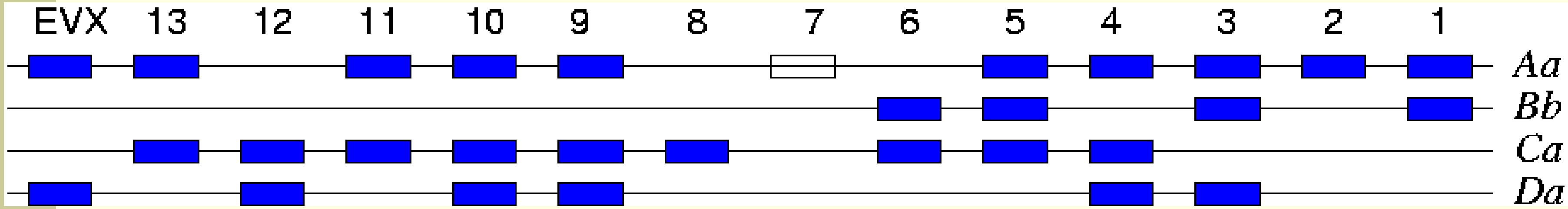
Ranking of anchor points to prioritize anchor points, e.g.

- anchor points from verified homologies -- higher priority
- automatically created anchor points (using CHAOS, BLAST, …) -- lower priority

Application: *Hox* gene cluster

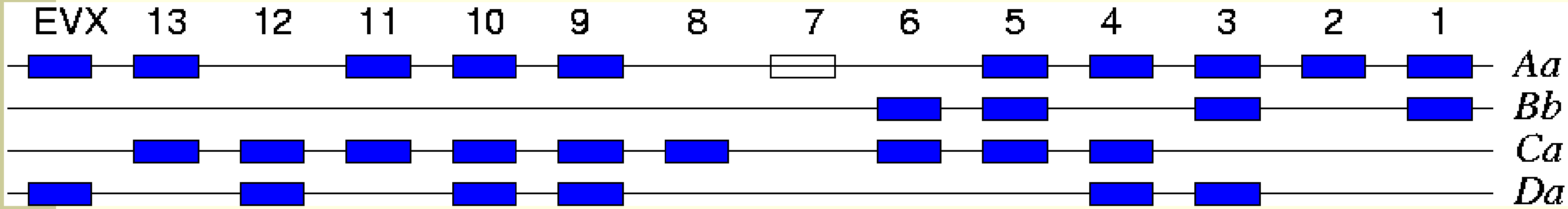


Application: *Hox* gene cluster



Use gene boundaries as anchor points

Application: *Hox* gene cluster



Use gene boundaries as anchor points

+ CHAOS / BLAST hits

Application: *Hox* gene cluster

	no anchoring	anchoring
Ali. Columns		
2 seq	2958	3674
3 seq	668	1091
4 seq	244	195
Score	1166	1007
CPU time	4:22	0:19

Application: *Hox* gene cluster

Example:

Teleost *Hox* gene cluster:

Application: *Hox* gene cluster

Example:

Teleost *Hox* gene cluster:

Score of anchored alignment 15 % higher than score of non-anchored alignment !

Application: *Hox* gene cluster

Example:

Teleost *Hox* gene cluster:

Score of anchored alignment 15 % higher than score of non-anchored alignment !

Conclusion: Greedy optimization algorithm does a bad job!

Application: Improvement of Alignment programs

Two possible reasons for mis-alignments:

Application: Improvement of Alignment programs

Two possible reasons for mis-alignments:

- Wrong **objective function**: *Biologically* correct alignment gets bad *numerical* score

Application: Improvement of Alignment programs

Two possible reasons for mis-alignments:

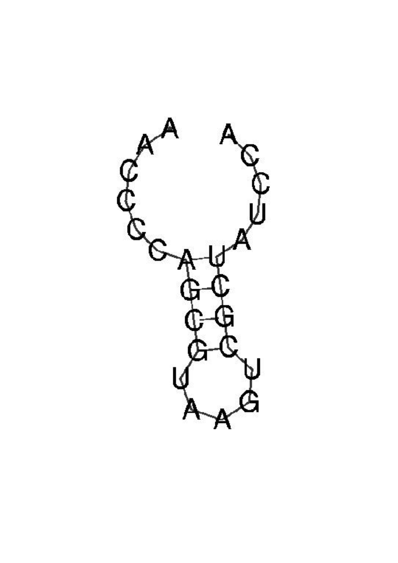
- Wrong **objective function**: *Biologically* correct alignment gets bad *numerical* score
- Bad **optimization algorithms**: Biologically correct alignment gets best numerical score, but algorithm fails to find this alignment

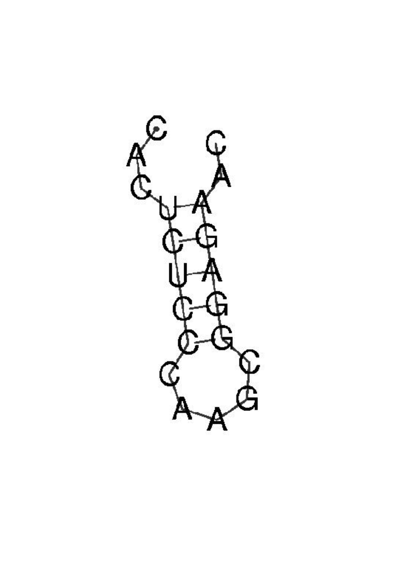
Application: Improvement of Alignment programs

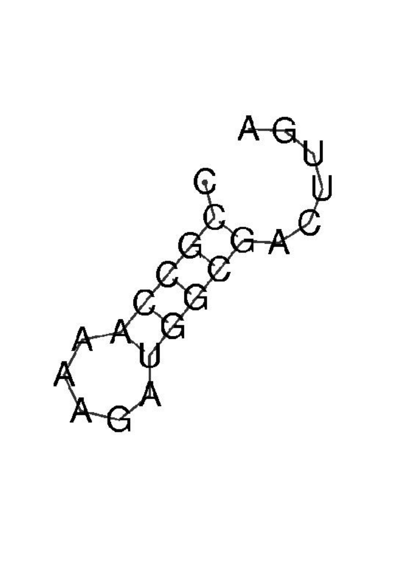
Two possible reasons for mis-alignments:

- Anchored alignments can help to decide

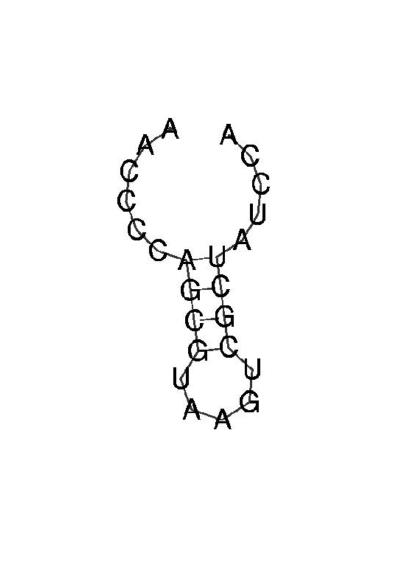
Application: RNA alignment

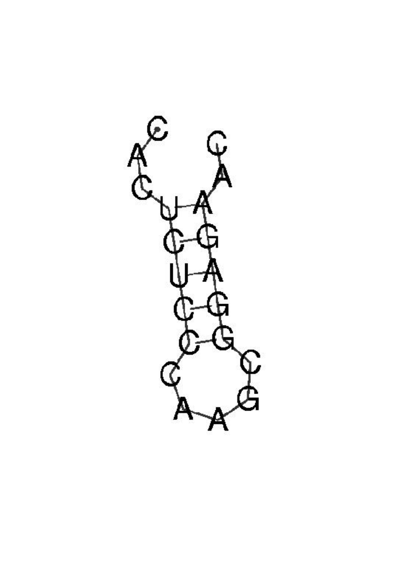


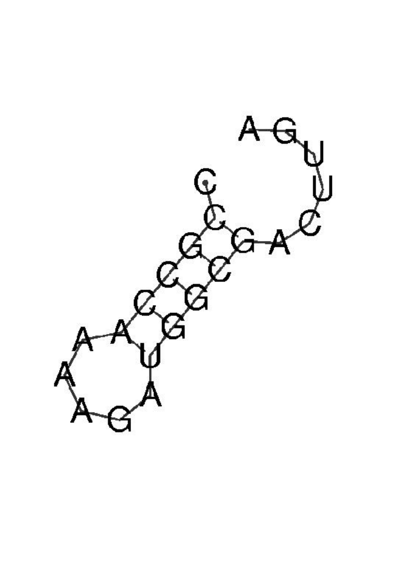




Application: RNA alignment



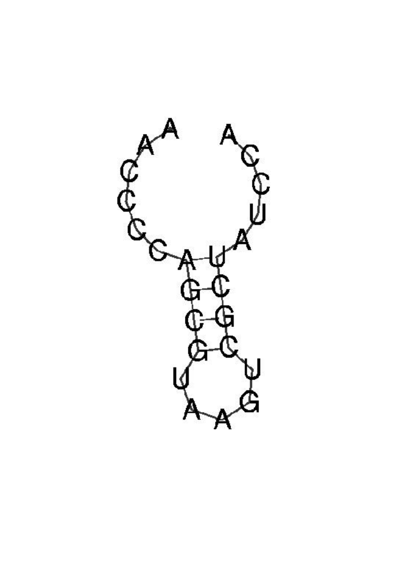


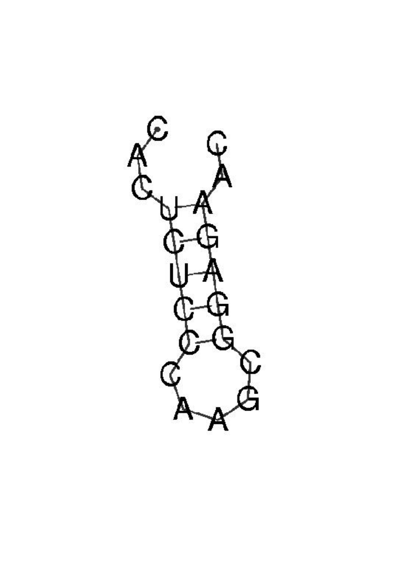


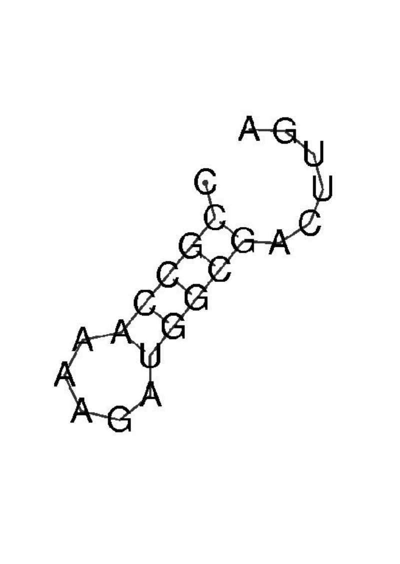
```
aa----CCCC AGC---GUAa gucgcuaucc a
cacucuCCCA AGC---GGAG Aac------- -
ccg----CCA AaagauGGCG Acuuga---- -
```

non-anchored alignment

Application: RNA alignment



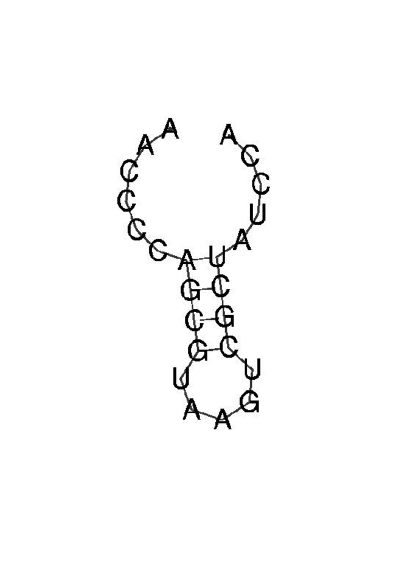


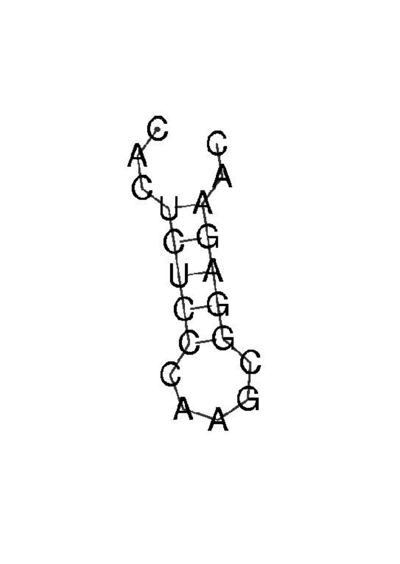


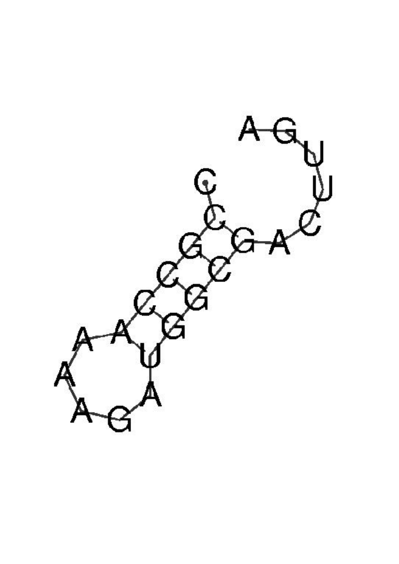
```
aa----CCCC AGC---GUAa gucgcuaucc a
cacucuCCCA AGC---GGAG Aac-------- -
ccg----CCA AaagauGGCG Acuuga---- -
```

structural motif mis-aligned

Application: RNA alignment



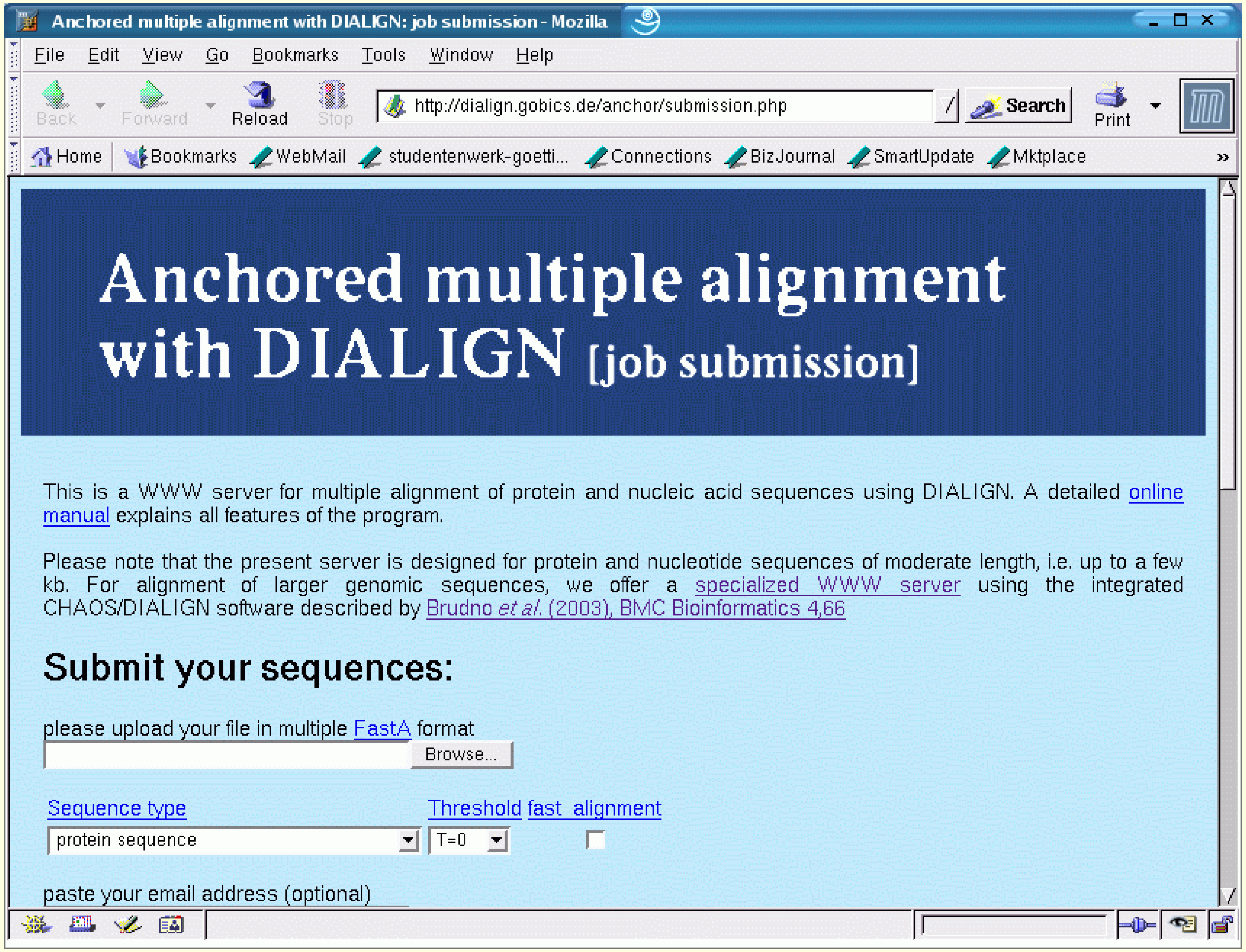




```
aaCCCCAGCG UAAGUCGCUA UCca--
--CACUCUCC CAAGCGGAGA AC----
----CCGCCA AAAGAUGGCG ACuuga
```

3 conserved nucleotides as anchor points

WWW interface at GOBICS (Göttingen Bioinformatics Compute Server)



Anchored multiple alignment with DIALIGN: job submission - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://dialign.gobics.de/anchor/submission.php
Search
Print
Home Bookmarks WebMail studentenwerk-goetti... Connections BizJournal SmartUpdate Mktplace

Anchored multiple alignment with DIALIGN [job submission]

This is a WWW server for multiple alignment of protein and nucleic acid sequences using DIALIGN. A detailed online manual explains all features of the program.

Please note that the present server is designed for protein and nucleotide sequences of moderate length, i.e. up to a few kb. For alignment of larger genomic sequences, we offer a specialized WWW server using the integrated CHAOS/DIALIGN software described by Brudno *et al.* (2003), BMC Bioinformatics 4,66

Submit your sequences:

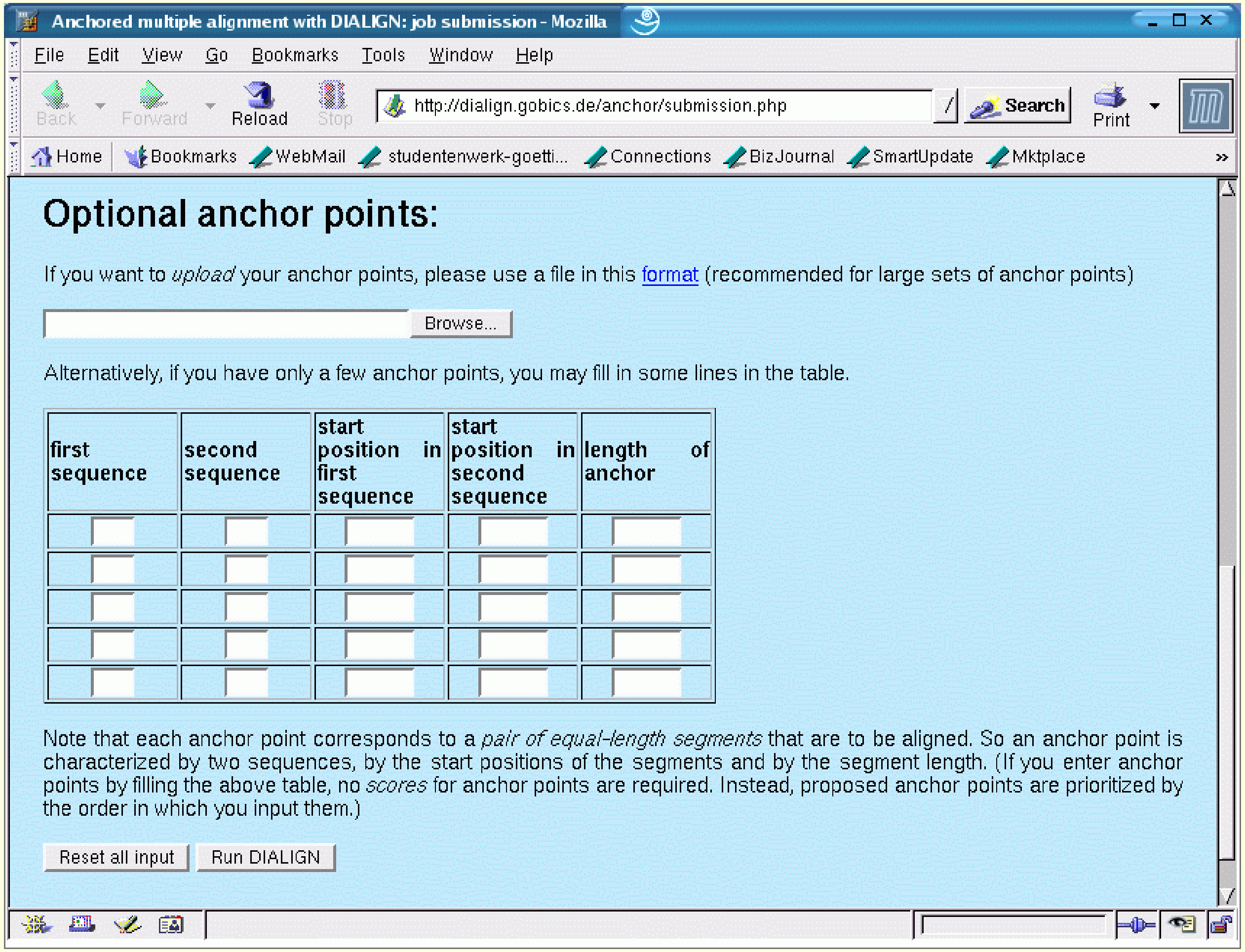
please upload your file in multiple FastA format

Browse...

Sequence type	Threshold	fast alignment
protein sequence	T=0	□

paste your email address (optional)

WWW interface at GOBICS (Göttingen Bioinformatics Compute Server)



Anchored multiple alignment with DIALIGN: job submission - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://dialign.gobics.de/anchor/submission.php
Search
Print
Home Bookmarks WebMail studentenwerk-goetti... Connections BizJournal SmartUpdate Mktplace

Optional anchor points:

If you want to *upload* your anchor points, please use a file in this format (recommended for large sets of anchor points)

Browse...

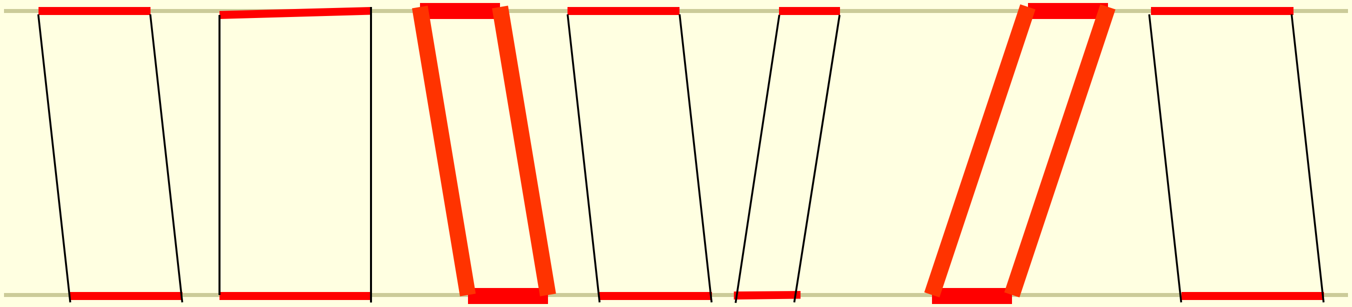
Alternatively, if you have only a few anchor points, you may fill in some lines in the table.

first sequence	second sequence	start position in first sequence	start position in second sequence	length of anchor

Note that each anchor point corresponds to a *pair of equal-length segments* that are to be aligned. So an anchor point is characterized by two sequences, by the start positions of the segments and by the segment length. (If you enter anchor points by filling the above table, no *scores* for anchor points are required. Instead, proposed anchor points are prioritized by the order in which you input them.)

Reset all input   Run DIALIGN

Anchored sequence alignment



Alignments between anchor points can be calculated independently on parallel processors

Anchored sequence alignment

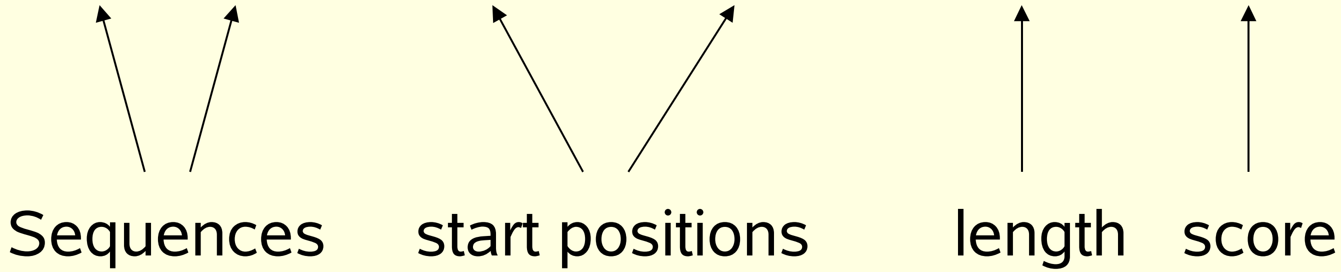
File containing anchor points:

```
1   3    215    231     5    4.5
2   3     34     78    23    1.23
1   4    317    402     8    8.5
```

Anchored sequence alignment

File containing anchor points:

```
1    3    215    231     5    4.5
2    3     34     78    23    1.23
1    4    317    402     8    8.5
```



Sequences start positions length score

```
-NLFVALYDfvasgdntlsitkGEKLRVLgynhn-----
kGVIYALWDyepqnddelpmkeGDCMTIIhrede-----
gYQYRALYDykkereedidlhlGDILTVNkgslvalgfs
-NFRVYYRDsrd------pvwkGPAKLLWkg--------
-drvrkksga---------awqGQIVGWYctnlt-----
```

Input: sequence data

Goal: align *biologically* related residues!
= residues related by structure, function, evolution

Anchored sequence alignment

WKKNADAPKRAMTSFMKAAY

WNLDTNSPEEKQAYIQLAKDDRIRYD

WRMDSNQKNPDSNNPKAAYNKGDANAPK

Anchored sequence alignment

WKKNADAPKRAMTSFMKAAY

WNLDTNSPEEKQAYIQLAKDDRIRYD

WRMDSNQKNPDSNNPKAAYNKGDANAPK

Anchored sequence alignment

```
WKKNAD-----APKRAMTSFMKAAY------------
WNLDTN-----SPEE------KQAYIQLAKDDRIRYD
WRMDSNQKNPDSNNP------KAAYN---KGDANAPK
```

Anchored sequence alignment

WKKNADAPKRAMTSFMKAAY

WNLDTNSPEEKQAYIQLAKDDRIRYD

WRMDSNQKNPDSNNPKAAYNKGDANAPK

Anchored sequence alignment

WKKNADAPKRAMTSFMKAAY

WNLDTNSPEEKQAYIQLAKDDRIRYD

WRMDSNQKNPDSNNPKAAYNKGDANAPK

Anchored sequence alignment

WKKNADAPKRAMTSFMKAAY

WNLDTNSPEEKQAYIQLAKDDRIRYD

WRMDSNQKNPDSNNPKAAYNKGDANAPK

Anchored sequence alignment

```
WKK-------NADAPKRAMTSFMKAA---Y-
WNLDT-----NSPEEKQAYIQLAKDDRIRYD
WRMDSNQKNPDSNNPKAAYN---KGDANAPK
```