Structural alignment of RNA sequences with the FOLDALIGN algorithm

Jakob Hull Havgaard

Division of Genetics and Bioinformatics
Department of Basic Animal and Veterinary Science
Faculty of Life Sciences
University of Copenhagen

Overview

Introduction Why is it so hard to align ncRNAs

FOLDALIGN The recursion, and the scoring scheme

Heuristics λ, δ, and bifurcations

Pruning Method and local alignment results

Backtrack Divide and conquer

Global alignment pruning

Closing remarks

Introduction

Compensating mutations: Base pairs can change with little effect on the structure of the molecule

The secondary structure is much more conserved

A good algorithm combines sequence similarity information with secondary structure information

A general algorithm proposed by Sankoff in 1985 is too slow for more than a few sequences

Even for two sequences the time complexity is $O(L_1^3 L_2^3)$ and the memory complexity is $O(L_1^2 L_2^3)$

FOLDALIGN

FOLDALIGN is an implementation of the pairwise Sankoff algorithm

It can make a local or global structural alignment of two sequences

There is now also a global multiple alignment version called foldalignM, which is based on the PMcomp algorithm (Torarinsson et al. Bioinformatics, in press)

Scoring scheme

- Single nucleotide substitutions
- Base pair substitutions
- Maximum energy co-folding

Recursion

$$D_{(i-1)(j+1),(k-1)(l+1)} = \max\{D_{ij,kl} + S_{n_i n_j, n_k n_l}, D_{(i-1)(j+1),(k-1)(l+1)}\} \quad \text{(a)}$$

$$D_{(i-1)(j+1),kl} = \max\{D_{ij,kl} + S_{n_i n_j, --}, D_{(i-1)(j+1),kl}\} \quad \text{(b)}$$

$$D_{ij,(k-1)(l+1)} = \max\{D_{ij,kl} + S_{--, n_k n_l}, D_{ij,(k-1)(l+1)}, \} \quad \text{(c)}$$

$$D_{(i-1)j,(k-1)l} = \max\{D_{ij,kl} + S_{n_i -, n_k -}, D_{(i-1)j,(k-1)l}, \} \quad \text{(d)}$$

$$D_{i(j+1),k(l+1)} = \max\{D_{ij,kl} + S_{-n_j, -n_l}, D_{i(j+1),k(l+1)}\} \quad \text{(e)}$$

$$D_{(i-1)j,kl} = \max\{D_{ij,kl} + S_{n_i -, --}, D_{(i-1)j,kl}\} \quad \text{(f)}$$

$$D_{i(j+1),kl} = \max\{D_{ij,kl} + S_{-n_j, --}, D_{i(j+1),kl}\} \quad \text{(g)}$$

$$D_{ij,(k-1)l} = \max\{D_{ij,kl} + S_{--, n_k -}, D_{ij,(k-1)l}\} \quad \text{(h)}$$

$$D_{ij,k(l+1)} = \max\{D_{ij,kl} + S_{--, -n_l}, D_{ij,k(l+1)}\} \quad \text{(i)}$$

$$D_{im,kn}{}^{*} = \max\{D'_{ij,kl} + E_{j+1,m,l+1,n} + S_{mbl}, D_{im,kn}\} \quad \text{(j)}$$

$^{*}: \forall\ \begin{matrix} j+1 < m \leq i+\lambda \\ l+1 < n \leq k+\lambda \end{matrix}$

$$D'_{ij,kl} = \begin{cases} D_{ij,kl} & \text{if } n_i \text{ \& } n_{j'} \text{ and } n_k \text{ \& } n_{l'} \text{ base pairs} \\ & \text{where } i < j' \leq j \text{ and } k < l' \leq l \\ \varnothing & \text{Otherwise} \end{cases}$$

$$E_{ij,kl} = \begin{cases} D_{ij,kl} & \text{If } n_i \text{ \& } n_j \text{ and } n_k \text{ \& } n_l \text{ base pairs} \\ \varnothing & \text{Otherwise} \end{cases}$$

Heuristics

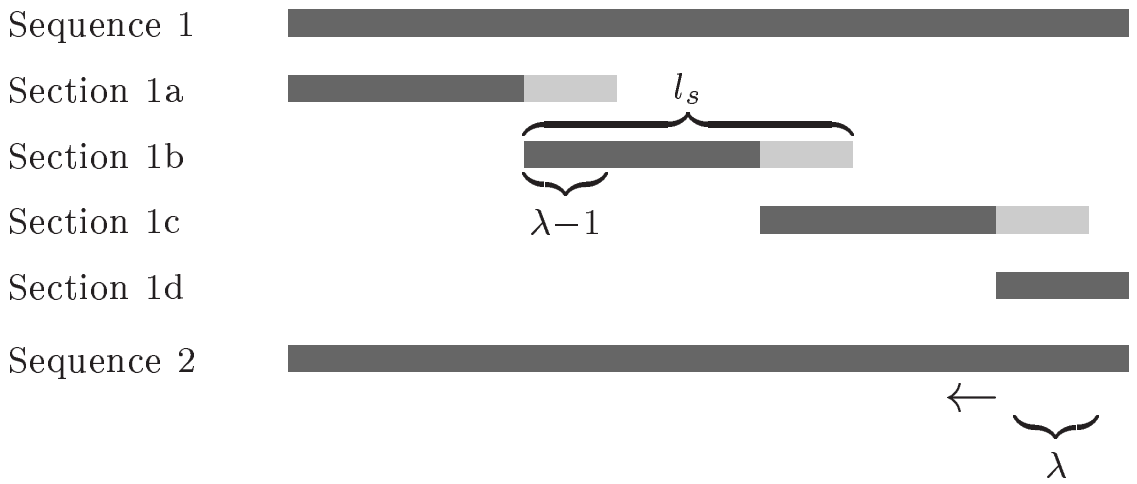
The full algorithm is very slow and needs huge amounts of memory

Heuristics (cheats) are used as work-arounds for these problems

FOLDALIGN uses four types of heuristics:

1. Maximum motif length — λ
2. Maximum subsequence length difference — δ
3. Bifurcation constraint
4. Pruning

λ and δ



Time complexity: $O(L_1^3 L_2^3) \rightarrow O(L_1 L_2 \lambda^2 \delta^2)$

Memory complexity: $O(L_1^2 L_2^2) \rightarrow O(L_1 L_2 \lambda \delta) \rightarrow O(\lambda^3 \delta)$

Bifurcation constraint

More heuristics are needed

The algorithm is still slow

Two types of heuristics are currently very popular:

1. Pre-alignment
2. Pre-folding

We take a different approach

Pruning

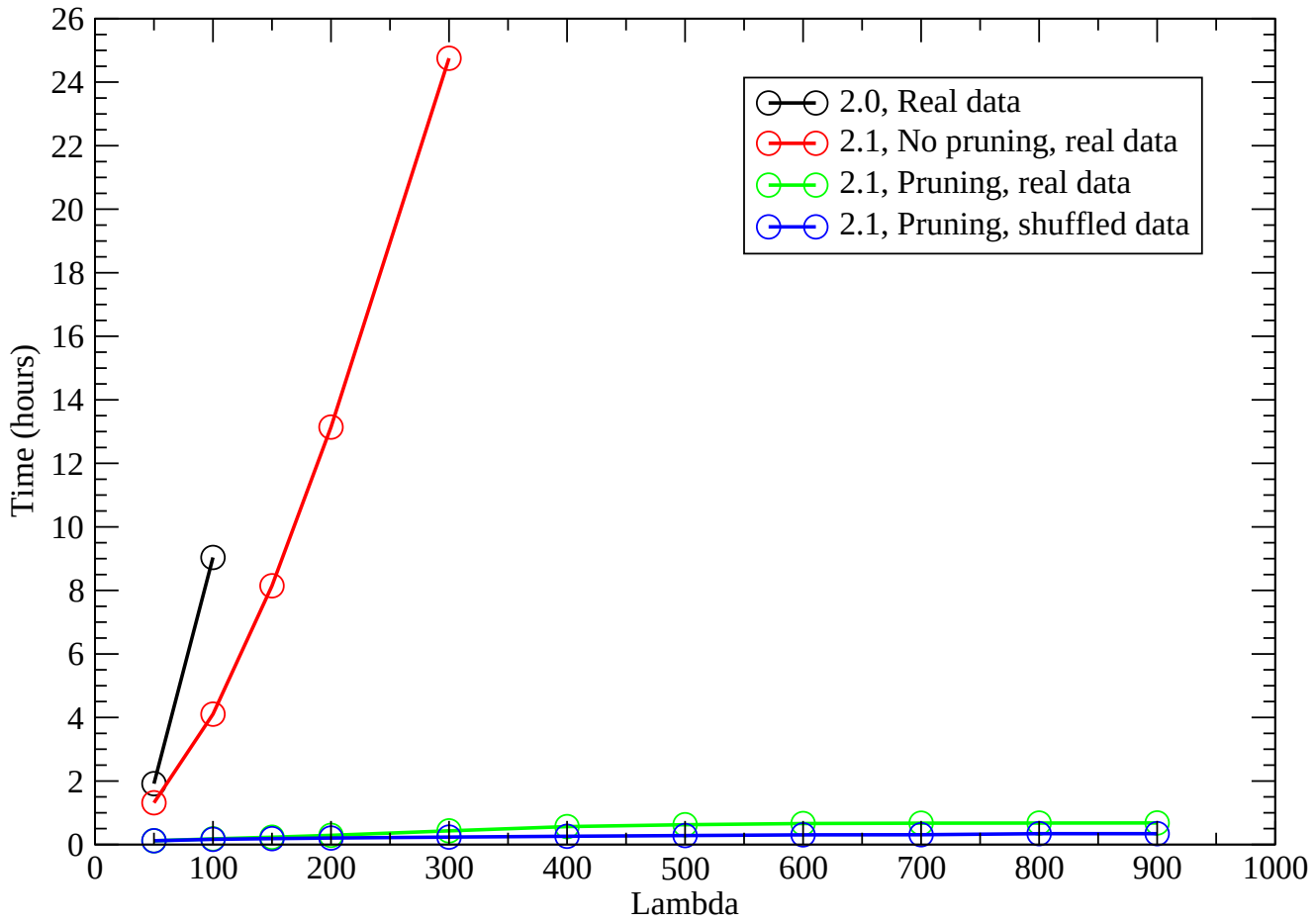
Lots of poor alignments is calculated

Don’t let them waste your time

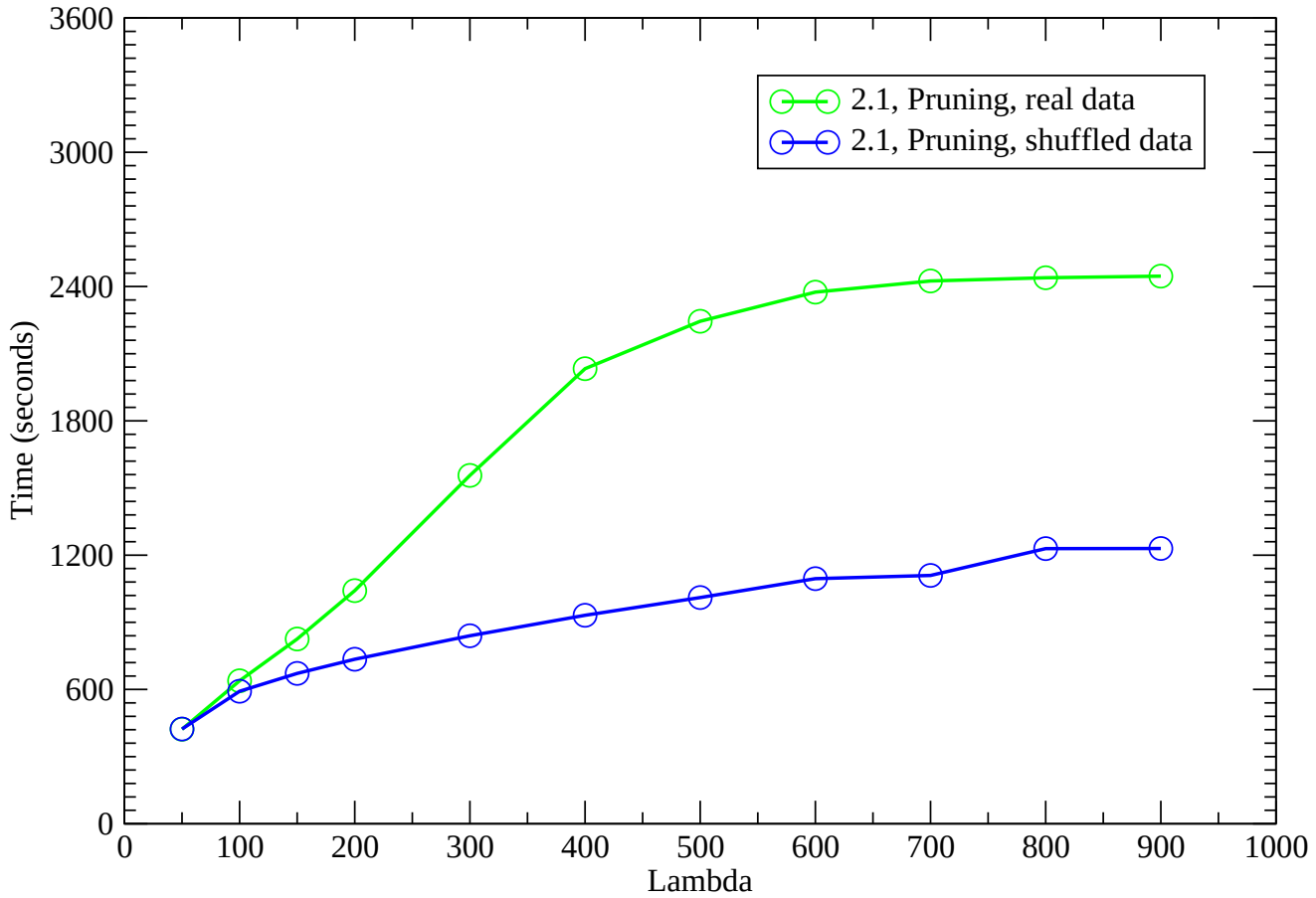
Use pruning!

A sub-alignment of a given length must have a score above a length dependent minimum score or it is removed (pruned)

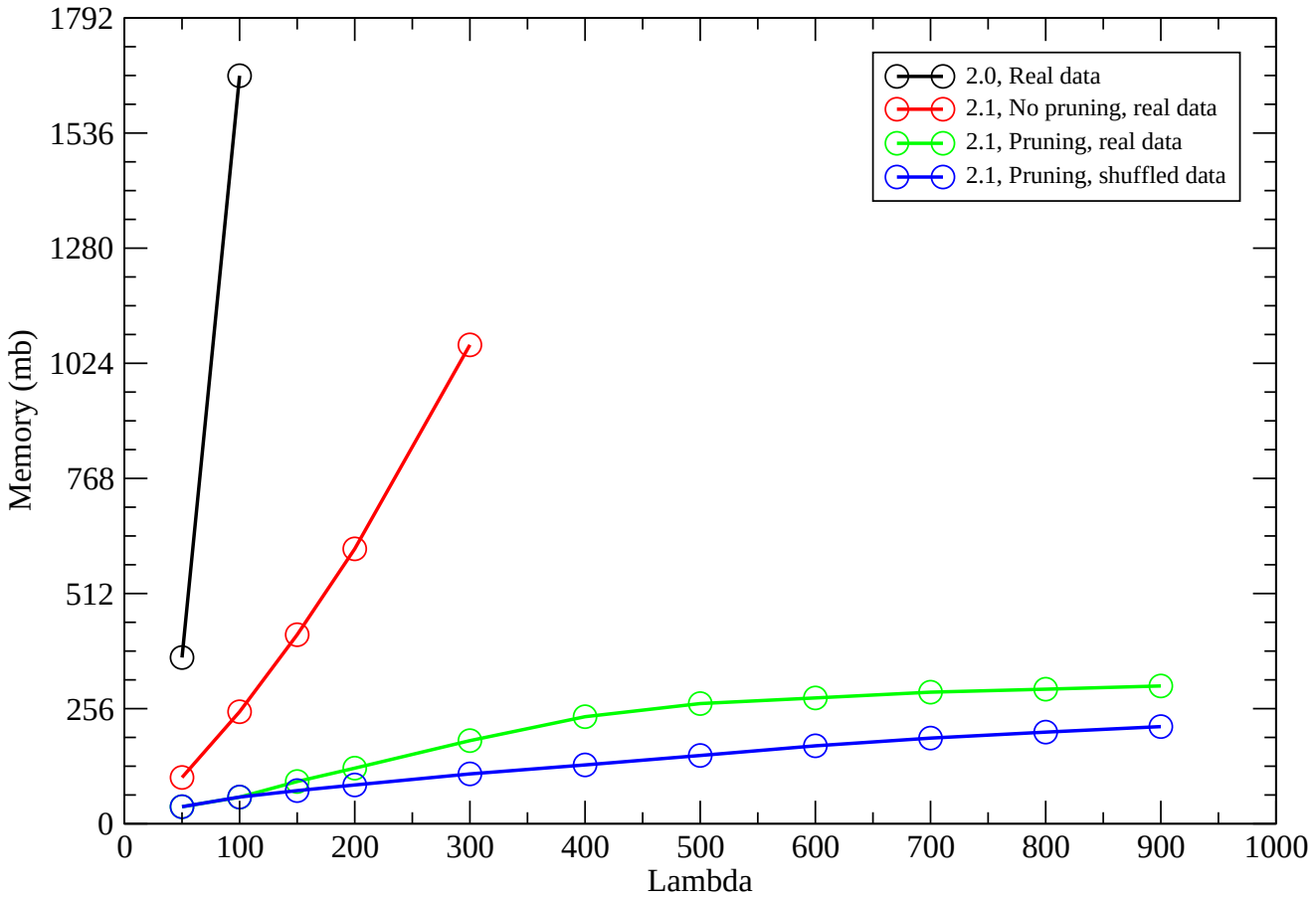
Pruning results - time



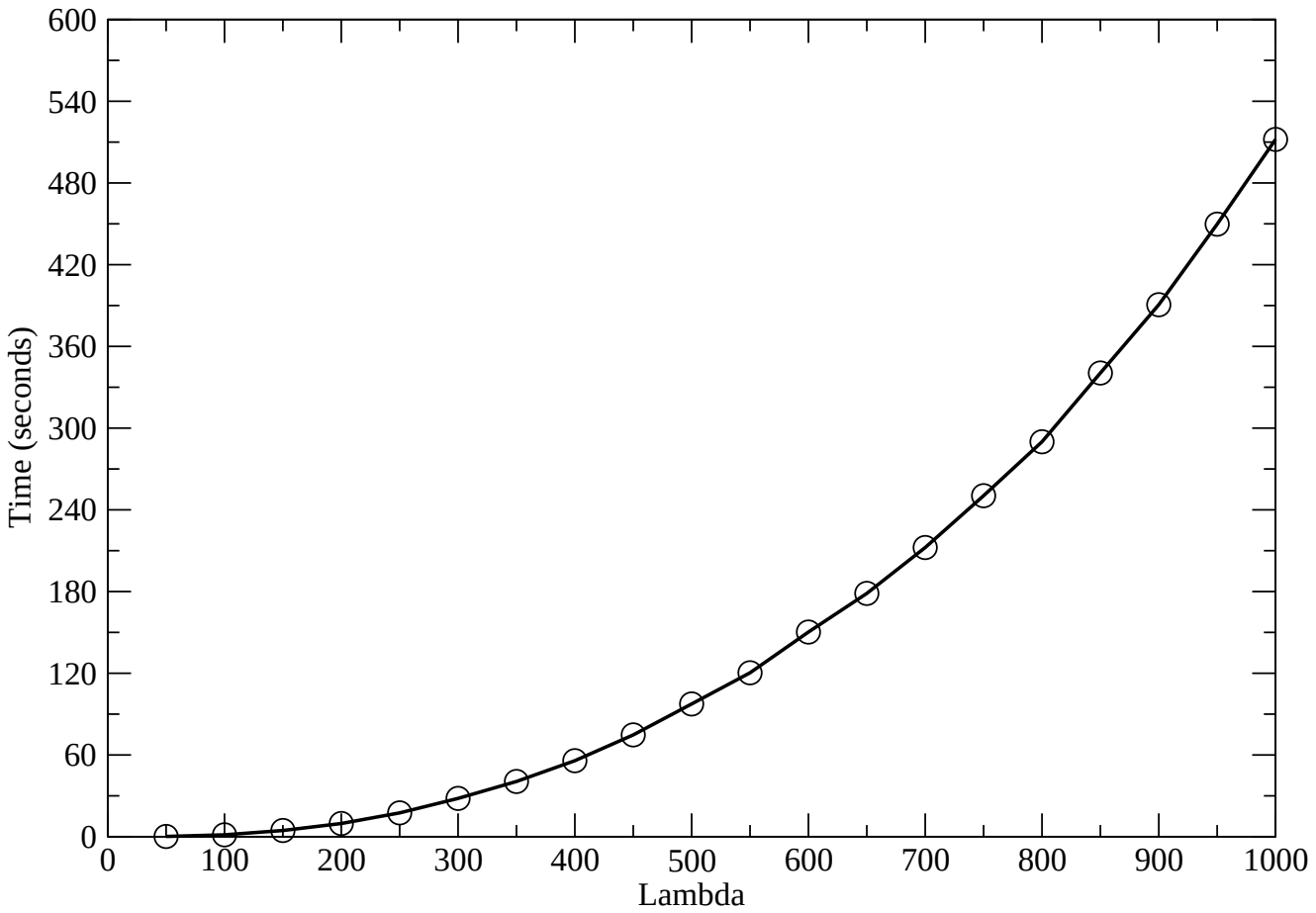
Pruning results - time detail



Pruning results - memory



Pruning results - full time



Localization

	No pruning					Pruning				
Type	P_t	P_f	N_f	PPV	Sens	P_t	P_f	N_f	PPV	Sens
5S rRNA	2	0	0	1.00	1.00	2	0	0	1.00	1.00
Purine	3	2	2	0.60	0.60	3	2	2	0.60	0.60
THI	12	11	9	0.52	0.57	13	11	8	0.54	0.62
U1	6	1	0	0.86	1.00	6	1	0	0.86	1.00
tRNA	171	75	72	0.70	0.70	165	72	78	0.70	0.68
Unknown		12					13			
Average				0.74	0.77				0.74	0.78

Backtrack — divide and conquer

Backtrack realignment uses more memory than the local alignment scan

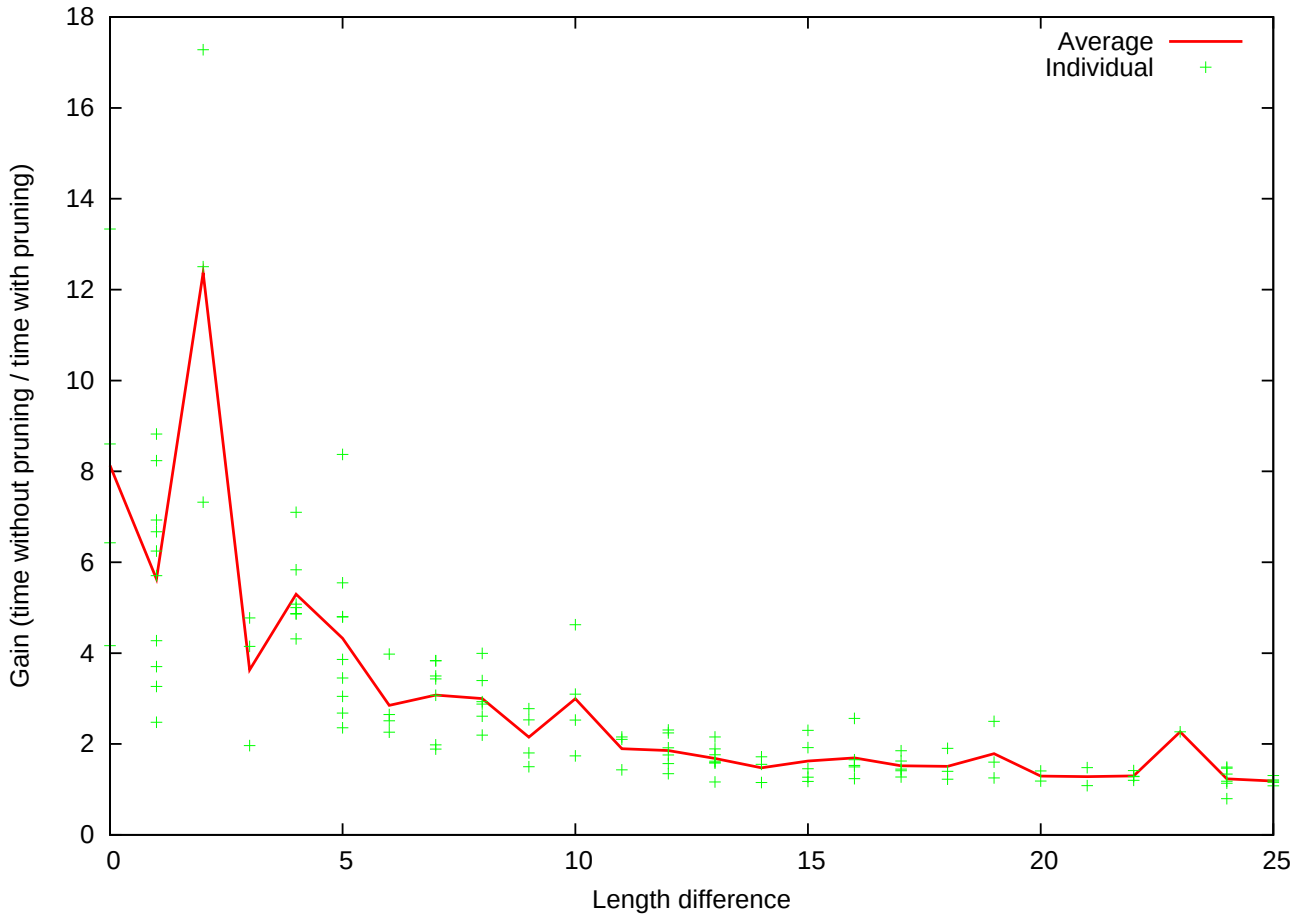
1. A local alignment scan finds the coordinates of the alignment
2. A pre-backtrack realignment is used to find all the bifurcation points in the conserved structure
3. The bifurcation points are used to split the structure into (hopefully) smaller unbifurcated segments
4. Each segment is realigned with the unbranched algorithm and backtracked in a normal fashion

Global alignment and pruning

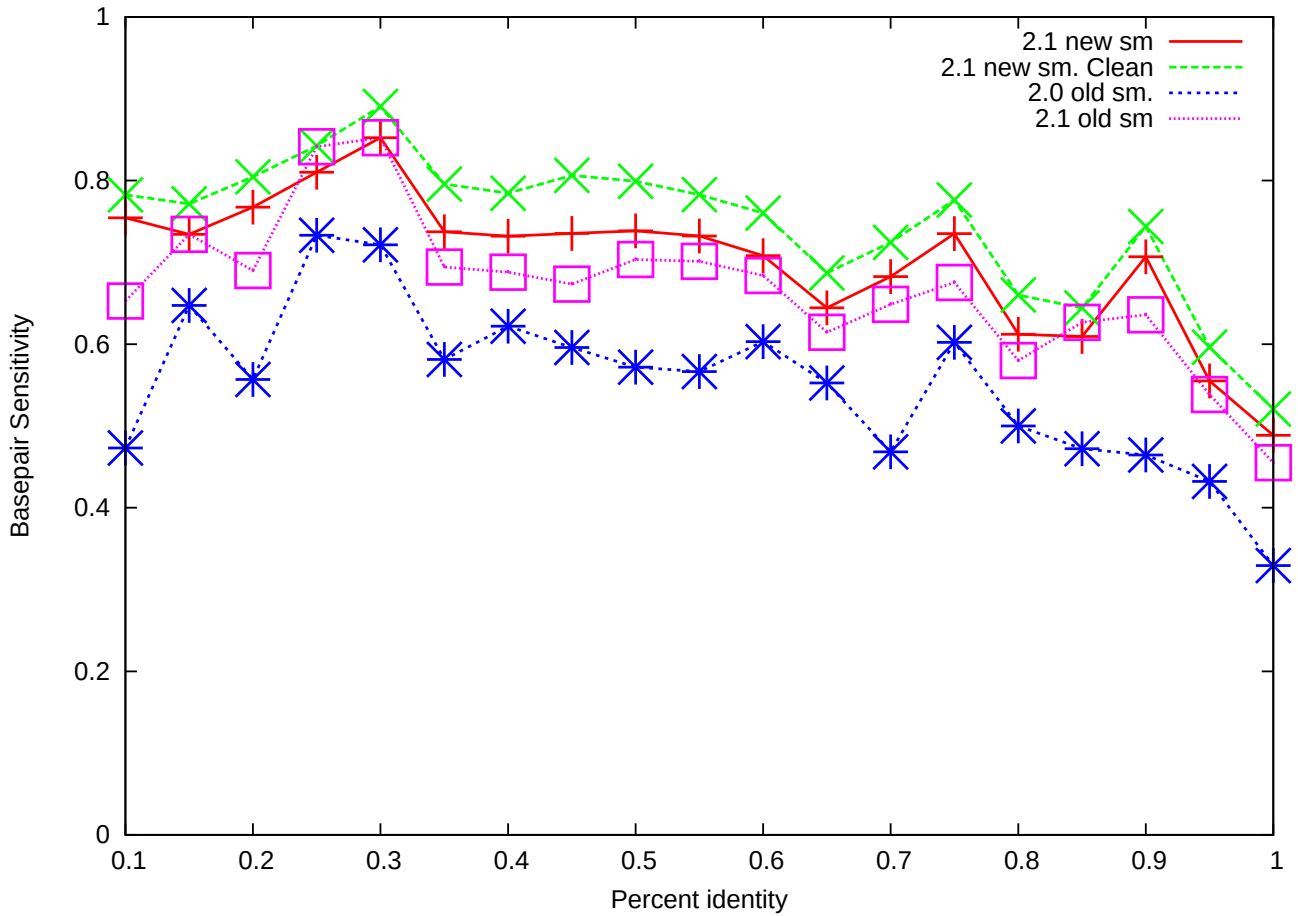
The pruning is too effective during global alignment. Too much is lost

$$\Theta_{global} = \Theta_{local}(l_1, l_2) - \min\{\mathrm{abs}(l_1 - l_2), \mathrm{abs}(L_1 - L_2)\} \times G_E$$

Global alignment - time



Global alignment - performance



Conclusion

FOLDALIGN is a tool for local and global alignment of RNA sequences

It is fast and memory efficient (for an implementation of the Sankoff algorithm)

It is easy to use

It can make pairwise local structural alignments

It makes good global alignments (Dowell et al. 2006, Gardner et al. 2005)

Its global structure predictions are as good as that of competing algorithms

Web-server, source code etc are available at http://foldalign.ku.dk

Acknowledgements

Jan Gorodkin, Elfar Torarinsson

Rune Lyngsø, and Gary Stormo