Master Thesis

Improving multiple sequence alignments using information from libraries of optimal pairwise alignments

Author(s):
Hunziker, Angela

Publication Date:
2003

Permanent Link:
https://doi.org/10.3929/ethz-a-004616597

Rights / License:
In Copyright - Non-Commercial Use Permitted
Improving Multiple Sequence Alignments
Using Information from Libraries of Optimal Pairwise Alignments

Angela Hunziker
August 21, 2003

Supervision:
Prof. Dr. G. Gonnet
Dr. P. von Rohr
## Contents

1 Summary ........................................ 5

2 Introduction ..................................... 6

3 Construction of the Optimal Pairwise Library .... 8
   3.1 Primary Library .................................. 8
   3.2 Extended Library .................................. 8

4 Multiple Sequence Alignment ....................... 11
   4.1 Circular Tours .................................. 11
   4.2 Probabilistic Algorithm ......................... 12
   4.3 The new implementation ......................... 14

5 Analysis ........................................... 16
   5.1 BAliBase ........................................ 16
      5.1.1 Reference 1: equidistant sequences ........ 16
      5.1.2 Reference 2: family with orphan sequences .. 17
      5.1.3 Reference 3: families of related sequences .. 17
      5.1.4 Reference 4: extensions at either end ....... 17
      5.1.5 Reference 5: internal insertions ............ 17
      5.1.6 Alignment scores ........................... 17
   5.2 MAlign with / without improvements ............ 18
      5.2.1 Reference 1: equidistant sequences ........ 18
      5.2.2 Reference 2: family with orphan sequences .. 20
      5.2.3 Reference 3: families of related sequences .. 22
      5.2.4 Reference 4: extensions at either end ....... 23
      5.2.5 Reference 5: internal insertions ............ 24
      5.2.6 The whole sequence set of BAliBase .......... 24
   5.3 Comparison with other alignment programs ...... 25
      5.3.1 Reference 1: equidistant sequences ........ 26
      5.3.2 Reference 2: family with orphan sequences .. 28
      5.3.3 Reference 3: families of related sequences .. 30
      5.3.4 Reference 4: extensions at either end ....... 31
5.3.5 Reference 5: internal insertions .......................... 32
5.4 ClustalW with improvements ................................. 33

6 Discussion and Conclusions .................................. 35
List of Figures

1  The primary library [1]  ........................................ 9
2  Extending the primary library  .................................. 10
3  The extended library  .............................................. 10
4  First step of the progressive pairwise alignment  ......... 12
5  Next step of the progressive pairwise alignment  .......... 12
6  Node with two leaves as children  ............................ 13
7  Node with one leaf and one internal node as children ...... 13
8  Node with two internal nodes as children ................. 14
9  The new implementation  .......................................... 15
10 Comparison with MAli gn, reference 1: < 25%, 20 – 40% and
    > 35%  .......................................................... 19
11 Comparison with MAlign, reference 1: the whole sequence set 19
12 Comparison with MAlign, reference 2: short, medium, long  20
13 Comparison with MAlign, reference 2: the whole sequence set 21
14 Comparison with MAlign, reference 3  .......................... 22
15 Comparison with MAlign, reference 4  .......................... 23
16 Comparison with MAlign, reference 5  .......................... 24
17 Comparison with MAlign, the whole sequence set ........... 25
18 An overview of the different multiple alignment algorithms 26
19 Comparison of various alignment programs, reference 1: <
    25%, 20 – 40%, > 35%  ........................................ 27
20 Comparison of various alignment programs, reference 1: the
    whole sequence set  ............................................ 28
21 Comparison of various alignment programs, reference 2: the
    whole sequence set  ............................................ 29
22 The alignment of small and large families when orphans are
    introduced  ..................................................... 29
23 Comparison of various alignment programs, reference 3  .... 30
24 Comparison of reference 1 with reference 3  ................ 31
25 Comparison of various alignment programs, reference 4  .... 32
26 Comparison of various alignment programs, reference 5  .... 33
27 Comparison of IPM, ICT and ClustalW  ........................ 34
1 Summary

The calculation of multiple sequence alignment has become very essential in molecular biology. A number of approaches have been developed to calculate the alignment with acceptable requirements of memory and CPU time.

T-Coffee is a recently developed meta-algorithm, which constructs a library of pairwise alignments to improve the subsequent computation of the multiple alignment. I implemented this improvement in Darwin and handed it over to the Darwin built-in function MAlign, which performs the multiple alignment using either circular tours or the probabilistic algorithm.

Three comparisons are made in this document using the BaliBase reference alignments as test cases. First, a comparison is made between the new implementation and MAlign performed either with circular tours or the probabilistic method. Then the new implementation is compared to other alignment programs: DIALIGN, SAGA, PRRP, PIMA and HMMT. Finally, the new implementation is compared to ClustalW with the improved library of pairwise alignments.

It is shown that the new implementation of the above described meta-algorithm outperforms all the other alignment programs with some exception where it performs equally well as other programs.

The global methods perform better than the local ones, when no insertions or extensions are introduced into the sequences. The performance of all the programs is affected by the length of the sequences, the degree of identity of the sequences and the presence of large insertions and deletions in the alignment. It is shown that there exists a lower bound at 20\% sequence identity for all the programs, where the quality of the alignments deteriorates. But alignment of the conserved regions is still possible even below that limit.
2 Introduction

The multiple alignment of protein sequences is one of the major problems in computational biology. It is used to find characteristic motifs and conserved regions in protein families, and it is applied to secondary structure prediction. A dynamic programming algorithm exists to solve this problem and guarantees a mathematically optimal alignment. But this method is limited to the alignment of a small number of short sequences since the computing time and the memory requirements grow exponentially with the number of sequences to be aligned.

Therefore various heuristic approaches have been developed, which use different strategies (progressive, iterative, mixed) based on very different algorithms. A traditional approach is the progressive pairwise alignment method, which builds the multiple alignment gradually by aligning the closest sequences first and successively adding the more distant ones. The closeness of the sequences is determined by the PAM distance. A number of alignment programs implementing this method exist, such as PIMA [9], ClustalW [8] and the Darwin built-in function MAlign. Iterative strategies have been implemented recently to refine and improve the initial alignment, for example DIALIGN [5], SAGA [6], HMMT [10] and PRRP [7] (which consists of progressive parts as well).

Multiple sequence alignments are typically computed by progressive pairwise alignment algorithms, which are computationally efficient. But all progressive pairwise alignment algorithms have the same problem that errors introduced during any stage of the computation cannot be corrected at later stages.

T-Coffee - a novel method for multiple sequence alignments - tries to circumvent this problem. It compiles a library of pairwise sequence alignments and improves it by comparing these pairwise alignments to all the remaining sequences. The pairwise sequence alignments are built by Lalign and ClustalW, whereas the multiple sequence alignment is computed using ClustalW.

A significant improvement is expected by compiling a library of pairwise alignments using dynamic programming, and by computing the multiple se-
quence alignment by either circular tours or the probabilistic algorithm. The information contained in the library is expected to improve the placement of gaps in the multiple sequence alignment.

This meta-algorithm for computing multiple sequence alignments was implemented in Darwin. The algorithm first compiles a library of optimal pairwise alignments and then builds the multiple alignment by using circular tours or the probabilistic method.

This new implementation was compared to existing implementations such as DIALIGN, SAGA and other methods.
3 Construction of the Optimal Pairwise Library

Let us assume we have four sequences A, B, C and D. From these sequences we want to construct an optimal pairwise library, which can then be used to improve an MSA (Fig. 2) [1].

3.1 Primary Library

The construction of the optimal pairwise library consists of two steps. First of all, a primary library is computed, which contains a set of pairwise alignments between all the sequences to be aligned. The second step improves the alignments of the primary library by comparing them to all other sequences. This results in the extended library, which is used to improve the multiple sequence alignment.

The pairwise alignments of the primary library are computed using conventional dynamic programming. The scores for that computation are based on the Dayhoff scoring system. Each pairwise alignment is assigned a primary weight, which equals the score in the alignment.

3.2 Extended Library

The extended library stems from the primary library. Each pairwise alignment of the primary library is examined further and compared to all the remaining sequences. The alignment of each character (amino acid) pair is checked with the corresponding character of the remaining sequences. This results in examining all the triplets involving the pair that is currently considered.

Consider the alignment A-B of sequences A and B of the example in Fig. 1. Each aligned character, e.g. the first aligned character G, of the alignment A-B, takes the primary weight of the alignment as the initial weight, e.g. \( W(A_G, B_G) = 92 \). Start with the first aligned character, G of sequences A and B, and look at the remaining sequences C and D (Fig. 2). In sequence C, the first character G is well aligned with the G of sequence A. The same is
true for the G of sequence B. Thus it can be said that the alignment of $A_G$ and $B_G$ is supported by sequence C. Hence the minimal weight of the alignment A-C and B-C is added to the weight $W(A_G, B_G)$, i.e. $W(A_G, B_G) = 92 + 85$. Then look at the same character G in sequence D. $A_G$ is aligned with a gap in sequence D, as well as $B_G$, which means that this does not give any additional information relative to $A_G$ and $B_G$ and it does not influence the weight $W(A_G, B_G)$. In summary, the weight associated with a character pair, e.g. $W(A_G, B_G)$, will be the sum of all the weights gathered through the examination of all the triplets involving that pair. The same is done for all subsequent positions.

This leads to a matrix (length of sequence A $\times$ length of sequence B) that is specific to the alignment A-B. This matrix consists of the weights that have been calculated during the examination of all the triplets involving the alignment A-B. Now the alignment A-B can be recomputed using conventional dynamic programming with this specific matrix instead of a general amino acid weight matrix. [1]

All these modified pairwise alignments together are the extended library (Fig. 3). These pairwise alignments are later used for improving the multiple sequence alignment.
Figure 2: Extending the primary library

Figure 3: The extended library
4 Multiple Sequence Alignment

Once the extended library has been built, as described in section 4, it can be used to improve the multiple sequence alignment. Two corresponding methods are used to compute an initial MSA - circular tours or the probabilistic algorithm. The circular tours method is based on progressive pairwise alignments in which the order of the pairwise alignments is determined by solving the traveling salesman problem on the all-against-all pairwise PAM-distance matrix. The probabilistic algorithm uses probabilistic ancestral sequences on a given evolutionary phylogenetic tree to compute the multiple alignment.

4.1 Circular Tours

The multiple sequence alignment by circular tours is built using progressive pairwise alignments, which are derived from the extended library. The order of the pairwise alignments is given by the solution of the traveling salesman problem. [2]

The circular tour is computed or approximated by a symmetric traveling salesman problem. In a traveling salesman problem, a number of cities and its distances are given. Each city has to be visited exactly once and the overall path must be minimal.

This is applied to the problem of finding a circular tour, where the cities correspond to the sequences to be aligned and the distances to the PAM distances of the pairwise alignments. The circular tour is then broken at the largest distance between two sequences, which results in the circular order.

The progressive pairwise alignment starts with sequences 1 and 2 of the circular order and aligns them. This alignment was already computed and stored in the extended library. Hence we just retrieve it from there. In the next step, sequences 2 and 3 of the circular order are aligned and the gaps are propagated. The old gaps are propagated into the new alignment and the new gaps into the old alignment, i.e. the gap in the alignment 1-2 (Fig. 4) is put into the newly aligned sequence 3 and the gap of the alignment 2-3 is propagated to the previously aligned sequence 1. This leads to a multiple alignment of 3 sequences.

Then take the pairwise alignment of sequence 3 and 4. Insert all the gaps
from sequence 3 of the alignment 3-4 (Fig. 5), which are not already present in sequence 3 of the multiple alignment 1-2-3 into all previously aligned sequences, i.e. sequences 1 to 3. Insert all the gaps from sequence 3 of the multiple alignment 1-2-3, which were present before the previous step into sequence 4. Add this newly aligned sequence 4 to the multiple alignment 1-2-3, which leads to a multiple alignment with 4 sequences.

4.2 Probabilistic Algorithm

The probabilistic algorithm does not compute the alignment with only sequences, but with probabilistic ancestral sequences (PAS). A probabilistic ancestral sequence is a sequence of probability vectors instead of amino acids. It consists of a matrix of $20 \times n$, where $n$ is the length of the alignment. The PASs are built according to a tree with the $n$ sequences as leaves. For each node of the tree, the corresponding PAS is computed. [3]

Consider a node with two leaves as children (Fig. 6). Each leaf is determined by an amino acid sequence. Each amino acid pair (amino acids at the
Figure 6: Node with two leaves as children

same position within the sequences) forms a column of the PAS of node $U$. Assume that the two amino acids $E$ and $F$ of the leaves are the first amino acids of the sequences. Hence the first column of the PAS at node $U$, $PAS_U$, equals the probability of any of the 20 amino acids denoted by $u$ changing to $E$ over distance $d_1$ times the probability of $u$ changing to $F$ over distance $d_2$. Thus

$$PAS_U[u][1] = Pr\{u \rightarrow E, d_1\} \times Pr\{u \rightarrow F, d_2\} = (M^{d_1})_{E,u} \times (M^{d_2})_{F,u}$$

where $M$ denotes the 1-PAM matrix. $PAS_U[u][1]$ is one element of the first column, where $u$ is one amino acid out of 20.

Now calculate $PAS_U[u][2]$, the second column of the PAS, by considering the second amino acids of the sequences at the leaves.

Figure 7: Node with one leaf and one internal node as children

The computation of the PAS of a node with one leaf and one internal node as children works similar (Fig. 7). The probability of amino acid $u$ changing to amino acid $v$ over distance $d_2$ is summed over all possible amino
acids $v$. Therefore

$$PAS_{V[u]} = Pr\{u \rightarrow E, d1\} \times \sum_v (Pr\{u \rightarrow v, d2\} \times PAS_{V[v]}) = (M^{d1})_{Eu} \times \sum_v ((M^{d2})_{vu} \times PAS_{V[v]})$$

for all amino acids $u$.

The PAS of a node with two internal nodes as children (Fig. 8) is computed as follows:

$$PAS_{V[u]} = \sum_v (Pr\{u \rightarrow v, d1\} \times PAS_{V[v]}) \times \sum_w (Pr\{u \rightarrow w, d2\} \times PAS_{W[w]}) = \sum_v ((M^{d1})_{vu} \times PAS_{V[v]}) \times \sum_w ((M^{d2})_{wu} \times PAS_{W[w]}) .$$

In this way, the PASs for all of the nodes of the tree are calculated bottom up. The multiple alignment is then done by starting at the root of the tree and traversing top down.

![Figure 8: Node with two internal nodes as children](image)

4.3 The new implementation

Both circular tours and the probabilistic algorithm are already implemented in Darwin by the function `MAAlign`. The construction of the optimal pairwise library is newly implemented by myself. This improved all against all matrix of pairwise alignments is then handed over to `MAAlign` to perform the multiple sequence alignment either by circular tours or the probabilistic algorithm. These improved methods are referred to as the improved proba-
bilistic method (IPM) and the improved circular tours method (ICT) in this document (Fig. 9).

![Diagram](image)

**Figure 9:** The new implementation
5 Analysis

This new implementation of MSA (IPM and ICT) is compared to other alignment programs, such as PRRP, SAGA, DIALIGN, PIMA and HMMT. A comparison is performed between MALign (MALign (probabilistic method): MAP and MALign (circular tours): MAC) and IPM and ICT. Finally, IPM and ICT are compared to ClustalW combined with the improved library described in chapter 3.

BaliBase provides a mechanism to perform these comparisons. BaliBase consists of an alignment database with over 100 reference alignments, which are divided into five hierarchical reference sets.

Reference 1, 2 and 3 are subdivided into three groups: Short (< 100 residues), Medium (200 – 300 residues) and Long (> 400 residues). [4]

5.1 BaliBase

BaliBase is a database of manually-refined multiple sequence alignments specifically designed for the evaluation and comparison of multiple sequence alignment programs. The alignments are categorized by sequence length, similarity, and by the presence of internal insertions and extensions at either end. [4]

BaliBase can be downloaded at
http://www-igbmc.u-strasbg.fr/BioInfo/BAliBASE

5.1.1 Reference 1: equidistant sequences

Reference 1 supplies equidistant divergent sequences of similar lengths, where the percent residue identity of any two sequences lies within a specified range and no large extensions or insertions have been introduced. Depending on the percent identity, the alignments are divided into three subgroups: < 25% identity, 20 – 40% identity and > 35% identity. Reference 1 studies the effect of sequence length and percent identity on the quality of the alignments.
5.1.2 Reference 2: family with orphan sequences

Reference 2 contains alignments of a family (closely related sequences with > 25% identity). Additionally, it contains up to three orphan sequences, which are distant members of the family with < 20% identity, but sharing a common fold. The stability of the family alignment when orphans are introduced into the sequence set is analyzed, as well as the quality of the alignment of the orphan sequences. It also examines the degree of disruption, which is produced by the introduction of orphans to the alignment of the family.

5.1.3 Reference 3: families of related sequences

Reference 3 studies the correct alignment of equidistant divergent families. It can be compared to reference 1, which aligns small numbers of equidistant divergent sequences. The reference alignments consist of up to four families, with < 25% identity of any two sequences from different families.

5.1.4 Reference 4: extensions at either end

Sequences of unequal lengths are examined in reference 4. Sequences with large extensions at either end are introduced to investigate whether the programs are capable of aligning the core blocks flanking the extensions. No large insertions are introduced here.

5.1.5 Reference 5: internal insertions

Reference 5 also contains sequences of unequal lengths, but the insertions are internal and not at the beginning or the end of the sequences.

5.1.6 Alignment scores

BaliBase uses two different scores to estimate the quality of an alignment compared to the BaliBase reference. The sum-of-pairs score (SP) determines the extent of the correctly aligned sequences in an alignment. Therefore, it increases with the number of sequences correctly aligned. The total-column score (TC) tests the ability of the program to align all the sequences correctly.
The SP score is used for both reference 1 and reference 2 since it determines primarily the quality of the alignment within the families. Reference 3, 4 and 5 use the TC score to estimate the quality of the alignment between the families.

5.2 MAlign with / without improvements

Two comparisons are made in this section. On the one hand, the probabilistic algorithm is compared to circular tours and on the other hand, the new implementations IPM and ICT are compared to MAP and MAC (MAlign with either the probabilistic algorithm or circular tours).

5.2.1 Reference 1: equidistant sequences

Reference 1 studies the effect of sequence length and percent identity on the quality of the alignments. Consider Fig. 10, where the y-axis denotes the SP score and the x-axis the percent identity between the sequences. It can be said that the quality decreases with decreasing identity. The greatest difference in the scores is observed in category < 25\%. The higher the percent identity between the sequences, the smaller the difference between the various programs gets.

It can be seen in Fig. 10 and Fig. 11 that the alignment score of circular tours is ranked higher than the probabilistic method in most cases. It can also be seen that IPM and ICT show higher scores than MAP and MAC.

Improvement decreases with increasing identity, which was to be expected, because the alignments of the sequences of category > 35\% are already near optimal for the programs MAP and MAC.
Figure 10: Comparison with MAalign, reference 1: < 25%, 20 – 40% and > 35%

Figure 11: Comparison with MAalign, reference 1: the whole sequence set
5.2.2 Reference 2: family with orphan sequences

Reference 2 tests the ability of the different multiple alignment programs to align divergent orphan sequences (10 – 20% identity of the orphans and the family) with a family of highly related sequences (> 25% identity). In addition, it examines the degree of disruption, which is produced by the introduction of orphans to the alignment of the family. First of all, the families are aligned with no orphans to provide a reference for comparison. The alignments are then constructed with one, two and three orphans. It results in the SP scores showing no significant reduction for the family alignments, when orphans are introduced. Only in a small number of cases, the presence of orphans leads to a loss of alignment quality, where the loss is greater for small families than for large ones.

The ability of the programs to correctly align an orphan sequence is also affected by the presence of other orphans in the sequence set. The quality of the alignment either increases or decreases in an unknown way.

![Graph](image)

Figure 12: Comparison with MAalign, reference 2: short, medium, long

The alignment scores of long sequences in Fig. 12 is ranked higher than of short sequences. Mostly, the core blocks, which define those regions that
can be reliably aligned, are aligned better in long sequences than in short sequences, which can explain this phenomenon.

![Graph showing comparison with MAalign](image)

Figure 13: Comparison with MAalign, reference 2: the whole sequence set

In contrast to reference 1, the probabilistic method (IPM and MAP) is ranked higher as circular tours (ICT and MAC) in this test (Fig. 13). Thus, the probabilistic method is more appropriate for aligning orphan sequences with a family sequence than circular tours.
5.2.3 Reference 3: families of related sequences

Reference 3 aligns small numbers of equidistant divergent families.

Fig. 14 does not show a higher TC score of ICT compared to MAC. This could result from the fact that the circular order of ICT is different from the one of MAC in some cases. The circular tour is calculated with the distances between the pairwise alignments, which may not be the same for ICT and MAC.

![Bar chart](image)

Figure 14: Comparison with MAalign, reference 3
5.2.4 Reference 4: extensions at either end

Sequences with large extensions at either end are examined in reference 4. As shown in Fig. 15, the probabilistic algorithm ranks higher than circular tours.

![Bar chart comparison with MAalign](image)

Figure 15: Comparison with MAalign, reference 4
5.2.5 **Reference 5: internal insertions**

Reference 5 contains sequences with internal insertions. Fig. 16 shows a high score of ICT, which significantly ranks higher than MAC. The difference between IPM and MAP is much smaller.

![Graph showing comparison between MAP, IPM, MAC, and ICT scores](image)

*Figure 16: Comparison with MALign, reference 5*

5.2.6 **The whole sequence set of BaliBase**

Considering the alignment scores of all the sequences of the BaliBase reference database (Fig. 17), ICT ranks slightly higher than IPM. Circular tours allows more improvements than the probabilistic method. This results from IPM taking only into account a small number of the pairwise alignments from the extended library. Only when building the profile of a node with two leaves as children, the pairwise alignment is taken from the extended library. The profile of a node with one leaf and one internal node as children is completely recalculated using the already calculated profile of the internal node and the amino acid sequence of the leaf. Therefore, no information is taken from the extended library in this case calculating the profile. On the
contrary, information is taken from the extended library in each step of the circular tours algorithm and thus more improvements are possible.

![Bar graph showing comparison](image)

Figure 17: Comparison with MAlign, the whole sequence set

### 5.3 Comparison with other alignment programs

IPM and ICT are now compared to other multiple sequence alignment programs, such as PRRP, SAGA, DIALIGN, PIMA and HMMT. These programs are divided into local and global alignment algorithms as well as into progressive and iterative ones, as shown in Fig. 18.

Global alignment programs attempt to align the sequences over their whole length, whereas local alignment programs try to search for the most conserved motifs to align first. Progressive alignment programs build up the multiple alignment step by step by aligning the closest sequences first and successively adding the more distant ones.

PRRP [7] optimizes a progressive, global alignment by iteratively dividing the sequences into two groups, which are subsequently realigned.

DIALIGN [5] is a local approach and constructs multiple alignments by comparing segment to segment rather than residue to residue. The segments
Figure 18: An overview of the different multiple alignment algorithms

are built to a multiple alignment using an iterative procedure.

The genetic algorithm SAGA [6] selects the alignment, which optimizes the COFFEE Objective Function (OF) [11]. The OF is a measure of the quality of the multiple sequence alignment. It evaluates the consistency between the multiple alignment and the library of ClustalW pairwise alignments.

PIMA [9] uses a local dynamic programming algorithm to align the most conserved motifs first. The order of alignment is determined by a sequential branching algorithm.

HMMT [10] maximizes the probability that a hidden Markov model (HMM) represents the sequences to be aligned.

5.3.1 Reference 1: equidistant sequences

Reference 1 studies the effect of sequence length and percent identity on the quality of the alignments. It supplies three groups of alignments depending on the percent identity of the sequences. The lower the percent identity the worse the quality of the alignment gets. The alignments of category < 25%, as can be seen in Fig. 19, present a significantly lower score than the alignments of category 20 – 40% and category > 35%. The higher the percent identity the smaller the difference of the scores between the different
alignment programs gets. IPM and ICT show quite the same scores in category > 35% than the other programs, whereas in category < 25% they rank significantly lower.

![Comparison of various alignment programs](image)

Figure 19: Comparison of various alignment programs, reference 1: < 25%, 20 – 40%, > 35%

Fig. 20 shows the scores of all the sequences of reference 1. The global methods PRRP and SAGA rank higher than the local methods DIALIGN and PIMA. HMMT produced the lowest score, whereas the score of IPM and ICT was intermediate.
5.3.2 Reference 2: family with orphan sequences

Reference 2 aligns a family with up to three orphan sequences. The SP score of all programs is shown in Fig. 21. The global alignment programs (PRRP, SAGA and HMM) perform better than the local ones (DIALIGN and PIMA) in this test. SAGA ranks above PRRP now, in contrast to reference 1. Both IPM and ICT outperform all the other alignment programs, which means that they provide very accurate and reliable alignments and they still perform stable when orphans are introduced into families.

The comparison of the alignment of one orphan against small and large families (four and 14-22 sequences) indicates a significant improvement of the alignment of large families opposed to the one of small families (Fig. 22).
Figure 21: Comparison of various alignment programs, reference 2: the whole sequence set

Figure 22: The alignment of small and large families when orphans are introduced
5.3.3 Reference 3: families of related sequences

Reference 3 aligns small numbers of equidistant divergent families of sequences. The alignment programs are compared using the TC score as the scoring function, since it is a good estimator for the quality of the alignments between families. As seen in Fig. 23, the iterative strategies PRRP and SAGA perform better in this test than the progressive alignment methods. IPM and ICT rank significantly below most of the other alignment programs, except HMMT, which ranks at the same low level.

![Figure 23: Comparison of various alignment programs, reference 3](image)

Reference 3 is now compared to reference 1, which aligns individual equidistant sequences. In order to perform this comparison, a new set of reference 1 alignments is constructed by selecting one sequence from each family in the reference 3 alignments. Fig. 24 shows the difference of their scores \( \text{score}_{ref3} - \text{score}_{ref1} \). Most of the alignment programs align families more successfully than only sequences except PIMA.
5.3.4 **Reference 4: extensions at either end**

Reference 4 aligns sequences with large extensions at either end. In this test, an inversion of the previous ranking of the alignment programs is observed. DIALIGN and PIMA, which implement a local strategy, now outperform the global methods PRRP and SAGA, as well as the new global implementations IPM and ICT (Fig. 25). The global algorithms often fail to locate the flanking core blocks, which results in a total misalignment of these sequences.
5.3.5 Reference 5: internal insertions

Reference 5 aligns also sequences of unequal length, but in contrast to reference 4, the insertions are internal and not at the beginning or the end of the sequences. The local program DIALIGN remains the top ranking program as in reference 4, but PIMA ranks lower than the global programs PRRP and SAGA this time. Only the two global programs IPM and ICT based on MAKalign indicate an even lower TC score (Fig. 26).
5.4 ClustalW with improvements

The improvements of the pairwise library are used so far together with the multiple sequence alignment program MALign either with circular tours or the probabilistic method. This improved pairwise library is now used together with ClustalW, a global alignment program, and is compared to IPM and ICT. Based on the progressive alignment method, ClustalW uses the Neighbor-Joining algorithm to construct a guide tree. Additionally, it supplies sequence weighting, position-specific gap penalties and a residue comparison matrix depending on the degree of identity of the sequences.

ClustalW does not use all the information from the extended library, similar to the probabilistic algorithm. It only uses some of the improved pairwise alignments. Nonetheless, as seen in Fig. 27, the improved ClustalW ranks higher than both IPM and ICT.

ClustalW first builds a tree with all the sequences as leaves. It then calculates the multiple sequence alignment bottom up using the pairwise alignment of the extended library only when the current node has two leaves as children. The improved ClustalW can still be ameliorated by optimizing
Figure 27: Comparison of IPM, ICT and ClustalW

the underlying guide tree.
6 Discussion and Conclusions

It is shown in this document that all the considered programs are capable of correctly aligning on average 86% of the residues in an alignment of sequences with > 20% identity. Below 20%, a significant loss of alignment quality occurs, where the programs correctly align only 47% of the residues on average. The limit at 20% constitutes a real barrier for all the programs. The produced alignments are often unreliable and large dispersions of the scores occur.

Programs using iterative strategies (PRRP, SAGA and DIALIGN) perform very successful in these tests. But the introduction of orphan sequences sometimes disturbs the iteration process and forces the iteration away from the correct alignment. In this case, the new implementations IPM and ICT outperform the other programs and remain stable while calculating the alignment. Only the iterative program HMMT does not reach the score of the others.

Global alignment methods (PRRP and SAGA, as well as IPM and ICT) produce the most accurate alignments in the tests involving equidistant sequences, families of sequences and the alignment of orphan sequences with a family. When large extensions and internal insertions are introduced, the local program DIALIGN is the most successful program at locating the flanking core blocks. But the alignment outside the conserved motifs remains unreliable as well.

The tests have also shown that the new implementations IPM and ICT perform up to 34% better than MAC and MAP. The probabilistic method improves the most when MAP already ranks high, as well as circular tours ameliorates the most when MAC ranks higher than MAP.

ClustalW executed with an improved library of pairwise alignments produces a more reliable multiple alignment than IPM and ICT. But it can even be improved by optimizing the order of the sequences to be aligned, which is determined by a guide tree.

The choice of an alignment program depends a lot on the set of sequences to be aligned. There does not exist a single best program, which produces an accurate and reliable alignment in all categories. The performance of the
programs is affected by the sequence length, the percent identity between the sequences and by the extensions and internal insertions.

The probabilistic algorithm does not use all the pairwise alignments of the extended library. Only when calculating the profile of a node with two leaves as children, the optimal pairwise alignment of the extended library is incorporated. In all other cases, the profile of a node is either computed using two profiles (the node has two internal nodes as children) or one profile and one sequence (the node has one internal node and one leaf as children). Thus, the improvements of the pairwise library are much less considered in calculating the multiple alignment than desired. Integrating the associated profiles in the extended library of pairwise alignments might be a possibility to increase the improvement for the probabilistic algorithm. But this remains subject to further research.

The accuracy and reliability of the programs decreases with decreasing identity between the sequences. It is therefore essential to improve the methods aligning sequences of $< 20\%$ identity.

The alignments may even be improved using gap heuristics after the multiple alignment to change the locations of the gaps. The alignment is more accurate and reliable if there exist gap blocks instead of only small gaps, which are scattered among the alignment. The gap heuristics of MAli gn did not change the alignment produced by IPM and ICT, which results from the improvements computed in IPM and ICT. These improvements already shift the gaps together to larger gap blocks and therefore more accurate and reliable alignments are produced.
References


