Structural bioinformatics

Extracting multiple structural alignments from pairwise alignments: a comparison of a rigorous and a heuristic approach

Erik Sandelin
Stockholm Bioinformatics Center, AlbaNova, Stockholm University, 106 91 Stockholm, Sweden

Received on July 14, 2004; revised and accepted on October 20, 2004
Advance Access publication November 5, 2004

ABSTRACT
Motivation: Multiple structural alignments (MSTAs) provide position-specific information on the sequence variability allowed by protein folds. This information can be exploited to better understand the evolution of proteins and the physical chemistry of polypeptide folding. Most MSTA methods rely on a pre-computed library of pairwise alignments. This library will in general contain conflicting residue equivalences not all of which can be realized in the final MSTA. Hence to build a consistent MSTA, these methods have to select a conflict-free subset of equivalences.

Results: Using a dataset with 327 families from SCOP 1.63 we compare the ability of two different methods to select an optimal conflict-free subset of equivalences. One is an implementation of Reinert et al.’s integer linear programming formulation (ILP) of the maximum weight trace problem (Reinert et al., 1997, Proc. 1st Ann. Int. Conf. Comput. Mol. Biol. (RECOMB-97), ACM Press, New York). This ILP formulation is a rigorous approach but its complexity is difficult to predict. The other method is T-Coffee (Notredame et al., 2000) which uses a heuristic enhancement of the equivalence weights which allow it to use the speed and simplicity of the progressive alignment approach while still incorporating information of all alignments in each step of building the MSTA. We find that although the ILP formulation consistently selects a more optimal set of conflict-free equivalences, the differences are small and the quality of the resulting MSTAs are essentially the same for both methods. Given its speed and predictable complexity, our results show that T-Coffee is an attractive alternative for producing high-quality MSTAs.

Availability: The software for Resolver, our implementation of Reinert et al.’s ILP formulation, and the dataset used in this study are available at http://www.sbc.su.se/~erik/resolver

Contact: erik@sbc.su.se

1 INTRODUCTION
Multiple structural alignments (MSTAs) provide position-specific information on the sequence variability allowed by a protein fold. This information can be exploited to better understand the evolution and function of proteins as well as the physical chemistry of polypeptide folding (Holm and Sander, 1996; Sillitoe and Orengo, 2003). Although a few approaches exist which consider all structures simultaneously when building the MSTAs (Leibowitz et al., 2001; Dror et al., 2003; Shatsky et al., 2004), most methods for creating MSTAs rely on a pre-computed library of pairwise alignments. These methods differ in the way they incorporate the information from the pairwise alignments: Several are based on the progressive approach, where a similarity-tree, based on the initial pairwise alignments, is followed when building the MSTA (Šali and Blundell, 1989; Russell and Barton, 1992; Ding et al., 1994); In Orengo and Taylor’s (1996) approach a consensus structure is built iteratively by, at each step, calculating new alignments between the consensus and the remaining structures and then merging the most similar structure into the consensus. Guda et al. (2001) use the pairwise alignments to select a master structure which the other structures are aligned to resulting in an initial MSTA. This MSTA is subsequently refined by Monte Carlo optimization. Recently, Ochagavia and Wodak (2004) have proposed a method which extracts three-protein alignments from the set of pairwise alignments and then progressively expands these to include additional proteins and more spatially equivalent residues.

A different view of the problem of building and MSTA from a set of pairwise alignments is to acknowledge that the pairwise alignments will in general contain conflicting residue equivalences not all of which can be realized in the final MSTA. In this view the task of building a consistent MSTA is translated into the task of selecting the most optimal conflict-free subset of equivalences. This view is indifferent to whether the equivalences originate from sequence or structural alignments and indeed this approach has been explored for building multiple sequence alignments: By representing the pairwise alignments as a graph, Kecskeméti (1993) could formulate the problem of finding the set of optimal equivalences as that of finding the maximum weight trace (MWT) in the alignment graph. This allowed him to develop a branch-and-bound algorithm which due to its memory-intensive nature was able to solve only small problem sizes. Later, Reinert et al. (1997) found that the MWT problem could be formulated as an integer linear program (ILP) which can be solved using less memory-intensive methods.

In this paper we implement Reinert et al.’s ILP formulation of the MWT problem to construct MSTAs. We call our implementation Resolver. The main goal of this paper is to compare its ability to select the most optimal conflict-free subset of equivalences with that of a heuristic and faster method, T-Coffee (Notredame et al., 2000). T-Coffee is of the progressive type but uses a heuristic enhancement of the original weights of the equivalences in order to incorporate information of all alignments while still retaining the speed and simplicity of the progressive approach. Although originally developed for creating multiple sequence alignments it can be fed with a pre-computed library of residue equivalences. Hence by feeding T-Coffee...
Extracting multiple structural alignments from pairwise alignments

Fig. 1. Overview of Resolver: 1. All-against-all pairwise alignments are performed for the four proteins in this example. 2. Residue equivalences are extracted and translated into a graph. 3. The graph is separated into connected components. 4. Non-overlapping components with no conflicting equivalences (components III and IV) represent columns in the final MSTA. 5. Overlapping components are merged (I and II). 6. Components containing conflicting equivalences are processed with the method described in Section 2.1.1. The thick solid edges illustrate a mixed cycle. By removing the edge (Q,K) between the second and third protein all mixed cycles in this component are removed. The resulting connected components can be incorporated into the final MSTA.

with the same set of weighted equivalences as Resolver, we can directly compare the performance of the two methods.

The comparison is carried out on a large and diverse test set consisting of 327 families from SCOP 1.63 (Murzin et al., 1995). For the 251 families for which Resolver successfully creates a MSTA our comparison shows that Resolver consistently selects a more optimal set of equivalences, irrespective of whether optimality is judged by the original weights of the equivalences or the enhanced weights used by T-Coffee’s heuristics. However, we also find that although the methods’ MSTAs have statistically significant differences, these differences are small and if T-Coffee’s MSTAs are pruned to contain only original equivalences the quality of the MSTAs of the two methods is essentially the same.

2 METHODS

2.1 Resolver

Resolver constructs a MSTA from a library of weighted residue equivalences obtained from pairwise structural alignments. Some of these equivalences will not be consistent with each other, and to select the most optimal subset of consistent equivalences, Resolver implements the ILP formulation of the maximum weight trace problem by Reinert et al. (1997). Fig. 1 shows an overview of the method, and a detailed description is given below:

1. Start with a library of weighted residue equivalences.
2. Construct a graph where each node is a residue and each edge represents a residue equivalence.
3. Use the depth-first-search algorithm (Cormen et al., 2001) to separate the graph into connected components.
4. Merge overlapping components. For a component \( A \) call the maximum/minimum residue number of sequence \( S_i \) in \( A \), \( \max(A,i)/\min(A,i) \). Then two components \( A \) and \( B \) are said to not overlap if either \( \max(A,i) < \min(B,i) \) or \( \min(A,i) > \max(B,i) \). Otherwise they are said to overlap and if so the columns corresponding to these two components cannot be ordered and hence they contain conflicting equivalences which must be resolved. Therefore they are merged into one component and subsequently subjected to the procedure described in Section 2.1.1.
5. Components containing a maximum of one member from each sequence are free from ordering conflicts and will form one column in the final MSTA.
6. Components with two or more members from any of the sequences contain ordering conflicts which need to be resolved (note that by definition this also includes the merged components described in point 4). This is done by subjecting each component to the procedure described in Section 2.1.1.
7. After removal of conflicting equivalences each component corresponds to one column in the final MSTA. The MSTA is created from the components by ordering the columns with a depth-first-search algorithm.

For a pair of structures in the final MSTA their alignment will differ from the original pairwise alignment in that removed equivalences have introduced gaps and in that induced equivalences might be present. An induced equivalence is obtained when two residues which are not aligned in the original pairwise alignment end up in the same column in the final MSTA because they are linked through other equivalences. To superimpose the structures we first extract all equivalences (original and induced) from the MSTA created by Resolver. Then we minimize the sum of the squared distances of these equivalenced residues with each structure represented by its four Euler parameters (Goldstein et al., 2002). The minimization is carried out by using the conjugate gradient method as implemented by Press et al. (1992) and with a convergence tolerance of \( 10^{-8} \).
2.1.1 Resolving equivalence conflicts Once we have identified the components with conflicting equivalences we need to resolve these conflicts. This is done by using Reinert et al.’s ILP formulation of the MWT problem (Reinert et al., 1997).

In this formulation one starts with $M$ sequences, $S_i = \{s_{i1}, s_{i2}, \ldots \}$, $i = 1, \ldots, M$, where $s_{ij}$ is residue $j$ in sequence $i$. For the general case, $S$ are the full sequences whereas in our case $S_i$ is the set of residues from sequence $i$ found in the given connected component. This set is ordered such that $s_{ij}$ always have a larger residue number in the original sequence than $s_{i,j+1}$.

Then we introduce a mixed graph $G = (V, E, H)$ where $V$ is a set of vertices (the sequences), $E$ is a set of undirected edges (the equivalences) and $H$ is a set of directed edges. The set of directed edges $H$ are needed to maintain the sequential ordering of the residues within each sequence.

A directed edge starts in a vertex corresponding to residue $s_{ij}$ and ends in the vertex corresponding to residue $s_{i,j+1}$. Ordering conflicts corresponds to the presence of mixed cycles in the graph (Reinert et al., 1997), i.e., cycles involving at least one directed edge. This is easy to see: An ordering conflict arises when equivalences connect two residues from the same sequence as both residues cannot be present in the same column in the final MST. This means there is a path between these two residues and when we add the directed edges this path turns into a (mixed) cycle. An example of a mixed cycle can be seen in Fig. 1.

For every edge $e \in E$ we define a binary variable $x_e \in \{0, 1\}$, which indicates whether $e$ is realized or not, and a weight $w_e$. The weights, $\{w_e\}$, are assigned the values specified in the initial library of weighted equivalences. Our goal of selecting the optimal set of equivalences which contains no ordering conflicts can then be formulated as an ILP:

$$\text{maximize } \sum_{e \in E} w_e x_e$$

subject to

$$\sum_{e \in C} x_e \leq |C| - 1, \quad \forall C$$

where $|C|$ is the set of mixed cycles and $|C|$ is the number of undirected edges in cycle C.

In general, there are exponentially many mixed cycles in a graph. Therefore following Reinert et al., we consider only those mixed cycles found for each sequence $i$ by calculating the shortest path between residues $s_{i,j+1}$ and $s_{i,j}$. If such a shortest path exists it corresponds to a mixed cycle (as $s_{i,j}$ and $s_{i,j+1}$ are connected through a directed edge) and is added to our list of mixed cycles, i.e. $|C|$ in Equation (2). The corresponding ILP is solved and we repeat this step for the solution found. If this solution contains no mixed cycles we are done. Otherwise, the new set of mixed cycles is added to the old set and the extended problem is solved. This is repeated until no more mixed cycles are found.

In this study we use version 4.0 of the publicly available software package lp_solve (ftp://ftp.isp.cs.technion.ac.il/pub/lp_solve) to solve the ILP. Solving an ILP is an NP-hard problem (Cormen et al., 2001), i.e. no polynomial-time algorithm exists for this problem. However, if the integer requirement on the variables is relaxed, a solution to the corresponding linear program (LP) can be found in polynomial time using the simplex method (Cormen et al., 2001). Therefore, lp_solve solves the ILP in two steps. In the first step the integer requirement is relaxed and the corresponding LP is solved using the simplex method. If the solution to the LP has $N$ variables with non-integer values, lp_solve continues to systematically search the $2^N$ possible solutions for the optimal solution. For this search lp_solve uses a branch-and-bound search tree (Ausiello et al., 1999) which reduces the computational cost compared to an exhaustive enumeration of the $2^N$ solutions. However, this step is still NP-hard and it might require an exponential number of steps to find the optimal solution.

2.2 T-Coffee

T-Coffee (Notredame et al., 2000) is a multiple sequence alignment strategy which tries to avoid the most serious pitfalls of the progressive alignment strategy. By first examining the pairwise alignments it enhances the weight of residue pairs whose alignment is consistent with the other alignments. This allows it to use the speed and simplicity of the progressive strategy while still incorporating global information in each step of building the alignment. More specifically, the weights are enhanced by looking for transitivity triplets. Three residues, A, B and C, from three different sequences, are said to form a transitivity triplet if, in the pairwise alignments, A is aligned to B, B to C, and C to A. With this approach it obtains a dramatic improvement in accuracy with little decrease in speed (Notredame et al., 2000).

The T-Coffee software has the option of feeding it with a pre-computed library of weighted residue equivalences. In this case, T-Coffee first enhances the weights of the library as described above and then performs the progressive alignment using the position-specific substitution matrix defined by this enhanced library. As suggested in T-Coffee’s user manual this feature can be utilized to calculate MSTAs by feeding it with a library of weighted equivalences obtained from pairwise structural alignments. Indeed, this feature has been used by several studies to create MSTAs (Dietmann et al., 2001; Ochagavia and Wodak, 2004; O’Sullivan et al., 2004). However, these studies do not make use of the option to assign individual weights to the equivalences, i.e. in these studies all equivalences have the same weight when entering the enhancement procedure described above. This means that conflicting equivalences will only be discriminated by their degree of consistency with the other equivalences (i.e. the result of the enhancement procedure). In contrast, in this study the equivalences in the pre-computed library will be assigned individual weights (obtained from the pairwise alignments) as this allows for further discrimination of the conflicting equivalences according to how ‘good’ or ‘bad’ they are in the original pairwise alignment. Furthermore, using individual weights allows us to directly compare Resolver’s and T-Coffee’s ability to select an optimal set of conflict-free equivalences since we can feed both methods with exactly the same library of weighted equivalences.

We use version 1.37 of T-Coffee and run it with default parameters except for ‘clean_aln’ and ‘cosmetic_penalty’ which are both set to zero. The former has been recommended by the authors of T-Coffee when using structural alignments as input (Ochagavia and Wodak, 2004). cosmetic_penalty is a parameter which regulates to what extent T-Coffee makes the unalignable portions of the alignment more pleasing to the eye. Here we set the value of this parameter to zero because then the objective functions optimized by T-Coffee and Resolver will be most similar, although not identical, (T-Coffee optimizes the sum of the enhanced weights rather than the original weights.) However, even with a zero value of this parameter, T-Coffee sometimes aligns unalignable residues. Hence we prune the final alignment such that only original equivalences (and the equivalences induced by these) remain.

2.3 Building the library of weighted equivalences

To build a MSTA for M proteins both Resolver and T-Coffee start from a library of weighted residue equivalences obtained from pairwise structural alignments. In this study, to calculate the pairwise structural alignments we use Structal (Subbiah et al., 1993). This method utilizes iterative dynamic programming to find the alignment. Starting from a random orientation of the two structures a similarity matrix $W$ is calculated. Each element, $w_{ij}$, in this matrix represents a pair of equivalenced residues, $i$ and $j$, and depends only on the distance $d_{ij}$ between the two residues:

$$w_{ij} = \frac{20.0}{1 + (d_{ij}/2.24)^2} \quad (3)$$

Applying dynamic programming on this matrix gives an alignment and a set of equivalences. The two structures are least-square fitted onto each other using these equivalences and a new similarity matrix is calculated. Dynamic programming is applied again and the whole process is repeated until convergence.

To build the library of weighted equivalences we first use Structal to calculate alignments for the $M(M-1)/2$ pairs of proteins and then from Structal’s
output we extract the residue equivalences, \( \varepsilon \), and the distances, \( d_i \). Using these distances we assign the weight \( w_e \) [Equation (3)] to each equivalence.

3 EVALUATION

The evaluation of the performance of MSTA methods can be divided into two parts: how well a particular method optimizes its objective function and how high the quality of the produced MSTAs are. Hence for the first part, in this study we look at both the sum of the weights of the equivalences [the objective function optimized in Resolver, Equation (1)], and the sum of the enhanced weights of the equivalences (the objective function optimized in T-Coffee), which we denote with \( W \) and \( W_c \) respectively. The enhanced weight of a particular equivalence is simply the original weight plus the weights of the equivalences in the transitivity triplets that this equivalence participates in (Notredame et al., 2000). Of interest in this optimization part is also \( f_{\text{equiv}} \), the fraction of the set of original equivalences that are realized in the final MSTA. The closer to 1 \( f_{\text{equiv}} \) is, the better the MSTA matches the original pairwise alignments.

Evaluating the quality of the MSTAs is a more delicate task as even for pairwise alignments defining a reliable measure of alignment quality has proven to be a difficult task (Eidhammer et al., 2000; Koehl, 2001). Hence, in this study we will mostly look at observables of the MSTAs with the hope of revealing systematic differences between the two methods.

In what follows we define the observables which we will study: \( N_{\text{equiv}} \) is the total number of equivalences in the final MSTA, both original and induced equivalences, \( N_{\text{orig}} \) is the number of original equivalences, whereas \( N_{\text{induce}} \) is the number of induced equivalences, i.e. equivalences not present in the original set of pairwise alignments. \( N_{\text{col}} \) and \( N_{\text{cand}} \) are the number of aligned columns, i.e. columns with at least one pair of aligned residues, and the number of core columns, i.e. columns with all proteins aligned. \( N_{\text{gap}} \) denotes the average number of gap openings in a MSTA, where the average is over pairs of structures in the MSTA. For a pair of structures, a gap opening is every instance where one residue is aligned but not the previous one. Gap openings are counted over both structures.

Then we look at geometric measures: \( cRMS \) denotes the root mean square distance between aligned pairs of \( C_a \)-atoms (in angstroms), where the mean is taken over all the equivalences in the MSTA. \( cRMS \) is clearly not a good measure of alignment quality as lower \( cRMS \) can always be achieved by aligning fewer residues. Therefore, with the caveat above in mind, we also look at the structural alignment score, SAS, (Subbiah et al., 1993; Levitt and Gerstein, 1998), which is an attempt to quantify alignment quality by balancing \( cRMS \) against alignment length. For a pairwise alignment, SAS is defined as

\[
\text{SAS} = \frac{100 \times cRMS}{N_{\text{equiv}}},
\]

and here we generalize SAS to multiple alignments by redefining SAS as

\[
\text{SAS} = \frac{100 \times cRMS}{N_{\text{equiv}}} \times M(M - 1)/2,
\]

where \( M \) is the number of structures in the alignment.

To quantify the differences between Resolver and T-Coffee, for an observable \( O \) we define the normalized difference \( \Delta_O \) as

\[
\Delta_O = \left( \text{Resolver} - \text{T-Coffee} \right) / \text{Resolver}.
\]

Furthermore to judge the significance of the differences, for each observable, we also calculate the Wilcoxon \( P \)-value which gives the probability that the observed distribution of \( \Delta_O \) has a median value of zero.

4 RESULTS

4.1 Resolver

For this study we want to select a large and diverse dataset. To this end, we select domains from SCOP 1.63 with a maximum of 30% sequence identity. With Resolver, we then try to create a MSTA for each of the 350 families belonging to structural classes a, b, c or d and which have three or more members.

An upper limit of 1 h was set for calculating all the pairwise alignments and producing a consistent MSTA. The experiments were carried out on a PC-cluster with Xeon Intel 1.4 GHz nodes. For 11 families Structural did not find any significant pairwise alignments and for 12 families only one significant alignment was found and these 23 families were thus excluded from the analysis. Resolver finished successfully for 251 of the remaining 327 families, while for 76 families Resolver failed to resolve the equivalence conflicts within the time limit. These 76 families ranged in size from 4 to 39 members with more than half of them, 40, having 10 or more members (see Fig. 2a). As noted by Reinert et al. the complexity of their ILP formulation is largely dictated by the size of the largest connected component in the alignment graph (Reinert et al., 1997). Indeed, if we look at the families for which Resolver failed, for all 76 families the largest connected component contained 10% or more of all the equivalences, and for more than half of the families, 41, it contained more than 80% of the equivalences (see Fig. 2b). This should be compared with the ideal case, where the set of original equivalences contains no ordering conflicts, and where each connected component should roughly contain a fraction of 1/\( N_{\text{col}} \) equivalences, where \( N_{\text{col}} \) is the number of aligned columns in the final MSTA. Furthermore, as is evident from Fig. 2 for the set of families for which Resolver fails, the distributions of family-size and size of largest connected component is markedly different from those of the successful families. In that respect Resolver’s problem with these families is not surprising.

For the remaining 251 families Resolver successfully created MSTAs using a total of 3.0 CPU hours. With one family, c.67.1.3, requiring 2612 s and 236 requiring less than 2 min and 224 less than 30 s. Although c.67.1.3 only contains 7 members and 6670 equivalences the three largest connected components contain 26, 16 and 10% of all equivalences respectively.

4.2 Resolver versus T-Coffee

To compare the performance of T-Coffee with Resolver we applied it to the same 251 families for which Resolver successfully created a MSTA. For each family we fed T-Coffee with exactly the same library of weighted equivalences as we used for Resolve. The MSTAs were subsequently pruned as described in Section 2.2 to obtain the final MSTA. Based on this MSTA the structures were then superimposed in the same way as for Resolver (see Section 2.1). The heuristics of T-Coffee allow the method to process the 251 families in only 1 min 37 s on an Intel Xeon 2.8 GHz processor. First we compare the abilities of the two methods to select the most optimal set of equivalences. When making this comparison one should keep in mind that technically T-Coffee and Resolver optimize different objective functions: T-coffee optimizes the sum of the enhanced weights, \( W_c \), whereas Resolver optimizes the sum of the original weights \( W \). However, in practice they are quite similar as T-Coffee’s enhancement of the weights plays a similar role as the mixed cycle constraints in Resolver’s ILP [Equation (1)]. In a set of conflicting
equivalences, the equivalences to be removed are mainly determined by how many mixed cycles (Resolver) or how few transitivity triplets (T-Coffee) an equivalence participates in. The weights, original or enhanced, only play a role when conflicting equivalences are equal in this respect. When this happens ranking the equivalences according to the enhanced weights, rather than the original weights, should only matter for those cases where the equivalences have very different surroundings (i.e. the transitivity triplets they participate in).

For 57 families T-Coffee and Resolver select an identical set of equivalences while for all remaining 194 families, Resolver selects a more optimal set of equivalences in terms of $W$. For T-Coffee’s objective function, there is 1 family with identical $W$ despite different sets of equivalences, 22 for which T-Coffee finds a more optimal set and 171 families for which Resolver’s set of equivalences is more optimal. From Fig. 3a we also see that the optimal set of equivalences selected by Resolver sometimes contains fewer original equivalences than T-Coffee’s set. There are 13 such cases and only 5 of them correspond to cases where T-Coffee’s solution has a higher $W$, which shows that the most optimal set of equivalences is not always the set that realizes most original equivalences.

Figure 3a also shows that the fraction of the original equivalences which can be realized in the final MSTA varies between 100 and 68%. One family for which only 68% of the equivalences can be realized is d.49.1.1. This family has three members, d1e8oa and d1e8od, and the three pairwise alignments between these structures are frustrated in the sense that they are mutually inconsistent: Structural aligns almost all (72 out of 74) residues between d1e8oa and d1e8od whereas d1e8oa is aligned to the first half of d1914 and d1e8oa is aligned to the second half of d1914. Hence only equivalences from two of the alignments can be realized in the final alignment. In this case it turns out that the alignment between d1e8oa and d1e8od is significantly worse than the others and the corresponding equivalences are removed when creating the MSTA.

Although Resolver is able to select a more optimal set of equivalences than T-Coffee, as can be seen in the left half of Table 1, Resolver’s $W$ and $W_{tc}$ are only on average 0.5 and 0.2% larger than T-Coffee’s. This indicates that the differences between the methods’ sets of equivalences are small. To see what influence these differences have, we next compare the observables of the methods’ MSTAs. Figure 3b–d shows a comparison of $N_{gap}$, $cRMS$ and SAS for the two methods’ MSTAs. From this figure it appears that the quality of the MSTAs is essentially the same for the two methods. However, when looking at $\Delta_{O}$ the Wilcoxon P-values reveals that there are small but significant differences in $N_{col}$, $N_{gap}$ and $cRMS$, as shown by Table 1. These differences are explained by the fact that Resolver’s MSTAs in most cases retain more of the original equivalences (as indicated by $N_{orig}$ in Table 1) and consequently should have more aligned columns which also means that the $cRMS$ has to increase. Furthermore, as the removal of an original equivalence means that a gap is introduced between the corresponding pair of proteins, it is no surprise that T-Coffee’s MSTAs have more gaps. Finally, Table 1 also reveals that there is no significant difference in SAS for the two methods, indicating that SAS is not sensitive enough to judge if the additional equivalences included in Resolver’s MSTAs (as compared with T-Coffee) increases or decreases the overall quality of the MSTAs.

Figure 3d shows that the dataset contains an outlier, i.e. a family for which Resolver performs significantly worse (in terms of $cRMS$) than T-Coffee. This family is a.47.1.2 which has four members: d1dn1b, d1fioa, d1hs7a and d1lvfa. d1dn1b and d1fioa consist of four helices whereas d1hs7a and d1lvfa consist of four helices whereas d1hs7a and d1lvfa consist of three helices. Just as for d.49.1.1 the pairwise alignments are frustrated: d1dn1b, d1fioa and d1hs7a align consistently with the helices of d1hs7a aligned to the first three of d1dn1b and d1fioa. d1lvfa’s alignments causes the frustration as its helices are all aligned to the helices of d1hs7a and the first three of d1dn1b, but for d1fioa the alignment is shifted one helix so that it aligns to the last three of d1fioa. As this alignment is highly significant Resolver decides to keep the alignment between d1lvfa’s and d1fioa’s last helices. However, the first helix of d1lvfa still remains aligned to the first helices of d1dn1b.
Extracting multiple structural alignments from pairwise alignments

Fig. 3. Comparison of MSTAs produced by T-Coffee and Resolver for the 251 SCOP families. Shown is (a) \( f_{\text{orig}} \), fraction of the original equivalences that are realized in the MSTA, (b) \( N_{\text{gap}} \), average number of gap openings, (c) \( cRMS \), root mean square deviation of the \( C_\alpha \) atoms of the equivalenced residues and (d) SAS, structural alignment score.

Table 1. Comparison of observables for Resolver’s and T-Coffee’s MSTAs

<table>
<thead>
<tr>
<th>Observable</th>
<th>Normal families ( \langle \Delta O \rangle )</th>
<th>Problematic families ( \langle \Delta O \rangle )</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>( W )</td>
<td>0.0051 ± 0.0005 ( 2.0 \times 10^{-33} )</td>
<td>0.021 ± 0.003 ( 9.8 \times 10^{-4} )</td>
<td>0.0005</td>
</tr>
<tr>
<td>( W_{\text{tc}} )</td>
<td>0.0022 ± 0.0004 ( 9.9 \times 10^{-22} )</td>
<td>0.009 ± 0.003 ( 0.005 )</td>
<td>0.0006</td>
</tr>
<tr>
<td>( N_{\text{eq}} )</td>
<td>0.0042 ± 0.0006 ( 5.4 \times 10^{-13} )</td>
<td>0.009 ± 0.004 ( 0.007 )</td>
<td>0.0004</td>
</tr>
<tr>
<td>( N_{\text{equiv}} )</td>
<td>0.0049 ± 0.0009 ( 1.05 \times 10^{-7} )</td>
<td>0.006 ± 0.006 ( 0.36 )</td>
<td>0.0006</td>
</tr>
<tr>
<td>( N_{\text{induce}} )</td>
<td>0.010 ± 0.005 ( 0.08 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( N_{\text{cut}} )</td>
<td>0.0054 ± 0.0009 ( 3.6 \times 10^{-9} )</td>
<td>0.019 ± 0.009 ( 0.08 )</td>
<td></td>
</tr>
<tr>
<td>( N_{\text{core}} )</td>
<td>0.002 ± 0.002 ( 0.09 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( N_{\text{gap}} )</td>
<td>(-0.039 ± 0.009 ) ( 2.5 \times 10^{-6} )</td>
<td>(-0.02 ± 0.05 ) (</td>
<td>1.0 )</td>
</tr>
<tr>
<td>SAS</td>
<td>0.007 ± 0.004 ( 0.12 )</td>
<td>(-0.04 ± 0.01 ) ( 0.02 )</td>
<td></td>
</tr>
<tr>
<td>( cRMS )</td>
<td>0.012 ± 0.004 ( 3.3 \times 10^{-4} )</td>
<td>(-0.03 ± 0.01 ) ( 0.05 )</td>
<td></td>
</tr>
</tbody>
</table>

Shown is the average normalized difference, \( \langle \Delta O \rangle \), for a number of observables. Also shown are Wilcoxon P-values which give the probability that the observed distributions of \( \Delta O \) have median values of zero. The left half shows data for the 251 families which Resolver successfully finished within the 1 h time limit, while the right half shows data for the additional 11 families completed when the time limit was extended to 20 h.

and d1hs7a resulting in d1lvfa being poorly superimposed on the other three structures. For T-Coffee, the equivalences in the alignment between d1lvfa and d1fioa gets the lowest rank, since they do not form any transitivity triplets, and none of them are realized in the MSTA. However, Resolver’s set of equivalences still have higher \( W \) and \( W_{\text{tc}} \) than T-Coffee’s, so this outlier is a result of the way the optimization problem is formulated rather than the failure of Resolver.

Finally, although T-Coffee easily produces MSTAs for the 76 families which Resolver previously failed on, these families could still be problematic for T-Coffee in that the quality of the produced MSTAs are lower than for the 251 previously studied families. Therefore, we applied the two methods to the 76 problematic families, this time extending the time-limit for Resolver to 20 h. Resolver now produced MSTAs for 11 of these families while it still failed for 65 families. Detailed inspection of these 65 families revealed that Resolver failed because the largest connected component gave rise to an ILP which the ILP-solver, lp_solve, stalled on, failing to produce a solution. Sampling a few of these ILPs further revealed that lp_solve stalls during the branch-and-bound phase. As described in Section 2.1.1
this step is NP-hard and might require an exponential number of steps if the ILP is too complex. Obviously, the complexity of these ILPs is beyond this limit and consequently for the ILP formulation to be applicable to these difficult families, the exact branch-and-bound procedure needs to be replaced by an approximate procedure. However, developing such a procedure is beyond the scope of this study.

For the 11 families which Resolver successfully finished we performed the same comparison with T-Coffee as we previously did for the 251 easier families. These 11 families range in size from four to nine proteins and the results of this comparison are shown in the right half of Table 1. In terms of W and Wc, Resolver’s solutions are more optimal than T-Coffee’s in 11 and 10 cases, respectively. Furthermore, T-Coffee’s solutions are now further away from optimality as is seen by the $\langle \Delta O \rangle$; for W and Wc which are four times larger than for the easier families. Furthermore, the differences of the selected sets of equivalences are now large enough to have an impact on the quality of the MSTAs: Resolver’s MSTAs now both have lower SAS and cRMS than T-Coffee’s. However, although the $\langle \Delta O \rangle$; for SAS and cRMS differ from zero by several standard deviations, the P-values are fairly high, indicating that 11 families are not enough to conclude if these observations are statistically significant.

5 DISCUSSION

In this paper we have addressed the problem of extracting a MSTA from a set of pairwise structural alignments. We have compared the abilities of two different methods to extract the most optimal conflict-free set of residue equivalences from the pairwise alignments. The methods compared are Resolver, an implementation of Reinert et al’s rigorous ILP formulation of the maximum weight trace problem, and the fast but heuristic T-Coffee. They were applied on a test set consisting of 327 families from SCOP 1.63 with a maximum of 30% sequence identity. Setting an upper time-limit of 1 h, Resolver finished successfully for 251 of these families and for the remaining 76 problematic families failure could be correlated to the largest connected component in the alignment graph containing a substantial fraction of the equivalences. Comparing the methods on the 251 families for which both methods produced a MSTA, we found that Resolver indeed finds more optimal sets of equivalences than T-Coffee, whether optimality is judged by Resolver’s objective function or T-Coffee’s objective function. Furthermore, we also showed that when these equivalences are assembled into a MSTA, Resolver’s alignments differ in a statistically significant way from T-Coffee’s. However, the differences are small and when the MSTAs are evaluated with SAS, an alignment quality measure which balances cRMS against the number of aligned residues, the differences disappear. Still, the small fraction of equivalences that differ between the methods could have an impact when the MSTAs are used in a direct application, such as for example protein structure prediction, but such a comparison is beyond the scope of this paper. Finally, reapplying Resolver on the 76 problematic families with an extended time-limit of 20 h, we found that Resolver could produce MSTAs for 11 of these families, while the remaining 65 were found to be inherently difficult for Resolver: the alignment graphs of these families gave rise to ILPs which the ILP-solver, Ip_solve, stalled on, without producing any solutions. For the 11 problematic families which Resolver finished we found that T-Coffee’s solutions were further away from optimality than for the easier families. Furthermore, for these families we now found an observable difference between the quality of the methods’ MSTAs with Resolver’s having lower cRMS and SAS than T-Coffee’s. However, we also found that 11 families were too few to conclude that this observation was statistically significant.

In summary, this study has shown that the heuristics of T-Coffee indeed are powerful: Given a set of pairwise alignments it selects a near-optimal set of conflict-free equivalences, which, combined with the method’s speed and predictable complexity, makes it an attractive choice for producing high-quality MSTAs.

This finding contrasts with the poor performance of T-Coffee observed by Ochagavía and Wodak (2004). However, the settings of their study are too different from ours to offer a reliable explanation for our disparate findings. In their study, T-Coffee was applied on only three but challenging examples which possibly represent difficulties not probed by our dataset. Furthermore, a different method was used to calculate the pairwise alignments and they did not assign individual weights to the equivalences fed to T-Coffee. Consequently, it would be of considerable interest to reapply T-Coffee to their examples with settings identical to our study (or vice versa) as that would provide valuable clues to the strengths and limitations of T-Coffee. Indeed, we have such work in progress.

Finally, in this paper we have not addressed the shortcomings of MSTA methods based on pairwise alignments. As pointed out by Shatsky et al., there are situations where a pairwise approach will fail to detect a pattern common to a set of proteins but where an approach considering all proteins simultaneously will succeed (Shatsky et al., 2004). The problematic family, d.47.1.2, encountered here (see Section 4.2) highlights this problem as the most optimal solution turned out to superimpose the structures poorly. A more systematic study of this issue would be of considerable interest. In particular, using a method which considers all structures simultaneously to generate pairwise libraries would facilitate a direct comparison between this method and pairwise methods.

ACKNOWLEDGEMENTS

E.S. acknowledges support from the Knut and Alice Wallenberg foundation. The author wishes to thank Olivia Eriksson for valuable discussions and the Biox Dell Supercluster for access to their computer resources. This work was in part supported by NIH grant GM-63817.

REFERENCES


