An iterative refinement algorithm for consistency based multiple structural alignment methods

Yu Chen and Gordon M. Crippen

Received on May 12, 2006; revised on June 21, 2006; accepted on June 23, 2006
Advance Access publication June 29, 2006

Abstract
Motivation: Multiple STructural Alignment (MSTA) provides valuable information for solving problems such as fold recognition. The consistency-based approach tries to find conflict-free subsets of alignments from a pre-computed all-to-all Pairwise Alignment Library (PAL). If large proportions of conflicts exist in the library, consistency can be hard to get. On the other hand, multiple structural superposition has been used in many MSTA methods to refine alignments. However, multiple structural superposition is dependent on alignments, and a superposition generated based on erroneous alignments is not guaranteed to be the optimal superposition. Correcting errors after making errors is not as good as avoiding errors from the beginning. Hence it is important to refine the pairwise library to reduce the number of conflicts before any consistency-based assembly.

Results: We present an algorithm, Iterative Refinement of Induced Structural alignment (IRIS), to refine the PAL. A new measurement for the consistency of a library is also proposed. Experiments show that our algorithm can greatly improve T-COFFEE performance for assembling 15 transglycosidases. Results on three other benchmarks showed that the algorithm consistently improves multiple alignment performance.

Availability: The C++ code of the algorithm is available upon request.

Contact: gcrippen@umich.edu

1 INTRODUCTION

MSTA can provide structural conservation information, which can help to solve biological problems such as fold recognition (Shi et al., 2001; Kelley et al., 2000) and reconstructing phylogenies (O’Donoghue and Luthey-Schulten, 2005). Up to now, many MSTA algorithms have been developed (Russell and Barton, 1992; Lupyan et al., 2005; Ye and Godzik, 2005; Ebert and Brutlag, 2006; Dror et al., 2003; Shatsky et al., 2002; Ochagavia and Wodak, 2004; Sandelin, 2005; Yang and Honig, 2000; Guda et al., 2001) and several multiple alignment databases have been built (Casbon and Saqi, 2005; Guda et al., 2006; Bhaduri et al., 2004).

MSTA is a difficult task because the search space is large and it grows exponentially as the number of structures to be aligned increases. It is also known that more than one biologically meaningful solution may exist for a pairwise structural alignment problem (Kolodny and Linial, 2004). Obviously, it is not possible to exhaustively search all possible combinations of all alternative pairwise alignments. Therefore, heuristics must be used to extract MSTA based on consistent pairwise alignments.

Although it is possible to align multiple structures simultaneously (Dror et al., 2003; Shatsky et al., 2002), most MSTA algorithms simplified the MSTA problem to the assembly of a pre-computed all-to-all PAL. One simple way to find a MSTA is to choose a pivot (master) structure and align all other structures (slaves) to it based on pairwise alignments (Akutsu and Sim, 1999; Levitt and Gerstein, 1998). If there is any error or inconsistency in the master–slave alignments, the final multiple structural superposition and alignment might be erroneous. CE-MC (Guda et al., 2001) uses a subsequent Monte Carlo step to increase the number of aligned columns and correct errors. Another widely used technique is the progressive approach (Yang and Honig, 2000; Ye and Godzik, 2005; Lupyan et al., 2005; Russell and Barton, 1992). Given a PAL, a binary guide tree can be built based on pairwise structural similarity scores. Then a MSTA can be generated by following the tree from leaves to the root. At each node in the tree, two sets of structures are joined based on either the closest pairwise alignment (Ye and Godzik, 2005; Yang and Honig, 2000) or a dynamic programming procedure (Lupyan et al., 2005). The aggregated MSTA from the progressive approach may contain errors owing to conflicts in the PAL. An error introduced in an early stage will be held in later MSTAs. Many progressive type MSTA algorithms have a further refinement step afterward to correct those errors. Usually in that step, MSTA and multiple superposition are refined iteratively (Lupyan et al., 2005; Russell and Barton, 1992). In PrISM, a round-robin dynamic programming was used where each structure was iteratively dropped from the MSTA and realigned to the MSTA based on interresidue distances in the current multiple superposition (Yang and Honig, 2000). In MAMMOTH-multi, the multiple structural superposition was refined each time after merging two branches, and the refined superposition was further used to correct alignment errors (Lupyan et al., 2005).

Correcting errors afterwards is not as good as avoiding errors in the first place. A wrong decision made in early stages may lead to a suboptimal final MSTA in a local minimum which cannot be refined later. Each alignment in the PAL is treated as a constraint with a weight associated with it. The consistency-based approach tries to minimize errors (or conflicts) by maintaining constraints unchanged as much as possible across a PAL. There are two different types of algorithms to derive consistent MSTAs, either progressively or globally.

T-COFFEE (Notredame et al., 2000) is a most widely used progressive type algorithm, which applies ‘triplet library extensions’
to ensure consistency. Given a pair of aligned residues, the algorithm searches the remaining sequences in the library to find any residue aligning to both residues (called a ‘triplet’), and the weight for a pair of aligned residues is determined by the sum of all triplets involving that residue pair. More consistent pairs will get higher weights and will more likely be aligned. Therefore alignment errors, especially errors occurring in early steps, are supposed to be reduced in each merging event in the progressive alignment. Although T-COFFEE was developed initially for multiple sequence alignment problems, it has been used to assemble pairwise structural libraries to generate MSTAs (3DCOFFEE: O’Sullivan et al., 2004). S4 (Casbon and Saqi, 2005), a MSTA database, was constructed via T-COFFEE assembly of PALs derived by SAP (Taylor and Orengo, 1989). The triplet idea can also be applied to structure information directly. MALECÔN searches for triplet structures that can be superimposed with the fewest conflicts and progressively adds more structures while maintaining the consistency (Ochagavia and Wodak, 2004).

Alternatively, consistency can be acquired by manipulating constraints globally by graph algorithms. A PAL can be interpreted as an alignment graph, where vertices represent residues in each protein and weighted edges represent pairwise alignments. Then graph problems like maximal weight trace (Sandelin, 2005) or graph clustering (Ebert and Brutlag, 2006) can be used to resolve conflicts and get consistency.

One problem for these consistency-based assembler programs is that the final performance still depends greatly on the quality of the initial PALs. If there is a large proportion of errors or inconsistencies in the PAL, it will be hard to differentiate between true signals and noise.

CBA tried to resolve conflicts in the initial PAL by realigning all pairwise structural alignments after progressively superimposing all structures (Ebert and Brutlag, 2006). Since the superposition was based on assembling initial pairwise alignments, the progressive approach may not result in the optimal conflict-free superposition and, therefore, it is not guaranteed to generate the best MSTA.

In this paper, we present a new iterative approach to solve the conflict problem. Instead of assembling data containing conflicts and trying to remove conflicts from an assembled MSTA, we try to make pairwise alignments as consistent as possible before the assembly step. We define a measure of the consistency of a PAL based on Equations (1) and (2). Let $RMSD(i, j)$ be the root mean squared deviation (RMSD) between structure $i$ and $j$ based on corresponding pairwise alignment after superimposing both structures to structure $p$. Given a master structure $p$, the average of all pairwise RMSDs in the corresponding master-slave superposition (aRMSD$_p$) can be used to represent the level of conflicts involving that structure [Equation (1)], and the mean of all possible aRMSD$_p$ (mRMSD) indicates the overall consistency in a PAL.

$$aRMSD_p = \frac{2}{(n-1)n} \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n-1} RMSD(i, j)$$

$$mRMSD = \frac{1}{n} \sum_{p=1}^{n} (aRMSD_p)$$

Ideally, if there is no conflict between master-slave alignments and slave-slave alignments, a structural pairwise alignment between two slaves can be induced from two corresponding master–slave alignments and low mRMSD will be expected. If conflicts exist, we need to refine pairwise alignments based on induced alignments. An induced alignment is defined based on the principle of transitivity: given a master structure $P$ and two pairwise alignments ($A, P$) and ($B, P$), the induced alignment ($A, B$) contains all residue pairs (from $A$ and $B$) that have common aligned residues in $P$. Since $P$ is the master structure and ($A, P$) and ($B, P$) were used to superimpose structures $A$ and $B$ onto $P$, residue pairs in the induced alignment ($A, B$) should be close in 3D space since they are supposed to be superimposed onto the same residue. Substitution of the original pairwise alignment ($A, B$) with the induced alignment ($A, B'$) will resolve conflicts among the triplet {A, P, B}. To avoid information loss during the alignment induction, an extra dynamic programming step was used to refine the induced alignment when necessary. By iteratively choosing the master structure, conflicts should be removed gradually and mRMSD of the PAL will be optimized until there is no further improvement.

2 METHODS AND MATERIALS

2.1 Algorithm

Construction of the PAL. All-to-all pairwise alignments are generated using CE (Shindyalov and Bourne, 1998) and results are transformed into T-COFFEE library format (Notredame et al., 2000). In both the original PALs and refined PALs, only one alignment is stored per structural pair. Therefore, one-to-one correspondence is guaranteed.

Refinement. IRIS is a deterministic iterative refinement algorithm consisting of four steps.

1. Master–slave superposition. A master structure is selected in a modified round-robin style: before each round, all structures are ordered in terms of aRMSD$_p$, and iterations always start from the structure with the fewest conflicts and end at the structure with the most conflicts. After each iteration, structures are sorted again. Multiple structural superposition is derived by superimposing every slave structure onto the master structure $P$. The superposition is done by Kearsley’s method (Kearsley, 1990).

2. Derivation of induced alignments. RMSDs of all pairwise alignments are calculated based on the current multiple structural superposition. If the RMSD between two structures $A$ and $B$ is larger than a cutoff (δ), then the triplet {A, P, B} is not consistent and needs to be refined. An induced alignment ($A, B'$) is derived based on pairwise alignments ($A, P$) and ($P, B$). If residue $A_i$ from $A$ and residue $B_j$ from $B$ are aligned to the same residue $P_k$ from $P$, the two residues $A_i$ and $B_j$ are aligned in the induced alignment.

3. Refinement of induced alignments. Structures $A$ and $B$ may share some common features that do not exist in the master structure $P$. Therefore if the original induced alignment is too short (<95% of the current pairwise alignment length), it is further extended by a dynamic programming routine similar to that used in CE (Shindyalov and Bourne, 1998). The similarity matrix $S$ is calculated between structures $A$ and $B$ [Equation (3)] $d_{\alpha \beta}$ is the distance between residue $i$ in structure $A$ and residue $j$ in structure $B$ after induced alignment based pairwise superposition. Therefore, pairs of aligned residues in an induced alignment get a higher score, and are more likely to be aligned. If the dynamic programming step still cannot produce an alignment long enough, the induced alignment is not used. Otherwise, the corresponding pairwise alignment is replaced by the induced alignment in the PAL.

$$S_{\alpha, \beta} = \alpha - d_{\alpha, \beta}$$
(4) Evaluation of consistency. After each iteration, the consistency of the updated PAL is evaluated based on mRMSD (Equation (2)). If the consistency has been improved, the current PAL is stored.

The program runs iteratively until no master–slave superposition can further improve the consistency of the current PAL. There are four adjustable parameters in IRIS: 8, α, open gap penalty and extension gap penalty. In this implementation, we used 4.0, 10.0, 5.0, 0.5, respectively. It is also possible to make α in Equation (3) a variable to favor residue pairs in similar structural environments (Chen and Crippen, 2005). However, since the current objective is to purely optimize 3D superposition based on coordinate information, we stick to CE-based PAL to make fair comparisons.

Detection of structural core. T-COFFEE v3.92 was used to assemble PAL into a column-wise MSTA. High weights were imposed on (structurally) aligned pairs (((100/n(n–1))/2, where n is the number of structures) to guarantee that those pairs remain aligned. Other weighting schema (O’Sullivan et al., 2004; Casbon and Saqi, 2005) have also been tested (see Results). The fully aligned columns from MSTA constitute the structural core (called the ‘MSTA core’).

Alternatively, structural cores are defined directly from the PAL. The alignment number Naln for a residue in a structure is defined as the number of pairwise alignments involving that residue in the PAL, and the maximum possible Naln is N – 1 for a PAL containing N structures (owing to the one-to-one correspondence). We selected the structure containing the most residues with Naln = N – 1 as the master structure and a ‘PAL core’ was derived based on master–slave pairwise alignments.

2.2 Performance criteria

It is difficult to compare alternative superpositions with different associated raw RMSD as well as different number of equivalences. Kolodny et al. reported several pairwise measurements to simultaneously consider both factors (Kolodny et al., 2005). One simple measurement is structural alignment score (SAS):

\[
\text{SAS}_{ij} = \frac{\text{RMSD}_{ij} \times 100}{N_{\text{mat}}},
\]

where \(N_{\text{mat}}\) is the number of equivalences in the pairwise alignment. In order to make an easy comparison to other methods, we used the core RMSD (cRMSD), the size of the structural core (nCore) and the multiple SAS (mSAS) to evaluate the performance. cRMSD is defined as the average of all pairwise RMSDs between all residues involved in the structural core [Equation (5)]. cRMSD is dependent on 3D superpositions. For convenience, nCore, \(N_{\text{mat}}\) is calculated based on the pairwise superposition of core residues in structures i and j.

\[
c\text{RMSD} = \frac{2}{n(n-1)} \sum_{i} \sum_{j} \text{RMSD}_{ij}
\]

mSAS is defined as a scaled cRMSD similar to SAS in Equation (4)

\[
m\text{SAS} = \frac{c\text{RMSD} \times 100}{n\text{Core}}
\]

2.3 Materials

Four benchmarks were used to evaluate the performance. The first benchmark is a subset of Lindahl’s fold recognition benchmark (Lindahl and Elofsson, 2000). All 15 TIM \(\alpha/\beta\) barrel domains belonging to the transglycosidase superfamily (Table 1) were selected with PDB files downloaded from author’s website. Three other benchmarks (globins, jelly rolls and OB folds) were obtained from MALECON (Ochagavia and Wodak, 2004) with PDB files downloaded from PDB website and domain parsed based on CATH 2.5.

CE was obtained from http://cl.sdsc.edu and T-COFFEE was obtained from http://igs-server.cnrs-mrs.fr. We compared the performance with several MSTA algorithms. POSA and MAMMOTH-multi were tested on their online servers. Other algorithms, including CBA, MASS, MultiProt and CE-MC, were downloaded from the corresponding websites and executed locally.

3 RESULTS

3.1 Transglycosidase benchmark (TIM barrel fold)

A total of 15 transglycosidases were used to benchmark the performance of MSTA algorithms. This benchmark is difficult because all 15 domains share very low sequence identity (<30%) with each other, and the chain lengths vary a lot (Table 1). In fact, the pairwise alignment of transglycosidases itself is a difficult problem (Williams et al., 2003). As a result, particularly high inconsistencies between pairwise alignments are found in this benchmark. Figure 1 shows a plot of all the 15 alignments (14 induced alignments and 1 pairwise alignment) between a pair of proteins in this benchmark in the original CE based PAL. The proportion of conflicts is so high that we can not find any obvious consensus/consistency among them. The initial mRMSD is 13.70 Å. Based on the original PAL, T-COFFEE can only produce a MSTA containing seven fully aligned columns with cRMSD of 2.99 Å.

In other T-COFFEE based MSTA algorithms, different weighting schema have been reported. 3DCoffee (O’Sullivan et al., 2004) used a constant weight of 100 for all structurally aligned residue pairs, while S4 (Casbon and Saqi, 2005) used a variable weighting scheme in which weights (<1000 and usually ~100) were dependent on how close the two residues being superimposed were in 3D space. These two alternative weighting schema put fewer constraints on structural based alignments. Using the 3DCoffee weighting scheme, T-COFFEE aligned 46 residues with cRMSD of 5.96 Å, while T-COFFEE aligned only 35 residues with cRMSD of 6.4 Å based on the S4 weighting scheme.

The PAL core does not depend on any column-wise MSTA and can be directly calculated based on the PAL. Surprisingly, with the same original PAL, we can detect a PAL core containing 75 aligned residues with cRMSD of 3.54 Å (Fig. 2), which easily outperformed T-COFFEE! This example shows that a consistency-based algorithm can easily fail when there is a high level of conflicts.

| Table 1. Transglycosidase benchmark |
|---|---|---|
| Domain | Description |
| 1amy d1amy_2 | Plant alpha-amylase |
| 1ppi d1ppi_2 | Animal alpha-amylase |
| 2aaa d2aaa_2 | Fungal alpha-amylases |
| 1hya d1hyb__ | beta-Amylase |
| 1bgl d1bgla5 | beta-Galactosidase, domain 3 |
| 1cbg d1cbg__ | Plant beta-glucosidase |
| 1ghr d1ghr__ | Plant beta-glucanases |
| 1cxh d1cdg_4 | Cycloextrim glycosyltransferase |
| 1eeb d1eece__ | Endocellulase E1 |
| 1edg d1edg__ | Endoglucanase CelA |
| 1elt d1elt__ | Endo-beta-N-acetylglucosaminidase |
| 2ebn d2ebn__ | Endo-beta-N-acetylglucosaminidase |
| 1lo d1hvq__ | Hevamine A (chitinase/lysozyme) |
| 1qba d1qba_3 | beta-N-acetylhexosaminidase |
| 1xyza d1xyza__ | Xylanase |
After IRIS, the conflicts in the PAL were greatly reduced (mRMSD = 4.30 Å) and consistent alignment patterns emerged in the alignment plot (Fig. 3). The lower conflict level boosted up the T-COFFEE performance: with the refined PAL, T-COFFEE can align 125 columns with cRMSD of 3.20 Å. Changing the T-COFFEE weighting scheme only slightly changed the result.

By using the 3DCOFFEE weighting scheme, T-COFFEE can align 131 residues with cRMSD of 3.34 Å. The PAL core also expanded to include 141 aligned residues with cRMSD of 3.93 Å (Fig. 4).

The transglycosidase benchmark was also tested on several other MSTA methods. Results are listed in Table 2. CE-MC is not used on this benchmark because CE-MC uses the POM data management system and requires the original PDB files, while the downloaded transglycosidase PDB files are SCOP domain files and only contain ATOM entries. With the aid of IRIS, the performance of structural core detection exceeds many other structure-based MSTA algorithms. As we can see from the table, we can detect a much larger structural core under a cRMSD comparable to or even better than for other algorithms.

3.2 Malecon benchmarks

The globin benchmark contains 15 structures. This benchmark was frequently used to evaluate the performance of an MSTA algorithm (Ye and Godzik, 2005; Lupyan et al., 2005). The original CE-based PAL is very consistent on this benchmark, with mRMSD of 2.74 Å. A typical alignment plot is shown in Figure 5, and we can clearly see a consistent alignment in the plot. T-COFFEE performed very well on this benchmark, aligning 93 residues with cRMSD of 2.18 Å. IRIS refinement only marginally improved the consistency (reduced mRMSD to 2.56 Å). T-COFFEE can align 97 residues with cRMSD of 2.24 Å. The initial PAL core contains 101 residues with cRMSD of 2.64 Å. Interestingly, after refinement, the size of the PAL core reduces to 92 with cRMSD of 2.27 Å.

Performance of other programs on this benchmark is shown in Table 3. As we can see, our CE-IRIS combination worked better than or approximately similar to all other methods, including CE-MC. Lupyan et al., (2005) reported they could align 133 residues with cRMSD of 1.56 Å. But that calculation was based on the 'loose core', which comprises all columns with >66% conservation (personal communication with the author).
The other two MALECON benchmarks, jelly rolls and OB folds, were much less frequently used than the globin benchmark owing to the large amount of variation in the PALs. As we can see, a large number of conflicts exist in both cases (mRMSDs equal to 13.06 and 8.01, respectively). mRMSDs were improved after IRIS refinement, although they were still fairly large (Table 4). Both MALECON and MAMMOTH-multi could not find any structural core in the jelly roll benchmark (Ochagavia and Wodak, 2004; Lupyan et al., 2005).

For a subset of jelly rolls (with six proteins removed), MALECON aligns 24 residues with cRMSD of 1.19 Å (mSAS = 4.95), while CE-IRIS-TCOFFEE aligns 40 residue with cRMSD 2.42 Å (mSAS = 6.05). For OB folds, MALECON can align 20 residues with cRMSD of 1.26 Å (mSAS = 6.3). Although we can at most align 52 residues (CE-IRIS, PAL core), the cRMSD (4.29 Å) is even higher (mSAS = 8.25). In either case, mSAS is high, suggesting large structural variations may exist in the two benchmarks.

### Table 2. Performance of MSTA algorithms on the transglycosidase benchmark

<table>
<thead>
<tr>
<th>Methods</th>
<th>nCore</th>
<th>cRMSD (Å)</th>
<th>mSAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE-IRIS-TCOFFEE (MSTA core)</td>
<td>125</td>
<td>3.20</td>
<td>2.56</td>
</tr>
<tr>
<td>CE-IRIS (PAL core)</td>
<td>141</td>
<td>3.93</td>
<td>2.78</td>
</tr>
<tr>
<td>CBA</td>
<td>73</td>
<td>2.4</td>
<td>3.28</td>
</tr>
<tr>
<td>MAMMOTH-multi</td>
<td>100</td>
<td>4.00</td>
<td>4.00</td>
</tr>
<tr>
<td>POSA (flexible)</td>
<td>58</td>
<td>2.91</td>
<td>5.01</td>
</tr>
<tr>
<td>POSA (rigid)</td>
<td>54</td>
<td>3.02</td>
<td>5.59</td>
</tr>
<tr>
<td>MultiProt</td>
<td>21</td>
<td>1.63b</td>
<td>7.76</td>
</tr>
<tr>
<td>MASSa</td>
<td>17</td>
<td>2.02b</td>
<td>11.88</td>
</tr>
</tbody>
</table>

*More than one alignment was reported and the longest alignment with the lowest RMSD was used for comparison. 
*Not cRMSD but the lowest aRMSD was reported.

The other two MALECON benchmarks, jelly rolls and OB folds, were much less frequently used than the globin benchmark owing to the large amount of variation in the PALs. As we can see, a large number of conflicts exist in both cases (mRMSDs equal to 13.06 and 8.01, respectively). mRMSDs were improved after IRIS refinement, although they were still fairly large (Table 4). Both MALECON and MAMMOTH-multi could not find any structural core in the jelly roll benchmark (Ochagavia and Wodak, 2004; Lupyan et al., 2005). For a subset of jelly rolls (with six proteins removed), MALECON aligns 24 residues with cRMSD of 1.19 Å (mSAS = 4.95), while CE-IRIS-TCOFFEE aligns 40 residue with cRMSD 2.42 Å (mSAS = 6.05). For OB folds, MALECON can align 20 residues with cRMSD of 1.26 Å (mSAS = 6.3). Although we can at most align 52 residues (CE-IRIS, PAL core), the cRMSD (4.29 Å) is even higher (mSAS = 8.25). In either case, mSAS is high, suggesting large structural variations may exist in the two benchmarks.

### Table 3. Performance of MSTA algorithms on the globin benchmark

<table>
<thead>
<tr>
<th>Methods</th>
<th>nCore</th>
<th>cRMSD (Å)</th>
<th>mSAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE-IRIS-TCOFFEE (MSTA core)</td>
<td>97</td>
<td>2.24</td>
<td>2.31</td>
</tr>
<tr>
<td>CE-IRIS (PAL core)</td>
<td>92</td>
<td>2.27</td>
<td>2.46</td>
</tr>
<tr>
<td>STAMPa</td>
<td>55</td>
<td>1.30</td>
<td>2.36</td>
</tr>
<tr>
<td>CE-MCa</td>
<td>95</td>
<td>2.37</td>
<td>2.49</td>
</tr>
<tr>
<td>CBA</td>
<td>83</td>
<td>2.13</td>
<td>2.57</td>
</tr>
<tr>
<td>MAMMOTH-multi</td>
<td>85</td>
<td>2.29</td>
<td>2.69</td>
</tr>
<tr>
<td>MASSa</td>
<td>59</td>
<td>1.73</td>
<td>2.93</td>
</tr>
<tr>
<td>POSAa</td>
<td>71</td>
<td>2.29</td>
<td>3.23</td>
</tr>
<tr>
<td>MultiProt</td>
<td>50</td>
<td>1.55b</td>
<td>3.10</td>
</tr>
</tbody>
</table>

*From Ye and Godzik (2005).
The result is different from the literature (see text).

### 3.3 Running time

The running time of IRIS depends on the inconsistency level of the original PAL, the average chain length, and the number of structures to be aligned. It takes 11 min for IRIS to assemble the 15 transglycosidases and 24 s to assemble the 15 globins on a Pentium IV CPU at 3.0 GHz. The time for refining a PAL is around 50% of the time needed to generate the PAL by CE (29 min for transglycosidases and 1 min for globins).

### 4 DISCUSSION

It might be better to use normalized RMSDs instead of the raw RMSD to measure inconsistency and to compare the quality of multiple alignments. However, since the length of the induced alignment does not change much during the refinement procedure, the normalized RMSD would be approximately proportional to RMSD, and therefore results should not change much.

A summary of comparisons of performance before and after IRIS refinement is shown in Table 4. From the table, we can see that IRIS refinement.
reference always improved alignment consistency (reducing mRMSD) and almost always enlarged the size of the core structure with little change in cRMSD. It is interesting to see that PAL cores have almost the same performance as MSTA cores. In particular, when there are many conflicts before refinement, PAL cores can often provide more information than MSTA cores assembled by T-COFFEE or other MSTA algorithms.

A possible explanation would be that the PAL core is determined by a particular master-slave superposition, while an MSTA core is determined by the overall consistency. When the overall consistency is tractable, consistency based alignment algorithms can perform very well, and a consensus MSTA core can be easily obtained. If there is little consistency in the PAL (as in the transglycosidases example), it is hard for a consistency-based method to determine which alignment should be selected for the final MSTA. The good performance of PAL indicates that the master–slave superposition may be a useful tool for extracting consistencies in such a situation.

T-COFFEE and consistency-based graph algorithms are useful tools to obtain column-wise alignment tables. But as shown in the transglycosidase example, those algorithms are prone to fail when the conflict level is high. This problem has rarely been addressed before, and special attention should be paid to multiple alignment databases using such techniques. For example, in the S4 database a large number of transglycosidases were assembled based on SAP-TCOFFEE protocol. Without proper quality controls, we can hardly trust such MSTA.

5 CONCLUSION
We described a new approach to solve the MSTA problem when a high level of conflicts between pairwise alignments exist. A measurement of conflict levels is proposed. Using our pairwise alignment refinement approach, 15 transglycosidases were aligned and a large structural core consisting of 120–140 residues was detected. Our experiments showed that T-COFFEE may fail when large proportions of conflict exist. Master–slave superposition may be more robust in such circumstances and may be used to refine the alignment library.

ACKNOWLEDGEMENTS
We thank Dr. David States for providing computing sources, and we thank Dr. Notredame for answering questions on T-COFFEE.

Conflict of Interest: none declared.

REFERENCES


