BMC Bioinformatics

BioMed Central

Research article

Iterative refinement of structure-based sequence alignments by Seed Extension

Changhoon Kim, Chin-Hsien Tai and Byungkook Lee*

Address: Laboratory of Molecular Biology, Center for Cancer Research, National Cancer Institute, National Institutes of Health, Bethesda, Maryland 20892, USA

Email: Changhoon Kim - kimchan@mail.nih.gov; Chin-Hsien Tai - taic@mail.nih.gov; Byungkook Lee* - bk@nih.gov

* Corresponding author

Published: 9 July 2009

BMC Bioinformatics 2009, **10**:210 doi:10.1186/1471-2105-10-210

Received: 2 February 2009

Accepted: 9 July 2009

This article is available from: http://www.biomedcentral.com/1471-2105/10/210

Abstract

Background: Accurate sequence alignment is required in many bioinformatics applications but, when sequence similarity is low, it is difficult to obtain accurate alignments based on sequence similarity alone. The accuracy improves when the structures are available, but current structure-based sequence alignment procedures still mis-align substantial numbers of residues. In order to correct such errors, we previously explored the possibility of replacing the residue-based dynamic programming algorithm in structure alignment procedures with the Seed Extension algorithm, which does not use a gap penalty. Here, we describe a new procedure called RSE (Refinement with Seed Extension) that iteratively refines a structure-based sequence alignment.

Results: RSE uses SE (Seed Extension) in its core, which is an algorithm that we reported recently for obtaining a sequence alignment from two superimposed structures. The RSE procedure was evaluated by comparing the correctly aligned fractions of residues before and after the refinement of the structure-based sequence alignments produced by popular programs. CE, DaliLite, FAST, LOCK2, MATRAS, MATT, TM-align, SHEBA and VAST were included in this analysis and the NCBI's CDD root node set was used as the reference alignments. RSE improved the average accuracy of sequence alignments for all programs tested when no shift error was allowed. The amount of improvement varied depending on the program. The average improvements were small for DaliLite and MATRAS but about 5% for CE and VAST. More substantial improvements have been seen in many individual cases. The additional computation times required for the refinements were negligible compared to the times taken by the structure alignment programs.

Conclusion: RSE is a computationally inexpensive way of improving the accuracy of a structure-based sequence alignment. It can be used as a standalone procedure following a regular structure-based sequence alignment or to replace the traditional iterative refinement procedures based on residue-level dynamic programming algorithm in many structure alignment programs.

Background

In searching for protein functions and in building homology models, it is desirable to have accurate sequence motifs and profiles [1-3], which are obtained from sequence alignments of homologous proteins. However, it is often difficult to obtain accurate sequence alignments based on sequence similarity alone when sequence similarity is low.

Therefore, structural alignments, when available, have been used to guide sequence alignments. Such structure-based sequence alignments have been used as the gold standard to evaluate pure sequence alignment methods [4,5] and to derive structural environment-specific substitution matrices which have been shown to be useful for detection of remote homologs and for sequence-structure alignments [6-9].

However, structure-based sequence alignments produced by different programs can be different even when the structures are similar [10,11]. There are a large number of instances wherein all or parts of the structure are shifted by 2 or 4 residues or even by an odd number of residues [12]. Some methods are probably quite good at detecting structural similarity, yet relatively poor in terms of the accuracy of the sequence alignment they produce [12].

DaliLite and VAST use a Monte-Carlo procedure after initial structural alignment [13,14], FATCAT and MATT adopt AFP (aligned fragment pair)-based dynamic programming without constructing initial structural alignments [15,16], and other programs mostly rely on residue-level dynamic programming algorithm according to various scoring schemes with or without initial rigid-body superposition [17-20].

We previously developed the SE (Seed Extension) algorithm which generates a sequence alignment from a superimposed structure pair without changing the superposition [21]. A number of other programs [22-25] also provide a similar function, but these programs use the dynamic programming algorithm and a gap penalty. We have shown that SE, which is not based on the dynamic programming algorithm and does not use a gap penalty, generates a more accurate alignment on average than programs that use a dynamic programming algorithm.

In this study, we report on the development of a fast refinement procedure, which can be used to improve an existing structure-based sequence alignment. The procedure, which we call RSE (Refinement with SE), is an iterative procedure that uses SE in its core. Using CDD (Conserved Domain Database) [26] "root node set" as the reference alignment [12], we show that appending the RSE procedure to a structure-based sequence alignment program improves the accuracy of the alignment for all 9 programs tested.

Results

Improvement of the overall alignment accuracy

In order to see if the RSE procedure improves or degrades alignments produced by different structure comparison programs, we ran the program to be tested with default options to obtain the structure-based sequence alignment for each structure pair. Then the sequence alignment and the corresponding structure pair were fed to the RSE program to obtain a new sequence alignment. We used the fraction of correctly aligned residues with shift error δ, $f_{CAR}(\delta)$, or $F_{CAR}(\delta)$, which is $f_{CAR}(\delta)$ averaged over all structure pairs in a superfamily, as the measure of accuracy of the alignment for each superfamily [12]. Since there were 96 superfamilies in the dataset (Table 1), we took the average over all superfamilies, $<F_{CAR}(\delta)>$ (angle brackets for averaging), as the measure of the overall accuracy of alignments for the whole dataset for a given method.

RSE procedure improved the alignment accuracy, as measured by $<F_{CAR}(0)>$, for all methods (Figure 1). The improvements were small for DaliLite and MATRAS but about 5% for CE and VAST. The alignments from FAST, LOCK2, and TM-align also improved even though these programs were designed to give high quality sequence alignments [17,18,20]. There were more alignments with accuracy gain than those with accuracy loss, except for DaliLite and MATRAS (Figure 2). The increase in the number of correctly aligned residues is large for many alignments, especially for CE, SHEBA, TM-align, and VAST, while a decrease, when happens, is always relatively small in magnitude, except for a few pairs for MATRAS.

The nature of the improvement varied among different methods. For CE, MATT and TM-align, RSE improved $<F_{CAR}(0)>$ but not $<F_{CAR}(8)>$ (Figure 1), which indicates that it is mostly alignment shift error that was reduced by the RSE procedure. For FAST and SHEBA-4, the improvements appear to be mainly correction of under-alignments, presumably by reducing the number of gaps, since $<F_{CAR}(8)>$ increased almost as much as $<F_{CAR}(0)>$ by the refinement.

MATT is a unique method in that it considers the flexibility of structures to improve the sequence alignment quality [16], but its overall accuracy with the root node set was still worse than that of DaliLite and could be noticeably increased by the RSE procedure (Figure 1). The RSE-augmented MATT, FAST, and SHEBA-4 achieved $<F_{CAR}(0)>$ values that were now comparable to that of DaliLite, which is a much slower program (Figure 3).

Table 1: Composition of the CDD root node set

SCOP class	Number of CDs†	Number of structure pairs
all-α (a)	11	326
all-β (b)	15	1721
α/β (c)	35	912
α+β (d)	26	510
others§(o)	9	122
total (t)	96	3591

§ Other than the four major classes.
† The 5 outlier superfamilies were excluded

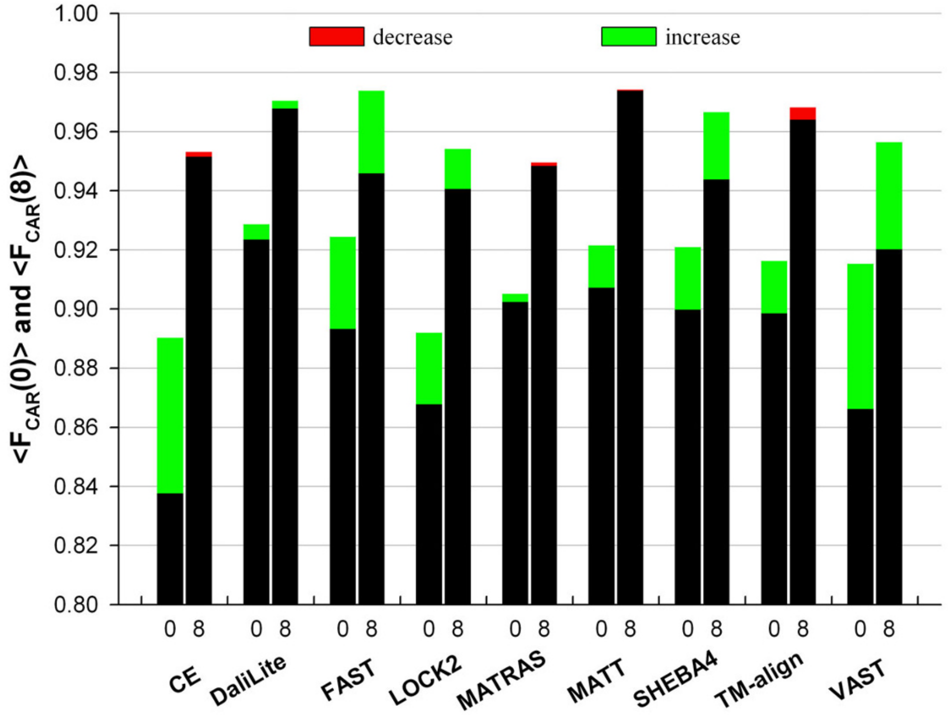


Figure 1
Average improvements of structure-based sequence alignments. The average accuracies of sequence alignment were computed for each method before and after refinement by the RSE procedure. The corresponding method name is given under each pair of bars along the x-axis, where the bars for <F$_{CAR}$(0)> and <F$_{CAR}$(8)> are marked with 0 and 8, respectively. The black portion together with the red tip, when present, represent <F$_{CAR}$(0)> or <F$_{CAR}$(8)> before the refinement. The green and red tips indicate the increment and the decrement, respectively, after the refinement.

Computing time

The times spent by the RSE procedure were nearly negligible compared to the total times spent by the programs to align the structure pairs: RSE took about 46 to 60 milliseconds of wall clock time per alignment on average (Figure 3). In order to measure time complexity for the RSE procedure in terms of CPU times, we focused on the refinement of CDD alignments, since the average wall clock times for all methods were similar (Figure 3). The number of cycles used by RSE to reach the final alignment varied for different structure pairs (Figure 4A) and affected the overall computing time. However, the CPU time per cycle showed linear dependence on the product of the query and target lengths (Figure 4B).

Dependence on structural types

In order to see if the improvement of the alignment depended on protein structural types, the structure pairs were grouped according to their SCOP class (Table 1) and <F$_{CAR}$(0)>s were computed for each class and method (Figure 5). The RSE procedure improved the <F$_{CAR}$(0)> for most classes (the green tips), but there were cases wherein <F$_{CAR}$(0)> decreased by a small amount (the red tips in the case of DaliLite and MATRAS). The <F$_{CAR}$(0)> increases were most prominent for FAST and VAST across all SCOP classes and for CE for the β-sheet containing classes. The alignments also improved for LOCK2, SHEBA-4, MATT and TM-align for all SCOP classes. The <F$_{CAR}$(0)> in "others" class in DaliLite increased to a comparatively large extent, indicating that certain defects in its alignments were effectively corrected.

Refinement of good and not-so-good initial input alignments

When RSE was run on the reference CDD alignment, f$_{CAR}$(0) decreased for 1589 out of 3591 pairs, making the <F$_{CAR}$(0)> value to decrease by approximately 5% (Table 2). Since CDD is being used as the standard, any change in alignment will reduce the <F$_{CAR}$(0)> value. However, the <F$_{CAR}$(0)> value remained higher than that from any structure comparison programs (see Table 2 and Figure 1).

RSE also improved the accuracy of the alignments from the pure sequence alignment program SSEARCH by 19% to about 67% and from the profile-profile alignment program SALIGN by 20% to 75% (Table 2). This shows that

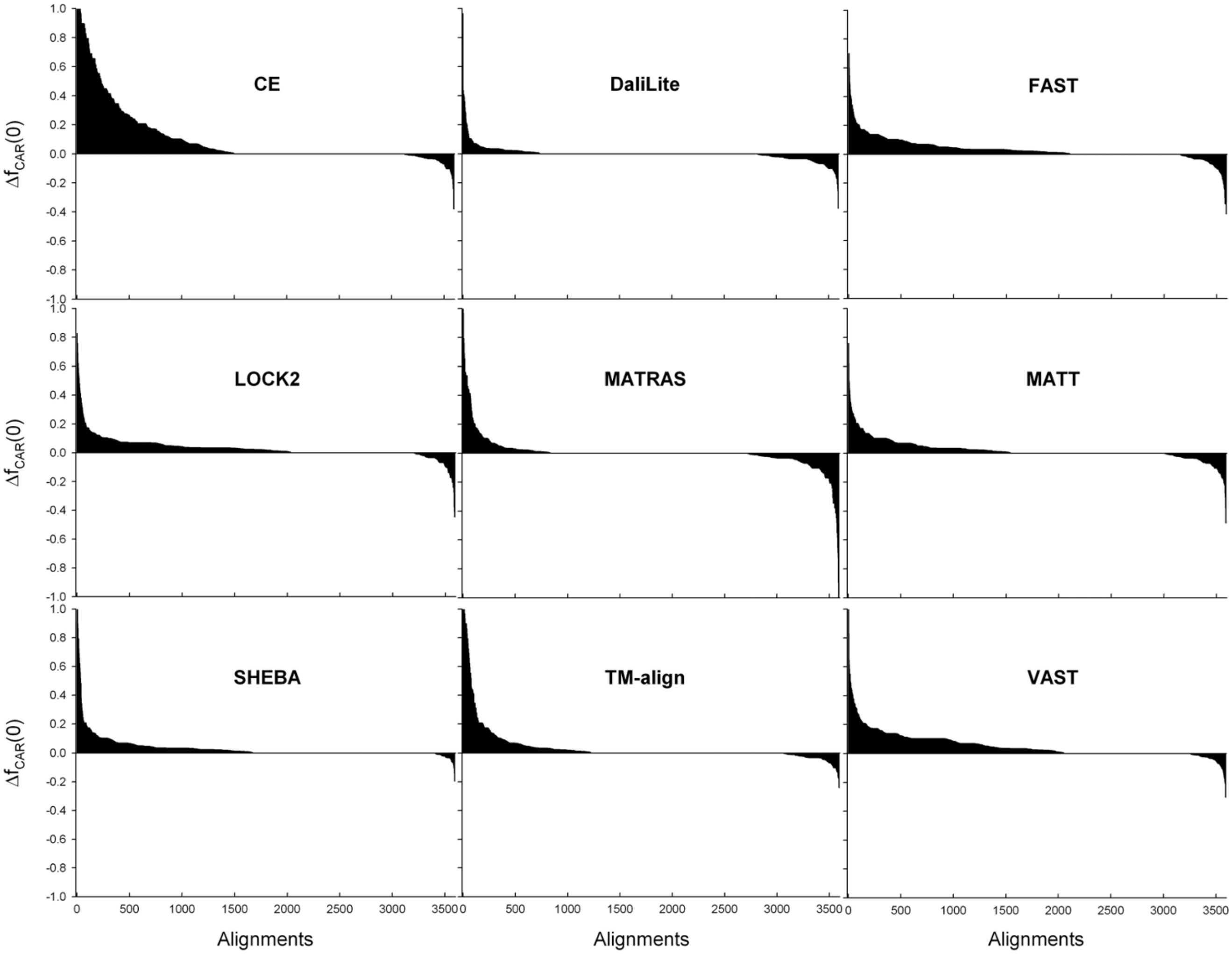


Figure 2

Change in accuracy of each individual alignment after refinement. The alignments from each method were sorted according to their $\Delta f_{CAR}(0)$, which is the $f_{CAR}(0)$ of the refined alignment minus that of the original alignment. The nine graphs, one for each method, are arranged in alphabetical order. The x- and y- axes in each graph represent the alignment and $\Delta f_{CAR}(0)$, respectively. The number of alignments for which $f_{CAR}(0)$ increased and decreased are, respectively, (1495, 469), (733, 770), (2130, 437), (2029, 383), (835, 867), (1542, 581), (1669, 182), (1221, 182), and (2054, 334), for CE, DaliLite, FAST, LOCK2, MATRAS, MATT, SHEBA, TM-align, and VAST, respectively.

RSE improves even a poor alignment. But the final accuracy attained was substantially lower than those from any structure comparison programs.

Comparison of improvements between SE and RSE

The performance of RSE was compared to that of the original SE (Figure 6). SE produced improved alignments for 7 methods, but poorer alignments for DaliLite and MATRAS. RSE made additional improvements for all methods, although the extent of the improvement varied for different methods.

To give concrete examples of improvement, Table 3 lists pairs in the immunoglobulin superfamily for which RSE made most improvement. It can be noted that many programs generate alignments in which no residue pairs are correctly aligned. Panels A and D in Figure 7 show the sequence alignments and the structural superposition, respectively, produced by CE for the pair given in Table 3. Shaded blocks in the sequence alignments indicate the residues aligned in the CDD reference alignment. Note that all the residues in the shaded blocks in panel A are shifted by one residue, resulting in an out-of-phase superposition of β-strands and the mis-alignment of the signature Cys residues of the immunoglobulin domains (panel D). For the pairs shown in Table 3, DaliLite, CE, MATRAS, SHEBA4, TM-align, and VAST produced sequence alignments with one residue shift, while FAST, MATT, and

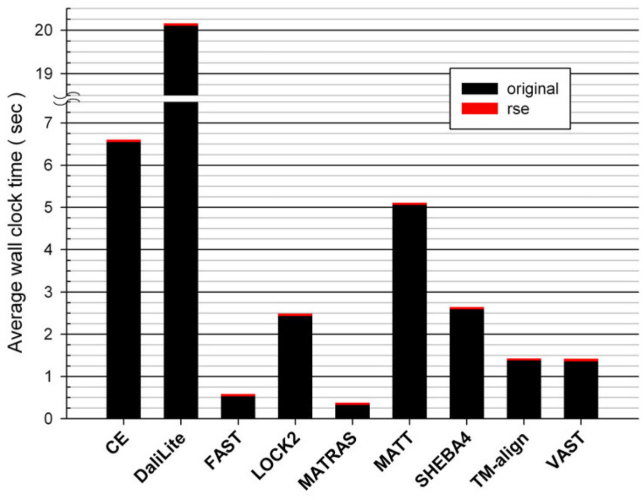


Figure 3

Average execution times. The total wall clock times for each method to align 3,591 pairs and for the RSE to refine them were recorded on Dual 2 GHz PowerPC G5 with 4GB memory, running Mac OS X version 10.3.9. The pre-processing times for MATRAS and VAST were not included. The x- and y-axes show the methods and the average times in seconds, respectively. The times taken by the methods and by the RSE are shown in black bars and red tips, respectively.

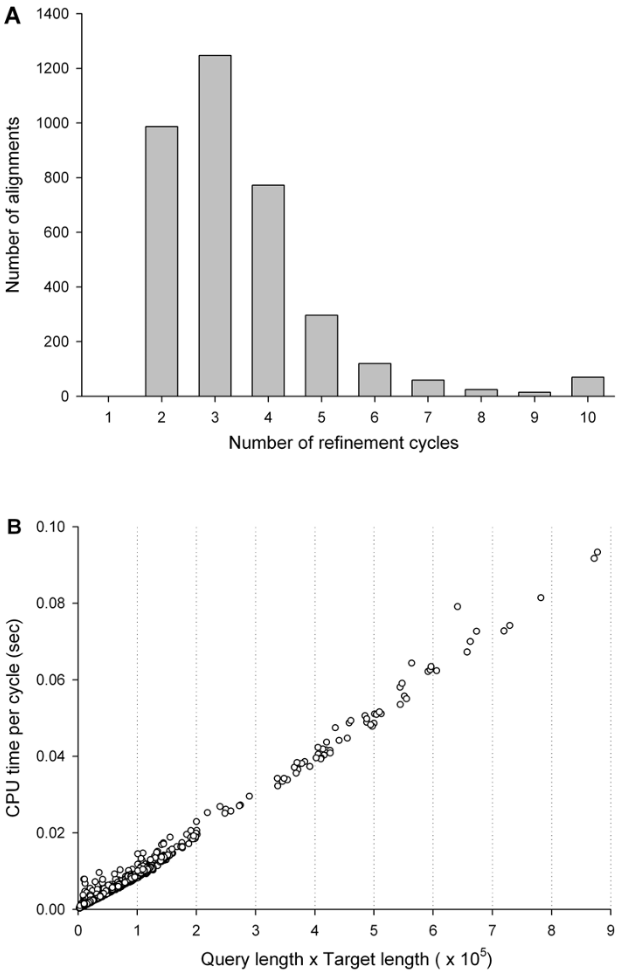


Figure 4

Time complexity for RSE. The number of refinement cycles and the CPU time per refinement cycle were recorded for each of the 3,591 CDD alignments. Panel (A) gives the histogram of the number of alignments vs. the number of refinement cycles. There is no alignment with one cycle because RSE always executes one additional final cycle (unless no alignment is found in the first cycle). Panel (B) gives the CPU times per cycle as a function of the size of the alignment matrix.

LOCK2 produced those with some residues shifted by two residues.

RSE could correct these alignments, unlike SE (panels B and C). Since SE just derives a sequence alignment from a given structural superposition without changing it, it cannot correct a bad superposition. In contrast, RSE iteratively adjusts the structural superposition, which can result in a large improvement.

Quality of the CDD alignments as the standard

In order to better understand the nature of the changes of the CDD alignments by the RSE procedure, many cases were visually inspected. There were 136 pairs (3.4% of all pairs) from 21 different superfamilies for which the $f_{CAR}(0)$ in RSE-refined CDD alignment decreased by more than 20%. As expected, some of these structure pairs were from the cd00531 (7 pairs) and cd01984 (6 pairs) superfamilies, for which our previous study [12] indicated that the CDD alignments were in error. For some pairs from two other superfamilies (cd00198 and cd00385), RSE again appeared to produce more reasonable alignments than CDD, in terms of the distances and orientations of side chains between aligned residues. Fourteen pairs including the worst three cases were from cd00688, which are made of α/α toroid structures (a barrel made of two layers of alpha-helices). Not all helices in these structures could be superposed simultaneously without ambiguity and RSE produced tilted alignments. There were 47 pairs from the three superfamilies having the $(\beta/\alpha)_8$ TIM-barrel structure (cd01292, cd00415 and cd00945), for which the inner layer of beta-strands were reasonably alignable but the outer helices were not. There were other helix-containing superfamilies (cd00389, cd00397, cd00198, and cd00385), for which at least one pair of alpha-helices was not unambiguously alignable. For some pairs in cd00158, CDD has pairs of residues aligned, which RSE could not align because they were too far apart from each other in an

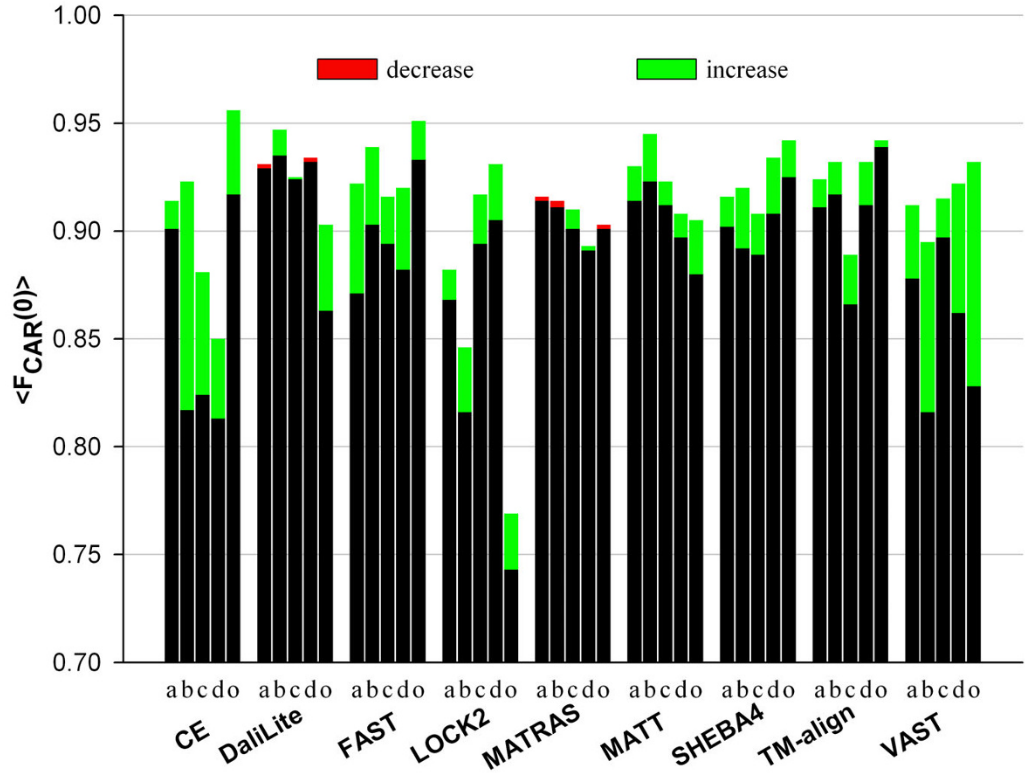


Figure 5

Dependence of improvement by the RSE procedure on the SCOP class. The structure pairs were grouped according to their SCOP class and then <$F_{CAR}(0)$> in each class was computed for each method before and after refinement by the RSE procedure. The SCOP class names in single characters are under each bar along the x-axis: **a**, **b**, **c**, and **d** for all-α, all-β, α/β, and α+β classes, respectively; **o** for the other (other than **a** to **d)** classes. Color scheme is the same as in Figure 1.

irregularly shaped region of the superposed structures. These were aligned in CDD presumably by sequence similarity.

Discussion

Structure-based sequence alignments are not as robust as one would like. In some cases, they can be inherently ambiguous. But more frequently different structure alignment programs generate alignments that contain errors that can be easily recognized by human experts. We showed in a previous study [12] that, the overall average accuracy of structure-based sequence alignments, as measured by <$F_{CAR}(0)$> with the CDD root node set as the reference, ranged from 81% to 89% depending on the program used. When the five outlier superfamilies [12] are excluded, it ranges from 84% to 92% (Figure 1). The two newly included methods, TM-align [20] and MATT [16], are not exceptional in this regard.

The RSE procedure reported here was designed to improve the structure-based sequence alignments. It uses the previously reported SE algorithm [21] to obtain a refined sequence alignment from an input alignment. SE is a heuristic algorithm that produces an alignment from two

Table 2: Average performance of the control methods

Methods	CDD+RSE	DaliLite		SSEARCH		SALIGN	
		-	+RSE	-	+RSE	-	+RSE
<$F_{CAR}(0)$>	0.946	0.923	0.929	0.471	0.665	0.548	0.748
<$F_{CAR}(8)$>	0.981	0.968	0.970	0.634	0.726	0.664	0.808
<$f_{CAR}(0)$>	0.963	0.928	0.929	0.326	0.494	0.411	0.592
<$f_{CAR}(8)$>	0.988	0.984	0.984	0.507	0.612	0.577	0.793

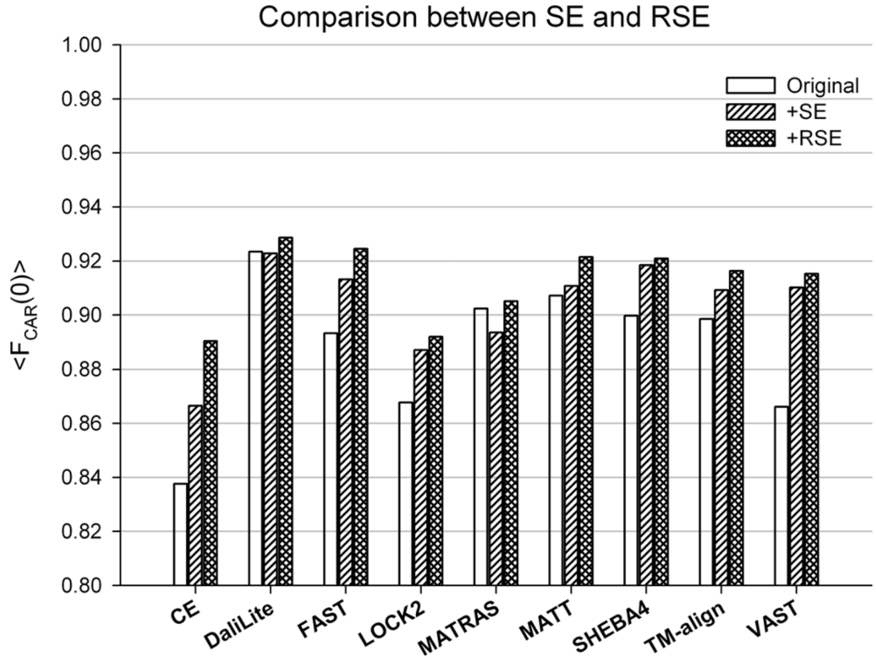


Figure 6

Average fraction of correctly aligned residues before and after modification by either SE or RSE. The average accuracies of sequence alignment were computed for each method before (white) and after modification either by SE (hatched) or RSE (crosshatched). The method name is given under each group of three bars along the x-axis. The y-axis gives <F_{CAR}(0)>, the average fraction of correctly aligned residues, averaged over the superfamily.

superimposed structures without using a gap penalty. Figure 1 shows that the average accuracy improved for all structure alignment programs tested by adding the RSE refinement procedure. Notably, alignments from MATT, which is a program that considers structural flexibility, could also be improved significantly by the RSE procedure, which does not explicitly consider structural flexibility. RSE reduced the shift error for most programs since the refinement increases $F_{CAR}(0)$ more than $F_{CAR}(8)$. For FAST and SHEBA-4, RSE seems to lengthen the alignment also since $F_{CAR}(8)$ and $F_{CAR}(0)$ increased to a similar extent. The alignments improved for structure pairs from all SCOP classes for most of the programs tested (Figure 5).

Table 3: The most improved case in immunoglobulin superfamily (cd00096) for each method

Program	Protein pair (SCOP domains)	$f_{CAR}(0)$		
		-	+SE	+RSE
CE	d1a6aa1-d1cdi_1	0.000	0.207	1.000
DaliLite	d1fg2b_-d1wioa2	0.000	0.207	0.966
FAST	d1a1ma1-d2ig2h1	0.345	0.000	0.862
LOCK2	d1c5da1-d1cid_1	0.172	0.414	1.000
MATRAS	d1ev2e2-d1i1ad2	0.000	0.000	1.000
MATT	d1cs6a1-d1f3jb1	0.241	0.138	1.000
SHEBA4	d1a2yb_-d1e4ka2	0.000	0.000	1.000
TM-align	d1f3jb1-d1vcaa2	0.000	0.207	1.000
VAST	d1a2yb_-d1cqka_	0.000	0.000	1.000

Impressively, the alignments from FAST, one of the fastest programs, could be improved to about the same level of accuracy as those from DaliLite, the best performer without RSE (Figure 1). The accuracies of MATT and SHEBA-4 also increased to similar levels. These improvements were achieved with nearly negligible increase in overall processing times (Figures 2 and 3). Therefore structure alignments can be done with substantially reduced computational cost without compromising accuracy by combining RSE with one of the fastest programs. Alternatively, the RSE procedure can be implemented to replace the traditional residue-based dynamic programming algorithm in a structure comparison program that uses it to improve both the accuracy and computing time.

An ideal refinement procedure will fix incorrectly aligned regions without degrading the correctly aligned ones (Figure 7). Unfortunately, RSE seems to degrade some alignments when compared to the CDD alignments (Figure 2). When the CDD alignment itself was used as the initial

A. CE alignment

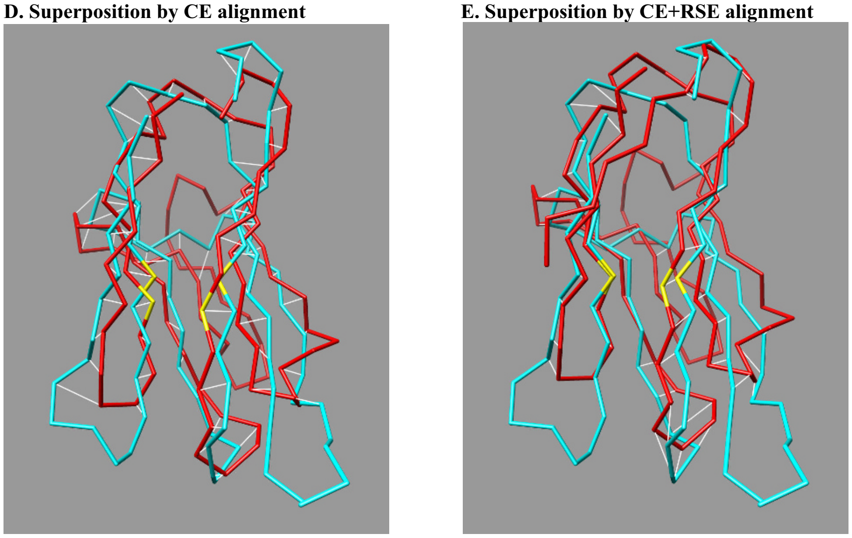
```
d1a6aa1 LREPNVLICFIDKFTPPVVNVTWLRNGKPVT----------------TGVSET--VFLpr
d1cdi_1 KKGDTVELTCTASQKKSIQFHWKNSNQIKILgnqgsfltkgpsklndRADSRRslWDQ--

d1a6aa1 edhlfRKFHYLPF-LPSTEDVYDCRVEHwgldEPLLKHWEF
d1cdi_1 -----GNFPLIIKnLKIEDSDTYICEVE----DQKEEVQLL
```

B. CE+SE alignment

```
d1a6aa1 itnvppevtvltnspvelrepnvlicfidkFTPPVVNVTWLRN-GKPVTtgvsetvflpr
d1cdi_1 tkkvvlgkkgdtveltcta-----------SQKKSIQFHWKNSnQIKILgnqgsfltkgp

d1a6aa1 edhlfrkfhylpflpstedvydcrvehwgldepllkhwef---------
d1cdi_1 sklndradsrrslwdqgnfpliiknlkiedsdtyicevedqkeevqllv
```

C. CE+RSE alignment

```
d1a6aa1 itnvppevtvltnspVELREPNVLICFIdkFTPPVVNVTWLRN-GKPVTtgvse------
d1cdi_1 tkkvvl---------GKKGDTVELTCTA--SQKKSIQFHWKNSnQIKILgnqgsfltkgp

d1a6aa1 ------------TVFlpredhlfRKFHYLPFlps---TEDVYDCRVEhwgldePLLKHWE
d1cdi_1 sklndradsrrsLWD--------QGNFPLIIknlkieDSDTYICEVE------DQKEEVQ

d1a6aa1 F--
d1cdi_1 Llv
```



D. Superposition by CE alignment

E. Superposition by CE+RSE alignment

Figure 7

An example of the refinement of the CE alignment. The structure-based sequence alignments by CE alone (A), CE followed by SE (B) and CE followed by RSE (C) are shown with the shaded blocks indicating CDD reference alignments. The disulfide-forming cysteine residue pairs are highlighted in yellow. The aligned and unaligned residues are indicated by uppercase and lowercase letters, respectively. The panels D and E show the superpositions by CE and after refinement by RSE, respectively. The aligned residue pairs are indicated by white lines. The cysteine residues are in yellow. The blue and red structures represent the SCOP domains d1a6aa1 and d1cdi_1, respectively.

alignment for an RSE procedure, <$F_{CAR}(0)$> and <$f_{CAR}(0)$> decreased to about 95% and 96%, respectively (Table 2). According to our visual inspection of a number of cases for which $f_{CAR}(0)$ fell to a value below 80%, the RSE procedure appears to have found an alternate alignment or to have corrected CDD errors in most cases. We expect that similar causes are at work for at least some of the cases seen in Figure 2 for which there is an apparent degradation of alignment accuracy.

RSE greatly improves the alignments from SSEARCH and SALIGN, which are non-structure-based, pure sequence-based alignment procedures (Table 2). This is to be expected since use of the structural information should improve the sequence alignment. One notes, however, that the average accuracy attained after the refinement is far below those of any of the structure alignment methods (Compare the numbers in Table 2 and the bar heights in Figure 1). This indicates that the outcome of the RSE procedure does depend on the quality of the input alignment. One can also note that there are about 7 to 11% error left after the RSE refinement of the alignments of all methods (Figure 1) and that no method reached the accuracy of refined CDD alignments (about 95% in Table 2). These observations imply that RSE could not correct certain errors of the input alignments. This could happen because some needed seed alignments could not be found from a poor initial superimposed structures and/or because of the constraints imposed by the inflexible, rigid body superposition of structures.

Conclusion

We devised a refinement procedure for structure-based sequence alignments, called RSE. It uses the SE algorithm, which produces a sequence alignment without using a gap penalty. When applied to the structure-based sequence alignments generated by various structure comparison/alignment programs, the average accuracy increased for all programs tested. This refinement procedure is fast enough to be routinely used as a supplemental procedure following a regular structure-based sequence alignment or to replace the traditional dynamic programming algorithm-based refinement procedure which is a part of many structural alignment programs.

Methods

The RSE procedure

We first briefly describe the SE algorithm [21]. Given a pair of superimposed structures A of length *m* and B of length *n*, define two *m* × *n* matrices *M* and *SP*. *M* is the matrix of average C_α distances defined as $M_{ij} = \sqrt{(d_{i-1,j-1}^2 + d_{i,j}^2 + d_{i+1,j+1}^2)/3}$, where d_{ij} is the distance between the C_α atoms of residue *i* of structure A and residue *j* of structure B. *SP* is the matrix of scalar products; $SP_{ij'}$ is the scalar product between two unit vectors which bisect the angles formed by three consecutive C_α atoms, (i-1, i, i+1) for structure A and (j-1, j, j+1) for structure B. A pair of residues (*i, j*) is a seed if its corresponding matrix element M_{ij} is the minimum in both the i^{th} row and the j^{th} column of the matrix and SP_{ij} is greater than 0. A set of consecutive (non-gapped) seeds defines a seed segment. The SE algorithm consists of the following steps:

1. Find *seeds* and *seed segments*.

2. Find *aligned segments* by extending *seed segments*.

3. Find the consistent set of *aligned segments* with the best score.

4. Discard all other aligned segments.

We modified the original SE algorithm slightly as follows. In the original SE, a seed segment was defined as a set of 3 or more consecutive seeds along a diagonal. In the new algorithm, we first label a seed at residue pair *i* and *j* as *tied* if there is another residue pair involving *i* or *j* such that $M_{i'j}$ - M_{ij} < 0.5 Å or $M_{ij'}$ - M_{ij} < 0.5 Å with a positive $SP_{i'j}$ or $SP_{ij'}$, respectively. Then we define a seed segment as a set of 2 (instead of 3 in the previous version) or more consecutive non-tied seeds. The tied seeds are ignored also during the extension of seed segments to obtain the aligned segments. We made this amendment because we observed instances wherein two, not three, consecutive residues are unambiguously aligned, isolated from other aligned regions. The following steps were newly appended.

5. Extend the surviving *aligned segments* after discarding the inconsistent aligned segments.

6. Change *tied seeds* to *extended pairs* if they do not overlap with already aligned residue pairs.

7. Repeat steps 3 to 5.

The reason for introducing step 5 is that there may be room for extension after removal of inconsistent segments. The additional steps 6 and 7 were used only in the last refinement cycle (see below) to pick up isolated pairs of alignable residue pairs.

For RSE, the sequence alignment by SE without steps 6 and 7 was followed by a rigid body superposition routine KABSCH [27,28]. This two-step process was repeated for up to 10 times until the alignment converged (until the last two alignments were the same). In the rigid-body superposition step, each aligned residue pair was weighted according to the distance d_{ij} between C_α atoms of the aligned residues: $w_{ij} = \frac{1}{1+(d_{ij}/d_0)^n}$ [29], where d_0 and *n* are adjustable parameters with default values of 3.0 Å and 2, respectively. Several combinations of d_0 (= 2.5 to 4.0 Å in 0.5 Å steps) and *n* (= 1 to 4) were tested, but the RSE procedure was rather insensitive to these parameters. During the iteration, the transformation matrix of the superposition that generated the best alignment, in terms of the number of aligned residues, was selected. The final sequence alignment was produced by an additional round of SE that included steps 6 and 7 after two structures were superimposed according to the chosen transformation matrix.

The RSE procedure accepts as input two superimposed structures or two independent structures with a sequence alignment, in which case a superposition is obtained through KABSCH procedure with unit residue weights. In this work, the RSE was run in the latter mode, since some structure alignment programs did not generate superimposed structures. Different programs produced sequence alignments in different formats, which had to be converted into a standard format (the FASTA format). The iterative refinement steps can be skipped by giving *-norefine* command line option, in which case the input superposition is used directly to generate the sequence alignment output. The program is downloadable from the following web site: http://lmbbi.nci.nih.gov/.

Reference alignments, structure alignment programs and time measure

We used the CDD (v.2.07) "root node set" introduced in our previous work [12] as the reference sequence alignments with corresponding SCOP domains. We chose this dataset because it is manually procured and because it includes many sequences that are sufficiently dissimilar that structure is needed for their accurate alignment. The 5 'outlier' superfamilies (cd00651, cd01345, cd02156, cd01284, and cd02688) were excluded, for which the CDD alignments were judged questionable as reference alignments [12]. The composition of the dataset is described in Table 1.

We included CE (Algorithm 1.0, Alignment calculator 1.02) [30], DaliLite_2.4.1 [13], LOCK2 [18], FAST [17], MATRAS (version 1.2) [19], MATT [16], SHEBA-4.0 [31], TM-align [20] and VAST (directly from Dr. Gibrat) [14]. We also included SSEARCH from FASTA3 package for pure sequence alignment [32] and SALIGN from Modeller (mod9v6) for profile-profile alignment [33]. The input multiple alignments for SALIGN were prepared from PSI-BLAST alignments (BLASTPGP [34] in blast-2.2.20 package), allowing up to 20 iterations with e-value cutoff of 0.0005 against nr database (as of 04/19/2009). Up to 1,000 sequences with most significant e-values were retained in the multiple sequence alignment. The parameter settings for PSI-BLAST were as described in Marti-Renom et.al. [33]. Otherwise, default values were used for all the programs.

In order to measure the execution times for the methods including the RSE procedure, time-stamps were recorded before and after system calls for the executables. For the CPU times per refinement cycle with CDD alignments, the elapsed time from after the initial structure superposition to the end of refinement cycles, which did not include the file I/O time, was divided by the number of refinement cycles. The CPU times for each alignment were averaged over three independent runs.

Authors' contributions

BL generated the original idea, CK executed the research, CHT participated in the modification of SE and in the development of the RSE software, and CK and BL wrote the paper. All authors read and approved the final manuscript.

Acknowledgements

This research was supported by the Intramural Research Program of the NIH, National Cancer Institute, Center for Cancer Research.

References

1. Yang AS, Honig B: **An integrated approach to the analysis and modeling of protein sequences and structures. III. A comparative study of sequence conservation in protein structural families using multiple structural alignments.** *J Mol Biol* 2000, **301(3)**:691-711.
2. Eidhammer I, Jonassen I, Taylor WR: **Structure comparison and structure patterns.** *J Comput Biol* 2000, **7(5)**:685-716.
3. Chakrabarti S, Lanczycki CJ, Panchenko AR, Przytycka TM, Thiessen PA, Bryant SH: **State of the art: refinement of multiple sequence alignments.** *BMC Bioinformatics* 2006, **7**:499.
4. Lassmann T, Sonnhammer EL: **Automatic assessment of alignment quality.** *Nucleic Acids Res* 2005, **33(22)**:7120-7128.
5. Edgar RC: **MUSCLE: multiple sequence alignment with high accuracy and high throughput.** *Nucleic Acids Res* 2004, **32(5)**:1792-1797.
6. Gelly JC, Chiche L, Gracy J: **EvDTree: structure-dependent substitution profiles based on decision tree classification of 3D environments.** *BMC Bioinformatics* 2005, **6**:4.
7. Goonesekere NC, Lee B: **Context-specific amino acid substitution matrices and their use in the detection of protein homologs.** *Proteins* 2008, **71(2)**:910-919.
8. Yang AS: **Structure-dependent sequence alignment for remotely related proteins.** *Bioinformatics* 2002, **18(12)**:1658-1665.
9. Bennett-Lovsey RM, Herbert AD, Sternberg MJ, Kelley LA: **Exploring the extremes of sequence/structure space with ensemble fold recognition in the program Phyre.** *Proteins* 2008, **70(3)**:611-625.
10. Godzik A: **The structural alignment between two proteins: is there a unique answer?** *Protein Sci* 1996, **5(7)**:1325-1338.
11. Mayr G, Domingues FS, Lackner P: **Comparative analysis of protein structure alignments.** *BMC Struct Biol* 2007, **7**:50.
12. Kim C, Lee B: **Accuracy of structure-based sequence alignment of automatic methods.** *BMC Bioinformatics* 2007, **8**:355.
13. Holm L, Park J: **DaliLite workbench for protein structure comparison.** *Bioinformatics* 2000, **16(6)**:566-567.
14. Madej T, Gibrat JF, Bryant SH: **Threading a database of protein cores.** *Proteins* 1995, **23(3)**:356-369.
15. Ye Y, Godzik A: **FATCAT: a web server for flexible structure comparison and structure similarity searching.** *Nucleic Acids Res* 2004:W582-585.
16. Menke M, Berger B, Cowen L: **Matt: local flexibility aids protein multiple structure alignment.** *PLoS Comput Biol* 2008, **4(1)**:e10.
17. Zhu J, Weng Z: **FAST: a novel protein structure alignment algorithm.** *Proteins* 2005, **58(3)**:618-627.
18. Shapiro J, Brutlag D: **FoldMiner and LOCK 2: protein structure comparison and motif discovery on the web.** *Nucleic Acids Res* 2004:W536-541.
19. Kawabata T: **MATRAS: A program for protein 3D structure comparison.** *Nucleic Acids Res* 2003, **31(13)**:3367-3369.
20. Zhang Y, Skolnick J: **TM-align: a protein structure alignment algorithm based on the TM-score.** *Nucleic Acids Res* 2005, **33(7)**:2302-2309.
21. Tai CH, Vincent JJ, Kim C, Lee B: **SE: An algorithm for deriving sequence alignment from a pair of superimposed structures.** *BMC Bioinformatics* 2009, **10(Suppl 1)**:S4.
22. Pettersen EF, Goddard TD, Huang CC, Couch GS, Greenblatt DM, Meng EC, Ferrin TE: **UCSF Chimera – a visualization system for exploratory research and analysis.** *J Comput Chem* 2004, **25(13)**:1605-1612.
23. Yang AS, Honig B: **An integrated approach to the analysis and modeling of protein sequences and structures. I. Protein structural alignment and a quantitative measure for protein structural distance.** *J Mol Biol* 2000, **301(3)**:665-678.
24. Kleywegt GJ, Jones TA: **A super position.** *CCP4/ESF-EACBM Newsletter on Protein Crystallography* 1994, **31**:9.
25. Russell RB, Barton GJ: **Multiple protein sequence alignment from tertiary structure comparison: assignment of global and residue confidence levels.** *Proteins* 1992, **14(2)**:309-323.

26. Marchler-Bauer A, Anderson JB, Derbyshire MK, DeWeese-Scott C, Gonzales NR, Gwadz M, Hao L, He S, Hurwitz DI, Jackson JD, *et al.*: **CDD: a conserved domain database for interactive domain family analysis.** *Nucleic Acids Res* 2007:D237-240.
27. Kabsch W: **A solution for the best rotation to relate two sets of vectors.** *Acta Crystallographica Section A* 1976, **32:**922-923.
28. Kabsch W: **A discussion of the solution for the best rotation to relate two sets of vectors.** *Acta Crystallographica Section A* 1978, **34:**827-828.
29. Gerstein M, Levitt M: **Comprehensive assessment of automatic structural alignment against a manual standard, the scop classification of proteins.** *Protein Sci* 1998, **7(2):**445-456.
30. Shindyalov IN, Bourne PE: **Protein structure alignment by incremental combinatorial extension (CE) of the optimal path.** *Protein Eng* 1998, **11(9):**739-747.
31. Jung J, Lee B: **Protein structure alignment using environmental profiles.** *Protein Eng* 2000, **13(8):**535-543.
32. Pearson WR: **Searching protein sequence libraries: comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms.** *Genomics* 1991, **11(3):**635-650.
33. Marti-Renom MA, Madhusudhan MS, Sali A: **Alignment of protein sequences by their profiles.** *Protein Sci* 2004, **13(4):**1071-1087.
34. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ: **Gapped BLAST and PSI-BLAST: a new generation of protein database search programs.** *Nucleic Acids Res* 1997, **25(17):**3389-3402.