2598–2606 *Nucleic Acids Research, 2006, Vol. 34, No. 9*
doi:10.1093/nar/gkl274

Refining multiple sequence alignments with conserved core regions

Saikat Chakrabarti, Christopher J. Lanczycki, Anna R. Panchenko, Teresa M. Przytycka, Paul A. Thiessen and Stephen H. Bryant*

National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bethesda, MD, 20894, USA

Received January 11, 2006; Revised February 19, 2006; Accepted April 3, 2006

ABSTRACT

Accurate multiple sequence alignments of proteins are very important to several areas of computational biology and provide an understanding of phylogenetic history of domain families, their identification and classification. This article presents a new algorithm, REFINER, that refines a multiple sequence alignment by iterative realignment of its individual sequences with the predetermined conserved core (block) model of a protein family. Realignment of each sequence can correct misalignments between a given sequence and the rest of the profile and at the same time preserves the family's overall block model. Large-scale benchmarking studies showed a noticeable improvement of alignment after refinement. This can be inferred from the increased alignment score and enhanced sensitivity for database searching using the sequence profiles derived from refined alignments compared with the original alignments. A standalone version of the program is available by ftp distribution (ftp://ftp.ncbi.nih.gov/pub/REFINER) and will be incorporated into the next release of the Cn3D structure/alignment viewer.

INTRODUCTION

The advent of large genome projects has led to an explosion of sequence data in public databases. In this connection, the establishment of structural, functional and evolutionary similarity between proteins and protein domains is a challenging task. Several domain databases are now available which combine homologous protein domains into the distinct families and represent them in the form of domain multiple sequence alignments. The accuracy of domain identification, protein classification and reconstruction of phylogenetic history of domain families crucially depends on the quality of underlying sequence alignments. Some domain resources, such as PFAM (1) and ProDom (2), rely on the automated methods of multiple sequence alignment while others, such as SMART (3) and CDD (4), employ careful manual intervention in constructing the domain models. The CDD database contains alignments that are carefully curated to be consistent with structure–structure alignments to preserve the conserved core of a protein domain family. Each curated CDD alignment records conserved features within the family members in terms of 'blocks', the regions where every sequence is aligned without the gaps.

Different methods have been proposed to produce a multiple sequence alignment. Some of them align all sequences simultaneously (5,6), while others apply a progressive alignment strategy (7–10). According to the latter, the sequences are aligned in a predetermined order dictated usually by the guide tree which groups similar sequences together with the subsequent addition of more dissimilar ones. This approach has been implemented in variety of programs and packages such as MULTALIGN (11), MULTAL (9) and CLUSTALW (10). While being widely accepted, progressive alignment has its own pitfalls as the misalignment made at previous stages can not be corrected afterwards and can propagate into serious alignment errors. Moreover, the final alignment strongly depends on the order of sequences being aligned. To overcome these flaws, iterative approaches have introduced the capacity to reconsider and realign previously aligned sequences at each iteration with the goal of improving the overall alignment score (7,12–19). Sequences are realigned in a random order and the iteration cycle ends as soon as a convergence criterion has been satisfied. While this strategy faces the problem of being trapped in a local minimum and producing suboptimal alignment (like most other multiple sequence alignment methods), it proves to be robust and produces more accurate alignments (14,18–19).

In this article we present an algorithm named REFINER that aims to refine an existing multiple alignment using the

*To whom correspondence should be addressed. Tel: 001 301 435 7792; Fax: 001 301 480 9241; E-mail: bryant@ncbi.nlm.nih.gov predetermined 'block model' of that domain family as a biologically relevant constraint on the search space. A block model represents conserved sequence/structure regions which are highly unlikely to contain gaps and are common to all family members. The refinement protocol works by iterative random selection and realignment of sequences with the family block model until the alignment score saturates to a stable value or until the iteration cycle terminates. Realignment of each sequence can correct misalignments between a given sequence and the rest of the profile by shifting the individual blocks on that sequence and at the same time preserves the family's overall block model (i.e. the conserved core regions). The latter constraint prohibits the insertion of gap characters in the middle of conserved blocks. Following a cycle of shifting for each sequence, the blocks in the block model can also be extended or shortened in size depending on the overall score improvement. The algorithm has been benchmarked against structure-based, manually curated (CDD) and un-curated (PFAM) alignments and shows an overall score improvement. Comparison of the algorithm against another independent alignment refinement method (19) showed better performance. The refinement is further tested and validated by checking the reliability in retrieval of functionally important sites and enhanced sensitivity for profile-based database searches compared with the original curated alignments. This method is reasonably fast and can realign several hundreds of highly diverse sequences within minutes. The refinement method also provides a means to detect the outlier sequences within an alignment and may thereby point a way towards new subfamily identification schemes.

MATERIALS AND METHODS

A benchmark to evaluate the accuracy of refinement algorithm

To test the overall performance of the refinement algorithm we used a collection of 362 manually curated 'parent alignments' (set_362) from the CDD version 2.00 (4). The current version of CDD is available at http://www.ncbi.nlm.nih.gov/Structure/cdd/cdd.shtml. A parent alignment corresponds to the most ancient (i.e. topmost) family in a domain family hierarchy, several hundred of which are currently defined by CDD. A smaller test set (set_94) of 94 multiple alignments from CDD (with more than five protein structure entries) was used to optimize the parameters for block extension cycles. In addition, we applied the refinement algorithm also on 900 un-curated PFAM (1) alignments [generated either by CLUSTALW (10) or T-Coffee (20)].

To compare the database search sensitivity of the Position Specific Scoring Matrices (PSSMs) computed from multiple sequence alignments before and after the refinement procedure, we first constructed a list of true positives for the conserved domain families from set_362. True positives here are defined as those proteins/domains which are structurally similar, as defined by the VAST algorithm (21,22), to the representative structure of CDD alignments. First, for each CDD alignment we chose a representative structure so that the CDD footprint on this structure and corresponding structural domain/chain boundaries [domain definitions from MMDB structure database have been used (23)] were consistent to a degree of 80% mutual overlap. By CDD footprint we mean the region on a structure between the first and the last residues aligned in CDD. For CDD alignments that have a corresponding MMDB structural domain, the VAST structure neighbors of an MMDB domain/chain are retrieved from the non-redundant set of MMDB chains. This set of 10 185 chains (db10185) was constructed by single-linkage clustering, based on BLAST E-values of 10^{-80} or less, from all the entries in the MMDB structure database (23). Finally, we checked the overlap of the CDD footprint and the VAST footprint (the regions between first and last residues aligned by VAST) for each structure from db10185. If the overlap comprised at least 80% of each footprint length, then that structure was recorded as being true positive and the VAST footprint was recorded as a valid region. At the end of this procedure 280 CDD family alignments (set_280) were selected which had at least one true positive entry.

Algorithm

We developed an algorithm that refines an existing alignment by systematically realigning each sequence to better match the profile constructed for the remaining sequences in the family. The refinement is constrained by the block model implicit in the family's multiple alignment, where a block model comprises an ordered set of one or more non-overlapping blocks. By definition, an aligned block in a multiple alignment is a region containing no gaps on any sequence; it is specified simply by a start position and a length in residues. An unaligned region between blocks is called a loop. While we discuss our algorithm in the context of a CDD multiple alignment that explicitly defines the conserved core region with a block model, it applies more generally since it is straightforward to preprocess any multiple alignment, e.g. a PFAM alignment, and derive a block model from its set of gap-free columns. A tool to preprocess a FASTA file can be downloaded with the REFINER application itself.

The algorithm performs one or more iterations of refinement, each of which in addition to iterative realignment of individual sequences contains a phase of 'block shifting' followed by a 'block editing' phase. The overall refinement method is described in the flow chart (Figure 1). The engine of the block shifting phase is a dynamic programming (DP) module (24) that incorporates constraints to align a sequence to the block model of a specified multiple alignment. Specifically, the DP module finds the optimal placement of every block in the block model on a protein sequence, using the PSSM of the specified multiple alignment to score each allowed arrangement of the blocks on the sequence. The shifted blocks retain the same relative order after DP. Since a block plays the role of a single residue in traditional DP algorithms (e.g. Smith–Waterman), the block-model-constrained DP we use is very fast even for long alignments. To prevent spurious long inserts, the distribution of loop sizes in the original multiple alignment is used to further constrain the possible block positions in the final alignment (24). In the results described herein no loop in the refined alignment is allowed to exceed the maximum loop length from the original alignment. In the block shifting phase of the refinement, the DP engine runs on each sequence of the original multiple alignment to set new block positions. The order in which

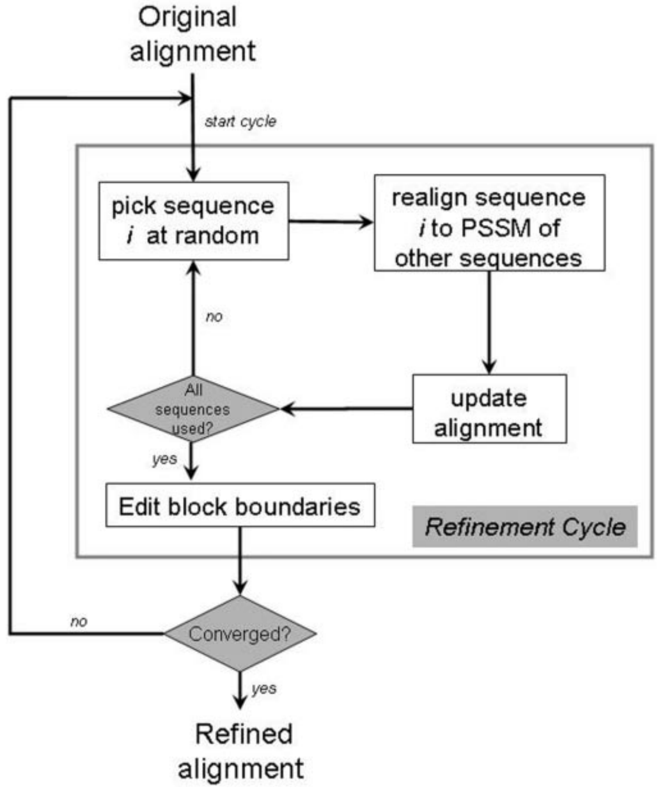


Figure 1. Flowchart of the refinement algorithm.

the sequences are refined is randomized to avoid bias and make the use of multiple iterations more effective. Convergence is declared when no further improvement of overall alignment score is observed or all iterations (maximum of five iterations are used in our benchmarking studies) have expired. The subsequent block editing phase examines each block across all sequences in the multiple alignment to see if it is advantageous to extend it at the N- and/or C-terminal end. For this to be possible, the adjacent loop regions must contain gap-free columns. Since this implies a block-length change, we do not use the quantitative methods from the block shifting phase to evaluate potential extensions. In the next section, we introduce a 3-fold heuristic criteria used to control block editing based on the statistical properties observed for block and loop regions

Objective functions

Several scoring functions are used to evaluate the overall alignment quality as well as the fitness of a particular sequence to the existing profile.

For an alignment of length *L* the 'row score', $\boldsymbol{S_r}$, is the sum of scores (derived from the PSSM) over all aligned positions of a particular row (sequence) *r*,

$$S_r = \sum_{i=1}^{L} \mathrm{PSSM}(i, AA_i^r). \tag{1}$$

Here the PSSM is indexed by alignment column *i* and the corresponding residues, *AA* from the sequence '***r***'. The 'alignment score', $\boldsymbol{S_N}$, for a multiple alignment with *N* rows is simply the sum of all row scores:

$$S_N = \sum_{r=1}^{N} S_r. \tag{2}$$

S_N is used as an objective function in the algorithm and S_r is employed by the DP engine to determine the refined block positions for row *r*, and in that context care is taken to evaluate S_r against the PSSM computed from a multiple alignment where row *r* is removed to avoid bias. Correcting an initial misalignment of a particular sequence typically improves the overall profile quality, measured in terms of S_N.

During block editing, the scores above are not appropriate to decide whether it is beneficial to add a new column(s) at a block terminus because (i) the decision involves all sequences of the alignment so that using S_r is not relevant and (ii) adding a new column would usually improve the overall score S_N whether truly beneficial or not. While large positive PSSM values usually correspond to regions of conservation in a multiple alignment, small PSSM values do not necessarily imply the location of badly aligned regions. Thus, we want to use a measure to ensure addition of only informative columns yet not overlook columns containing well-conserved residues with relatively small substitution scores.

To standardize the parameters for block extension we examine all columns in blocks from CDD alignments in the set_94 to establish heuristics based on observed differences in PSSM values between alignment columns in blocks and those that are not. Based on this analysis we developed a 3-fold criterion used to evaluate block extension events during refinement: after each block shifting phase, columns in loop regions adjacent to blocks are scrutinized and added to existing blocks until a column fails the 3-fold block extension test.

First, the distribution pattern of the median PSSM values for residues in block-forming columns from set_94 indicates that for a column to be added, the residues aligned in that column should have a median PSSM matrix value of at least three (for details see Results and Supplementary Data, Figure SM5). Next, we examine both the frequency of occurrence of negative scores and the relative weight of negative PSSM values in block-forming columns considering that the frequency and relative weight of negative scores (unfavorable substitutions) in the PSSM should be minimized for conserved block-forming columns. Characterization of the block-forming columns in set_94 suggests that the following threshold values for these two additional parameters are characteristic of alignment columns where block extension is predicted to be beneficial. The frequency of negative scores for a block-forming column, computed simply as the ratio of the number of sequences with a negative PSSM value to the number of alignment rows, should not exceed 0.3. The relative weight of negative scores, computed as the absolute sum of negative PSSM values in a column divided by the sum of all PSSM values in that column, also should not exceed 0.3. It has been observed that all the three criteria should be used together to achieve better performance on block extension. Data supporting these choices for block editing parameters are presented in Supplementary Data (Figure SM5). The distribution of the PSSM values for non-block columns are found to be distinctively different from the block forming columns within the dataset set_94 (data not shown).

Measuring the performance of refined alignment

If the block shifting phase of the refinement has an effect on a sequence, the position of some of the blocks of the alignment on that sequence must change. Changes in position of a block '*b*' have been recorded as a 'relative block shift' *B*, calculated as

$$B = \frac{\Delta_{\mathrm{shif}t}}{l_b}, \tag{3}$$

where for a given sequence $\Delta_{\mathrm{shift}}$is the difference in the position of block *N*-terminus before and after the refinement and ℓ_b is the length of the block.

The refined alignments are tested for their ability to retrieve previously identified functionally important sites. We used version 2.00 of the CDD alignment models to collect information on the location of functionally important sites that had been manually recorded by CDD curators during detailed literature surveys. Nearly 7100 functionally important sites ('features' in CDD alignments) were retrieved from the set_362 multiple alignment models. The original alignment of each of the functionally important columns (FICs) was compared against that found after applying our refinement method. This benchmarking study provides an important standard of truth by which to gauge the accuracy and quality of the alignments produced by our refinement algorithm.

We used two different search methods, HMMER (25) and SALTO_global (24) to test the ability of the refined sequence profiles to find the corresponding VAST neighbors in the dataset of 10 185 structural chains. The sensitivity–specificity analysis was performed by calculating the receiver operating characteristic (ROC) curves and ROC statistics. For a given protein family one can calculate the fraction of detected true positives and false positives at each similarity measure cut-off (*E*-value for HMMER or raw score for SALTO_global). Sensitivity here is defined as a number of detected true positives divided by the overall number of true positives in a database. The specificity is calculated as a ratio between the number of found false positives and the overall number of false positives in the database. To compare sensitivities of profiles before and after the refinement we measure the sensitivity at 1 and 5% of false positive rates.

RESULTS

Improvement of alignment scores

The alignment refinement algorithm has been applied to 362 CDD multiple alignments and 900 PFAM (1) alignments derived by both CLUSTALW (10) and T-Coffee (20) [See Supplementary Data for lists]. Each sequence within these alignments was realigned using the refinement procedure. The overall alignment score has been calculated (Equation 2) and the relative score improvement upon refinement is plotted in Figure 2. As can be seen from this figure, in most cases for both curated (75% of CDD alignments) and un-curated alignments (66 and 58% for CLUSTALW and T-Coffee derived PFAM alignments) the alignment score has been improved even without implementing the block editing phase of the procedure, and the block editing phase improves the performance by additional 10–12%. However, higher occurrences of negative improvement of alignment score are observed in un-curated (CLUSTALW and T-Coffee) alignment compared to curated (CDD) alignments. Our algorithm executes five iterations (or less if converged) and in 70% of the cases the overall score improves from iteration to iteration (Figure SM1 in Supplementary Data) which shows the effectiveness of applying the iterative scheme in the refinement. It seems that improvement of the alignment score largely depends on the sequence diversity of the input alignments. Our results indicate that 66% of all the improved alignments fall within 10–30% average sequence identity range (Figure SM2 in Supplementary Data).

It should be noted that the realignment of each sequence generally improves both the fitness of that particular sequence to the existing PSSM (row score, Equation 1) and the overall alignment score (Equation 2). Grossly misaligned sequences can therefore be detected by analyzing the difference in the row score before and after the refinement. Although the detected misalignment can be automatically corrected in some cases (if the row score has been improved), there can be sequences that do not belong to a given domain family (the row score can go down). Hence, by applying the iterative refinement one can identify such family outliers and remove them. As expected,

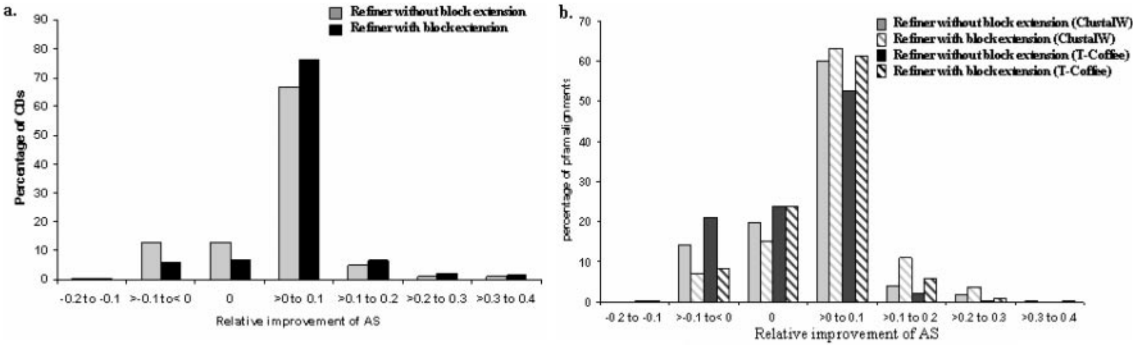


Figure 2. Improvement of alignment score after refinement. The histogram of relative improvement of alignment score (AS, Equation 2) for curated CDD alignments (**a**) and un-curated PFAM alignments (**b**) with or without block extension.

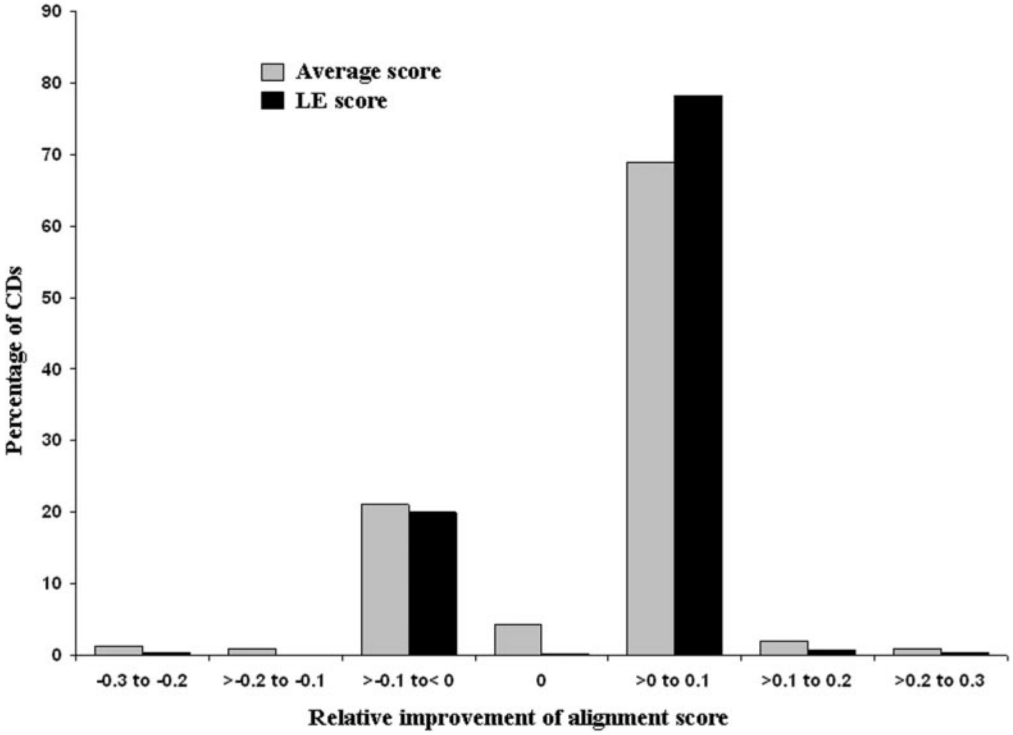


Figure 3. Comparison of performance of refinement. Histograms of relative improvement of alignments score achieved by our refinement method (REFINER) over RF algorithm of Wallace *et al*. (19) refinement package. Alignments are evaluated by both average and LE scores (19). Relative improvement of alignment score is measured as the difference between the final scores after application of REFINER and RF method divided by the final score obtained by RF method.

we showed that the number of such sequences is quite low in manually curated CDD alignment (Supplementary Data, Figure SM3).

Comparison of REFINER with other independent refinement method

We have compared the performance of our refinement method (REFINER) with an iterative method for improvement of multiple sequence alignment developed by Wallace *et al*. (19). We applied their algorithm to our dataset (set_362) and compared the improvement obtained by their and our method. We applied the Remove first (RF) algorithm from the Wallace *et al*. method using both average and log expectation (LE) scores to evaluate the alignments (19). Figure 3 shows the relative improvement of alignment (represented by average score and LE score) obtained by the REFINER over the method of Wallace *et al*. (19). For most cases (72 and 80% of CDs for average and LE scores, respectively) our alignment refinement method provides better performance in terms alignment score compared to that achieved by RF algorithm.

Alignment score changes with respect to the block shift

To illustrate how the alignment score (Equation 2) changes with respect to the magnitude of adjustments made in the alignment, we plot the relative improvement of block score (25% or higher) versus the block shift (Figure 4). The block shift parameter is a convenient way to quantify the changes in the alignment since the refinement algorithm shifts individual sequences for a better alignment without altering the length and number of blocks. As can be seen from this figure the highest score improvement is only observed for a small block shift of <10 residues. At the same time if the alignment is considerably changed upon the refinement, the overall score almost does not increase. This may indicate that moderate adjustments of the curated alignments made by the refinement algorithm are almost always beneficial while large alignment shifts may be detrimental given how reasonable the original structure-based (CDD) alignments are. This is further supported by the observation of similar correlation between the improvement of overall alignment score and the block shift (Figure SM4, Supplementary Data). The examination of scores of individual blocks before and after the refinement can also be useful in identifying the problematic blocks which either should be removed or realigned manually.

The extension of the blocks, as previously mentioned, may also improve the overall alignment score; altogether 25% of all blocks from 68% of CDD alignments can undergo extension. Figure 5 shows that adding one or two residues at the ends of blocks does not lead to a score increase whereas more extensive editing of block boundaries (up to 10 residues at either block terminus) can improve the alignment score. Stretching the blocks over the whole alignment would yield very small improvement as well. This in turn implies that the current CDD block model is specific enough for a given family but there are conserved columns in the inter-block regions which should also be taken into account.

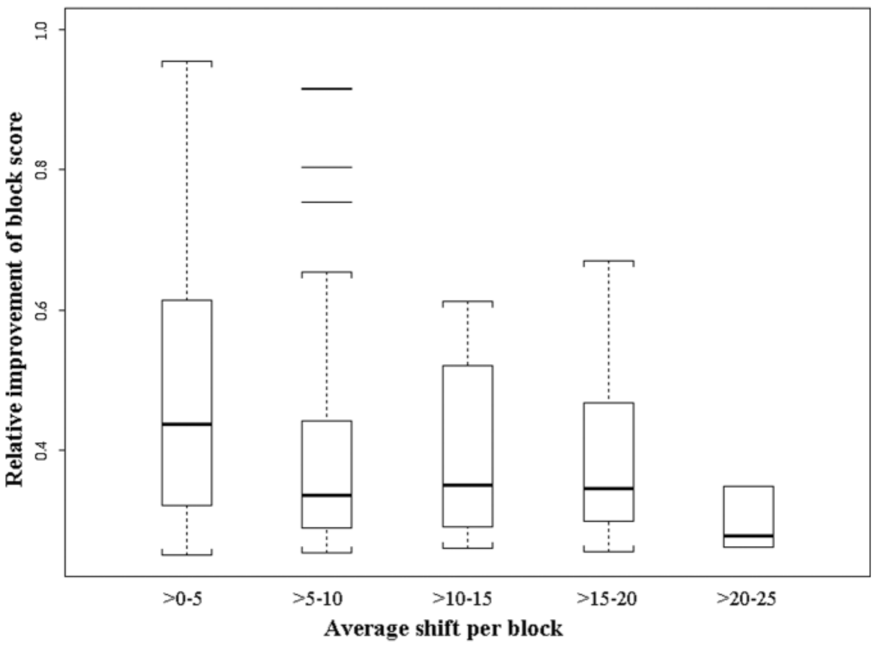


Figure 4. Effect of block shift on block score. The relative improvement (using structure-based, curated CDD alignments) of score of 25% or higher per block is plotted versus the block shift. The central line in each box shows the median value, the upper and lower boundaries of individual box show the upper and lower quartiles, and the vertical lines extent to a value of 1.5 times the inter quartile range. Outlier values are shown outside the whiskers. Values on top of each box provide the percentage of data points for each block shift bin.

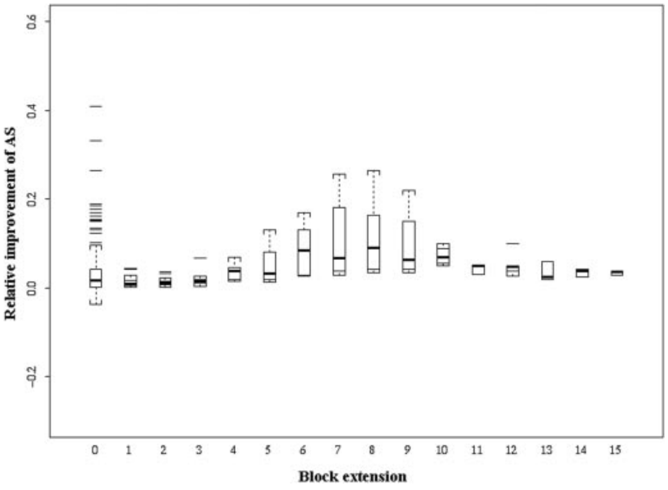


Figure 5. Effect of block extension on alignment score. The relative improvement of alignment score (AS, Equation 2) is shown for each bin of the block extension. Block extension is calculated as the sum of the extended columns for all blocks within a CDD alignment.

Quality control by recovering the functionally important columns

The accuracy of the refinement algorithm has been tested further in terms of the retrieval of previously identified FICs from set_362 structure-based, curated alignments. We assume that the FICs should not change much upon the refinement since they represent manually annotated conserved sites for a family and can be used as a standard of truth. It has been shown previously that fully automated procedure of functional site prediction can recover most of the annotated FIC from CDD for similar but not identical test set (26). FICs were extracted (see Materials and Methods) from the original CDD alignments and compared against that derived after applying the refinement on the same CDD alignment. Figure 6 shows that majority (~70%) of FICs remain unchanged after the refinement; i.e. the refinement algorithm does not perturb and successfully reproduces the exact alignment for important residues. We also carefully examined those cases where the refinement algorithm introduced some changes in the alignment of FICs. In fact, 67% of changed FICs have a higher score after the refinement as compared with the original alignment (Figure 6 inset), indicating some improvement is possible even at these sites.

Figure 7 shows an example of the alignment of the members of Bowman–Birk type proteinase inhibitor (BBI) family before and after the refinement. FICs are highlighted by yellow in the figure. While the refinement algorithm improves the overall score for the whole alignment (~5%), it also changes the alignment of FICs so that score of some FICs increases by 24%. As can be seen from this figure the score increase is caused by the correction of misaligned Cys residues which form the disulfide bond in all the representative structures of this family. Therefore, in this case the refinement algorithm was able to not just reproduce the FICs but also improve the alignment for some of them.

Comparison of the sensitivity/specificity of PSSMs before and after the refinements

One way to validate the multiple alignment method is to examine its performance of produced alignment or sequence profile in homology-based database searches. We used two

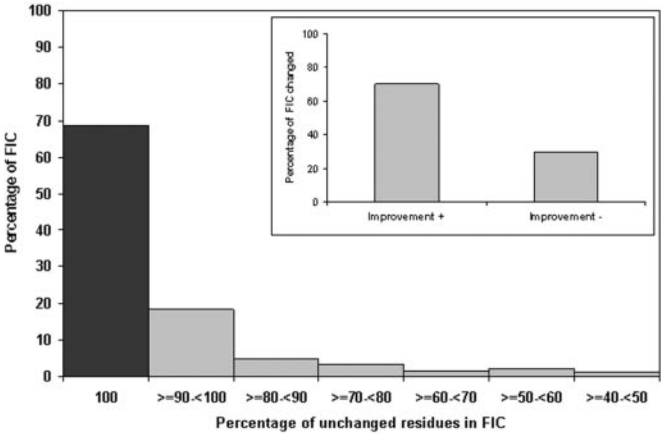
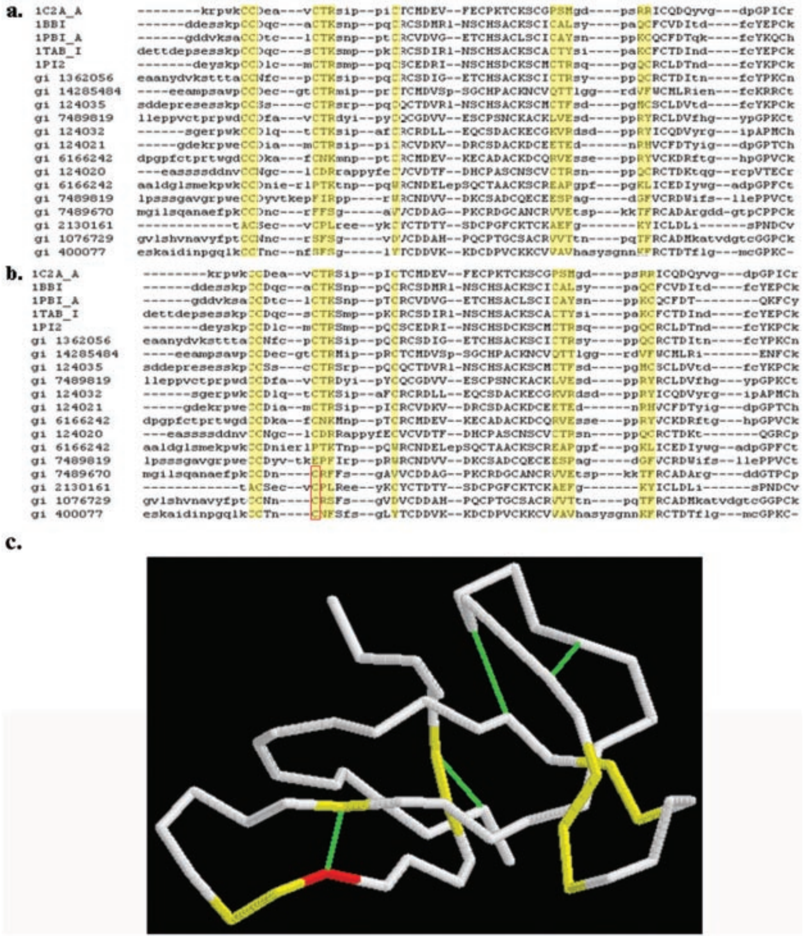


Figure 6. Quality control by testing the recovery of FIC. Alignments of FICs are compared before and after the refinement. The automated refinement procedure could reproduce the exact same alignment that was obtained by careful manual curation for most (shown by black box) of the FICs. In addition, majority of the changed FICs show better score (inset, improvement+) when compared against the score derived before the refinement.

methods to perform the database searches namely HMMER and SALTO_global (24). The HMM models were constructed using global mode of HMMER 2.3.2 (25). SALTO_global represents a global version of SALTO algorithm presented in the Materials and Methods section (24) and requires all blocks to be aligned in the final multiple sequence alignment. Each HMM model and PSSM derived from the alignments before and after the refinement procedure were used to search the 'non-redundant' database of protein chains (db10185).

The sensitivities of finding VAST neighbors by two search methods HMMER and SALTO_global at 1 and 5% error rates are given in Table 1. It is clear from the table that the sensitivities of sequence profiles/HMMs have increased upon the refinement; this result is more pronounced when the HMMER is used to search the database. Although the overall improvement in sensitivity is not dramatic (~5%) it implies that the refinement algorithm produces better alignments. To investigate this further we calculated the correlation between the relative improvement in alignment score and relative improvement in sensitivity (Table 2). This table shows that there is statistically significant correlation between the



```
a.  1C2A_A       ---------krpwkCCDea---vCTRsip--piCTCMDEV--FECPKTCKSCGPSMgd----psRRICQDQyvg---dpGPICr
    1BBI         ------ddesskpCCDqc---aCTKsnp--pqCRCSDMRl-NSCHSACKSCICALsy----paQCFCVDItd----fcYEPCk
    1PBI_A       ------gddvksaCCDtc---lCTKsnp--ptCRCVDVG--ETCHSACLSCICAYsn----ppKCQCFDTqk----fcYKQCh
    1TAB_I       dettdepsesskpCCDqc---sCTKsmp--pkCRCSDIRl-NSCHSACKSCACTYsi----paKCFCTDInd----fcYEPCk
    1PI2         -------deyskpCCDlc---mCTRsmp--pqCSCEDRI--NSCHSDCKSCMCTRsq----pgQCRCLDTnd----fcYKPCk
    gi 1362056   eaanydvkstttaCCNfc---pCTKsip--pqCRCSDIG--ETCHSACKSCICTRsy----ppQCRCTDItn----fcYPKCn
    gi 14285484  ----eeampsawpCCDec--gtCTRmip--prCTCMDVSp-SGCHPACKNCVQTTlgg---rdVFWCMLRien---fcKRRCt
    gi 124035    sddepresesskpCCSs----cCTRsrp--pqCQCTDVRl-NSCHSACKSCMCTFsd----pgMCSCLDVtd----fcYKPCk
    gi 7489819   lleppvctprpwdCCDfa---vCTRdyi--pyCQCGDVV--ESCPSNCKACKLVEsd----ppRYRCLDVfhg---ypGPKCt
    gi 124032    ------sgerpwkCCDlq---tCTKsip--afCRCRDLL--EQCSDACKECGKVRdsd---ppRYICQDVyrg---ipAPMCh
    gi 124021    -----gdekrpweCCDia---mCTRsip--piCRCVDKV--DRCSDACKDCEETEd------nRHVCFDTyig---dpGPTCh
    gi 6166242   dpgpfctprtwgdCCDka---fCNKmnp--ptCRCMDEV--KECADACKDCQRVEsse---ppRYVCKDRftg---hpGPVCk
    gi 124020    ---eassssddnvCCNgc---lCDRrappyfeCVCVDTF--DHCPASCNSCVCTRsn----ppQCRCTDKtqg--rcpVTECr
    gi 6166242   aaldglsmekpwkCCDnie-rlPTKtnp--pqWRCNDELepSQCTAACKSCREAPgpf---pgKLICEDIywg--adpGPFCt
    gi 7489819   lpsssgavgrpweCCDyvtkepFIRpp----rWRCNDVV--DKCSADCQECEESPag-----dGFVCRDWifs--llePPVCt
    gi 7489670   mgilsqanaefpkCCDnc---rFFSg-----aVVCDDAG--PKCRDGCANCRVVEtsp---kkTFRCADArgdd-gtpCPPCk
    gi 2130161   ------------tACSec---vCPLree--ykCYCTDTY--SDCPGFCKTCKAEFg-------KYICLDLi------sPNDCv
    gi 1076729   gvlshvnavyfptCCNnc---rSFSg-----vDVCDDAH--PQCPTGCSACRVVTtn----pqTFRCADMkatvdgtcGGPCk
    gi 400077    eskaidinpgqlkCCTnc--nfSFSg-----lYTCDDVK--KDCDPVCKKCVVAVhasysgnnKFRCTDTflg---mcGPKC-
b.  1C2A_A       ---------krpwkCCDea--vCTRSip--pICTCMDEV--FECPKTCKSCGPSMgd----psRRICQDQyvg---dpGPICr
    1BBI         ------ddesskpCCDqc--aCTKSnp--pQCRCSDMRl-NSCHSACKSCICALsy----paQCFCVDItd----fcYEPCk
    1PBI_A       ------gddvksaCCDtc--lCTKSnp--pTCRCVDVG--ETCHSACLSCICAYsn----ppKCQCFDT---------QKFCy
    1TAB_I       dettdepsesskpCCDqc--sCTKSmp--pKCRCSDIRl-NSCHSACKSCACTYsi----paKCFCTDInd----fcYEPCk
    1PI2         -------deyskpCCDlc--mCTRSmp--pQCSCEDRI--NSCHSDCKSCMCTRsq----pgQCRCLDTnd----fcYKPCk
    gi 1362056   eaanydvkstttaCCNfc--pCTKSip--pQCRCSDIG--ETCHSACKSCICTRsy----ppQCRCTDItn----fcYPKCn
    gi 14285484  ----eeampsawpCCDec-gtCTRMip--pRCTCMDVSp-SGCHPACKNCVQTTlgg---rdVFWCMLRi--------ENFCk
    gi 124035    sddepresesskpCCSs---cCTRSrp--pQCQCTDVRl-NSCHSACKSCMCTFsd----pgMCSCLDVtd----fcYKPCk
    gi 7489819   lleppvctprpwdCCDfa--vCTRDyi--pYCQCGDVV--ESCPSNCKACKLVEsd----ppRYRCLDVfhg---ypGPKCt
    gi 124032    ------sgerpwkCCDlq--tCTKSip--aFCRCRDLL--EQCSDACKECGKVRdsd---ppRYICQDVyrg---ipAPMCh
    gi 124021    -----gdekrpweCCDia--mCTRSip--pICRCVDKV--DRCSDACKDCEETEd------nRHVCFDTyig---dpGPTCh
    gi 6166242   dpgpfctprtwgdCCDka--fCNKMnp--pTCRCMDEV--KECADACKDCQRVEsse---ppRYVCKDRftg---hpGPVCk
    gi 124020    ---eassssddnvCCNgc--lCDRRappyfECVCVDTF--DHCPASCNSCVCTRsn----ppQCRCTDKt--------QGRCp
    gi 6166242   aaldglsmekpwkCCDnierlPTKTnp--pQWRCNDELepSQCTAACKSCREAPgpf---pgKLICEDIywg--adpGPFCt
    gi 7489819   lpsssgavgrpweCCDyv-tkEPFIrp--pRWRCNDVV--DKCSADCQECEESPag-----dGFVCRDWifs--llePPVCt
    gi 7489670   mgilsqanaefpkCCDn----CRFFs---gAVVCDDAG--PKCRDGCANCRVVEtsp---kkTFRCADArg----ddGTPCp
    gi 2130161   ------------tACSec--vCPLRee--yKCYCTDTY--SDCPGFCKTCKAEFg-------KYICLDLi------sPNDCv
    gi 1076729   gvlshvnavyfptCCNn----CRSFs---gVDVCDDAH--PQCPTGCSACRVVTtn----pqTFRCADMkatvdgtcGGPCk
    gi 400077    eskaidinpgqlkCCTn----CNFSfs--gLYTCDDVK--KDCDPVCKKCVVAVhasysgnnKFRCTDTflg---mcGPKC-
c.
```

Figure 7. Improvement of alignment after refinement. Alignments of Bowman-Birk type proteinase inhibitor (BBI) family (CDD code: cd00023) derived (**a**) before and (**b**) after the refinement show marked improvement. Block forming columns are displayed in capitals where functional important residues are boxed in yellow. One of the conserved cysteine sites is shown in red in (b) and (c) where probable misalignments are corrected in number of sequences. (**c**) Displays backbone representation of the structure of hydrolase inhibitor (pdb code: 1C2A). Functional important sites are marked in yellow and disulfide bonds are shown in green.

Table 1. Sensitivity values estimated from the ROC curves at 1 and 5% error rates (fraction of false positives)

Search method	Error rate	Before_Refiner	After_Refiner	After_Refiner_ext[a]
SALTO_global	1%	0.43	0.44	0.45
	5%	0.49	0.50	0.51
HMMER	1%	0.47	0.48	0.50
	5%	0.54	0.56	0.58

[a]Refiner with BLOCK extension.

Table 2. Correction between the relative improvement of alignment score and sensitivity

Search method	PSSM used	Correlation cofficient	
		At 1% FP	At 5% FP
SALTO_global	After_Refiner	0.35*	0.38*
	After_Refiner_ext	0.42*	0.47*
HMMER	After_Refiner	0.45*	0.40*
	After_Refiner_ext	0.58*	0.49*

All values are shown over the increases from search derived with original (before refinement) alignments.
*Statistical significance or P-values < 0.05.

alignment score and sensitivity, where the latter is being used as an indication of alignment accuracy. Even though database search methods may not be sensitive enough to capture the entire difference between the profiles before and after the refinement, we can conclude that the improvement observed in the alignment score can be the result of the alignment improvements made by the refinement algorithm.

DISCUSSION

We developed a new algorithm to refine a multiple alignment of protein sequences. Our approach assumes that the alignment is represented by a set of conserved regions or blocks that are aligned without gaps. One example of such a block-based alignment constitutes the structure-based, manually curated CDD alignment, which can be used to infer the domain organization, predict functional sites or model structures of unknown proteins. In this case the refinement algorithm can be used as a tool to assist CDD curation and as a standalone program to refine a multiple sequence alignment. We have also applied our refinement algorithm on un-curated multiple alignments (PFAM dataset) derived by CLUSTALW (10) and T-Coffee (20).

In this article we demonstrated that our refinement algorithm improves the input alignment not only when it is applied to un-curated alignment but also to a curated alignment. Since the input alignment represents a rather accurate alignment and assumes block structure, we have to develop sensitive, dedicated methods to measure the alignment improvement. Therefore, in addition to measuring the improvement by alignment score increase we analyzed the statistics of block extensions and shifts, quality of recovering FICs and sensitivity of sequence profiles based on refined alignments. We showed that the refinement algorithm shows an improvement with respect to all these measures. We also showed better performance of our method when compared with an iterative alignment refinement method (19).

In summary, our approach provides a fast and accurate method for refinement of existing block-based alignments. In particular, this method can be used to refine alignments by correcting local misalignments and to find sequence outliers, which do not fit the domain family model.

SUPPLEMENTARY DATA

Supplementary data are available at *NAR* Online.

ACKNOWLEDGEMENTS

The authors thank Dr Maricel G. Kann for kindly providing a standalone version of SALTO_global program. This work was supported by the Intramural Research Program of the National Library of Medicine at National Institutes of Health/DHHS. Funding to pay the Open Access publication charges for this article was provided by Intramural Research Program of the National Library of Medicine at National Institutes of Health/ DHHS.

Conflict of interest statement. None declared.

REFERENCES

1. Bateman,A., Birney,E., Cerruti,L., Durbin,R., Etwiller,L., Eddy,S.R., Griffiths-Jones,S., Howe,K.L., Marshall,M. *et al.* (2002) The Pfam protein families database. *Nucleic Acids Res.*, **30**, 276–280.
2. Servant,F., Bru,C., Carrere,S., Courcelle,E., Gouzy,J., Peyruc,D. and Kahn,D (2002) ProDom: Automated clustering of homologous domains. *Brief. Bioinformatics*, **3**, 246–251.
3. Letunic,I., Goodstadt,L., Dickens,N.J., Doerks,T., Schultz,J., Mott,R., Ciccarelli,F., Copley,R.R., Ponting,C.P. and Bork,P. (2002) Recent improvements to the SMART domain-based sequence annotation resource. *Nucleic Acids Res.*, **30**, 242–244.
4. Marchler-Bauer,A., Panchenko,A.R., Shoemaker,B.A., Thiessen,P.A., Geer,L.Y. and Bryant,S.H. (2002) CDD: a database of conserved domain alignments with links to domain three-dimensional structure. *Nucleic Acids Res.*, **30**, 281–283.
5. Lipman,D.J., Altschul,S.F. and Kececioglu,J.D. (1989) A tool for multiple sequence alignment. *Proc. Natl Acad. Sci. USA*, **86**, 4412–4415.
6. Stoye,J., Moulton,V. and Dress,A.W. (1997) DCA: an efficient implementation of the divide-and-conquer approach to simultaneous multiple sequence alignment. *Comput Appl Biosci.*, **13**, 625–626.
7. Hogeweg,P. and Hesper,B. (1984) The alignment of sets of sequences and the construction of phyletic trees: an integrated method. *J. Mol. Evol.*, **20**, 175–186.
8. Feng,D.F. and Doolittle,R.F. (1987) Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J. Mol. Evol.*, **25**, 351–360.
9. Taylor,W.R (1988) A flexible method to align large numbers of biologicalsequences. *J. Mol. Evol.*, **28**, 161–169.
10. Thompson,J.D., Higgins,D.G. and Gibson,T.J. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, **22**, 4673–4680.
11. Barton,G.J. and Sternberg,J.E. (1987) A strategy for the rapid multiple alignment of protein sequences. Confidence levels from tertiary structure comparisons. *J. Mol. Biol.*, **198**, 327–337.
12. Gotoh,O. (1996) Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments. *J. Mol. Biol.*, **264**, 823–838.
13. Notredame,C. and Higgins,D.G. (1996) SAGA: sequence alignment by genetic algorithm. *Nucl. Acids Res.*, **24**, 1515–1524.
14. Heringa,J. (2002) Local weighting schemes for protein multiple sequence alignment. *Comput. Chem.*, **26**, 459–477.

15. Berger,M.P. and Munson,P.J. (1991) A novel randomized iterative strategy for aligning multiple protein sequences. *Comput. Appl. Biosci.*, **7**, 479–484.
16. Katoh,K., Misawa,K., Kuma,K. and Miyata,T. (2002) MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Res.*, **30**, 3059–3066.
17. Do,C.B., Mahabhashyam,M.S., Brudno,M. and Batzoglou,S. (2005) ProbCons: probabilistic consistency-based multiple sequence alignment. *Genome Res.*, **2**, 330–340.
18. Wang,Y. and Li,K.B. (2004) An adaptive and iterative algorithm for refining multiple sequence alignment. *Comput. Biol. Chem.*, **28**, 141–148.
19. Wallace,I.M., O'Sullivan,O. and Higgins,D.G (2005) Evaluation of iterative alignment algorithms for multiple alignment. *Bioinformatics*, **21**, 1408–1414.
20. Notredame,C., Higgins,D.G. and Heringa,J. (2000) T-Coffee: a novel method for fast and accurate multiple sequence alignment. *J. Mol. Biol.*, **302**, 205–217.
21. Madej,T., Gibrat,J-F. and Bryant,S.H. (1995) Threading a database of protein cores. *Protein Struct. Funct. Genet.*, **23**, 356–369.
22. Gibrat,J.F., Madej,T. and Bryant,S.H. (1996) Surprising similarities in structure comparison. *Curr. Opin. Struct. Biol.*, **6**, 377–385.
23. Chen,J., Anderson,J.B., DeWeese-Scott,C., Fedorova,N.D., Geer,L.Y., He,S., Hurwitz,D.I., Jackson,J.D., Jacobs,A.R., Lanczycki,C.J. *et al.* (2003) MMDB: Entrez's 3D-structure database. *Nucleic Acids Res.*, **31**, 474–477.
24. Kann,M.G., Thiessen,P.A., Panchenko,A.R., Schaffer,A.A., Altschul,S.F. and Bryant,S.H. (2005) A structure-based method for protein sequence alignment. *Bioinformatics*, **21**, 1451–1456.
25. Eddy,S.R. (1998) Profile hidden Markov models. *Bioinformatics*, **14**, 755–763.
26. Panchenko,A.R., Kondrashov,F. and Bryant,S. (2004) Prediction of functional sites by analysis of sequence and structure conservation. *Protein Sci.*, **13**, 884–892.