Protein Engineering, Design & Selection vol. 19 no. 3 pp. 129–133, 2006
Published online January 19, 2006 doi:10.1093/protein/gzj005

SHORT COMMUNICATION

Variable gap penalty for protein sequence–structure alignment

M.S.Madhusudhan[1], Marc A.Marti-Renom[1], Roberto Sanchez[2] and Andrej Sali[1,3]

[1]Departments of Biopharmaceutical Sciences and Pharmaceutical Chemistry and California Institute for Quantitative Biomedical Research, University of California at San Francisco, San Francisco, CA 94143 and [2]Structural Biology Program, Mount Sinai School of Medicine, Box 1677, 1425 Madison Avenue, New York, NY 10029, USA

[3]To whom correspondence should be addressed.
E-mail: sali@salilab.org

The penalty for inserting gaps into an alignment between two protein sequences is a major determinant of the alignment accuracy. Here, we present an algorithm for finding a globally optimal alignment by dynamic programming that can use a variable gap penalty (VGP) function of any form. We also describe a specific function that depends on the structural context of an insertion or deletion. It penalizes gaps that are introduced within regions of regular secondary structure, buried regions, straight segments and also between two spatially distant residues. The parameters of the penalty function were optimized on a set of 240 sequence pairs of known structure, spanning the sequence identity range of 20–40%. We then tested the algorithm on another set of 238 sequence pairs of known structures. The use of the VGP function increases the number of correctly aligned residues from 81.0 to 84.5% in comparison with the optimized affine gap penalty function; this difference is statistically significant according to Student's *t*-test. We estimate that the new algorithm allows us to produce comparative models with an additional ~7 million accurately modeled residues in the ~1.1 million proteins that are detectably related to a known structure.

Keywords: comparative protein structure modeling/gap penalty function/homology modeling/sequence–structure alignment

Introduction

Accuracy in the alignment of nucleic acid and protein sequences is key to a number of biological problems, including those of gene annotation, phylogeny determination, protein structure modeling and protein function annotation. A widely used method for aligning two sequences of residues is based on dynamic programming (Needleman and Wunsch, 1970; Sellers, 1974; Smith and Waterman, 1981). Dynamic programming optimizes a scoring function that depends on residue–residue substitution scores and penalties for the creation and extension of gaps.

Methods to improve the accuracy of alignment focused on both aspects of the scoring function. To improve residue matching scores based on the simple Dayhoff-type matrices (Dayhoff *et al*., 1978), environment-dependent substitution matrices (Shi *et al*., 2001) and sequence profile matching (Marti-Renom *et al*., 2004) were proposed. Another group of improvements involve the gap penalty. Typically, an affine gap penalty (AGP) function of the form $g = u + vl$ is used. This function depends on the gap initiation and extension parameters, u and v, and on the number of residues in the gap, l. The parameters of the AGP have been exhaustively optimized (Barton and Sternberg, 1987). In addition, a linear gap penalty function dependent on the structural environment of the gap (Lesk *et al*., 1986), exponential gap penalty forms (Qian and Goldstein, 2001; Goonesekere and Lee, 2004), local alignments with monotonically increasing gap penalties (Mott, 1999) and a user-defined arbitrary gap penalty function (Dewey, 2001) were described.

With an application to comparative protein structure modeling (Marti-Renom *et al*., 2000; Madhusudhan *et al.*, 2005; Moult, 2005) in mind, we are particularly interested in methods that align a protein sequence (target) to a related sequence of known structure (template). The accuracy of comparative protein structure modeling is directly dependent on the accuracy of the target–template alignment (Madhusudhan *et al*., 2005). In this paper, we describe a dynamic programming algorithm with a variable gap penalty (VGP) function that penalizes insertions and deletions between positions that are buried, located within the same regular secondary structure segment and distant in space. We begin by outlining the algorithm, the datasets used in training and testing and measures of alignment accuracy (Methods). We then optimize the parameters of the VGP function and compare its alignment accuracy with that of the optimized AGP function, using as a reference the corresponding structure-based alignments (Results). In addition, several sample alignments using the AGP and VGP functions are compared to illustrate advantages of the VGP function in comparative modeling. Finally, we discuss the VGP function, the corresponding algorithm and their benefits to comparative modeling (Discussion).

Methods

Alignment algorithm

We introduce a dynamic programming algorithm to obtain an optimal alignment between one or more pre-aligned protein sequences (i.e. sequence block) with one or more pre-aligned protein structures and sequences (i.e. structure block). The distinguishing feature of the algorithm is that it can use a VGP function of an arbitrary form and still guarantee a globally optimal solution. The algorithm is implemented in the SALIGN command of the program MODELLER-8 (Sali and Blundell, 1993) (http://salilab.org/modeller/modeller.html). The implementation works for both the global and local

alignment (Needleman and Wunsch, 1970; Smith and Waterman, 1981; Sankoff, 1983) and can utilize either similarity or dissimilarity residue substitution scores.

The problem of the optimal alignment of two sequences (or two blocks of sequences) as addressed by the dynamic programming algorithm (Needleman and Wunsch, 1970; Sellers, 1974; Smith and Waterman, 1981) is as follows. Given two sequences (or blocks of sequences) of length N and M, respectively, a scoring matrix of dimensions $N \times M$ is constructed. Each element $W_{i,j}$ of this scoring matrix is the score for substituting (aligning) residue i in the first sequence with residue j in the second sequence. Substitution scores are taken from standard residue substitution matrices, such as the BLOSUM series of matrices (Henikoff and Henikoff, 2000). The scoring matrix can also be constructed by comparing the sequence profiles at each aligned position (Marti-Renom *et al.*, 2004). The goal is to align the residues from the two sequences so as to optimize the overall alignment score. The alignment score is a sum of scores corresponding to the matched residues and penalties for occurrences of unmatched residues (i.e. gaps). The gap penalty function is usually the AGP $g = u + vl$, where parameters u and v are constant penalties for opening and extending a gap, respectively, and l is the length of the gap.

The recursive dynamic programming equations for the global alignment of the structure block with the sequence block, using a VGP function, are as follows:

$$D_{i,j} = \mathrm{Min}_{\max(0,i-L)\leqslant i'<i,\max(0,j-L)\leqslant j'<j}\left(D_{i',j'} + G_{i,j,i',j'} + W_{i,j}\right)$$

$$D_{i,0} = G_{i+1,1,0,0}$$

$$D_{0,j} = G_{1,j+1,0,0}$$

$$W_{M+1,j} = 0$$

$$W_{i,N+1} = 0$$

The last four equations are the initial conditions for the recursion defined in the first equation. M and N are the lengths of the structure and sequence blocks, respectively, L is the maximum allowed gap length, G is the VGP function of any form and $W_{i,j}$ is the dissimilarity residue substitution score for positions i and j from the structure and sequence blocks, respectively. To obtain W, we use the 20×20 residue substitution matrix BLOSUM62, transformed into dissimilarity scores and scaled to lie between 0 and 1000. The dynamic programming scoring matrix D is calculated for $i = 1$ to $M + 1$ and $j = 1$ to $N + 1$. The optimal score for the global alignment of two blocks, d, corresponds to the smallest element in $D_{M+1,0<j\leqslant N+1}$ and $D_{0<i\leqslant M+1,N+1}$. The residue equivalence assignments (i.e. alignment) are obtained by backtracking in matrix D, starting from the element d (Needleman and Wunsch, 1970). The recursive equations for the local alignment and/or dissimilarity substitution scores are omitted from this paper, but are trivial to derive given the equations above (Durbin, 1998).

Gap penalty function

The function G is the VGP function for simultaneous insertions from positions i' to i in the structure block and from positions j' to j in the sequence block (Figure 1). If $i' = i - 1$ (or $j' = j - 1$), there is no insertion in the structure (sequence) block.

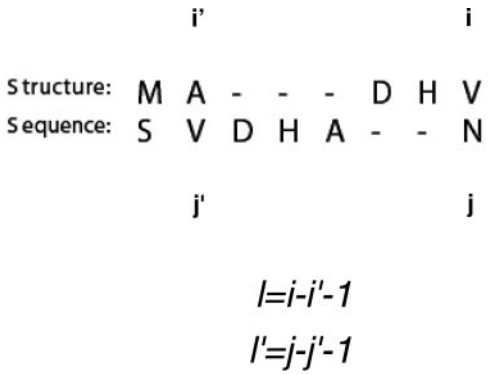

Fig. 1. Definition of a gap. i, i' and j, j' are residue positions on the structure and sequence, respectively. A single gap can include deletions and insertions on either the sequence or structure and is defined by all four indices i, i', j and j'.

The variable gap penalty function used in this study is defined recursively:

$$G_{i,j,i',j'} = \begin{cases} 0, & l = 0 \text{ and } l' = 0 \\ R(i,i')u + (l + l')v - \min(l,l')t & l > 0 \text{ or } l' > 0 \end{cases}$$

$$l = \begin{cases} i - i' - 1 & 0 < i < M \\ \max(0, i - i' + 1 - e) & i' = 0 \text{ or } i = M + 1 \end{cases}$$

$$l' = \begin{cases} j - j' - 1, & 0 < j < N \\ \max(0, j - j' + 1 - e) & j' = 0 \text{ or } j = N + 1 \end{cases}$$

$$R(i,i') = 1 + \left(W_H H_i + W_S S_i + W_B B_i + W_C C_i + W_d P_{i,i'}\right)$$

where l and l' are the lengths of the insertions in the structure and the sequence blocks, respectively, v is the gap extension penalty, u is the gap opening penalty, t is the diagonal gap penalty (Altschul *et al.*, 1997) and e is the maximum number of overhanging residues at the sequence termini that are not penalized for gaps if not aligned. R is the function that modulates the gap opening penalty depending on the structural environment at the position of the insertion. R is at least 1, but can be larger to make the opening of gaps more difficult in the following circumstances: within helices or strands, at buried positions, in straight backbone segments and between two structurally distant residues. The values of H, S, B and C lie between 0 and 1, while $P \geqslant 0$. W_i are the weights of these five properties in R.

H_i is the average value for helical content at position i in the structure block. The numerical value of H_i in every sequence is either 1 or 0 depending on whether the conformation from positions i' to i is helical or not. S_i is a similar measure for the occurrence of a β-strand from positions i' to i.

B_i is the average burial of the residue from position i' to i in the structure block. Residue burial is defined as $1 - a$, where a is the fractional side-chain solvent accessibility on a scale from 0 to 1 (Sali and Overington, 1994).

C_i is the average backbone straightness of residues in the structure block from positions i' to i:

$$C_i = \begin{cases} 1 & \text{if } H_i = 1 \text{ or } S_i = 1 \\ f(\theta) & \text{otherwise} \end{cases}$$

$$f(\theta) = 1 - \min[180^\circ, \max(0^\circ, \theta)]/180^\circ$$

where the angle θ lies in the range 0–180° and is defined by the least-squares lines through Cα atoms $i - 3$ to i and from $i + 3$ to i.

$P_{i,i'}$ depends on the proximity of the two residues spanning the gap:

$$P_{i,i'} = \max(0, d - d_0)^{\gamma}$$

where d is the distance between Cα atoms at positions i' and i averaged over all structures in the structure block, d_0 is an empirical constant corresponding to the distance below which there is no increase in the opening gap penalty and γ is an empirical constant.

Optimized values for all nine parameters (u, v, W_i, d_0 and γ) were obtained by a grid search (see below). The VGP function is reduced to the special case of the AGP function when all weights W_i are set to 0.

Training and testing datasets

DBAli (Marti-Renom *et al*., 2001) was mined to create two sets of pairwise alignments of structures, the first to optimize (train) the VGP parameters and the second to test the accuracy of the resulting alignments. The training and testing sets, containing 240 and 238 alignments, respectively, spanned the sequence identity range 20–40%, with the root mean square deviation (r.s.m.d.) on structural superposition of at most 2.0 Å for at least 80% of the Cα atoms. None of the alignments were of protein sequences with less than 80 residues. The Protein Data Bank (PDB) chain identifies percentage sequence identities, Cα r.s.m.d.s and structure overlap on structural alignment are listed separately for the two sets in supplementary material (http://salilab.org/sup.p.m.at/msm_a2d).

Alignment accuracy

The accuracy of an alignment was measured by superposing the native structures, extracted from the PDB (Berman *et al*., 2002), as implied by the alignment. A rigid-body least-squares superposition of all the Cα atoms was done using the SUPERPOSE command of MODELLER (Sali and Blundell, 1993). Second, the percentage of structurally equivalent positions was defined as the percentage of the Cα atoms in the shorter of the sequences that are within 4 Å of the equivalent atoms in the superposed structure (structure overlap or SOV) (Marti-Renom *et al*., 2004).

Test of statistical significance

Because the distribution of alignment accuracy difference between affine gap penalty alignments and variable gap penalty alignments was approximately Gaussian, the Student's t-distribution statistics allow us to compute whether the estimated average difference is statistically significant (Rees, 1987; Marti-Renom *et al*., 2002). Accordingly, the lower and upper bounds on the average alignment accuracy difference of the whole population of alignments are given by

$$\mu_{u,l} = D \pm \frac{t(n-1, c)S}{\sqrt{n}}$$

where D is the average alignment accuracy difference, S is the standard deviation of the alignment accuracy differences, $t(n-1,c)$ is the Student's t-statistic for $n-1$ degrees of freedom and at a confidence level of c, which in this case is set to 95%. $n = 238$ is the number of alignments in the sample. When 0 lies between μ_l and μ_u, the difference between the performances of the two methods (affine gap penalty and variable gap penalty) is not statistically significant at the given confidence level.

Availability and efficiency

The VGP algorithm is a part of SALIGN, an alignment module in MODELLER-8 (Sali *et al*., 1993) (http://salilab.org/modeller). For a pair of proteins with ∼200 residues each, dynamic programming with AGP is essentially instantaneous. On the other hand, dynamic programming with the variable gap penalty function allowing for arbitrary gap lengths takes ∼2 min on a PC with a 3.6 GHz Pentium 4 CPU. When the maximum gap length is limited to 30 residues, the run time is reduced to ∼15 s.

Results

In this section, we first determine the nine parameters used in the VGP function. We then examine the accuracy of the optimized function on a test set of alignments. Finally, we illustrate the effectiveness of the new algorithm by two examples.

Optimization of gap penalty parameters

The parameters of the variable gap penalty function are too many to optimize simultaneously. Therefore, the parameters were divided into three sets, grouping together the weights for average helicity, strandedness, burial and straightness of residue positions in the structure block (W_H, W_S, W_B and W_C); the weight for gap spanning distance, optimal gap spanning length and the exponent (W_d, d_0 and γ); and the diagonal gap penalty (t). For each set, a grid search in parameter space was performed. The values of parameter sets not being optimized were held fixed at 0 initially and at previously optimized values subsequently. Parameter optimization was terminated on the convergence of the average alignment accuracy score for the training set of alignments. The optimized values for the parameters were $u = -100$, $v = 0$, $W_H = 3.5$, $W_S = 3.5$, $W_B = 3.5$, $W_C = 0.2$, $W_d = 4.0$, $d_0 = 6.5$, $\gamma = 2.0$ and $t = 0$. To speed up the computation, we set the maximum gap length to 30 residues based on the maximum gap length in the structure-based alignments in the test and training sets. The two parameters of the AGP were also optimized similarly.

Comparison with affine gap penalty

We compared the results obtained from using the optimized VGP function with those obtained using the optimized AGP function (Figure 2). The accuracy of the alignments was measured by using the SOV measure (see Methods). In a majority of the cases, the VGP performs better than the AGP. Specifically, for the 238 testing alignments, the new algorithm performed better in 157 cases, whereas it was worse in 47 cases. In 34 cases, the difference in performance was indistinguishable. Although the difference between the average accuracies of the AGP and VGP (81 and 84.5%, respectively) is small, it is statistically significant according to Student's t-test (Marti-Renom *et al*., 2002).

Illustrative examples

In two examples, we compare the results obtained with the VGP and AGP with the structural alignment of a pair of structures (Figure 3). The AGP incorrectly introduces gaps within segments of regular secondary structure. These misalignments

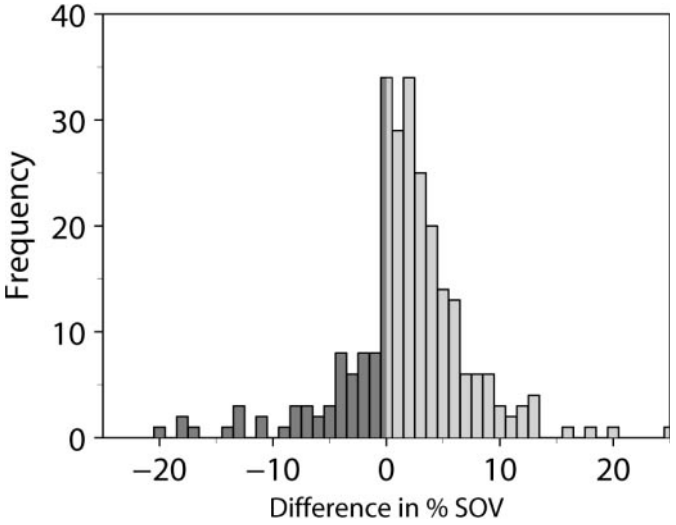

Fig. 2. Distribution of the difference in alignment accuracy between dynamic programming with optimal AGP and optimal VGP. The difference in the alignment accuracy is expressed as a percentage of the SOV. Each point in the histogram corresponds to one alignment in the benchmark. The cases where the structure-dependent VGP function performs better than the AGP are indicated in white.

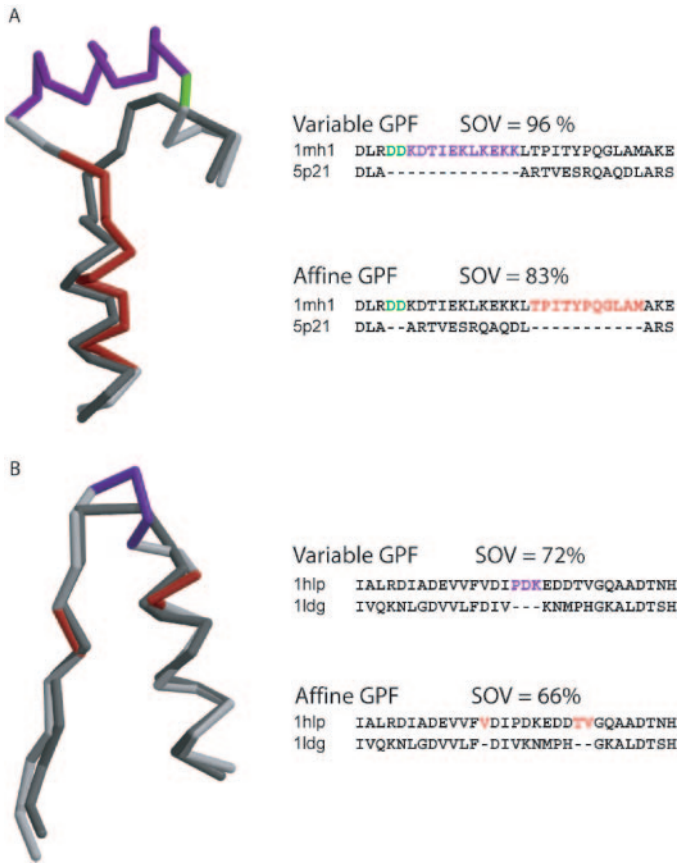

Fig. 3. Sample improvements in alignment accuracy using dynamic programming with a VGP instead of AGP. (**A**) Small G protein (PDB code 1mh1, colored light gray) is superimposed on H-ras p21 (PDB code 5p21, colored dark gray). (**B**) Malate dehydrogenase (PDB code 1hlp, colored light gray) is superposed on L-lactate dehydrogenase (PDB code 1ldg, colored dark gray). Superpositions are according to the SCOP structural alignments. Residues inserted using the AGP and the VGP are colored red and purple, respectively. Residues are colored green if they are insertions according to both alignments. The SOV under 4 Å for whole structures is also displayed.

are corrected when the VGP function is used. The accuracies of the VGP and AGP alignments in comparison with the structural alignment were 96 and 83% in the first example and 72 and 66% in the second example, respectively. The examples also illustrate that the VGP alignment can align residues correctly at the expense of maximizing sequence identity, a common problem of the AGP alignments, especially when the sequences are remotely related (<30% sequence identity).

Discussion

We have described and tested a structure-dependent VGP function for aligning a sequence to a structure. The method relies on a modified dynamic programming algorithm that can use a gap penalty function of any form. The VGP function was constructed to reflect common knowledge about the preferred environment of gaps in structure-based alignments of proteins. First, we penalize the occurrence of gaps within regular secondary structure segments. Second, we also penalize the gaps in buried and straight backbone segments because the cores of structures are usually more conserved than their exposed, floppy loop regions. Finally, we penalize gaps that span long distances in space more than those that span short distances because local changes in structure are more tolerated in evolution than larger changes. The benchmark demonstrates that an optimized VGP function performs better than an optimized AGP function (Figures 2 and 3): The use of the VGP function increases the number of correctly aligned residues from 81 to 84.5% in comparison with the optimized AGP function; this difference is statistically significant according to Student's *t*-test. Moreover, it is possible that further refinement of the functional form of the VGP, enabled by the generality of our dynamic programming algorithm, will yield even more accurate alignments.

Our algorithm is useful in comparative protein structure modeling where a key step is to align the sequence to be modeled to a sequence of a related template structure. Although the gains in terms of the number of correctly aligned positions within a certain distance cut-off appear to be small, the benefits telescope in a large-scale application, such as our comprehensive MODBASE database (Pieper *et al*., 2006). This database stores comparative models for domains in 1.1 million of the 1.8 million unique sequences in UniProt (as at May 2005) (Bairoch *et al*., 2005). If the 2% increase in the number of correctly aligned residues is assumed for all models in MODBASE, the new alignment protocol would result in an additional 7 million correctly modeled residues; for a protein with an average length of 200 residues, this increase in coverage is equivalent to 35 000 newly modeled proteins.

There is still plenty of room for improving sequence–structure alignment accuracy. As the next step, we will utilize the structure-dependent VGP function in the profile–profile alignment (Yona and Levitt, 2002; Edgar and Sjolander, 2004; Marti-Renom *et al*., 2004) based on environment-dependent residue substitution tables (Shi *et al*., 2001).

Acknowledgements

Discussion with our group members, especially with Drs Narayanan Eswar and Ursula Pieper, is greatly appreciated. We acknowledge funding by The Sandler Family Supporting Foundation and NIH GM54762, GM62529, DE016274 and GM62529 and also hardware gifts from IBM, Intel and Sun.

References

Altschul,S.F., Madden,T.L., Schaffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) *Nucleic Acids Res.*, **25**, 3389–3402.

Bairoch,A., *et al*. (2005) *Nucleic Acids Res.*, **33**, D154–D159.

Barton,G.J. and Sternberg,M.J. (1987) *Protein Eng.*, **1**, 89–94.

Berman,H.M., *et al*. (2002) *Acta Crystallogr.*, **D58**, 899–907.

Dayhoff,M., Schwartz,R. and Orcutt,B.C. (1978) in Daytoff,M. *et al.* (eds), *Atlas of Protein Sequence and Sructure*, National Biomedical Research Foundation, Washington, DC.

Dewey,T.G. (2001) *J. Comput. Biol.*, **8**, 177–190.

Durbin,R., Eddy,S.R., Krogh,A. and Mitchison,G. (1998) *Biological Sequence Analysis: Probilistic Models of Proteins and Nucleic Acids*, Cambridge University Press, Cambridge.

Edgar,R.C. and Sjolander,K. (2004) *Bioinformatics*, **20**, 1301–1308.

Goonesekere,N.C. and Lee,B. (2004) *Nucleic Acids Res.*, **32**, 2838–2843.

Henikoff,S. and Henikoff,J.G. (2000) *Adv. Protein Chem.*, **54**, 73–97.

Lesk,A.M., Levitt,M. and Chothia,C. (1986) *Protein Eng.*, **1**, 77–78.

Madhusudhan,M.S., Marti-Renom,M.A., Eswar,N., John,B., Pieper,U., Karchin,R., Shen,M.-Y. and Sali,A. (2005) in Walker,J.M. (ed.), *The Proteomics Protocols Handbook*, Humana Press Inc., Totowa, NJ.

Marti-Renom,M.A., Stuart,A.C., Fiser,A., Sanchez,R., Melo,F. and Sali,A. (2000) *Annu. Rev. Biophys. Biomol. Struct.*, **29**, 291–325.

Marti-Renom,M.A., Ilyin,V.A. and Sali,A. (2001) *Bioinformatics*, **17**, 746–747.

Marti-Renom,M.A., Madhusudhan,M.S., Fiser,A., Rost,B. and Sali,A. (2002) *Structure (Camb.)*, **10**, 435–440.

Marti-Renom,M.A., Madhusudhan,M.S. and Sali,A. (2004) *Protein Sci.*, **13**, 1071–1087.

Mott,R. (1999) *Bioinformatics*, **15**, 455–462.

Moult,J. (2005) *Curr. Opin. Struct. Biol.*, **15**, 285–289.

Needleman,S.B. and Wunsch,C.D. (1970) *J. Mol. Biol.*, **48**, 443–453.

Pieper,U., *et al*. (2006) *Nucleic Acids Res.*, **34**, D291–D295.

Qian,B. and Goldstein,R.A. (2001) *Proteins*, **45**, 102–104.

Rees,D. (1987) *Foundation of Statistics*, Chapman Hall, New York, NY.

Sali,A. and Blundell,T.L. (1993) *J. Mol. Biol.*, **234**, 779–815.

Sali,A. and Overington,J.P. (1994) *Protein Sci.*, **3**, 1582–1596.

Sellers,P.H. (1974) *SIAM J. Appl. Math.*, **26**, 787–793.

Shi,J., Blundell,T.L. and Mizuguchi,K. (2001) *J. Mol. Biol.*, **310**, 243–257.

Smith,T.F. and Waterman,M.S. (1981) *J. Mol. Biol.*, **147**, 195–197.

Yona,G. and Levitt,M. (2002) *J. Mol. Biol.*, **315**, 1257–1275.

Received October 31, 2005; revised December 2, 2005; accepted December 2, 2005

Edited by Valerie Daggett