doi:10.1016/S0022-2836(03)00309-7 J. Mol. Biol. (2003) 328, 593–608

JMB

Available online at www.sciencedirect.com

SCIENCE DIRECT®

In silico Protein Recombination: Enhancing Template and Sequence Alignment Selection for Comparative Protein Modelling

Bruno Contreras-Moreira, Paul W. Fitzjohn and Paul A. Bates*

Biomolecular Modelling Laboratory, Cancer Research UK London Research Institute Lincoln's Inn Fields Laboratories 44 Lincoln's Inn Fields London WC2A 3PX, UK

*Corresponding author

Comparative modelling of proteins is a predictive technique to build an atomic model for a given amino acid sequence, on the basis of the structures of other proteins (templates) that have been determined experimentally. Critical problems arise in this procedure: selecting the correct templates, aligning the query sequence with them and building the non-conserved surface loops. In this work, we apply a genetic algorithm, with crossover and mutation, as a new tool to overcome the first two. *In silico* protein recombination proves to be an effective way to exploit the variability of templates and sequence alignments to produce populations of optimized models by artificial selection. Despite some limitations, the procedure is shown to be robust to alignment errors, while simplifying the task of selecting templates, making it a good candidate for automatic building of reliable protein models.

Keywords: protein structure prediction; comparative modelling; template selection; alignment errors; genetic algorithm

Introduction

Globular proteins with similar amino acid sequences have similar structures. The exponential function relating sequence similarity to structural divergence was first derived by Chothia and Lesk in 1986.[1] This allows modelling proteins (queries) at the atomic level based on structural knowledge of their homologues (templates). In addition, the accuracy of a model can be estimated from this function. For proteins around 90% identical in sequence, the root-mean-square deviation (rmsd) for the backbone of their superimposed core is expected to be below 0.5 Å. If the sequence identity drops to 30%, the expected rmsd is around 4 Å. There are many exceptions to this rule but, in general, it is a good way to estimate the accuracy of models, as experiments such as EVA demonstrate.[2,3]

Abbreviations used: rmsd, root-mean-square deviation; SCOP, structural classification of proteins; PDB, Protein Data Bank; CASP, critical assessment of techniques for protein structure prediction; SSE, secondary structure elements; pssm, position-specific scoring matrix; *e*-value, expectation value.

E-mail address of the corresponding author: paul.bates@cancer.org.uk

http://www.bmm.icnet.uk

Apart from this natural limitation for protein structure prediction based on a single template, modellers still have several pitfalls to face: searching for and selecting templates, sequence alignment between query and template, side-chain placement and loop building. Many different ideas have been applied to each of these tasks, but as the Fourth Critical Assessment of Techniques for Protein Structure Prediction (CASP4) meeting concluded,[4] the first two remain the most critical.

There are several methods and computer programs for protein comparative modelling, such as MODELLER,[5,6] SwissModel,[7] 3D-JIGSAW,[8] FAMS[9] and EsyPred3D,[10] but they all share a common generic algorithm.[3,11] The procedure can be summarised as follows.

Step 1. Template search and selection based on sequence similarity to the query.

Step 2. If there are several possible templates, calculate a multiple structural alignment.

Step 3. Align the query sequence to the single or multiply aligned templates.

Step 4. Construct a model for the core of the query structure based on the alignment.

Step 5. Build non-conserved loops connecting secondary structure elements (SSE).

Step 6. Refine the complete model.

0022-2836/03/$ - see front matter

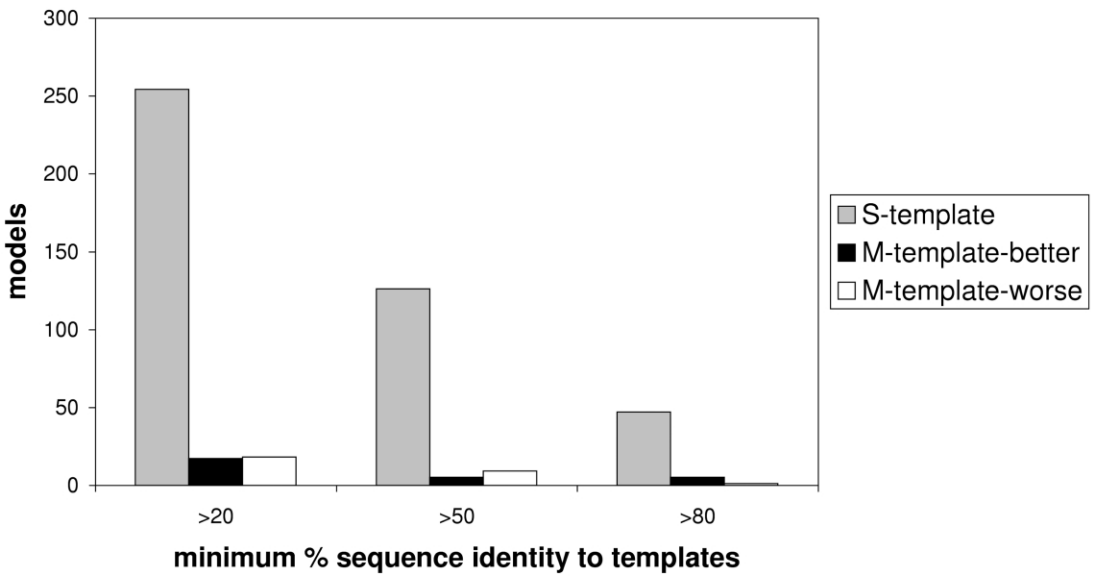

Figure 1. Single *versus* multiple template performance for comparative modelling. The program 3D-JIGSAW was used to build models on the basis of structural alignments between query and template(s). This eliminates potential sequence alignment errors. Models were built using between one and five templates from the same SCOP family, with sequence identities ranging from 80–100%, 50–100% and 20–100% (*X*-axis). The *Y*-axis corresponds to the total number of models in each bin. Multiple-template models are compared to the best single-template model, and are considered significantly better or worse if their rmsd values after superimposition upon the experimental structure are at least 0.6 Å different.

The main problems highlighted in CASP4 affect the first three steps. Steps 4–6 will only improve the quality of the model if minimal errors occur at the initial stages.

The aim of the current work is to design a modelling procedure that automatically minimizes errors during the steps 1–3. This problem can be described as solving a combinatorial optimization problem in template and alignment space. Because genetic algorithms have been applied successfully for optimization problems[12] by mimicking chromosomal mutation and recombination, we chose this algorithmic approach. In recent years, genetic algorithms have been used by many groups to study protein folding, protein docking and alignment optimization.[13–18] Furthermore, recent experiments have shown the possibility of generating new, viable and useful protein folds *via* protein fragment shuffling.[19,20]

Here, a genetic algorithm is applied to comparative modelling. We call our method *in silico* protein recombination, as it is a way to combine different templates and alignments. It simulates artificial genetic selection on a population of single-template models created from different templates and different sequence alignments per template. Fitness for each member of the population is defined as a simple function of solvent accessibility and residue–residue pair potentials on a simplified side-chain representation. Due to the relatively long computational time required, the number of alignments per template used throughout this work had to be small, in the range of five to ten (see Materials and Methods). As discussed later, this new method permits the identification of more favourable alignments and tertiary structure conformations.

Results

In this section some experiments that led us to test our new approach are described. In particular, we concentrated on the first three steps of the generic comparative modelling procedure; template selection, query to template alignment and single/multiple template modelling. For this, we use our program 3D-JIGSAW, which has been shown to be competitive in previous CASP editions[8,21] and in a continual online assessment of comparative modelling (EVA†). We do not consider that the results presented here are significantly sensitive to the choice of a particular comparative modelling program, since these models are used only to build the initial population for the recombination procedure (see below). Only after presenting this preliminary analysis can the value of the protein recombination experiments be appreciated.

Single *versus* multiple template modelling: what is the advantage?

In theory, due to the greater coverage of conformational space, using more than one template should generate a model that is more accurate than any of the individual templates. However,

† http://cubic.bioc.columbia.edu/eva

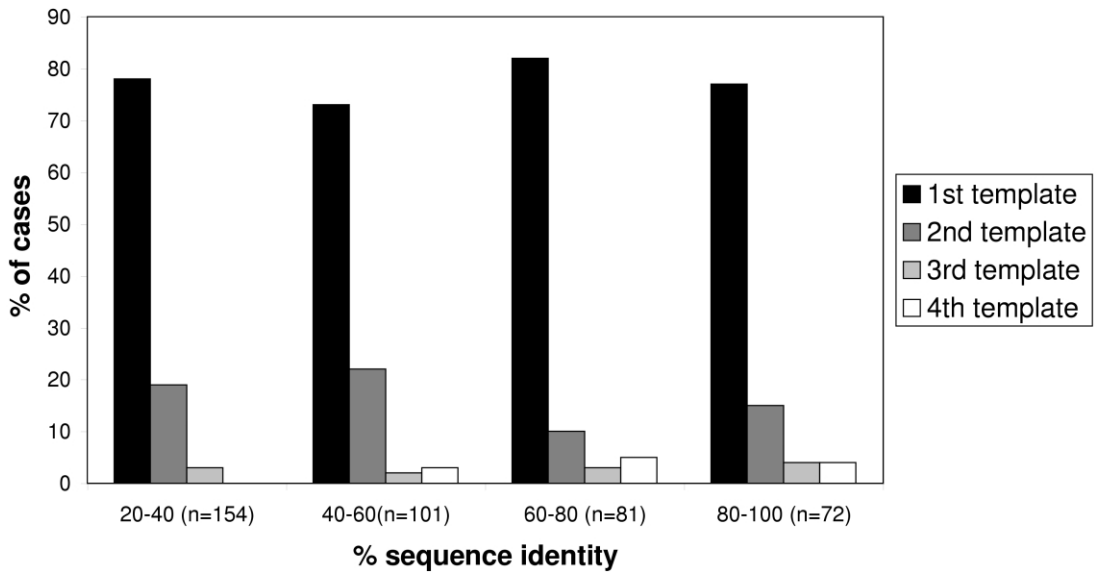

Figure 2. Up to four potential templates to build a model are ranked according to sequence identity with the query sequence. A model is constructed from each and then compared to the experimental structure. The *X*-axis shows four sequence identity bins between query and template. The *Y*-axis states the percentage of cases in which the first, the second, the third or fourth-ranked templates yield the best model. Interestingly, around 25% of the time, the highest ranked template does not produce the best model. This is observed along the whole sequence identity range.

CASP4 showed that only very occasionally were multi-template models more accurate than single-template models. The reasons for this are the choice of templates and sequence alignment errors.[4,21,22] As the limited number of targets for comparative modelling in CASP4 precluded definitive conclusions, we performed a simple experiment using 3D-JIGSAW. (1) From each of 271 SCOP families,[23] one protein domain (query) was selected randomly to be modelled, the remainder were used as potential templates. Two different models were constructed; one using the template with the highest level of sequence identity with the query and the other using up to five templates. Each query was aligned with its respective template(s) on the basis of their known atomic coordinates, in order to minimize alignment errors. (2) Both models were compared to the experimental structure.

From the results presented in Figure 1, it can be concluded that our current methodology is not taking full advantage of the possibility of using several templates to build comparative models. In general, multiple-template models are no better than their corresponding ideal single-template models and, indeed, can be considerably worse. A minimum difference of 0.6 Å was used to compare rmsd measures between models. This value was chosen because it has been found to be the maximal backbone variability observed either in protein structures solved under different crystal lattices, or comparing NMR and crystallographic structures.[24,25] Only in a marginal proportion of cases were multiple-template models found to improve over the ideal single-template model (maximum improvement observed was 1.66 Å), showing no preference for any region in the sequence identity range. On the other hand, multiple-template models could be significantly worse (maximum deviation observed was 1.92 Å) with a comparable frequency.

Because these results are similar to those obtained in CASP4 for all the participant methodologies, it is tempting to think this is actually a limitation of the generic method itself. In other words, single-template models, on average, appear more accurate, provided that the optimal template can be identified. Errors in the template(s) alignment with the query may be disregarded as the reason for this, because the models in the experiment had been built from structural alignments. The next step in the analysis was then to investigate ways to classify the available templates.

How to select templates

Following the principle of "similar sequences have similar folds", quantified by Chothia & Lesk,[1] it seems reasonable to rank the possible templates to build a model, using their sequence identity with the query. Indeed, one of the more successful programs for comparative modelling, SwissModel,[7] weights the contribution of each template to the final model using exactly this criterion. This rule has been used for the experiment described above, but only after it was decided to test its validity. For this, we simulated the construction of single-template models for 392 SCOP domains. Up to four different models for each were constructed using different templates. Each set of models was then compared to the experimental structure, and the results are shown

in Figure 2. This trivial experiment allowed us to estimate the difficulty of selecting templates. Perhaps surprisingly, errors in choosing the optimal template are equally likely for each of the sequence identity ranges used, with a frequency of approximately 25%. If the optimal sequence alignment could be found, sequence identity would indeed be a good template classifier (results not shown), suggesting that alignment errors mask the identification of the best template. Similar difficulties are encountered if templates are ranked according to expectation values, based on similarity scores, as shown in the last subsection of Results. As a consequence, being unable to identify the optimal template routinely forces us to consider multiple templates in model building.

The optimal sequence alignment is not always the best for modelling

As indicated above, probably the most persistent problem in comparative modelling is aligning the query sequence with the template(s). The main information types usually available for these alignments are sequence and secondary structure. Here, we analysed how often the optimal sequence alignment between query and template, calculated through dynamic programming,[26] corresponds to the model with the lowest rmsd from the experimental structure.

Using a simple procedure (see Materials and Methods),[27] five alternative alignments were produced for each of 58 single-template models, with sequence identities with the templates ranging from 15% to 82%. For each of these alignments, a model was constructed and then compared to its corresponding experimental structure. The highest level of sequence identity alignment provided the lowest rmsd model in 42 cases, but the remaining 16 cases would have been modelled more accurately using a suboptimal alignment. These suboptimal alignments have a range of sequence identities with their templates, from 15% to 51%.

These results suggest that suboptimal alignments (and perhaps other alternative alignments) should be considered routinely in model construction rather than relying on the single optimal sequence alignment. Indeed servers such as EsyPred3D[10] try to improve comparative modelling by considering alternative and consensus alignments. Of course, this raises the question of how to identify the best alignment. We have not found rules that help at the sequence level, so we must move to the structure level and search for something such as a simple threading function.

A simple fitness function to compare protein structures

We first tested a simplified representation of proteins, depicting residues as backbone plus side-chain centroid, and scoring the internal packing according to statistically derived atom–atom potentials.[28] We chose these potentials because they explicitly consider backbone to side-chain centroid contacts, thus not accounting for specific rotamers. It is a coarse and relatively quick method to score models. Both these features are important for the subsequent use of these potentials in our model-building procedure (see Results). On its own, this function was not able to correlate energies and rmsd of protein models consistently (results not shown). Because proteins fold in solution, a solvation term[29,30] was added to the fitness function in a ratio of 1:1. This term is the sum of residue solvent-accessible areas (as calculated using the program NACCESS[31]) multiplied by tabulated amino acid solvation free energies.[32] As an initial test to evaluate how efficient this fitness function is, we applied it to the models built in the suboptimal alignment experiment (see above) to identify the best alignment: it correctly identified optimal models in 51 out of 58 cases. Further investigation was carried out to optimize the ability of this function to assess protein conformations and to weight the two terms, but eventually a 1:1 weighting seemed to be at least as good as other linear and non-linear combinations (results not shown).

Protein recombination: a way to combine different templates and different alignments

With the above fitness function, we were then able to try the following modelling approach: to use all available templates and different alignments for each of them with the query, expecting to get an optimized final conformation. The way genetic information is combined in Nature seemed the most appropriate for our purposes, since proteins, like DNA, are linear molecules. As we are applying this mechanism to proteins, it was decided to call this new approach *in silico* protein recombination (see Figure 3). The algorithm is a close analogy to genetic variability generation. The recombination relies on two given protein models being superimposed and a crossover chosen between them at a random point (outside of regular SSE). In genetic terms, each protein would be a sister chromatid. Mutation is a way of generating novel molecular conformations. This is done here by averaging the coordinates of two given models. Although a choice of reasonable recombination and mutation rates is important, the algorithm is critically dependent on the quality of the fitness function; it is, after all, this function that the genetic algorithm seeks to optimize.

Testing an ideal fitness function: limits of the method

It was necessary to show the usefulness of this algorithm by first using an ideal fitness function. In the present context this function is rmsd (see Materials and Methods), since we know beforehand the experimental structure of the proteins

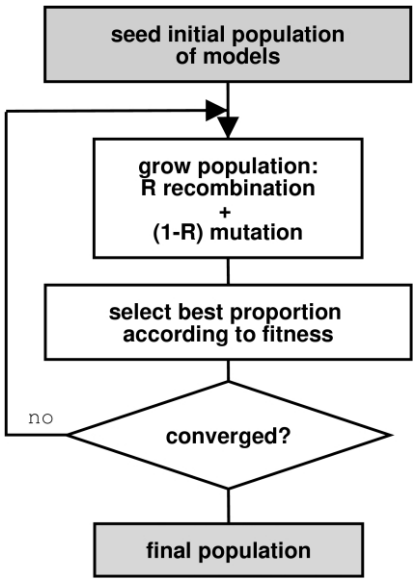

Figure 3. *In silico* protein recombination flowchart. *R* and $1 - R$ are probabilities.

we are trying to model. An experiment was set up to model 163 SCOP domains using their family relatives as templates. Sequence-based alignments were used to build these models (see Materials and Methods). The domains consisted of 32 α, 44 β, 44 α/β and 45 α + β protein folds. The results (Table 1) show that using several templates in this way permits building models that, on average, are not significantly more accurate than the optimal template (improvement of 0.46 Å), but never worse. However, in some cases the improvement is significant (up to 2.33 Å), mainly because of loop choices. For models with no templates with over 40% of sequence identity, the average improvement becomes significant (0.88 Å). From a population point of view, using this algorithm, models in the last generation show a consistent improvement (2.6 Å better than the initial population).

A second important conclusion of this experiment was that mutation does not contribute significantly to the gain in accuracy, as noticed in similar genetic algorithm approaches.[18] Finally, because we use rmsd as a fitness function, this experiment shows that our algorithm could not improve further, regardless of the fitness function we apply.

Table 1. Benchmark of *in silico* protein recombination using rmsd to the experimental structure as fitness function

	Δ Average rmsd (Å)	Δ Best template rmsd (Å)	Generations
A. *Up to 100% identity: N = 163*			
Best	−7.49 (−7.60)	−2.33 (−1.77)	1 (3)
Mean	−2.60 (−2.53)	*− 0.46 (−0.39)*	8 (8)
Worst	*− 0.16 (−0.23)*	*− 0.04 (0)*	15 (14)
B. *Up to 40% identity: N = 50*			
Best	−7.49 (−7.60)	−2.33 (−1.77)	2 (4)
Mean	−2.77 (−2.67)	−0.88 (−0.78)	10 (9)
Worst	*− 0.48 (−0.3)*	*− 0.05 (0.01)*	17 (18)

A, Models using templates of any sequence identity; B, only templates below 40% sequence identity were used. Values in parentheses correspond to simulations using only recombination, otherwise mutation has been applied also. The first column shows the final average population rmsd with respect to the initial rmsd values. The second column shows the evolution of rmsd with respect to the optimal template, had we identified it. Non-significant differences are shown by the use of italics. The last column shows the number of generations needed to reach convergence.

Testing the method to correct alignment errors using the simple fitness function

The next experiment was set up to gain insights into the ability of this method to correct alignment errors using a real fitness function. Eight SCOP domains were selected: two α (d1a03a_ and d1a8h_1; shortened to A1 and A2), two β (d1qfja1 and d2phla1; B1 and B2), two α/β (d1pmt_2 and d1poxa2; C1 and C2) and two α + β (d1pne__ and d1a5r__; D1 and D2) folds. For each of them, models were built using their known experimental structures as templates. Variable patches of the query sequence were shifted randomly one, two, three or four positions with respect to its correct place in the otherwise perfect sequence alignment. Thus, every initial modelling population was composed of partially wrong protein models and was fed into the recombination program. The number of models used for the initial populations was five. Five replications for each of the eight sets were performed. Figure 4 shows that this algorithm is able to recombine models to yield better-alignment models, suggesting that it is robust enough to overcome alignment errors if partially correct alignments are present in the initial population. Again, this reinforces the view that using models constructed from different alignments should result in more favourable protein conformations. A more detailed analysis of this experiment, illustrating a typical protein recombination simulation, is shown in Figure 5, taking d1pne__ as an example. In this instance, after generating an initial population in which every member had serious alignment errors, a recombination experiment spanning over 13 generations converged onto a final population in which members had perfect alignments, with rmsd from the ideal model of 0.8 Å (0.05 Å for the backbone). Crossover points found in the final models are shown in the multiple structural alignment of the initial models (Figure 5A) and in a molecular representation (Figure 5B).

Recombining models built from different templates and alternative alignments

Protein recombination experiments were set up to model the same previous eight SCOP domains (A1 to D2, see above). To build the initial population of models for each simulation we used

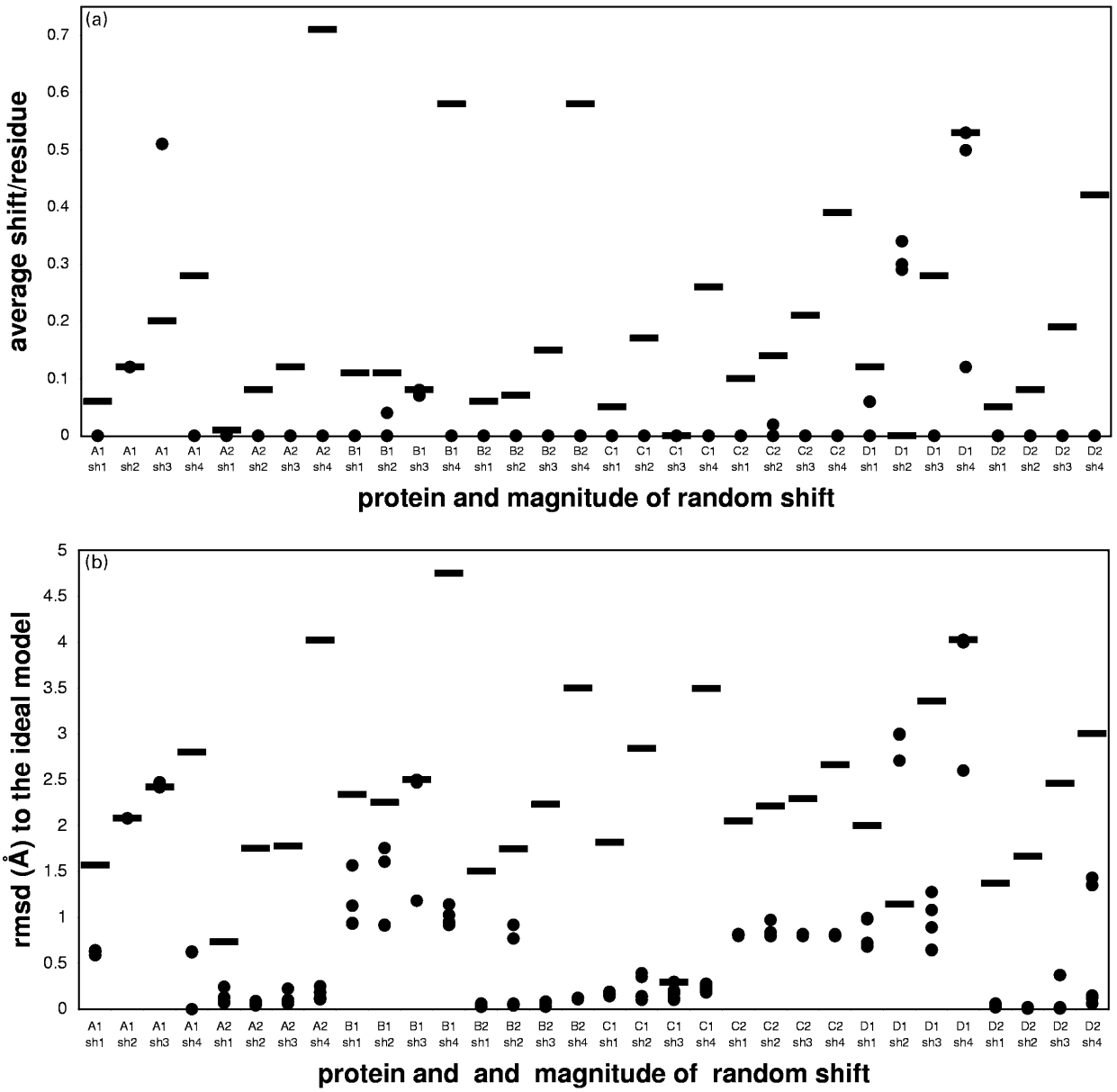

Figure 4. Protein recombination is able to generate optimal alignments and more accurate models from populations of models obtained from randomly shifted template alignments. Eight model populations (for sequences A1, A2, B1, B2, C1, C2, D1 and D2) were created using randomly shifted alignments. For each sequence, four different populations were generated, using shifts of one, two, three and four residues. Finally, each population was recombined five times. Final population averages (marked as periods) for each experiment are shown in the same column. (a) Note that alignment shifts tend to disappear upon recombination with respect to the best initial model (marked as −). (b) At the same time, rmsd from the known experimental structure tend to diminish.

single-template models built from alternative alignments (with the same template) and from several templates in their corresponding SCOP families. The number of models used for their initial populations ranged from 10–102. In addition, to analyse how different recombination runs for the same input can be, each initial population was used to start ten independent recombination processes. The results are shown in Figure 6. The picture arising from this experiment is that alignment shifts are minimized upon recombination and can go beyond the best initial model in the population. At the same time, final populations average rmsd values are comparable to the best initial model seeded. Furthermore, these simulations pointed out the importance of running the same population of models through recombination several times to exploit the capability of the method fully. Since this is a population-based method, a population answer should be provided. This can be achieved by running independent simulations on the same input. Analysis of these experiments showed that, on average, rmsd between independent runs are not significant, so

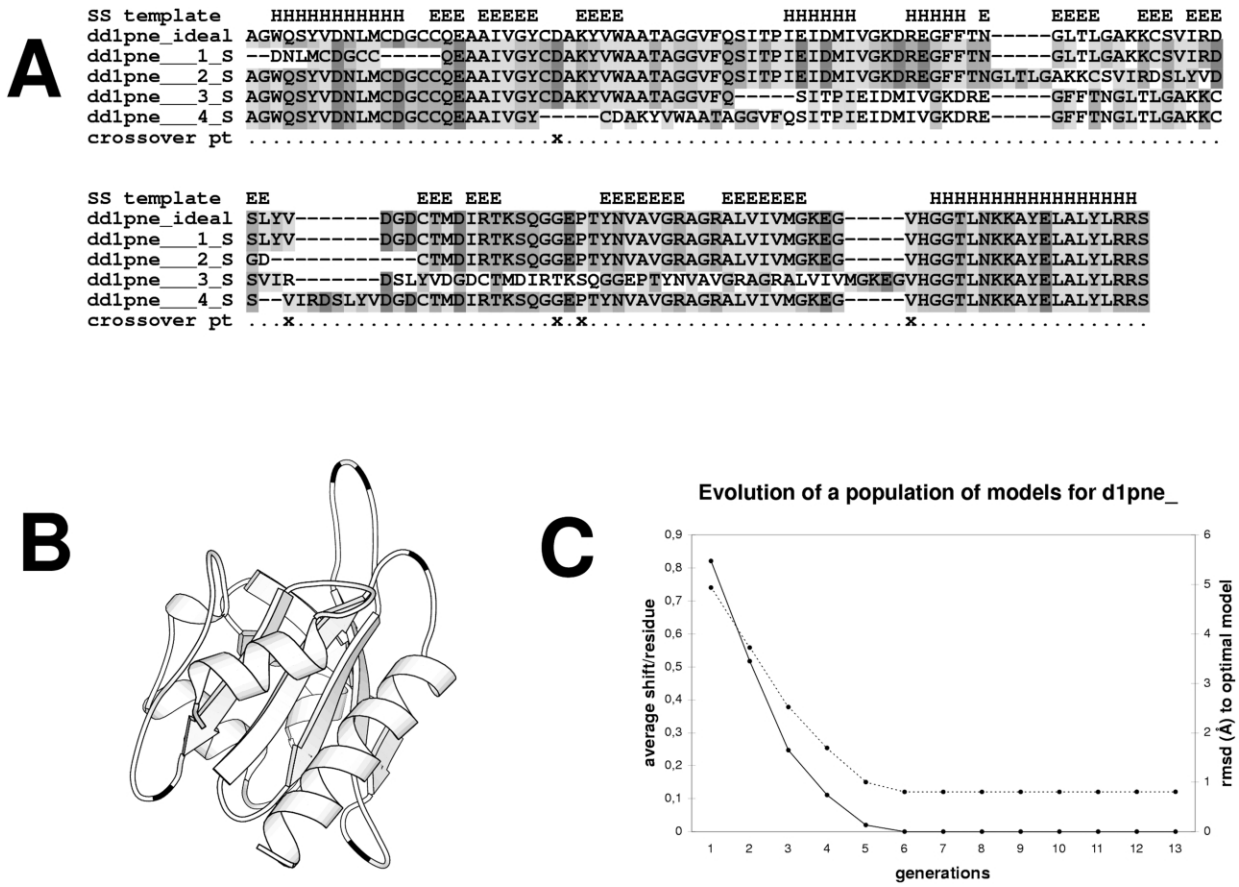

Figure 5. Protein recombination experiment in detail. Four shifted-alignment models (1_S, 2_S, 3_S and 4_S) for d1pne__ (cow profilin, 1PNE) were generated. Their rmsd from the non-shifted conformation (ideal model) ranged from 2.3 Å to 9 Å and their average alignment shift per residue from 0.16 to 2.68. The structural alignment of these initial models with the known structure of d1pne__ is shown in A. The top row shows the STICK-assigned secondary structures for the template (H for α-helix and E for β-sheet). The x in the bottom row mark frequently observed cross-over points in models in the final population, displayed in space in B. The experiment rmsd and shift profiles are shown in C, After 13 generations the simulation converged and the final population has an average rmsd of 0.8 Å from the experimental d1pne__ structure (only 0.05 Å from the backbone) and no alignment shift.

they could be considered as ensembles of protein conformations, analogous to NMR structures.

Large-scale benchmark of the method

To conclude the benchmark of the method, a large-scale protein recombination experiment was tested on a set of 130 SCOP domains (27 α, 38 β, 26 α/β and 39 α + β protein folds). Domains were modelled using their family relatives as templates and only one sequence alignment per template. Due to computing time limitations only, one independent run was performed for each of the 130 populations. Despite this handicap, the algorithm produces final populations of models that are comparable to the best initial model (see Figure 7 and Table 2) and that are consistently better than the initial population (around 1 Å). In 92% of the cases (89% for models built from templates 40% or less identical in sequence), final population models are not significantly different from the best initial model. However, as expected from the reference experiment, using rmsd as a perfect fitness function, no improvement is seen beyond this limit. The good news is that the algorithm converges onto protein conformations close to the optimal model, suggesting that our method sorts templates better than sequence identity measures and that there is no need to select templates for modelling. The bad news is that more favourable protein conformations, according to the fitness function, do not always correspond to lower rmsd states (see Figure 8B for an example) and that, on average, the algorithm is not taking full advantage of the expected possibilities of combining different templates. To some extent this was predictable, since only one alignment per template was used for this experiment, making the method comparable to 3D-JIGSAW in that respect. A more detailed analysis follows.

Improvements in accuracy

After recombining 130 sets of single-template models, only three final populations have conformations significantly better than the optimal

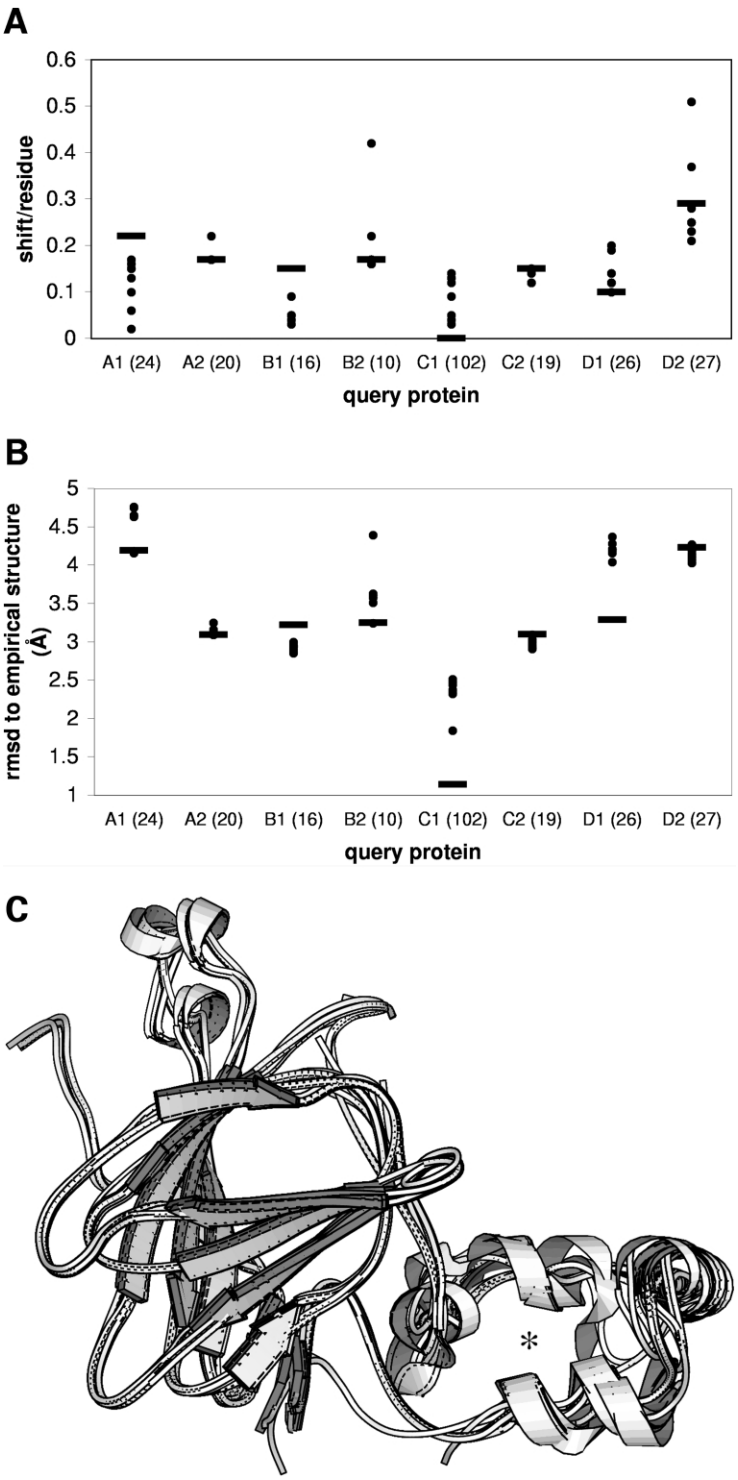

Figure 6. Alternative alignments and different templates improve the performance of protein recombination. For each of eight protein model sets, ten recombination replications were carried over and their final population averages are shown in the same column. (A) Note that alignment shifts tend to diminish upon recombination with respect to the best initial model (marked as −). (B) On the other hand, rmsd improvements are not equally consistent. (C) Structural similarity of eight final population models for protein B2, with the range of rmsd shown in B. Note that the precise region where the major differences are found (*) is a small, flexible, helical subdomain interacting closely with the next monomer when this seed storage protein (2PHL) hexamerizes.

template model (over 0.6 Å of rmsd difference). Inspection of these models and others with minor improvements (30 recombination experiments) shows that the improvements come from choosing alternative surface loop conformations or from small subdomain movements. Figure 8A shows one example in which the final population in the experiment achieved an rmsd from the known structure of the protein that is significantly better (0.89 Å) than the model built using the best template. In this case the improvement comes from the relative orientation of two subdomains from different templates that have been arranged together. Nevertheless, it is clear that, on average, models in populations do not improve their rmsd over that of the optimal template model. The value of this method is that it converges consistently around the optimal template's conformations, and these cannot be identified routinely.

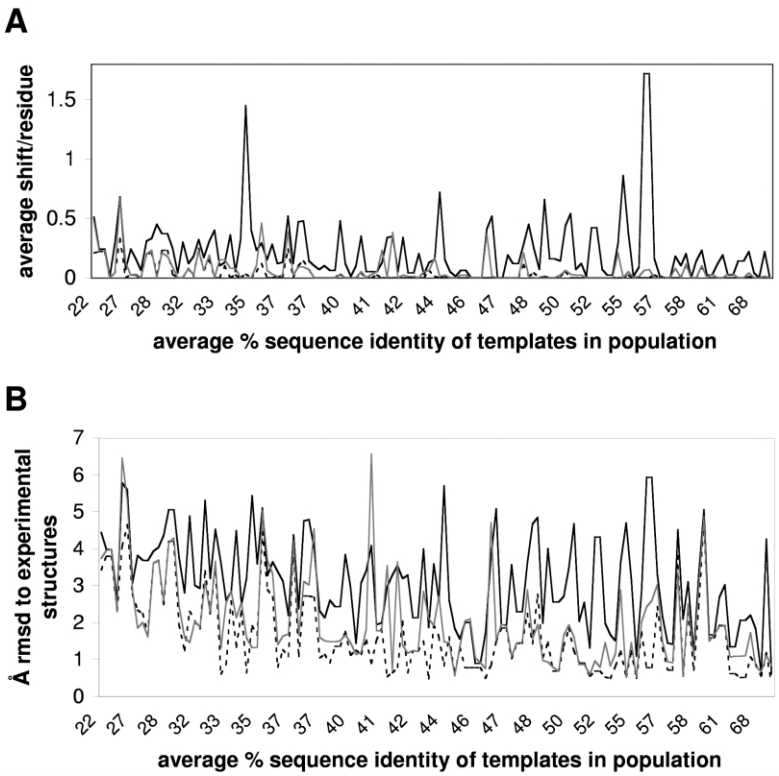

Figure 7. Performance of *in silico* protein recombination in a set of 130 unique experiments designed to model SCOP domains. Each model comes from a single sequence-aligned template. (A) Average population alignment shift measures are compared at the beginning (black continuous line) and when the algorithm converges (grey continuous line). Final populations of models are significantly better than initial (see Table 2) and the degree of improvement is limited by the best initial model (black broken line) had we known it beforehand. (B) Average population rmsd from experimental structures for each SCOP domain is compared at the beginning and at the end of each experiment. Final population rmsd values are often over the best initial model, but differences are not significant in 120 out of 130 experiments (see Table 2).

Improvements in alignments

As a result of this experiment, it may be concluded that populations improve their average alignment shift (with respect to their structural alignment) through rounds of fitness selection and recombination. On average, this improvement is about 0.16 shift per residue (see Table 2), but the ceiling of this improvement is again usually dictated by the optimal template model. Figure 9 shows how observed improvements in population energies correlate to average alignment shifts and rmsd changes through recombination experiments. A linear correlation between energy improvement and alignment shift change is found (Figure 9A), but the interdependency between energy evolution and rmsd change (Figure 9B) is less clear, and can be approximated only tentatively by a logarithmic function.

Benchmark including PSI-BLAST alignments

To compare our results to those obtained with a standard alignment program, PSI-BLAST,[33] we reinvestigated the same test set of 130 SCOP domains. This time only templates found by PSI-BLAST, and that were less than 40% identical with

Table 2. Benchmark of *in silico* protein recombination using our simple fitness function

	Δ Average rmsd (Å)	Δ Best template rmsd (Å)	Δ Average shift (shift/residue)	Δ Best template shift (shift/residue)	Generations
A. *Up to 100% identity: N = 130*					
Best	−4.17	−0.88	−1.66	−0.18	11
Mean	−1.06	*0.4*	−0.16	0.02	24
Worst	2.47	5.66	0.17	0.37	30[a]
B. *Up to 40% identity: N = 44*					
Best	−4.13	−0.88	−1.41	−0.18	12
Mean	−0.98	*0.24*	−0.2	0.05	25
Worst	0.67	2.37	0.17	0.44	30[a]

A, Models using templates of any sequence identity; B, only templates below 40% sequence identity were used. The first column shows the final average population rmsd with respect to the initial rmsd values. The second column shows the evolution of rmsd with respect to the optimal template, had we identified it. Non-significant differences are shown by the use of italics. The third column shows the final average alignment shift with respect to the initial population. The fourth column highlights the same value now with respect to the best template. The last column shows the number of generations needed to reach convergence. Overall, in 92% of the simulation experiments the final population has an average rmsd from the experimental structure comparable to the model built from the best template, meaning that this method consistently identifies the best templates. If only the 40% subset is considered, the figure drops slightly, to 89%.

[a] Maximum generations allowed.

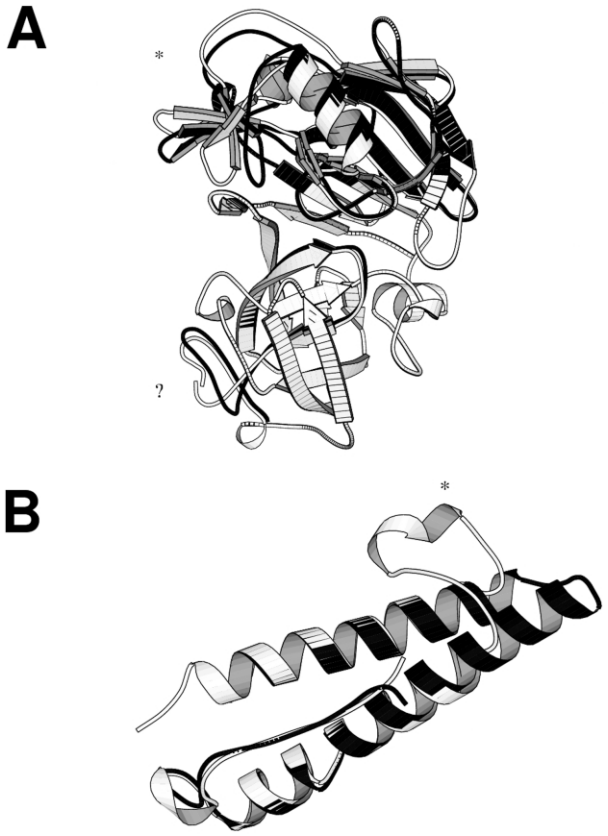

Figure 8. Limitations of the algorithm. Global rmsd improvements come usually from surface loop movements (these are intrinsically flexible anyway) or small subdomain movements, as can be seen (A) in the experiment to model d1apr__ (mould acid protease 2APR) from a population of 11 models built from different templates from the same SCOP family. The final population model is depicted in white, while the best initial model is shown in black (* points to the main differences observed comparing the two models and ? shows a broken loop, a common side-effect of protein recombination). The worst rmsd result obtained in our protein recombination benchmark is shown in B, where it was attempted to model d1dt0a1 (superoxide dismutase N-terminal domain in 1DT0) from an initial population of eight models. The simulation yields a final population rmsd of 5.35 Å while the optimal template model (shown in black) is only 0.89 Å away from the known experimental structure. In this particular example, the long loop (*) is taken from a template (1MNG) whose crystallographic contacts bent the helical bundle.

the query sequence, were used (see also Materials and Methods). Alignments were taken directly from the program's output and subsequent models built using 3D-JIGSAW. These were added to models built using the same templates, but with our own alignments that consider secondary-structure information. The aim of the experiment was to compare the final population of recombined models to the PSI-BLAST-based model constructed from the alignment with the best *e*-value. The first observation from this experiment is that only 54 out of 130 domains can be modelled within these constraints, since templates for the remaining could not be found using default parameters. On this reduced dataset, recombined populations of models tend to be, on average, 0.51 Å closer to the corresponding experimental molecular structure than the best *e*-value PSI-BLAST-based model. More importantly, the corresponding difference in alignment shift was, on average, 0.42 shift/residue better than the PSI-BLAST model. However, in three cases, the recombination protocol did not improve beyond the PSI-BLAST alignment; indeed the original PSI-BLAST aligned models had better agreement with the experiment in some exposed loops. Again, these results suggest that further improvements can be made to the energy function.

This experiment suggested that simply taking the best *e*-value, and associated template, from a standard PSI-BLAST output, does not necessarily produce the best model. On average, in these 54 examples, models built from the best *e*-value alignment were 0.81 Å worse than the best models built from the ensemble of templates found by PSI-BLAST. In other words, *e*-values are not necessarily a good indication of how good a model would be, as shown in Figure 2 for sequence identity. This observation holds for alignment accuracy, since the best models in terms of *e*-value are, on average, 0.58 shift/residue worse than the corresponding best model.

Reinvestigating the fitness function

Finally, to investigate our fitness function when applied to recombination experiments, these 54 populations of models were taken to further assess the contribution of the solvation term. This was done by recombining these populations with and without this term in the energy function. The comparison of these simulations provides a clear conclusion: inclusion of the solvation term yields better recombinant models in terms of deviation to the experimental structures and alignments shift in 22 out of 54 domains; the remainder are very similar. On average, selecting models without the solvation term yields models that are 0.4 Å worse than those selected including it. Alignments are further displaced by an average of 0.05 shift/residue.

Discussion

The results presented here provide insights into two recurrent problems in protein comparative modelling; selecting templates and alignment errors. The novel methodology proposed here deals with both simultaneously and, despite some deficiencies, it is found to be robust to alignment errors. It classifies possible protein conformations confidently for a given sequence on the basis of its homologous partners in the structural database, the templates. These two features are crucial to automation of the construction of comparative

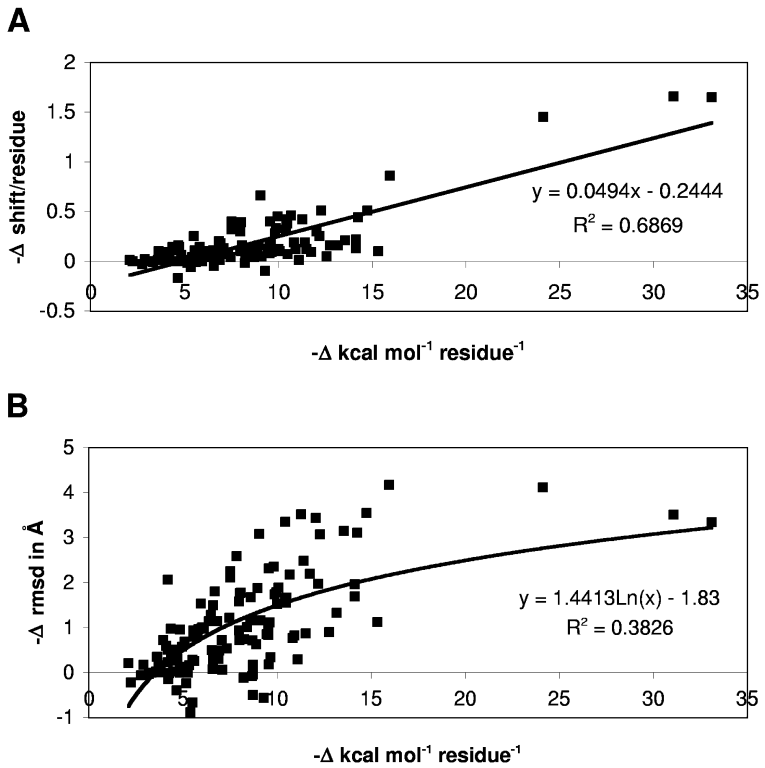

Figure 9. Correlations between energy improvements in populations and alignment and rmsd improvements calculated on data from 130 recombination experiments. (A) A linear correlation is found for the change in average alignment shift, suggesting that it could be predicted, to some degree, from experimental energy profiles. (B) The correlation to rmsd is weaker and is modelled with a logarithmic function only tentatively, suggesting that it would be of little value to predict rmsd improvements from energy profiles.

models. Nevertheless, comparison of the fitness function with the ideal suggests that further improvements can be made to this function. Some limitations and applications of this algorithm are discussed below.

Applications

As shown in the analysis of the results, the method presented here improves the alignment accuracy of protein comparative models and avoids the step of selecting templates, since models from all possible templates can be used. If these models are to be used as guides for site-directed mutagenesis experiments, one of the most popular applications,[11] alignment accuracy is essential to target the correct residues. Comparative models have been applied to fit protein structures into electron microscopy density maps of single molecules or supramolecular complexes,[34–37] and alignment accuracy is therefore important to place the corresponding protein parts into the experimental data.

A different application of modelling, at the population level, would be to gain insights into fold flexibility within a given molecule or even across families, because members of the same population of models can have geometrical differences that cannot be penalized at the level of fitness. This could simply be pointing out the weakness of the fitness function used, but recent papers,[30,38] using different functions and different approaches, such as the Metropolis rule, propose that sequence or structure ensembles represent the nature of a given protein fold more faithfully.

The most important feature of this methodology is its ability to recover alignment errors and to generate different alignments from those contained in the initial population. This could be used to combine comparative models obtained from different sources, templates and alignments to get, not a consensus answer (something other programs already do[39]), but a model close to the optimal template that could correct alignment errors found in the initial population. Indeed, this feature has been confirmed by the relatively promising results obtained by our group using protein recombination in CASP5, particularly within the remote homology section, where alignment errors are more frequent (B.C.M. *et al.*, unpublished results).

Limitations

The presented algorithm has several limitations, the most obvious being the fitness function. Improvements to it will be translated into improvements of the algorithm performance, within the limits defined in our benchmark using an rmsd function as a way to calculate fitness. This means that the algorithm can potentially take advantage of better fitness functions found by the community in the future or those already described in the literature.[29,30,40] However, better functions typically require more computing time, which may limit their practical applicability. In addition, because

the algorithm creates new protein conformations every generation by "cut and paste", if finer energy functions were used, it would be necessary to minimize protein conformation energies every generation, adding yet more computational overhead to the process. The fitness function used for this work was chosen as it is fast to calculate at the price of being less accurate. This has the benefit that population members need not be minimized every generation. Despite this, protein recombination experiments can still last for hours in a worst-case scenario (see Materials and Methods). As a consequence, in a practical situation, models generated by *in silico* protein recombination often need to be minimized, particularly to fix broken loops. In general, the energy function used and the run-time checks are sufficient to produce models with minor stereochemical problems that can be fixed with a subsequent full-atom minimization algorithm.

The second limitation of the method is the search for meaningful alternative alignments to the modelling templates. We have shown the ability of the method to recover from some alignment errors and to improve the population alignment accuracy, but the condition is that partially correct alignments should be present in the initial population. If all the initial alignments for, say, helix1 are wrong, the method would not be able to provide an accurate conformation for that part of the protein. This suggests that models used for recombination experiments should cover different reasonable alignment possibilities. Unfortunately the total number of possible sequence alignments is vast and no hint can be given about the minimal alignment set required to solve the problem. Sub-optimal alignment strategies, like that used in our experiments,[27] and different alignment procedures could be used, since it is accepted that different sequence alignment tools usually give different answers to the same non-trivial alignment problem and often each of them would give optimal alignments in particular cases but not in others.[41]

Finally, the stochastic nature of the algorithm implies that slightly different answers for the same input can be obtained. This can be utilized to provide useful information concerning fold flexibility, as discussed above, but would of course require additional computing time.

The role of mutation

One of the findings of this work is the secondary role of mutation, compared to recombination, in generating useful conformation variability. This would, in theory, undermine the capacity of the method to generate novel protein conformations, substantially different from any of the templates used. Of course, this is related to the way the mutation mechanism is implemented, and because the current method is simply an averaging procedure, with no attempt to correct generated distorted side-chains, we believe it is possible to increase the contribution of mutation. It would imply quality checks after averaging or, as with SWISS-MODEL,[7] averaging only the C^{α} atoms and then reconstructing the rest of the residue.

To test if variability generated by other means could improve the performance of the method, a protein recombination experiment was done in which the original sets of initial models were used to generate extra compatible protein conformations using the method CONCOORD.[42] The results (not shown) were not significantly different, so we concluded that mutation, in this context and with this fitness function, is secondary to recombination. Similar observations have been made in related contexts.[18]

Crossover and secondary structure elements

An important feature of the method is the choice of crossover points between models. In this algorithm, crossover is permitted to occur only out of regular SSE, as defined by STICK,[43] a program that assigns secondary structure states on the basis of vectors that represent the topology of the fold. The reason for this, is that protein geometry would otherwise be distorted seriously, requiring additional efforts to reconstruct reasonable conformations. This was avoided for strictly practical reasons and there is no reason to believe that genetic recombination, to which this algorithm is analogous, occurs only outside of DNA regions coding for regular SSEs.

Conclusion

The method presented here is a novel way to explore the space of sequence alignment and template variability simultaneously for comparative modelling applications. In spite of some limitations, such as the small number of alternative sequence alignments used and relatively high computing requirements, the procedure is found to be robust to alignment errors, making it an attractive tool for automatic model construction. The method is capable of providing compatible conformations for the same sequence. Finally, this algorithm would benefit from future improvements to sequence alignment and model building techniques, and of course the growth of the Protein Data Bank.

Materials and Methods

3D-JIGSAW flowchart for building comparative models

The program 3D-JIGSAW[8] builds models in a series of steps.

Step 1. Search for templates.

Step 2. Align template(s) to query sequence using sequence and secondary structure information.

Step 3. Trim alignments to exclude gaps from secondary structure elements (SSE).

Step 4. Take aligned SSEs from templates and look for loops to connect all the possible combinations of SSE along the query sequence.

Step 5. Mutate sequences to the actual query sequence and add rotamers for each side-chain.

Step 6. Mean-field selection of SSE backbone fragments and side-chains.

Step 7. Fix breaks and minimize energy of the complete model.

For the experiments described here, step 1 is bypassed.

Protein test sets from SCOP

For every experiment described here, protein families from SCOP 1.55[23] were selected randomly from the four major classes (337 α, 276 β, 374 α/β and 391 α + β protein families). Only a non-redundant fraction (90% sequence identity cut-off) of protein domains in each family, according to the ASTRAL database,[44] was considered.

To benchmark *in silico* protein recombination, using the simple fitness function explained below, the following SCOP domains were selected as query proteins to be modelled using proteins in the same family as templates (27α, 38β, 26 α/β and 39 α + β): d1pbk__(4), d1pama2(7),d1pne__(6),d1poxa2(3),d2phia_(16),d1pina2(4), d1pvxa_(6),d1pvaa_(5),d1psra_(6),d1ppn__(13),d1a75a_(5), d1a5da2(9),d1a25a_(5),d1a33__(6),d1a03a_(6),d1a0aa_(4), d1a0ca_(3),d1a1s_1(4),d1a81a1(15),d1ad3a_(2),d1adwa_(9), d1ae7__(16),d11bga_(8),d2abl_2(14),d2act__(13),d1acz__(7), d1qaua_(6),d2aaib2(8),d2aaib1(7),d1an8_2(6),d1an4a_(4), d1qnna2(10),d1qnga_(8),d1qo8a3(3),d1aoza3(2),d1aoa_2(3), d1aoga1(7),d1alo_3(5),d1allb_(11),d1alla_(11),d1ala__(9), d1qlca_(6),d1qk1a1(4),d1qkka_(9),d1qh7a_(6),d1aisa2(7), d1aisa1(5),d1ain__(9),d1aw0__(5),d1aw1a_(8),d1awpa_(3), d1awca_(4),d1qpca_(5),d2apr__(11),d1qqya_(8),d1qqka_(5), d2ay1a_(6),d1ayaa_(16),d1b26a2(2),d1b2pa_(5),d1b06a2(10), d1b1xa1(8),d1b8za_(3),d1bg3a3(3),d1bg0_1(5),d1be9a_(4), d2bb2_1(9),d1bb9__(15),d1rbla2(5),d1rblm_(5),d1bc4__(9), d1blxb_(4),d1bla__(5),d1bjwa_(6),d1bkja_(3),d1bkb_2(2), d1bh6a_(6),d1bhda_(3),d1bwva2(5),d1bwya_(13),d8ruci_(5), d1burs_(5),d2rspa_(4),d1rp1_2(5),d1bzsa_(8),d1bxta2(6), d1bxsa_(2),d1c4zd_(4),d1c1da1(2),d1c9ha_(4),d1cf5a_(6), d1ce7a_(6),d1scha_(4),d1clh__(8),d1ck7a2(8),d1sw6a_(4), d2ctha_(6),d1ste_1(3),d1crka1(5),d1srra_(9),d1crb__(13), d1cs8a_(14),d1csee_(6),d1cpcb_(12),d1cpn__(2), d1cpt__(3), d1cyda_(3),d1d6aa_(7),d1d3ca2(7),d8dfr__(3),d1teha1(6), d1tcda_(8),d1dn2a2(14),d1tnra_(3),d1dot_1(7),d1dlpa2(6), d1dmxa_(3),d1dt0a1(8),d1duvg2(4),d1duxc_(5),d2trxa_(7), d1dssg2(5),d1dsya_(5),d1tx4b_(11),d1e3pa2(2),d1e3ia1(6), d1e1oa1(2), d1u9aa_(4),d1ef5a_(3),d1egza_(4). The number of templates used in each case is indicated in parentheses.

For the detailed analysis presented in Results, only eight SCOP families were considered, two of each class. Each of them contained several templates with a variable degree of sequence identity with the query. They were: d1a03a_ (rabbit calcyclin, 1A03); d1a8h_1 (*Thermus thermophilus* methionyl-tRNA synthetase, 1A8H); d1qfja1 (*Escherichia coli* flavin oxidoreductase, 1QFJ); d2phla1 (*Phaseolus vulgaris* seed storage protein, 2PHL); d1pmt_2 (*Proteus mirabilis* glutathione transferase, 1PMT); d1poxa2 (*Lactobacillus plantarum* pyruvate oxidase, 1POX); d1pne__ (bovine profilin, 1PNE) and d1a5r__ (human small ubiquitin-related protein SUMO-1, 1A5R).

Single *versus* multiple-template modelling

The 271 families from SCOP were selected randomly. A draw was made to select one protein domain (query) in each family to be modelled using the other proteins, in the same family, as templates. Templates in each family were ranked by sequence identity with the query. Only the first would then be used for single-template models, or down to the first five for multiple-template models.

To bypass alignment errors in this experiment, the query sequence was aligned with the best template using the known molecular structure (taken from the Protein Data Bank[45]), permitting us to reach conclusions concerning the templates. In particular, query and best template were aligned structurally and superimposed in space using distance-driven dynamic programming, a previously tested approach.[46,47] In our implementation, two given C^{β} atoms are considered to be equivalent if the distance between them is less than 3 Å.

When more than one template was used, a multiple structural alignment was built and only the leader sequence was then aligned to the query.

Optimal and suboptimal sequence alignments

When query and template sequences were needed to be aligned, we used the evolutionary sequence profile of the query as scoring matrix complemented with their three-state (α-helix, β-sheet and coil) secondary structure matching. Sequence profiles (position-specific scoring matrices, pssm) were computed after five iterations of PSI-BLAST[33] against the nr database† with default parameters. The secondary structure for the template was assigned by running the program DSSP[48] on the original set of coordinates taken from the PDB database.

The secondary structure of the query was predicted using its sequence profile and the program PSI-PRED.[49] To compute the score for any $cell_{ij}$ in the alignment matrix the log-odd for residue j (template) in the query pssm (position i) was taken and 1 added if their residue secondary structure states matched. After filling the matrix, it is normalized so that the maximum value of any cell is 1. Then the dynamic programming procedure, as modified by Gotoh,[50] proceeds using a gap opening penalty of 1.0 and an extension penalty of 0.25.

After computing the optimal alignment, the pssm is used to calculate the average log-odd score (or bit-score) per residue. Alignments were considered for the experiments only if their bit-score was over 2.0. This cut-off was chosen after a benchmark of sequence alignment techniques (data not shown).

To generate suboptimal alignments, the guidelines explained in detail previously[27,51] were followed to implement an iterative dynamic programming function that discovers non-trivial suboptimal alignments by penalizing positions aligned in previous iterations. After computing one alignment trace, aligned residues are marked to be penalized in the next iteration. The penalty chosen for the next iterations was −0.1.

† http://www.ncbi.nlm.nih.gov

Atomic deviation measures: rmsd

For the first three experiments explained in Results (covering single *versus* multiple templates) selecting templates and alternative alignments, the reported rmsd values were obtained after superimposing pairs of models with the program SSAP.[52] These measures correspond to average deviations between all pairs of equivalent C^{α} atoms.

For the recombination experiments, the rmsd calculations are now based on C^{β} and are calculated as part of our structural superposition routine described above. This rmsd function is:

$$\mathrm{rmsd}(p,q) = \sqrt{\frac{\sum_{i=1}^{n}(p_i - q_i)^2}{n}}$$

where p and q are sets of n C^{β} atoms in Cartesian space.

Both measures are based on all the equivalent pairs of residues obtained after aligning two sequences, including loops.

Fitness function = internal contacts + solvation energies

The fitness function is a free energy estimate based on two components: residue–residue contacts and solvation energies:

$$\mathrm{fitness}(p) = \mathrm{contacts}(p) + \mathrm{solvation}(p)$$

To calculate the first term, protein models are simplified so that every residue is represented as only four pseudo-atoms (CO, C^{α}, NH and R, the side-chain centroid). A total contact energy (all-against-all) for each model can then be calculated quickly by using precomputed statistical atom–atom potentials using a soft Lennard–Jones type function. The representation and the statistical potentials used have been described.[28]

The solvation term is calculated as the sum of all-atom side-chain solvent-exposed area (calculated with NACCESS†) multiplied by tabulated empirical residue solvation free energies.[32]

To compare protein models with different lengths, total energies are divided by the number of residues.

Genetic algorithm: recombination + mutation

The flow diagram of the algorithm is shown in Figure 3. Here, we explain the details of the implementation. To begin, an initial population of protein models is required, composed of more than one member. These models can be generated from different templates and/or different query to template sequence alignments. Then the population is grown until the selection size is reached (50 members in our recombination experiments). At this point, the fitness is estimated for every member of the population. Only a given proportion (typically 75%) of protein models is selected as seed for the future generation (founder population). This process is iterated until the founder population becomes homogeneous (in this work, defined as when the fitness difference between the best and worst founder is less than 0.1 kcal mol^{-1} $\mathrm{residue}^{-1}$ (1 cal = 4.184 J)).

A population grows to reach selection size by selecting two mating protein models randomly. In the draw, members of a population have a probability of selection that increases proportionally to the number of siblings they have, according to:

$$\mathrm{prob}(p) = \frac{\mathrm{number\_siblings}(p) + 1}{\mathrm{size}(P) + \mathrm{total\_number\_siblings}(P)}$$

where p is a member of the population P.

With a probability of R (0.9 in this experiment) these two mates will undergo recombination, otherwise they will generate a mutant model.

A recombination event starts by superimposing the two mates as follows. (1) C^{β} superposition based on sequence alignment. Because they have identical sequences, this alignment is trivial. (2) Refinement based only on equivalent residues; tolerance is set to twice the average C^{α}–C^{β} distance (3.61 Å).

Once this complex is formed, a random event is needed, the selection of the crossover point. For this, only regions with no regular secondary structure, as defined by STICK‡ are considered. The recombinant protein is made of the N terminus of one protein and the C terminus of the other; the boundary is the crossover point. The program always cuts and pastes proteins in coil regions and therefore the geometry of loops involved in crossover events may be severely affected. No attempt is made to fix them on run-time in the present implementation.

The mechanism for mutation is simply an all-atom Cartesian space average of the two selected mates, once they have been superimposed. Some mutant proteins may have seriously distorted geometries.

After a reproduction event, the program does a simple phi, psi, omega angle analysis to reject sibling proteins with bad stereochemistry (more than one main-chain break and more than 4% non-planar peptide bonds).

A protein recombination experiment can take from five minutes to several hours (running serial C++ code on a 2.4 GHz PentiumIV desktop PC running Linux) depending on the size of the sequence to model and the population. Thus, it is usually more expensive than building models using traditional methodologies. The most time-consuming step of the algorithm is growing each population, but this could be done in parallel if a farm of computers is available by performing one reproduction event per node.

Alignment shift calculation

To calculate the quality of the alignments in the protein recombination experiment, the resulting models in each population were aligned structurally with their corresponding real structures, as taken from the PDB database. Taking these alignments as references, the average number of shifts per aligned residue is computed. As models and real structures have identical sequences, this computation is trivial. An average shift of 0 means that the real structure and the model can be superimposed optimally using their corresponding sequence alignment. A value of 1 would mean that every residue is displaced, on average, by one residue.

† http://wolf.bms.umist.ac.uk/naccess

‡ http://mathbio.nimr.mrc.ac.uk/ftp/wtaylor/stick

Generation of models from shifted alignments

The sequence for each of the eight query SCOP domains (described above) was used as input for the interactive form of the web server 3D-JIGSAW† and five alignments with the top template (100% identical in sequence) were shifted one, two, three or four positions to either side of a randomly selected residue before building the models. The resulting complete models were used in the recombination experiment.

Building models from PSI-BLAST output

PSI-BLAST version 2.2.5 was used with default parameters. The database used was the same as that used by our 3D-JIGSAW server, prepared by merging PFAM[53] and PDB sequences. Five iterations were used and the output was parsed to extract the alignments with a maximum of eight templates. Models were built from these alignments using 3D-JIGSAW. The average *e*-value of the alignments used was 8E-03. PSI-BLAST models were, on average, 1.7 residues shorter than corresponding models aligned by our procedure.

Figure preparation

The Figures showing protein structures were prepared with Rasmol[54] and MOLSCRIPT.[55]

Software

A test version of *in silico* Protein Recombination is available‡.

Model sets used in this work are available from the authors on request.

Acknowledgements

We thank the Biomolecular Modelling Group for their input and Marc Offman for testing the algorithm extensively on difficult CASP5 targets.

References

1. Chothia, C. & Lesk, A. M. (1986). The relation between the divergence of sequence and structure in proteins. *EMBO J.* **5**, 823–826.
2. Eyrich, V. A., Marti-Renom, M. A., Przybylski, D., Madhusudhan, M. S., Fiser, A., Pazos, F. *et al.* (2001). EVA: continuous automatic evaluation of protein structure prediction servers. *Bioinformatics*, **17**, 1242–1243.
3. Contreras-Moreira, B., Fitzjohn, P.W., Bates, P.A. (2002). Comparative modelling: an essential methodology for protein structure prediction in the post-genomic era. *Appl. Bioinformatics*, **1**.
4. Tramontano, A., Leplae, R. & Morea, V. (2001). Analysis and assessment of comparative modeling predictions in CASP4. *Proteins: Struct. Funct. Genet. Suppl.*, 22–38.
5. Sali, A. & Blundell, T. L. (1993). Comparative protein modelling by satisfaction of spatial restraints. *J. Mol. Biol.* **234**, 779–815.
6. Fiser, A., Do, R. K. & Sali, A. (2000). Modeling of loops in protein structures. *Protein Sci.* **9**, 1753–1773.
7. Guex, N., Diemand, A. & Peitsch, M. C. (1999). Protein modelling for all. *Trends Biochem. Sci.* **24**, 364–367.
8. Bates, P. A. & Sternberg, M. J. (1999). Model building by comparison at CASP3: using expert knowledge and computer automation. *Proteins: Struct. Funct. Genet.* **37**, 47–54.
9. Ogata, K. & Umeyama, H. (2000). An automatic homology modeling method consisting of database searches and simulated annealing. *J. Mol. Graph. Model.* **258–272**, 256–305.
10. Lambert, C., Leonard, N., De Bolle, X. & Depiereux, E. (2002). ESyPred3D: prediction of proteins 3D structures. *Bioinformatics*, **18**, 1250–1256.
11. Marti-Renom, M. A., Stuart, A. C., Fiser, A., Sanchez, R., Melo, F. & Sali, A. (2000). Comparative protein structure modeling of genes and genomes. *Annu. Rev. Biophys. Biomol. Struct.* **29**, 291–325.
12. Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd edit., Springer, New York.
13. Unger, R. & Moult, J. (1993). Genetic algorithms for protein folding simulations. *J. Mol. Biol.* **231**, 75–81.
14. May, A. C. & Johnson, M. S. (1994). Protein structure comparisons using a combination of a genetic algorithm, dynamic programming and least-squares minimization. *Protein Eng.* **7**, 475–485.
15. Pedersen, J. T. & Moult, J. (1995). *Ab initio* structure prediction for small polypeptides and protein fragments using genetic algorithms. *Proteins: Struct. Funct. Genet.* **23**, 454–460.
16. Morris, G. M., Goodsell, D. S., Huey, R. & Olson, A. J. (1996). Distributed automated docking of flexible ligands to proteins: parallel applications of AutoDock 2.4. *J. Comput. Aided Mol. Des.* **10**, 293–304.
17. Rabow, A. A. & Scheraga, H. A. (1996). Improved genetic algorithm for the protein folding problem by use of a Cartesian combination operator. *Protein Sci.* **5**, 1800–1815.
18. Xia, Y. & Levitt, M. (2002). Roles of mutation and recombination in the evolution of protein thermodynamics. *Proc. Natl Acad. Sci. USA*, **99**, 10382–10387.
19. Riechmann, L. & Winter, G. (2000). Novel folded protein domains generated by combinatorial shuffling of polypeptide segments. *Proc. Natl Acad. Sci. USA*, **97**, 10068–10073.
20. Broo, K., Larsson, A. K., Jemth, P. & Mannervik, B. (2002). An ensemble of theta class glutathione transferases with novel catalytic properties generated by stochastic recombination of fragments of two mammalian enzymes. *J. Mol. Biol.* **318**, 59–70.
21. Bates, P. A., Kelley, L. A., MacCallum, R. M. & Sternberg, M. J. (2001). Enhancement of protein modeling by human intervention in applying the automatic programs 3D-JIGSAW and 3D-PSSM. *Proteins: Struct. Funct. Genet. Suppl.*, 39–46.
22. Venclovas, C. (2001). Comparative modeling of CASP4 target proteins: combining results of sequence search with three-dimensional structure assessment. *Proteins: Struct. Funct. Genet. Suppl.*, 47–54.

† http://www.bmm.icnet.uk/servers/3djigsaw
‡ http://www.bmm.icnet.uk/servers/3djigsaw/recomb/index.html

23. Murzin, A. G., Brenner, S. E., Hubbard, T. & Chothia, C. (1995). SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.* **247**, 536–540.
24. Brunger, A. T. (1997). X-ray crystallography and NMR reveal complementary views of structure and dynamics. *Nature Struct. Biol. 4 Suppl.*, 862–865.
25. Montelione, G. T., Zheng, D., Huang, Y. J., Gunsalus, K. C. & Szyperski, T. (2000). Protein NMR spectroscopy in structural genomics. *Nature Struct. Biol. 7 Suppl.*, 982–985.
26. Needleman, S. B. & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* **48**, 443–453.
27. Saqi, M. A. & Sternberg, M. J. (1991). A simple method to generate non-trivial alternate alignments of protein sequences. *J. Mol. Biol.* **219**, 727–732.
28. Robson, B. & Osguthorpe, D. J. (1979). Refined models for computer simulation of protein folding. Applications to the study of conserved secondary structure and flexible hinge points during the folding of pancreatic trypsin inhibitor. *J. Mol. Biol.* **132**, 19–51.
29. Holm, L. & Sander, C. (1992). Evaluation of protein models by atomic solvation preference. *J. Mol. Biol.* **225**, 93–105.
30. Koehl, P. & Levitt, M. (1999). *De novo* protein design. I. In search of stability and specificity. *J. Mol. Biol.* **293**, 1161–1181.
31. Hubbard, S. J. & Thornton, J. M. (1993). *NACCESS, Computer Program*, Department of Biochemistry and Molecular Biology, University College, London.
32. Eisenberg, D. & McLachlan, A. D. (1986). Solvation energy in protein folding and binding. *Nature*, **319**, 199–203.
33. Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W. & Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucl. Acids Res.* **25**, 3389–3402.
34. Zhang, X., Shaw, A., Bates, P. A., Newman, R. H., Gowen, B., Orlova, E. *et al.* (2000). Structure of the AAA ATPase p97. *Mol. Cell*, **6**, 1473–1484.
35. Wriggers, W. & Chacon, P. (2001). Modeling tricks and fitting techniques for multiresolution structures. *Structure (Cambridge)*, **9**, 779–788.
36. Elcock, A. H. (2002). Modeling supramolecular assemblages. *Curr. Opin. Struct. Biol.* **12**, 154–160.
37. Aloy, P., Ciccarelli, F. D., Leutwein, C., Gavin, A. C., Superti-Furga, G., Bork, P. *et al.* (2002). A complex prediction: three-dimensional model of the yeast exosome. *EMBO Rep.* **3**, 628–635.
38. Zagrovic, B., Snow, C. D., Shirts, M. R. & Pande, V. S. (2002). Simulation of folding of a small alpha-helical protein in atomistic detail using worldwide-distributed computing. *J. Mol. Biol.* **323**, 927–937.
39. Lundstrom, J., Rychlewski, L., Bujnicki, J. & Elofsson, A. (2001). Pcons: a neural-network-based consensus predictor that improves fold recognition. *Protein Sci.* **10**, 2354–2362.
40. Janardhan, A. & Vajda, S. (1998). Selecting near-native conformations in homology modeling: the role of molecular mechanics and solvation terms. *Protein Sci.* **7**, 1772–1780.
41. Elofsson, A. (2002). A study on protein sequence alignment quality. *Proteins: Struct. Funct. Genet.* **46**, 330–339.
42. de Groot, B. L., van Aalten, D. M., Scheek, R. M., Amadei, A., Vriend, G. & Berendsen, H. J. (1997). Prediction of protein conformational freedom from distance constraints. *Proteins: Struct. Funct. Genet.* **29**, 240–251.
43. Taylor, W. R. (2001). Defining linear segments in protein structure. *J. Mol. Biol.* **310**, 1135–1150.
44. Brenner, S. E., Koehl, P. & Levitt, M. (2000). The ASTRAL compendium for protein structure and sequence analysis. *Nucl. Acids Res.* **28**, 254–256.
45. Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H. *et al.* (2000). The Protein Data Bank. *Nucl. Acids Res.* **28**, 235–242.
46. Russell, R. B. & Barton, G. J. (1992). Multiple protein sequence alignment from tertiary structure comparison: assignment of global and residue confidence levels. *Proteins: Struct. Funct. Genet.* **14**, 309–323.
47. Gerstein, M. & Levitt, M. (1996). Using iterative dynamic programming to obtain accurate pair-wise and multiple alignments of protein structures. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **4**, 59–67.
48. Kabsch, W. & Sander, C. (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, **22**, 2577–2637.
49. Jones, D. T. (1999). Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol. Biol.* **292**, 195–202.
50. Gotoh, O. (1982). An improved algorithm for matching biological sequences. *J. Mol. Biol.* **162**, 705–708.
51. Saqi, M. A., Bates, P. A. & Sternberg, M. J. (1992). Towards an automatic method of predicting protein structure by homology: an evaluation of suboptimal sequence alignments. *Protein Eng.* **5**, 305–311.
52. Taylor, W. R. & Orengo, C. A. (1989). Protein structure alignment. *J. Mol. Biol.* **208**, 1–22.
53. Sonnhammer, E. L. L., Eddy, S. R. & Durbin, R. (1997). Pfam: a comprehensive database of protein domain families based on seed alignments. *Proteins: Struct. Funct. Genet.* **28**, 405–420.
54. Sayle, R. A. & Milner-White, E. J. (1995). RASMOL: biomolecular graphics for all. *Trends Biochem. Sci.* **20**, 37–376.
55. Kraulis, P. J. (1991). MOLSCRIPT: a program to produce both detailed and schematic plots of protein structures. *J. Appl. Crystallog.* **24**, 946–950.

Edited by J. Thornton

(Received 13 December 2002; received in revised form 27 February 2003; accepted 28 February 2003)