Using structure to explore the sequence alignment space of remote homologs

Andrew Stephen Kuziemko

Submitted in partial fulfillment of the
requirements for the degree
of Doctor of Philosophy
under the Executive Committee of the Graduate School of
Arts and Sciences

COLUMBIA UNIVERSITY
2011

Abstract

"Using structure to explore the sequence alignment space of remote homologs"

Andrew Kuziemko

The success of protein structure modeling by homology requires an accurate sequence alignment between the query sequence and its structural template. However, sequence alignment methods based on dynamic programming (DP) are typically unable to generate accurate alignments for remote sequence homologs, thus limiting the applicability of modeling methods. A central problem is that the alignment that would produce the best structural model is generally not optimal, in the sense of having the highest DP score. Suboptimal alignment methods can be used to generate alternative alignments, but encounter difficulties given the enormous number of alignments that need to be considered. We present here a new suboptimal alignment method that relies heavily on the structure of the template. By initially aligning the query sequence to individual fragments in secondary structure elements (SSEs) and combining high-scoring fragments that pass basic tests for 'modelability', we can generate accurate alignments within a set of limited size.

Chapter 1 introduces the field of protein structure prediction in general and the technique of homology modeling in particular. One subproblem of homology modeling — the sequence to structure alignment of proteins — is discussed in Chapter 2. Particular attention is given to descriptions of the size, density and redundancy of alignment space as well as an explanation of the dynamic programming technique and its strengths and weaknesses. The rationale for developing alternative alignment techniques and the unique difficulties of these methods are also discussed.

Chapter 3 explains the methodologies of S4 — the alternative alignment program we developed that is the main focus of this thesis. The process of finding alternative alignments with S4 involves several steps, but can be roughly divided into two main parts. First, the program looks for combinations of high-similarity fragments that pass basic rules for modelability. These 'fragment alignments' define regions of alignment space that can be searched more thoroughly with a statistical potential for a single representative for that region. The ensemble of alignments that is thus created needs to be evaluated for accuracy against the correct alignment. Current methods for doing so, as well as adjustments to those methods to better suit the realm of remote homology alignments, are discussed in Chapter 4. A novel measure for determining similarity between alignments, termed the inter-alignment distance (IAD) also is developed. This measure can be used to assess quality, but is also well-suited to finding redundant alignments within an ensemble.

In Chapter 5, the results of testing S4 on a large set of targets from previous CASP experiments are analyzed. Comparisons to the optimal alignment as well as two standard alternative alignment methods, all of which use the same similarity score as S4, demonstrate that S4's improvement in accuracy is due to better sampling and filtering rather than more sophisticated scoring. Models made from S4 alignments are also shown to significantly improve upon those made from optimal alignments, especially for remote homologs. Finally, an example of a sequence to structure alignment offers an in depth explanation of how S4 finds correct alignments where the other methods do not.

Chapter 6 describes a set of three experiments that paired S4 with the model evaluation tool ProsaII in a homology modeling pipeline. There were two primary objectives in this project. First, we wanted to test different methods for finding remote homologs that could serve as input to S4. And second, we evaluated the use of ProsaII as a method for discriminating between good and bad models, and thus also between homologous and non-homologous templates. The first two experiments are essentially blind searches for homologous sequences and structures. The third experiment takes remote templates returned by PSI-BLAST and uses S4 and ProsaII to find alignments and determine whether the template is a structural homolog. While S4 was able to find homologs in the blind searches, the alignment/model quality and level of discrimination was found to be higher when the input to the pipeline came from a set of structures produced by a template selection method.

Finally, Chapter 7 discusses the consequences of this research and suggests future directions for its application.

Contents

List of Figures

List of Tables

To my first teachers, my parents, David and Maria

Acknowledgements

First and foremost, I would like to thank Prof. Barry Honig for the opportunity of working with him in his group. Barry has set a wonderful example for all his students of what it means to be a professional scientist. His combination of curiosity along with a healthy dose of skepticism has made his lab an exciting place to learn the practice of science.

If a thesis could have a second author, this one's would be Donald Petrey. With his extensive knowledge of structure prediction and related fields, I was able to get instant feedback on practically any idea. I came to rely on this benefit of working with Donald, as well as his patience. It is a tremendous help to know an expert who is always willing to discuss your project.

This research builds upon the work of Christopher Tang. Fortunately for me, we overlapped in the lab for several years, during which time I was able to learn a great deal from Chris. I am the beneficiary of his clear and thoughtful understanding of this field.

There were days when a month's worth of work culminated in results that were worse than before. And others when a bug in my project remained elusive longer than I could stay good-natured. On these days it was very nice to have good friends in the lab, especially Sean West, Peng Liu, Klara Felsovalyi, Jeremie Vendome, Katie Rosa and Rachel Kolodny. In particular, Sean and Rachel went above and beyond the requirements of friendship to take time helping me with problems large and small. For this and because they very rarely told me to 'just Google it', I am very grateful.

Lastly, I would like to thank my parents, David and Maria Kuziemko, and my sister, Ilyana. Their encouragement throughout my education has made every goal easier to reach.

1

Introduction: Approaches to protein structure prediction

The huge disparity between the organic compounds from which living things are made and the complex forms and behaviors they possess is truly one of the wonders of nature. At what level does a lifeless organization of atoms become an ordered arrangement in support of a living thing? At a macroscopic level, the maxim of form following function is clear to any observer. Over the millennia, natural selection has 'designed' structures — birds' wings, vertebrate vision systems, the human brain — that seem ideally suited to their tasks. One of the wonders of biology is that this phenomenon of ideally formed functional structures continues down to the molecular level. Proteins (and other large biological macromolecules) are perhaps the smallest example of atoms arranged with a clear purpose.

The reductionist inclinations of scientists lead us to look for simpler truths behind complex phenomena. As is often the case, however, these investigations lead to more difficult questions. In the case of biology, while the discovery of the central dogma of a DNA sequence determining the composition of proteins makes possible a mechanistic explanation of natural selection, it too has led to many more questions. Why do some changes in the protein sequence have a large effect on protein structure and function and others do not? Is it possible to avoid the trial and error of natural selection and modify existing proteins toward a human-conceived purpose? Has nature fully explored all the basic shapes of proteins, or are there other possible forms — and novel functions associated with them — that are yet to be discovered? Significant progress could be made in answering all these questions if one central dilemma could be solved: how can we determine the three dimensional structure of a protein based on its sequence alone?

1.1 The protein folding problem

In 1961, C. B. Anfinsen's study of bovine pancreatic ribonuclease (1) revealed important insights into the nature of protein folding. He showed that an active sample of the enzyme could be denatured completely, thereby losing almost all activity, and then allowed to refold with the denaturant removed, which restored enzyme activity and, presumably, the protein's native secondary and tertiary structure. The critical conclusion from this experiment was that, since the experiment was performed *in vitro*, the information for a polypeptide chain to fold into a stable, active protein resided in the chain itself.

As a result, structural biologists began investigating how the three-dimensional structure of a protein arises from its one-dimensional sequence. Though much research has been done on this question and significant progress has been made (summarized below), the development of a comprehensive and consistently accurate solution to the protein structure prediction problem has not yet been achieved.

1.1.1 Levinthal's Paradox and the thermodynamic hypothesis

The biggest difficulty toward progress in the protein folding problem is the enormous number of possible sequences and possible structural conformations of each sequence. Unlike DNA with its small alphabet of four nucleotides and relatively consistent double-helical shape, proteins have a far larger degree of sequential and structural variability. Twenty amino acids and several sterically accessible states for each amino acid in a structure lead to an immense set of possible sequences and shapes.

This difficulty was described most incisively by Levinthal (2). He performed a simple calculation of the number of states available to a protein containing N amino acids. If we conservatively estimate each amino acid to have three possible states of their ϕ and ψ angles (as observed from Ramachandran mapping), then the protein will have approximately 3^N conformations available to it. If $N = 100$, we calculate that there are approximately 10^{47} distinct structures it can form. Of course, even if these conformations could be individually sampled by the protein very quickly, it would not be possible for a protein to fold on the timescale of milliseconds to seconds observed in nature. This has come to be known as 'Levinthal's paradox', though it was more of a demonstration that it is simply impossible for protein folding to occur by a random, brute force exploration of possible conformations. Rather, there must be a 'folding landscape' that encourages a polypeptide to move toward the native fold.

Another important contribution from Anfinsen suggests a shape for this landscape. The 'thermodynamic hypothesis' (3) claims that the native state of a protein sequence will possess the lowest possible free energy of all possible conformations at normal environmental conditions. This implies that the native state lies at the bottom of a potential well and should be stable to small perturbations.

1.1.2 Protein 'spaces'

This set of possible conformations accessible to a folding polypeptide is often referred to as structure space. This is a somewhat loosely defined term since structures cannot easily be conceived of as points in any kind of multi-dimensional Euclidean space. Still, there are clearly pairs of structures that are very similar and those that are very different and it is useful to think of these structures as points that are close together or far apart. There even exist distance measures between structures, such as RMSD, which make the notion of a structure space more intuitive.

Another space that we will make reference to here is sequence space. With 20 amino acid types, the size of this space of possible protein sequences is similarly daunting (20^N). The task in protein structure prediction is to take the single point in sequence space represented by the 'query' sequence and map it to the correct point (or a nearby one) in structure space. Incidentally, the related field of protein design involves the inverse folding problem, where the goal is to find sequences that will fit a given 3-dimensional structure (4, 5, 6). The goal here is truly the opposite: map a location in structure space to the corresponding one (or ones) in sequence space.

Later in Section 2.1.1 we will discuss yet another kind of space, alignment space, which describes the number of different possible correspondences *between* two sequences. While the sheer enormity of these possibilities is often a source of frustration for the structural biologist, the exquisite complexity of the natural world springs from the ability of its building blocks to combine and interact in myriad different ways.

1.1.3 Theories of protein folding

Returning to Levinthal's paradox, a graphical depiction of the search for the native conformation in the vast structure space available is shown in Figure 1.1a. The immediate problem with this hypothesis is the same embedded in Levinthal's paradox: the space is too vast and too flat for an unfolded protein to ever find the native conformation. Levinthal himself proposed a solution to this problem that postulated the existence of a folding pathway consisting of a series of partially folded intermediates (7), shown schematically in Figure 1.1b.

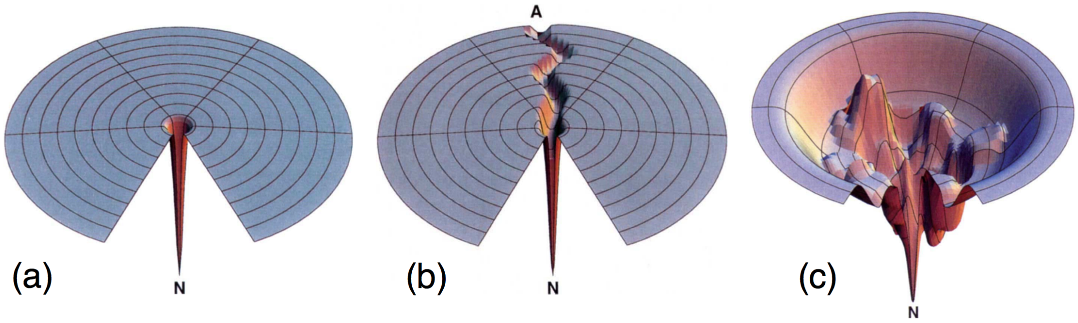


Figure 1.1: Different models of the folding landscape - (a) In the 'golf course' model, structure space is shown here as a featureless landscape with a single point with low free energy surrounding the native conformation, N. (b) The pathway model allows the unfolded protein to find the native conformation via a series of partially folded intermediates. (c) The folding funnel demonstrates that essentially all initial unfolded states lead to the native state while encountering a varied potential surface. Diagram from Dill and Chan (8).

This pathway approach also has limitations, perhaps the most obvious of which is that the denatured state does not represent a particular conformation at one end of the pathway, but rather the entire ensemble of all non-native folds. Further studies showed that structure space should be viewed as generally funnel-shaped, with the native conformation occupying the point of lowest free energy. Zwanzig *et al* (9) showed that if native conformations of individual amino acids are favored by even a few kT, then the random search through structure space becomes a biased one that guides the protein toward its native state. Lattice simulations of folding (10) demonstrated a quick 'hydrophobic collapse' to a 'globule' state followed by a slow, rate-limiting passage through a large number of transition states to the final native conformation. Others showed evidence for 'hierarchic folding' in which proteins folded one sub-structure at a time (11, 12). Studies such as these suggested more complex conceptions of what a hypothetical folding funnel should look like, such as the example in Figure 1.1c.

1.2 From sequence to structure

1.2.1 Biophysical approaches to protein folding

The studies discussed thus far suggest that: (a) the sequence contains sufficient information to determine the 3D structure, (b) the mechanism of moving from an extended state to a folded protein likely involves the formation of semi-stable intermediates of increasingly lower free energy and (c) the native conformation can be recognized by virtue of being the structure with the lowest free energy for its sequence. The minimum free energy requirement is responsible for several other observed properties of proteins, such as the hydrophobicity of the core and the propensity of different residues to form α-helices or β-sheets. However, given the enormous number of conformations available to even a small polypeptide and the difficulty in identifying the short-lived intermediate states, solving the protein folding problem remains currently out of reach from a purely biophysical approach (13).

Despite its current impracticality, a biophysical approach to the protein folding problem remain attractive, perhaps because, at least in principle, it should be possible. The protein sequence is known and the interactions of the different residue types with each other and the solvent are quite well understood (14, 15). But the number of interactions to consider as the length of the protein grows limits the utility of folding simulations to small polypeptides (13, 16, 17, 18). To increase this upper limit on protein length, it is possible to create 'coarse-grained' energy functions that sacrifice physical detail to lower the computational complexity while, hopefully, losing little accuracy (19). Recent work in these areas shows promise, but still remains applicable only to relatively small proteins, even when immense computing power is available (20).

These biophysical approaches are similar in that they generally do not limit themselves to a particular region of structure space. That is, they rely on the physical realism of their energy functions to move the protein toward its native conformation. The other approach to the structure prediction problem makes an explicit assumption about the general shape of the protein in question. These homology modeling techniques have the advantage of being able to return near-atomic resolution structures for longer protein sequences, but are limited to those cases where a close structural homolog, or 'template' protein, has not only already been determined by crystallographic or NMR techniques, but is closely related enough in sequence to be identified as a structure neighbor. Homology modeling techniques therefore limit their scope to the neighborhood of structure space defined by the template. While clearly reducing the number of conformations to sample, an incorrect choice of template or a flawed alignment guarantees an inaccurate model. (See Section 1.3 for more detail.)

1.2.2 Similarity and dissimilarity between proteins

In approaching an intractable problem like protein folding, it is often helpful to examine special cases that may lend insight to the general problem. As the amount of sequence and structure information available has grown, it is possible to use sequence and structure alignment techniques to determine the similarity between many pairs of proteins. Using these measures, we can identify two interesting classes of protein pairs: those that share high structural homology while having little sequence similarity and those that possess very different structures while having highly similar sequences.

Membership in the latter group is considerably more rare, but their existence alone is intriguing. For example, Alexander and colleagues designed two proteins with 88% identity, but very different structures (21). Protein pairs of this kind shine light on what is *different* about the residues in several key positions that account for the large variations in structure.

There are many more examples of the other extreme: proteins with structural similarity despite little sequence homology. In fact, there is a strong evolutionary reason for this phenomenon to occur. Since structure determines function, it is not surprising that random mutations could create very different sequences while natural selection maintained the protein's function and structure. This divergent evolutionary explanation is probably more likely, but it is also possible for this phenomenon to arise from two unrelated sequences that converge on the same structural solution to a functional need. As opposed to pairs in the opposite extreme, these proteins with high structural homology and little sequence similarity let us ask what is *common* among residue types and positions that lead to a similar overall shape from essentially disparate sequences.

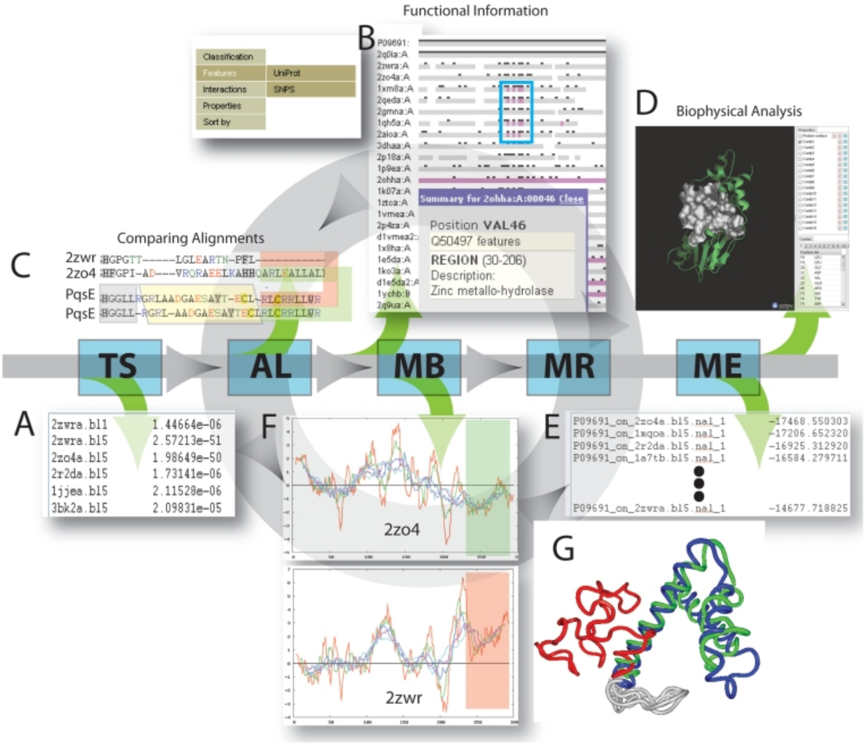
Many pairs of proteins can be found in this latter group. In fact, using a definition of structural similarity that allows for small differences while preserving the overall shape, one finds that a large majority of structural homologs have sequence identities in the range of 0-10% (22). This preponderance of low sequence identity structural homologs is perhaps not surprising since the number of possible structures — or at least the number of structures occurring in nature — is thought to be limited (23, 24, 25), while the number of unique sequences is extremely large.

1.3 Homology modeling

The task of structure prediction can be greatly simplified if a structure is available that has clear homology to a query sequence of interest and can therefore serve as a template. Assuming the correspondence of structurally equivalent residues is clear, one can build a model of the query by simply replacing template residues with their matches in the query. Biophysical or other techniques can be used to fill in the short segments where the template does not provide a guide, such as loop regions between SSEs and amino acid sidechains. This process is known as homology modeling.

There are many different areas in which homology models are being used, including: identifying active and ligand binding sites, determining protein-protein interfaces, refining models with data from experimental sources (such as NMR spectroscopy), providing a structural explanation for previous experimental observations, and using a model to plan new experiments (26). Though all these ends could be served by determining the structure with X-ray or NMR techniques, homology modeling can return an answer much more quickly. In addition, homology modeling can be successfully used for proteins that are difficult to solve experimentally.

The subject of this thesis is the sequence to structure alignment of proteins within the framework of homology modeling. So a brief overview of the steps involved in making a homology model will help put the work described here in its proper context. It will be helpful to refer to Figure 1.2, which is a flowchart of a homology modeling 'pipeline' developed by Norel and colleagues (27). The overall purpose of such a program is to take as input a query protein sequence at one end and produce its three dimensional structure as output at the other. This process breaks down into discrete steps, which is useful from the standpoint of modularity (e.g., one model-building technique can be swapped for another). Also, producing a structure in a step-wise fashion allows for the intervention of human expertise at a particular stage.



```
2zwr HGPGTT-----LGLEARTN-PFL-----------
2zo4 HFGPI-AD---VRQRAEELKAHHQARLEALLALI
PqsE HGGLLRGRLAADGAESAYT-ECL-RLCRRLLUR
PqsE HGGLL-RGRL-AADGAESAYTECLRLCRRLLUR
```

```
2zwra.bl1     1.44664e-06
2zwra.bl5     2.57213e-51
2zo4a.bl5     1.98649e-50
2r2da.bl5     1.73141e-06
1jjea.bl5     2.11528e-06
3bk2a.bl5     2.09831e-05
```

```
P09691_on_2zo4a.bl5.nal_1      -17468.550303
P09691_on_1mqoa.bl5.nal_1      -17206.652320
P09691_on_2r2da.bl5.nal_1      -16925.312920
P09691_on_1a7tb.bl5.nal_1      -16584.279711
            •
            •
            •
P09691_on_2zvra.bl5.nal_1      -14677.718825
```

Figure 1.2: Flowchart of a homology modeling pipeline - The blue boxes denote the five steps of homology modeling: template selection (TS), alignment (AL), model building (MB), model refinement (MR) and model evaluation (ME). The input to the first step (TS) is a protein sequence and the output from the final step (ME) is the sequence's predicted structure. For all other steps, their input is the output of the preceding step. The pictures flanking the pipeline demonstrate the type of output generated and possible opportunities for human intervention. Figure from Norel et al (27).

1.3.1 Template selection

The first task of homology modeling is the selection of an appropriate template for the query sequence. Along with the alignment of the sequence to the structure in the following step, this is the most critical step of the process. A significant error in template selection simply cannot be fixed at a later stage. For example, there is no correct alignment of an entirely α-helical sequence to a structure composed primarily of β-strands. And the resulting alignment and model would be far too flawed for current model refinement techniques to fix.

The several methods that exist to find templates from the Protein Data Bank PDB (28) are fairly reliable when a structure with high sequence homology is available. As identity continues to decrease below $\sim$30% identity, however, the task of recognizing the appropriate template becomes increasingly difficult (29). Methods such as BLAST (30) and PSI-BLAST (31) are often used for finding templates. But optimal alignment methods (32) can be used as well if their scores can be easily interpreted in terms of a likelihood estimate or 'e-value' (expectation value). Other programs (33) align the query sequence to the entire protein data bank (PDB) and return only alignments to those templates they deem homologous, thus implicitly performing template selection.

The criteria for determining success in template selection is whether the method reliably finds structures with clear structural homology to the query. Of course, this evaluation — or any evaluation in homology modeling — is only possible in test cases where the query structure is actually known, but withheld from the modeler. The definition of ‘clear structural homology’ is somewhat subjective, but some metrics dictate that: a large fraction of the query sequence can be aligned (high coverage); all or nearly all query SSEs have a topologically equivalent SSE in the template; an alignment to the template exists that can produce a model that is similar (as determined by methods such as RMSD, PSD (34) and TM-Score (35)) to the native query structure.

These criteria all use a fairly dry and mathematical definition of similarity. Homology modeling can also produce useful results without returning a model that is close to the native structure in terms of RMSD. For example, a partially-correct model that accurately depicts an active site or binding cleft while containing errors in other regions can still suggest follow-up experiments and also give insight into the functional role of the protein.

1.3.2 Alignment

Once a suitable template has been chosen, no other step in the homology modeling process is as critical for success as the determination of a quality alignment. In fact, the alignment stage represents the most significant source of error in current prediction efforts (36). The alignment of two sequences contains a combination of aligned residues and gaps. The aligned residues represent those portions of the sequences that are claimed to be structurally homologous. That is, the paired residues should occupy roughly the same position in space when the corresponding structures are aligned. The gaps in a sequence alignment are equally important in that they describe the portions of a sequence that do *not* have a homolog in the other sequence.

Sequence alignment methods thus have to perform correctly the different yet related tasks of identifying both homologous and non-homologous regions. Aligning non-homologous portions of the target and template sequences will not only lead to that portion of the model being incorrect, but typically ensures significant errors in other parts of the alignment and model. These two tasks of a sequence alignment method are handled by its two main components: a scoring function that rates the homology of each pairing of template and query residues and a gap penalty that determines the likelihood of gaps occurring at different positions in the alignment. It is the interplay of these two components that determines the results of most alignment methods. As the research discussed in this thesis is itself an alignment method, the topic of sequence alignment will be discussed in more detail in Chapter 2.

1.3.3 Model building

The job of a model building method is to convert a template structure into a model of the query sequence according to the correspondence of residues given by an alignment. A simple approach might begin with the template structure and change the type of each amino acid into that of the query residue to which it is aligned. Assuming one is only interested in the backbone of the model, this series of 'mutations' would actually build a fairly good model given that the query and template are quite similar and nearly all the query residues are aligned to the template.

In practice, however, we need methods that can build models from templates and alignments with more ambiguity. Alignments between more remote homologs will contain insertions (portions of a query that are not modelable based on the template) and deletions (portions of a template with no correspondence to the query and therefore no role in the model). The issue of modeling sidechains is another complication. The primary difficulty arises from the closely packed protein core, which presents a jigsaw-like puzzle to solve once the template's sidechains are swapped for the query's. Lastly, while the positions of secondary structure elements (SSEs) are typically better conserved between homologous proteins, the loops connecting them are not. The template often offers no reliable guide in these regions, requiring the model building method to construct loops *ab initio.*

One approach to solving these problems was developed by Xiang and Honig (37, 38) through the implementation of an 'artificial evolution' technique. The template is changed one residue at a time to incorporate changes in residue identity, insertions and deletions. A quick round of energy minimization follows each 'mutation'. This process continues until the template structure has been fully converted into the template.

Another approach developed by Sali and Blundell (39) uses the alignment and template structure to develop a detailed set of constraints, such as inter-residue distances, that define features of the model. A final model is then obtained by searching for a structure which satisfies the constraints most easily.

1.3.4 Model refinement

Model refinement techniques can be thought of as a separate step, though many of its techniques are incorporated into model building tools. The goal of this step is to find problematic regions of the model and make changes that bring the model closer to the native structure. Several major targets for model refinement methods are discussed below.

Loop regions are difficult to model because they contain no defined secondary structure and often lack a clear correspondence in the template structure. There are two main approaches to modeling loops. The first approach searches a database of structures to find loops that are similar in length, residue composition, solvent exposure and stem-to-stem distance (40, 41). The other approach builds loops *ab initio* using a variety of torsional sampling approaches (42). Nearly all loop modeling techniques need to use a loop closure method (43) as well as an evaluation function, such as a statistical or energy-based potential.

Sidechain prediction is another problem that must be addressed during the model refinement stage of a homology modeling procedure. These methods typically use a rotamer library to represent the commonly observed sidechain conformations for each residue type. A sampling algorithm and a scoring function (again usually statistical or energy-based) are then applied to find the set of rotamers with the best score (37, 44).

In contrast to sidechain prediction, which makes relatively small changes to the model without disturbing its backbone, several methods attempt to adjust the relative positions of entire secondary structure elements. The overall approach here is also sampling and scoring: moving the problematic SSE and using a scoring function to evaluate the potential or statistical energy at each position. But the large number of residues involved requires careful decisions to be made regarding which torsional angles will be constrained and which will be free to rotate (45, 46, 47).

Generally speaking, model refinement techniques can improve models in small ways, but cannot fix large errors. And because they seek to explore a portion of the vast space of the model's possible conformations, they are often computationally intensive. Considering the limited ability of refinement techniques to improve models, the importance of finding the best sequence-to-structure alignment is apparent.

1.3.5 Model evaluation

The final step of homology modeling is different from the first four in that it does not attempt to build or improve the model in any way. Instead, a model evaluation method is concerned with answering either of two questions. First, if presented with an ensemble of models, which could have resulted from an alternative alignment method (see Section 2.3) or from combinations of different methods used at each of the previous four steps, can the model evaluation technique rank these models effectively and identify the model that is closest to native? And second, can our model evaluator give reliable estimates of the likelihood of a particular model to be native?

The difference in these two tasks is somewhat subtle, but important. In performing homology modeling in the context of the CASP experiments (36) where the goal is simply to build the closest possible model to native, a model evaluation tool that can put models in a reliable rank order of distance from native is sufficient. However, in a more realistic context in which the query structure is truly unknown and not just withheld from the modeler, a model evaluation tool that can predict whether or not a structure is close to native is more useful. A program with that ability does not merely rank all models, but includes a prediction that several of them are likely nearly correct or that all of them contain significant errors. In this more realistic scenario, a positive result from a reliable model evaluation script can validate the output from all previous steps in the homology modeling pipeline. Because of its importance, model evaluation has been well studied and several different approaches have been developed.

One common approach is the use of statistical potentials to determine the quantity and quality of native-like interactions between residues in a model.

The statistical potential itself is calculated from a large database of structures. Information such as the types of residues present and the distances between pairs of residues are used to calculate potentials of mean force. These potentials can be seen as a preference for certain residue types to be at a particular distance from each other, therefore implicitly accounting for features of proteins such as the clustering of hydrophobic residues in the core. A particularly successful implementation of this concept is the DFIRE method (48), which returns an energy term to represent the stability of the model. These energies can be ranked, with the lowest energy model presumably the closest to native.

Another successful use of statistical potentials is implemented in the Prosa method (49). The fitness of a sequence for a particular protein shape or 'fold' is calculated with the statistical potential and compared to the score found from the same calculation on many different folds. A z-score can be thus determined for the model, which can be converted into the probability of the model being in approximately the correct fold (50, 51). Additionally, the fitness of each residue for its position in the structure can be analyzed separately. A string of poor-scoring residues may suggest a misfolded region of the protein. In this manner, it is possible to iterate between model evaluation and refinement methods to locate and fix errors.

One final approach to model evaluation also measures a sequence's fitness for a particular fold, but through the construction of a 3D profile instead of statistical potentials (4). These 3D profiles measure the solvent exposure, polarity and local secondary structure of each residue position in a structure. Since different amino acid types have different propensities for each of these characteristics, a 1D profile for a sequence can be developed. The 3D and 1D profiles can then be aligned using dynamic programming methods (see Section 2.2) to calculate a score for the alignment. In this manner, the fitness of a sequence for a particular structure is determined. In an evaluation context, a 3D profile can be prepared from a model and then used to score the model's own sequence (52).

2

Introduction: Finding homology through sequence alignment

As the name implies, homology modeling relies on the successful determination of the homologous correspondence between a template structure and a query sequence as described in an alignment. Successful completion of the preceding step of template selection will return a protein structure that *is* homologous to the query, but it is the job of the alignment procedure to determine exactly *how* the sequence and structure are related. Specifically, an alignment tells us which residues in the two sequence do and do not correspond structurally. In this chapter, we will examine some of the issues facing all alignment methods and discuss the making of alternative alignments in particular.

2.1 Alignment Space

Many problems in computational biology can be put in terms of a 'space' — a set of possibilities with a single or a small cluster of correct answers. Since the space is typically far too large to examine each member individually and because there are often many wrong answers that seem plausible, it is important to develop sophisticated sampling and scoring functions. A good sampling method will return a subset of the space that is collectively representative of the whole while containing only a small fraction of the number of elements. The scoring function has the task of discriminating between the 'good' and 'bad' examples presented to it. Both tasks are important, of course. If we think of the correct answer as a needle in a haystack, then we will never discover the needle without a good sampling method and we will not recognize it if we do without a reliable scoring function.

Along with loop modeling, sidechain prediction, inverse folding experiments, and SSE and ligand docking, the alignment of two sequences can also be described as a sampling and scoring problem. In this section, we will describe the nature of the alignment space to be sampled, since it is common to all alignment methods. These methods differentiate themselves primarily in terms of their scoring function, which will be discussed in subsequent sections in this chapter.

2.1.1 The size of alignment space

Like structure space or sequence space discussed earlier (Sections 1.1.1 and 1.1.2), alignment space is also difficult to visualize. Notions of similarity or distance will therefore require unique definitions suited to this space. (See Chapter 4 for more details on alignment comparison methods.) One place to begin, however, is by asking the simple question: how many unique alignments exist between sequences of given lengths?

This can be answered by examining the alignment matrix in Figure 2.1. By thinking of an alignment as both a path through this matrix and as a sequence of paired residues from the query and template, we can begin to see the solution. Let us first ask, for template and query sequences of length N and M, respectively, with $N \geq M$[1], how many unique alignments can we find with length k? That is, how many paths exist through the alignment matrix containing exactly k aligned pairs. We begin by noting that we must pick k query residues and k template residues, with $k \leq M$ since the number of aligned pairs cannot exceed the length of the shorter sequence. Also, there are $\binom{N}{k}$ unique ways to select k template residues and $\binom{M}{k}$ ways to select k query residues. One such pair of choices is denoted by the green arrows in Figure 2.1. Since each combination of k template residues and k query residues results in a unique alignment, we see that there will be $\binom{N}{k}\binom{M}{k}$ alignments with k pairs. The total size, S, of

[1] Assigning the longer sequence to the template is an arbitrary choice. The resulting size of alignment space is the same if the query is the longer sequence.

alignment space is therefore

$$S = \sum_{k=1}^{M} \binom{N}{k} \binom{M}{k} \qquad (2.1)$$

In practice, there are often a few constraints within alignment methods that prevent some alignments in this space from being explored, such as only allowing gaps that connect pairs containing either adjacent query or template residues. Even so, this formula represents a good estimate of the overall size of alignment space. To demonstrate how quickly this formula grows, for $M = N = 50$, $S \approx 10^{29}$, but when the sequences are both 100 residues in length, $S \approx 10^{591}$.

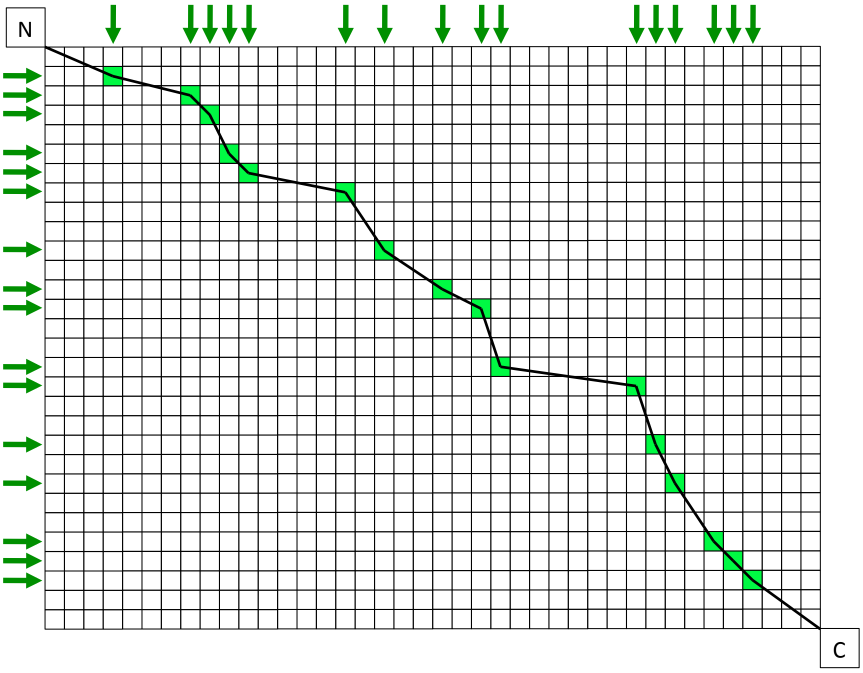


Figure 2.1: How many alignments can we make? - Using the variables from Equation 2.1, here we have a matrix with $N = 40$ and $M = 30$. A particular alignment is shown with length $k = 16$. For this particular matrix, the total size of alignment space, S, is $\sim 10^{19}$

[1] This derivation is based on discussions with Christopher Tang.

2.1.2 Density and redundancy in alignment space

The huge numbers from these calculations give rise to a feeling of disbelief: can there really be *so many* possible alignments to check? In fact, the disbelief is warranted since many of these alignments will be nearly identical. This is clear if one considers a smaller version of the alignment counting problem described above. Figure 2.2 depicts an alignment with a missing region that covers 10 query and 10 template residues. In an alignment of sequences of length 200 or 300, 10 x 10 residues may seem like a small region, perhaps covering a long loop between SSEs. But using equation 2.1 with $M = N = 10$, we find that there are over 180,000 unique paths through this region. Such a situation occurs commonly when a template loop region contains little similarity to the query. To further complicate matters, there could be several such regions in an alignment of sequences that are 200-300 residues in length. Even if one applies some restrictions on sampling, the combination of multiple occurrences of ambiguous portions of alignment space will quickly lead to an explosion in the number of alternate alignments.

It may be difficult to quantify exactly the density of an abstract space like alignment space, but the exercise above suggests that it is very dense in the sense of containing many alignments similar to each other. This density presents a challenge for an alignment sampling method: if the scoring function requires some amount of computational effort for each alignment it processes and we have reasonable limits placed on the total computational time of finding an alignment,

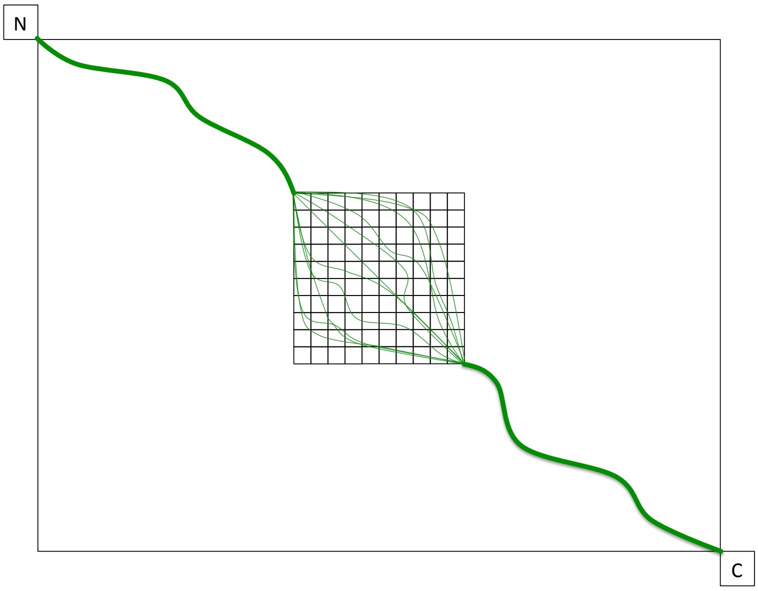


Figure 2.2: Many alternatives can arise from small regions of ambiguity - The heavy green lines represent portions of an alignment that are common to a set of alternatives. The 10 x 10 square is a region of ambiguity in the alignment that we wish to sample.

how can we ensure that time is not wasted checking redundant alignments? This issue of redundancy is of central importance to methods that generate an ensemble of alignments for a single query-template pair. Developing a novel approach to its mitigation was a major motivating factor for the work described here.

2.1.3 What is the correct alignment?

Assuming that our template selection method has successfully chosen a structure with actual structural homology for our query sequence, then, even though the 'haystack' is immense, there must be a 'needle' in it somewhere. The redundancy of alignment space further assures us that if there is a single alignment that best represents the correspondence of the homologous regions between query and template, then there will be many other alignments that are very nearly as good.

But does a single, ‘best’ alignment really exist? In practice, when assessing the quality of an alignment method, one must select a particular correct alignment for the sake of comparison. The alignment typically used for this is the one resulting from an alignment of the native query and template structures. This structure-based sequence alignment pairs residues based on their proximity in 3D space or their presence in topologically similar SSEs. But the existence of different structure alignment methods (34, 38, 53, 54) that can generate slightly different results implies that there is not a single correct answer.

For the purposes of homology modeling, it seems clear that the best alignment would be the one from which the best model can be made. Of course, this introduces new variables regarding which method is used to make the model and which model quality score is used to evaluate its similarity to the native structure. Though the differences between these methods is typically small, one should bear in mind that the concept of the correct alignment is not perfectly precise.

One source of this imprecision should be noted in particular. Even close structural homologs can have portions that share no clear correspondence. This often occurs in loop regions, which can be very dissimilar despite spanning the gap between well-aligned SSEs. Most structure alignment programs will nonetheless attempt to produce an at best meaningless alignment in these regions. Determining the quality of the output from an alignment method based partly on its performance in finding the same meaningless correspondence of residue in regions of non-homology diminishes the value of the exercise. As is discussed in Section 4.1.3, care can be taken in alignment quality evaluations to avoid this problem.

2.2 The dynamic programming algorithm

It is impossible to discuss sequence alignment methods without first explaining the dynamic programming (DP) algorithm, which is the basis for nearly all alignment methods. DP is a general technique suited to solving problems which are easily subdivided into smaller problems. It has been used in numerous applications in different fields, but we will limit our discussion here to its application to the problem of sequence alignment, which was first developed by Needleman and Wunsch in 1970 (55).

2.2.1 Outline of the algorithm

If the overall problem faced in sequence alignment is to find a highest-scoring or 'optimal' path through the alignment matrix according to certain scoring rules, then the subproblem into which DP divides this task is as follows: how can we calculate the optimal alignment back to the N-terminus and its score from a given residue pair (such as the red square in Figure 2.3) if we are given the optimal alignments and their scores from all adjacent pairs (the orange, blue and green squares in the figure)? If we could answer this question, then an approach to finding the optimal global alignment (i.e., the alignment that connects the N- and C-termini) is immediately apparent. We need only start at the N-terminus and solve this small sub-problem repeatedly at each position in the matrix as we move further out. We will eventually end at the C-terminus, at which point the optimal path back to the N-terminus and its score have just been obtained.

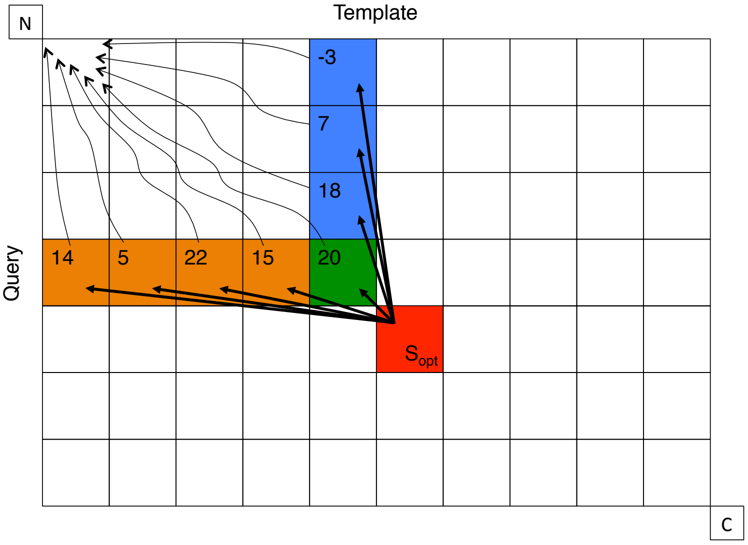


Figure 2.3: The dynamic programming sub-problem - All the cells in the matrix represent pairings of individual template residues (columns) and query residues (rows). Additionally, each cell has associated with it a similarity score value (not shown). As depicted here, the sub-problem requires us to find the highest scoring alignment between the red cell and the N-terminus via a connection to an adjacent cell involving either an insertion gap (blue cells), a deletion gap (orange cells) or no gap (green cell). The wavy lines from each adjacent cell represent the highest-scoring alignment back to the N-terminus and the number in each cell is the score of that alignment. The solution to the sub-problem will result in the determination of such an alignment for the red cell and its associated score, S_{opt}.

The basic step of the DP algorithm as applied to sequence alignment is depicted in Figure 2.3. But let us first discuss several things that are not shown. First, each column and row corresponds to a single residue in the template and query sequences, respectively. Each square thus represents a particular pairing of a template and query residue. A similarity score assigns a value to this match. Residues that are similar in amino acid type or other characteristics are given higher scores. Also, since homologous sequences generally have few regions of non-homology, there should be relatively few gaps in the alignment. A gap is represented by the arrows to the blue cells, which represent *insertions* of query residues between template residues, and the arrows to the orange cells, which represent *deletions* of template residues. If the red cell followed the green cell in an alignment, no gap would result since consecutive residues in both sequences would be paired. A gap penalty function would assign negative values to the insertions and deletions (not shown) that would generally be larger for longer gaps. These two functions, the similarity score and gap penalty, are the two defining features of optimal alignment methods. Variations in them account for essentially the only difference between most alignment methods, since almost all use the DP algorithm.

Let us return to the DP sub-problem of determining the best first step for an alignment back to the N-terminus that begins at the red cell. We will assume that we have already determined the optimal alignments from all adjacent cells (represented by the wavy lines in Figure 2.3) and their scores (the numbers given in the adjacent cells). We now have to determine the gap penalty for moving from the red cell to any of these adjacent cells. When we subtract the gap penalty from each adjacent cell's optimal alignment score and add the red cell's own residue-residue similarity score, we obtain the highest possible score for an alignment from the red cell to the N-terminus that passes through the adjacent cell. We choose to connect the red cell with the adjacent cell that produces the alignment with the highest score. This score will become the optimal score for the red cell, labeled S_{opt} in the figure, for later iterations of the algorithm. The problem is also solved for all cells above and to the left of the red cell. When these cells have their optimal alignments and scores determined, the algorithm moves one step further out. In this manner, we proceed from the N-terminus to the C-terminus. Upon reaching the C-terminus, we perform this procedure one final time and again calculate S_{opt}, which in this case is the score of the global optimal alignment. To find the alignment itself, we need only follow the sequence of arrows back to the N-terminus and record each alignment pair we pass through.

The alignment thus found is termed ‘optimal’ because it truly is the single highest-scoring alignment — as defined by the similarity score and gap penalty — of all alignments in the alignment space of the two sequences. However, it is very important to note that ’highest-scoring’ is not synonymous with ’best’ or ’correct’. It means only that the optimal alignment has the highest score for a given scoring function. Though essentially all optimal alignment methods rely on DP, they differ in how they calculate residue-residue similarity and gap penalties. Different approaches to these two functions are explained below.

2.2.2 The similarity score and gap penalty

A key step in the DP-based alignment procedure, or any other conceivable alignment method, is the determination of accurate similarity scores in each cell of Figure 2.3. A successful alignment technique cannot rely on a similarity function which fails to assign higher values to homologous pairings than non-homologous ones. Similarly, a gap penalty which over-penalizes regions of actual non-homology within the correct alignment or makes the opposite mistake and permits gaps too easily will also cause errors in a DP-based method.

There have been a number of advances in measuring residue-residue similarity. Individual residue-based scoring functions were replaced with more complex profile-profile (32, 56, 57) and environment-dependent methods (4, 58, 59). Gap penalties have also become more sophisticated. The earliest treatment of gaps is with an affine (or linear) gap penalty that is still common today. A linear gap penalty requires two parameters: a gap initiation penalty (g_i) for beginning a gap in the alignment and a gap extension penalty (g_e) for continuing one. Thus, a gap of length l would receive a penalty of $g_i + g_e(l - 1)$. Prompted by the realization that penalties of this type typically over-penalize long gaps, several studies have described the probability of a gap as a function of its length or location in the template structure with the goal of penalizing it appropriately (60, 61, 62). These probability functions have included power laws (63), multiexponentials (64) and bilinear approximations (65).

2.2.3 Advantages and disadvantages

The primary benefit to DP-based optimal alignment methods is immediately apparent to anyone who has used one: they work quickly. An optimal alignment can normally be returned within seconds, even for long sequences. Optimal alignment methods are also quite accurate when aligning sequences with 30% or more sequence identity (29).

A drawback to using a DP-based alignment procedure is the inherently local perspective used in determining each pair in the alignment. Returning to Figure 2.3, we note that the connections from the red cell to its adjacent cells are determined without considering either the pairs that have already been aligned (represented by the wavy lines back to the N-terminus) or the pairs that have yet to be aligned between the red cell and the C-terminus. In a sense, the DP algorithm is blind to everything except choosing the best connection between adjacent residue pairs. While it is true that the optimal alignment score in each of the adjacent cells implies a measure of the quality of the alignment to that point, it does not give any information about the specific residue pairings in that alignment.

For example, consider a case in which the alignment from a particular adjacent cell in Figure 2.3 back to the N-terminus contains residues representing a particular β-strand in the template. It is possible that a connection from the red cell to this adjacent cell will be a deletion which, though higher-scoring than all other possible connections, contains a deletion of a strand that is paired in the template with the strand already aligned. The result will be an alignment that leads to a model with an unpaired strand, a situation not found in nature. Another error that results from this local perspective is the possibility of building alignments that lead to non-compact or extended structures. The DP algorithm only sees the next connection and not the fact that it might be aligning the query sequence to distant, unconnected regions of the template structure.

Attempting to take issues of this type into account within the DP algorithm while maintaining its efficiency is not possible since doing so would mean the sub-problems of the algorithm are no longer independent. In terms of Figure 2.3, taking into account these global alignment concerns would mean that the optimal alignment back to the N-terminus could be different depending on whether the red cell or another adjacent cell is connected next in the alignment. Partly for this reason, most alignment methods are applied to higher homology cases where these issues are less troublesome. Another disadvantage of DP-based methods becomes apparent when we attempt to extend our search of alignment space beyond the optimal alignment.

2.3 Alternatives to the optimal alignment

As the similarity of two sequences decreases, it is expected that the probability of the optimal alignment containing an error will increase. A reasonable solution to this problem is to find alternatives to the optimal alignment, one of which should be close to correct. Since the optimal alignment has the highest score by definition, these alternative alignments will necessarily have lower scores and are thus often termed ‘suboptimal’. However, it is important to note that they are only suboptimal in the context of a DP sequence alignment score; the expectation is that one of them will produce a more accurate model than the optimal alignment when evaluated in structural terms.

2.3.1 Alternative alignment methods

A variety of suboptimal sequence alignment methods have been reported. Waterman (66) produced an ensemble of alternative alignments by modifying the dynamic programming algorithm to return all alignments with scores within a small difference, δ, from that of the optimal alignment. However, the difference between the DP scores of the structure-based alignment and the optimal sequence alignment can be significant, especially for remote homologues. Increasing δ until it encompasses a nearly correct alignment often produces an unmanageably large ensemble. Keeping δ small returns a more reasonable number, but the alignments tend to deviate negligibly from the optimal alignment.

Other methods have adapted Waterman’s approach to return only a ‘representative collection’ of alignments within δ from the optimal (67). Saqi and Sternberg (68) developed a method that also followed Waterman, but decremented the residue-residue similarity scores in the alignment matrix for each pairing in each alignment found. This approach directly penalizes an alignment that is similar to one previously determined, resulting in a more diverse ensemble. John and Sali (69) used genetic algorithm operators to splice and re-combine alignments in order to achieve the same goal. Chivian and Baker (70) produced suboptimal alignments by systematically varying the parameters in their optimal alignment method. Each alignment in their returned ensemble was therefore optimal (i.e. highest-scoring) under a different set of conditions.

One problem faced by all suboptimal methods is how to adequately sample the gigantic space of possibilities. Jaroszewski et al (71) sought to explore the size of alignment space between several pairs of small and medium-sized proteins (seven or fewer template secondary structures). The space was further constrained by only enumerating 'significantly different' alignments, thus disallowing gaps in template secondary structures and ignoring alignment variations in loop regions. Even with these restrictions, billions of alternative alignments were needed to find the correct alignment in some cases.

2.3.2 Difficulties in producing alternative alignments

The results of this study by Jaroszewski and colleagues adds detail to the problem of the enormous size of alignment space described in Section 2.1.1 and the related problem of alignment redundancy discussed in Section 2.1.2. For cases when the optimal alignment is significantly different from the correct alignment — a common occurrence in low homology sequence pairs — it is safe to conclude that DP-based alternative alignment methods cannot sample far enough away from optimal to return a correct or nearly correct alignment within a reasonable number of alternatives.

A further difficulty in using DP-based methods to produce alternative alignments is that, in practice, the density of alignment space becomes a computational problem: the number of alignments generated grows extremely quickly, which requires an equally large amount of memory and processing time. That is, it is not only impractical for a human researcher to sift through billions of alternative alignments, but at some point the number overwhelms the computer as well.

Finally, it should be noted that the problem associated with the local perspective of optimal DP methods discussed in Section 2.2.3 are equally true of DP-based suboptimal methods. The result is that a large portion of the ensemble of alternative alignments will therefore contain a variety of structural errors (e.g., unpaired strands, extended structures and over-stretched loops). These alignments are not modelable in the sense that these errors would result in poor structural models.

It is evident, therefore, that a suboptimal approach that is well-suited to finding the correct alignment between remotely homologous sequences would incorporate an improved sampling method, a technique for clustering similar alignments and a method for recognizing and avoiding regions of alignment space that contain structural errors. The goal of such a method would be to return a small number of modelable alignments free from structural errors that are sufficiently different from each other so as to ensure that many regions of alignment space are sampled — and not just the region surrounding the optimal alignment.

To create such a method, many questions must be resolved. How do we define redundancy between alignments? How can we sample alignment space broadly while keeping the number of returned alignments small? How can we check for alignments that would lead to bad models without performing the time-consuming task of modeling each one? The remainder of this thesis will describe a new alternative alignment method, the creation of which was motivated by the desire to find answers to these questions and address the shortcomings in the DP-based methods mentioned above.

3

Methods: The S4 Algorithm

Our approach to suboptimal alignments can be broadly divided into two parts: identifying promising regions of alignment space and then searching each region for a single best alignment. In both, the philosophy of initially casting a wide net and then narrowing the results is evident. In identifying promising regions of alignment space, we begin by finding the highest-scoring (i.e., most similar) fragments from across the alignment matrix, which will form the basis of all alignments in the final ensemble. The number of possible alignments through these fragments is still very large, however. A set of rules and filters is applied at this point to not only reduce this number significantly, but ensure that the remaining 'fragment alignments' are all biologically likely and modelable. Because the algorithm focuses on broadly sampling shifts of the query to template secondary structures, we call this method S4.

A similar approach is applied in the second part of S4 in which the fragment alignments are converted into full alignments. The fragments are used to define a narrow region of alignment space. A set of several hundred alignments are found within that region and the best one (as determined by a statistical potential) is chosen to represent that region. Again, a wide net is cast — the density of alignment space allows for many alignments to be found even in a narrow region — and then we narrow the results to a single representative.

A flowchart of the algorithm is given in the six steps of Figure 3.1. In terms of the two parts discussed above, Steps 1-4 identify the promising regions of alignment space and Steps 5 and 6 search those regions for a single best representative. Each step is discussed in detail below. Finally, an illustrative example is given that demonstrates these steps.

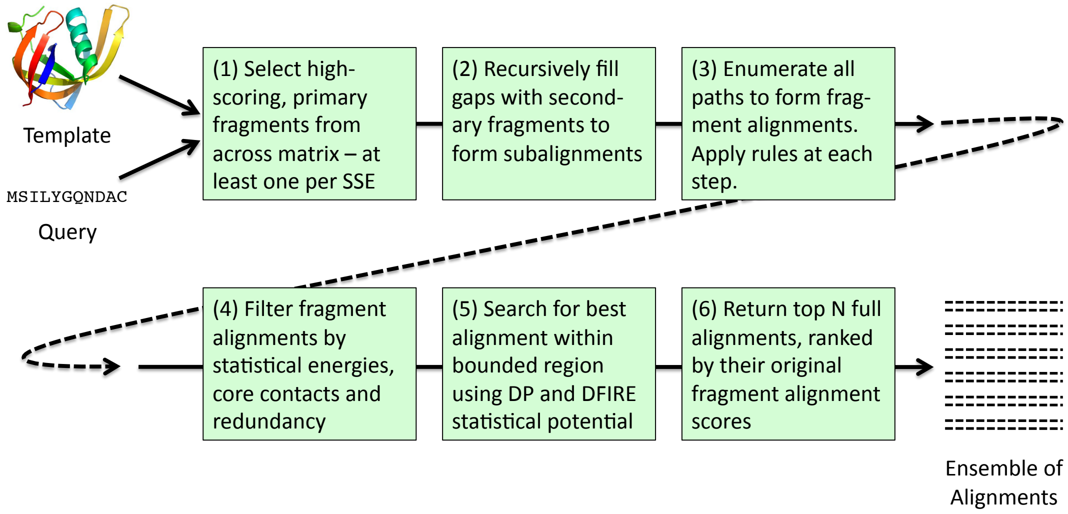

Template
MSILYGQNDAC
Query
(1) Select high-scoring, primary fragments from across matrix – at least one per SSE
(2) Recursively fill gaps with second-ary fragments to form subalignments
(3) Enumerate all paths to form frag-ment alignments. Apply rules at each step.
(4) Filter fragment alignments by statistical energies, core contacts and redundancy
(5) Search for best alignment within bounded region using DP and DFIRE statistical potential
(6) Return top N full alignments, ranked by their original fragment alignment scores
Ensemble of Alignments

Figure 3.1: Flowchart of the S4 algorithm - The numbers of the steps are used in the text and in the caption of Figure 3.2

3.1 The algorithm step-by-step

3.1.1 Steps 1 & 2: Selecting Fragments

Figure 3.2 illustrates the S4 algorithm and shows a typical dynamic programming matrix with the query sequence along the side and the template sequence across the top. The template sequence is divided into columns defined by its secondary structure elements. The highest scoring set of consecutively aligned residues within each diagonal of a column is called a fragment. Each fragment's score is the sum of the residue-residue similarity scores of the aligned pairs it contains, calculated based on the HMAP profiles (32) of the query and template sequences. To start the alignment process, the fragment from each column with the highest normalized score (the profile-profile similarity score divided by the length of the fragment) is added to a set of primary fragments (black lines in Figure 3.2). Next, several of the highest scoring remaining fragments from across all columns are selected to be primary fragments.

To avoid a combinatorial explosion of the number of possible alignments, the number of primary fragments must be kept low. However, by limiting this number, fragments that may be necessary to generate the correct alignment might not be available in the primary fragment set, especially in the case of remote homologs. In order to connect primary fragments in non-consecutive SSEs, we perform a recursive search for 'secondary' fragments (gray lines in the figure) to fill in the region defined by the fragment endpoints. For example, in Figure

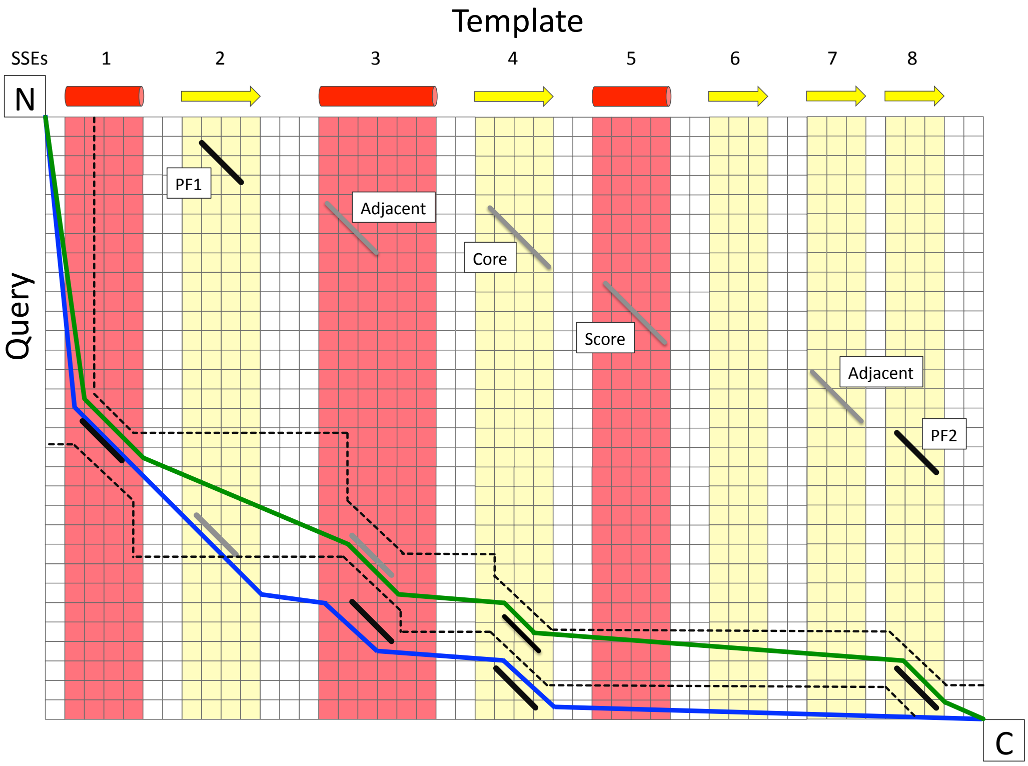


Figure 3.2: The S4 alignment matrix - The alignment matrix is depicted here with the template and its SSEs on the horizontal axis and the query sequence on the vertical. The black diagonal lines represent the high-scoring primary fragments chosen in **Step 1**. The gray lines are the secondary fragments which are found through a recursive method of filling the gaps between primary fragments in **Step 2**. In **Step 3**, the algorithm begins at the N-terminal of both sequences (upper-left corner) and proceeds to enumerate all paths to all primary fragments as well as all paths of secondary fragments between them. Rules and filters are applied during this enumeration, resulting in a set of valid fragment alignments that spans from the N- to the C-terminal (lower-right corner). The sets of fragments along the green and blue lines depict two examples of fragments alignments. Filters based on statistical energies and core contacts are applied to the full set of fragment alignments in **Step 4**. Boundaries are drawn narrowly around each fragment alignment and a single best full alignment is chosen to represent the region in **Step 5**. The final set of N alignments, where N is supplied by the user, is returned in **Step 6**.

3.2, two secondary fragments are chosen for being the highest-scoring secondary fragments that are adjacent to primary fragments PF1 and PF2. (An adjacent fragment is contained in a neighboring SSE and is on the same or a nearby diagonal.) Other secondary fragments are chosen by virtue of being high-scoring or in an SSE without which the alignment rules would be violated (e.g., a missing core strand). This process continues recursively until all gaps between fragments have been filled in. The procedure then steps back to associate secondary fragments into chains stretching from PF1 to PF2.

3.1.2 Steps 3 & 4: Enumerating fragment alignments

A fragment alignment is simply a set of fragments in order from the N- to C-terminal. It will by necessity contain primary fragments and typically several secondary fragments as well. The fragment alignments created in this step of the algorithm will form the basis of full alignments in the final step. To enumerate all such fragment alignments, S4 connects the N-terminal pseudo-fragment (upper-left corner of Figure 3.2) to each downstream primary fragment (either directly or through subalignments of secondary fragments). This process progresses to further downstream primary fragments until all alignments end at the C-terminal pseudo-fragment (bottom-right corner of Figure 3.2). After any connection is established, if an alignment fails to meet one of the conditions described below, the enumeration process is discontinued for that particular path (though some tests, such as exceeding a minimum contact order, can only be applied when the C-terminal is reached). It should be noted that this enumeration process is considered during the selection of primary and secondary fragments so that the number of fragment alignments to be enumerated is roughly 10 million.

Some of the conditions placed on the fragment alignments are based on the properties of the alignment itself and some are based on a 3D pseudomodel of the query. A pseudomodel for a set of fragments is constructed by simply copying the backbone and Cβ coordinates of residues of the template mutated to the identities of the corresponding aligned residues in the query (unaligned residues are ignored). Each of these conditions is described below.

Coverage: We are generally not interested in alignments that pair a very small number of residues. Therefore, only alignments in which 10% of the shorter sequence is aligned to the longer sequence are retained. Since only residues in template SSEs are counted in S4, this fraction represents a somewhat more stringent condition than it may initially appear.

Contact order: The contact order for a pseudomodel is defined here as the percentage of its SSE residues containing a Cβ that lies within 6Å of a Cβ from a residue in a different SSE. Fragment subsets whose pseudomodel has a contact order less than 65% of the contact order of the template itself are rejected. Making this threshold relative to the template ensures that extended models will not be built from compact templates, but if the template itself is extended, the fragment alignment will be kept.

Strand pairing: There are two general rules governing the pairing of beta strands in homologous proteins that can be used to eliminate bad alignments (72). First, a paired strand in the template should not become unpaired in the pseudomodel. Second, a core strand of a β-sheet in the template must be present in the pseudomodel if its flanking strands are also present. These simple ideas allow us to recognize many un-modelable fragment alignments.

Loop lengths: Fragment subsets are rejected if there are not enough residues in the query sequence to bridge the gap between any two consecutive fragments. Specifically, we require that $3.3\text{Å}\times(q_f - q_p) > d(t_p, t_f)$ where q_p is the index of the final query residue of the fragment preceding the loop, q_f is the index of the first query residue following the loop, and $d(t_p, t_f)$ denotes the distance in the template structure between the Cα atoms of the corresponding, aligned template residues. 3.3Å is a conservative estimate of the Cα-to-Cα length of a single amino acid residue.

Three other measures were used to eliminate fragment subsets that are unlikely to produce good models: preserved core contacts, query energy and template energy. For an alignment to be kept, all three of these measures must have values above the 66th-percentile for each measure and one of these three values had to surpass the 90th-percentile. The measures are listed below.

Preserved core contacts: A pair of residues in the template structure is considered to be a core contact if both residues in the pair are buried (60%

or more of surface area inaccessible), have Cβ atoms that are within 6Å and are both hydrophobic (amino acid types A, F, G, I, L, M, P, W, V and Y). An alignment that pairs hydrophobic amino acids in the query with template residues in a core contact generates a preserved core contact.

Statistical energy of query residues: An implementation of the DFIRE statistical potential (48) was used to evaluate each alignment by using the Cα and Cβ positions from the template with the amino acid types of the aligned query residues. A pseudo-Cβ position was determined for glycine residues based on the Cβ position in alanine. Loop residues were not considered in either the calculation of the statistical energies or in the tabulation of inter-residue distances that form the basis of this implementation of DFIRE. The value thus calculated, called the query energy, and the proximity of the alignment to native were found to be highly correlated.

Statistical energy of template residues: Similar to evaluating the statistical energy of the pseudomodel, we calculate the energy of the aligned template residues, which we term the template energy. In effect, this is a subset of the total statistical energy of the template structure. The motivation behind this is to recognize and remove alignments that pair query residues with an unlikely combination of template SSEs. This often occurs when the template is a multi-domain protein and the query is a single domain. In these cases, the highest scoring fragments may be spread out across multiple domains of the template in a structure that does not resemble a folded protein. Calculating this value allows S4 to eliminate many such alignments.

Redundancy: Lastly, to decrease the redundancy of the final results, some fragment alignments are removed due to their similarity to a higher-scoring alignment. Fragment alignments are considered redundant if they align to the same template SSEs, have all corresponding fragments within a shift of 4, and an inter-alignment distance (IAD) of less than 1. See Section 4.2 for an explanation of the IAD.

3.1.3 Steps 5 & 6: Constructing full alignments

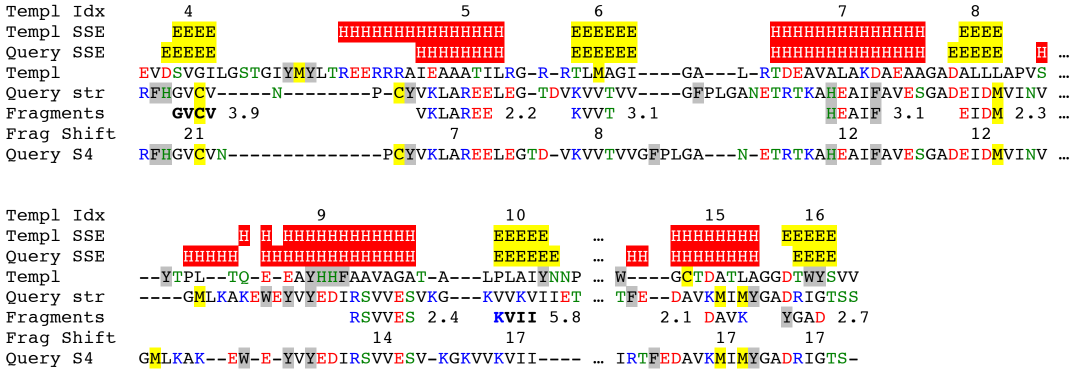
At this stage in the process, no full alignments in the normal sense have been created, only fragment alignments, which are just lists of fragments representing a region of alignment space. A round of alignment sampling using the full sequences of the query and template generates a final alignment from each fragment alignment. In this final step, alignments are restricted to a specific region of the dynamic program matrix. The boundaries of the region extend 3 residues above and below the fragments in each fragment alignment. The loop regions are constrained only by the boundaries of the surrounding fragments (dashed lines in Figure 3.2). Alignment sampling is carried out using the constrained Waterman approach, implemented in the program HMAP. (See Section 5.3.2 for an explanation of the constrained Waterman method.) Again, a pseudomodel is constructed for each alignment which is scored with DFIRE as described above.

The alignment with the best/lowest energy is selected to represent the original fragment alignment.

The S4 algorithm typically generates thousands of fragment subsets. A single, full alignment is generated for each one, starting with the highest-scoring, until N unique alignments have been found, where N is chosen by the user. The score of an alignment is simply the sum of the similarity scores of the paired residues in the original fragment alignment minus a flat penalty for each inserted residue. The insertion penalty only applies to residues inserted between template residues and is therefore used to encourage insertions at the termini. Deletions are not penalized since we found that structural considerations alone enabled us to disallow unreasonable gaps without an explicit penalty. For the results described in Chapters 5 and 6, a cap of 1,000 alignments was applied for each query/template pair, though evaluations of the top 100 alignments are given as well.

3.2 An illustrative example

Figure 3.3 depicts a portion of an alignment of two TIM barrel proteins that highlights many of the steps in the S4 algorithm. Referring to the flowchart in Figure 3.2, we can see that Step 1 succeeded in selecting two correct fragments in the initial set of primary fragments, shown in bold under template SSEs 4 and 10. (Not shown are two other correct primary fragments that were also found in SSEs 1 and 11.) The primary fragment in SSE 4 is correct, while the one in SSE 10 is shifted by two. The high similarity scores for these two fragments (relative to other fragments in their columns) is evidenced by their high z-scores (3.9 and 5.8), which are listed alongside the fragment. By virtue of being among the highest-scoring (i.e., most similar) fragments in the alignment matrix, these fragments were chosen to become primary fragments.



```
Templ Idx       4                        5           6                     7           8
Templ SSE      EEEE           HHHHHHHHHHHHHHH      EEEEEE            HHHHHHHHHHHHHH   EEEE
Query SSE     EEEEE                  HHHHHHHH      EEEEEE            HHHHHHHHHHHHHH  EEEEE   H …
Templ       EVDSVGILGSTGIYMYLTREERRRAIEAAATILRG-R-RTLMAGI----GA---L-RTDEAVALAKDAEAAGADALLLAPVS …
Query str   RFHGVCV-----N--------P-CYVKLAREELEG-TDVKVVTVV----GFPLGANETRTKAHEAIFAVESGADEIDMVINV …
Fragments      GVCV 3.9              VKLAREE 2.2   KVVT 3.1               HEAIF 3.1   EIDM 2.3 …
Frag Shift      21                      7            8                     12          12
Query S4    RFHGVCVN--------------PCYVKLAREELEGTD-VKVVTVVGFPLGA---N-ETRTKAHEAIFAVESGADEIDMVINV …

Templ Idx                   9                10                15       16
Templ SSE            H H HHHHHHHHHHHH       EEEEE    …      HHHHHHHH  EEEEE
Query SSE       HHHHH  HHHHHHHHHHHHHH       EEEEEE   …  HH  HHHHHHHH   EEEE
Templ       --YTPL--TQ-E-EAYHHFAAVAGAT-A---LPLAIYNNP … W----GCTDATLAGGDTWYSVV
Query str   ----GMLKAKEWEYVYEDIRSVVESVKG---KVVKVIIET … TFE--DAVKMIMYGADRIGTSS
Fragments                      RSVVES 2.4   KVII 5.8       2.1 DAVK   YGAD 2.7
Frag Shift                       14          17                  17       17
Query S4    GMLKAK--EW-E-YVYEDIRSVVESV-KGKVVKVII---- … IRTFEDAVKMIMYGADRIGTS-
```

Figure 3.3: An illustrative example of the S4 algorithm - The top five lines show the correct alignment of the template and query sequences, along with the pattern of their SSEs and the index of the template SSEs for reference. The Fragments line shows the fragments selected by S4 that led to the best alignment returned out of 1000. The number following the fragment is its z-score, which describes the similarity of the fragment relative to others in the same column/template SSE. The 'Frag Shift' line gives the diagonal of the DP matrix on which each fragment lies. As defined here, a shift of 0 would be assigned to the diagonal which begins at the N-terminus of both sequences in the upper-left corner of the DP matrix. An increase (or decrease) in the fragment shift from left-to-right in the figure implies an insertion (or deletion). The last line is the final S4 alignment that was returned based on the fragments shown.

During Step 2 of the algorithm, pairs of primary fragments are connected by subalignments consisting of combinations of secondary fragments. Many such subalignments will be found in the empty region between each pair of primary fragments, though a large fraction of these will be filtered out due to problems such as missing strands and over-stretched loops. The subalignment of secondary fragments that led to the correct S4 alignment are shown under SSEs 5-9. These fragments were chosen due to their relatively high scores and adjacency to primary fragments. An example of the latter point can be seen with the secondary fragment in SSE 9. Its shift (or diagonal on the DP matrix) is only different by 3 (14 vs. 17) from the primary fragment in SSE 10. By virtue of being relatively high-scoring, on a nearby diagonal and in a neighboring SSE, it is selected as a secondary fragment. This process continues recursively, selecting the high-scoring adjacent secondary fragment in SSE 8 and then in SSE 7. However, between the primary fragment in SSE 4 and the secondary fragment in SSE 5, there is a significant deletion of 14 residues (the shift of the fragments decreases from 21 to 7). The secondary fragment in SSE 5 is not adjacent and therefore must have been found either due to its own high score or because it is adjacent to the secondary fragment found in SSE 6, which is quite high-scoring and a necessary core strand. In this manner, the gaps between primary fragments are filled in. The particular subalignment shown between SSEs 4 and 10 is just one of several that were found to connect the primary fragments. The different reasons for choosing secondary fragments results in S4's sampling of a range of insertions and deletions.

Step 3 enumerates all combinations of both primary fragments and the secondary fragments which often connect them. The result is a set that contains many thousands of fragment alignments, such as the one shown in the Fragments line of Figure 3.3. The fragment alignment shown here, which led to a correct final alignment, also passed all of S4's alignment rules, which are checked during the enumeration process.

After eliminating many fragment alignments for redundancy and low statistical scores in Step 4, in Step 5 the algorithm searches for the best full alignment to represent each remaining fragment alignment. A bounded region is drawn narrowly around the fragments (as shown in Figure 3.2) and the Waterman alternative alignment technique is used to find a set of full alignments. These are evaluated as pseudomodels by DFIRE and the best/lowest-energy alignment is selected. As we can see in the 'Query S4' line of Figure 3.3, which shows the final S4 alignment that resulted from the fragments above it, this step not only fills in the gaps in loop regions, but often improves the alignment to template SSEs. Of the three fragments that were shifted from their correct positions (SSEs 10, 15 and 16), two of these (15 and 16) were corrected during Step 5. It should also be noted that this search of a narrow region of alignment space did not disturb the correct alignment of the other six fragments shown. This process continues until the top N alignments, ranked by their original S4 scores calculated in Step 3, are determined.

4

Methods: Techniques for Alignment Comparison

In the previous discussion of sequence alignment space (Section 2.1), it was noted that this concept is a difficult one to visualize. A consequence of this problem is that though any casual observer may be able to identify two alignments that look very similar, a precise definition of the similarity or distance between alignments is somewhat elusive.

Like sequence and structure space, which have measures such as percent identity and RMSD, respectively, to quantify similarity, we need a corresponding tool for alignment space. Such a measure would allow us to accomplish two related goals: assessing alignment quality and determining thresholds for clustering alignments to reduce redundancy. The first of these is accomplished by comparing an output alignment from our method to the correct/structure-based alignment. And the second goal is achieved by using the similarity measure to compare alignments in an ensemble to each other.

4.1 Alignment Comparison Scores

4.1.1 FDx

The study of sequence alignments is an established field with several existing measures for determining the accuracy of a sequence alignment. By far the most common is the FDx score (73).

$$\text{FDx} = \frac{\text{Number of pairs aligned within } x \text{ residues}}{\text{Number of aligned pairs in the reference alignment}} \tag{4.1}$$

This measure, also referred to as ALx, is popular perhaps due to its simplicity. An FD0 of 90% means that 9 out of 10 aligned pairs are identical between the two alignments. An FD2 of 100% indicates that all aligned pairs in the two alignments are within two residues of each other.

If the upside of using the FDx score is its simplicity, the downside is that this simple measure ignores some of the nuance of alignment space. Figure 4.1 demonstrates this problem. Compared to the correct, green alignment, the blue alignments will have equally good FD0 scores of ∼50%, but one is clearly much more similar. In another example, the red alignments will both have FD0 scores of 0%, though only one is truly a bad alignment. The problem with using the FDx scores is that the score is not proportional to our intuitive notion of similarity. While we can be sure that alignments with an FD0 of 99% will be unambiguously similar, as we see in Figure 4.1, alignments with an FD0 of 50% or even 0% may or may not be.

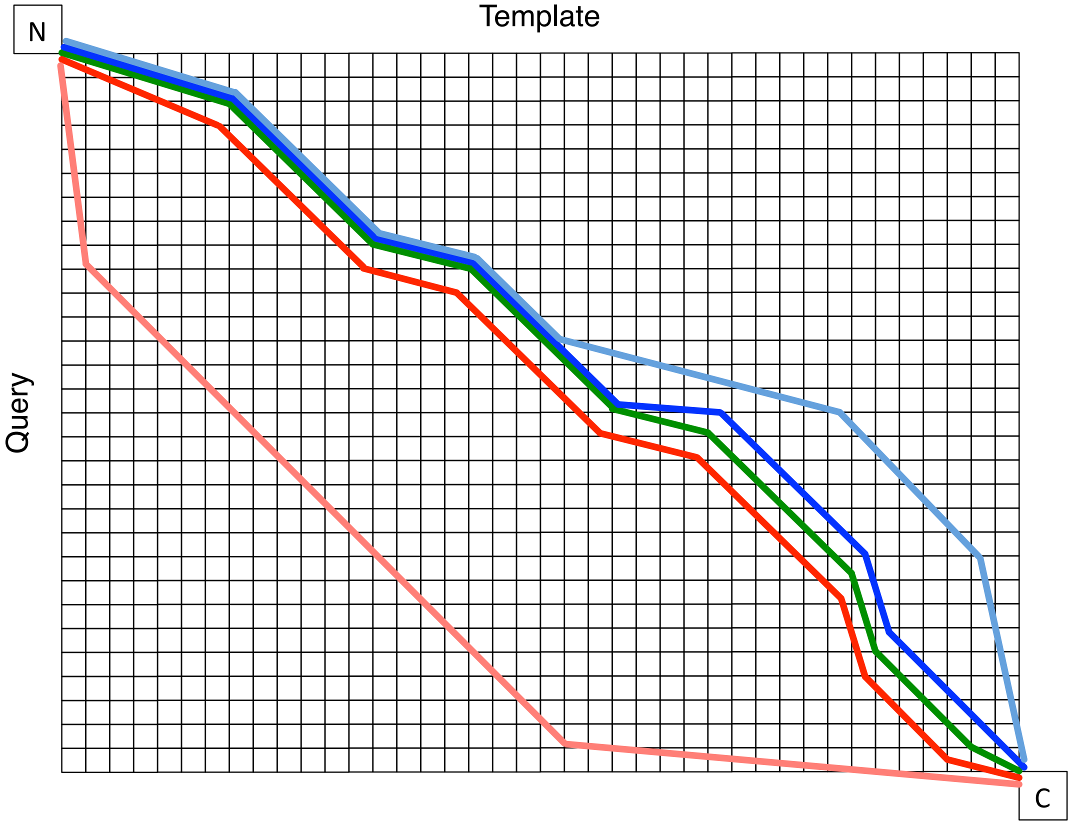


Figure 4.1: Problems measuring similarity with the FDx score - Using the green alignment as a reference, both blue alignments will have an FD0 score of 50% and both red alignments will have an FD0 score of 0%.

FD2 or FD4 scores are a bit more flexible, of course, but the problem described in Figure 4.1 still applies. The fact that the FDx score uses a particular cutoff, beyond which an error of x+1 in a residue is penalized equally to one that is much more seriously in error, is the reason for this confusion.

Depending on the situation in which an alignment is being performed, one may want to sample many alignments (as is the case in aligning remote homologs) and then later cluster alignments to reduce redundancy. The FDx scores are not well-suited to this task. For example, we may wish to cluster together the green, dark blue and dark red alignments in Figure 4.1. Using a threshold of 40%, for example, both blue alignments would be clustered with the green alignment, which is undesirable. However, there is no threshold that clusters only the three alignments that are clearly similar.

4.1.2 Other alignment comparison methods

Some new alignment scoring methods have been developed to address some of the problems discussed above. A method developed by Cline *et al* (74) gives the highest scores to pairs that are identical between two alignments, like FD0, but also gives some credit to pairs that are shifted slightly, up to a user-adjustable cutoff (set to five by the method's developers). In a sense, this score is similar to an average of several FDx scores, with the weight of the score decreasing as x increases.

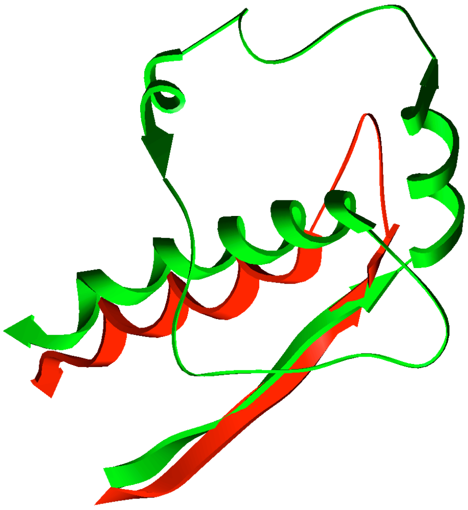
Another method developed by Chen and Kihara (75) treats the alignment as a path through the alignment matrix (such as those depicted in Figure 4.1). The horizontal and vertical distance between the two alignments (i.e., shifts in the template and query, respectively) are measured and the average of these two distances, called the ALD, is recorded. The ALD is calculated for each aligned position in one sequence. Finally, all the ALDs at each position are themselves averaged to return the final measure, termed the gALD. This method gets closer to the notion of literally measuring how far apart two sequences are. Importantly, it would seem to return more sensible results for the problematic alignment comparisons depicted in Figure 4.1.

One last possibility for comparing alignments should be mentioned since our interest is not just the building of quality alignments, but specifically the use of sequence alignment to predict protein structures. One could determine the similarity between two sequence-to-structure alignments by building models of each and using a model evaluation tool such as TM-score (35) or RMSD to compare the similarity of the resulting models. This method would likely be able to discern which alignments in Figure 4.1 are similar and which are not. While this is a suitable method for determining alignment quality in comparison to a structure-based sequence alignment, results of which we in fact report in Section 5.4.2, it is not a reasonable way of accomplishing the second goal of alignment comparison methods. It would be too slow to process thousands of alignments in an ensemble for clustering by building a model for each one.

4.1.3 FD Scores for Remote Homologs

As noted above, the FD0 score is overly sensitive to even single residue shifts away from the correct alignment. But in remote homology cases, though the overall structural similarity may be clear, the precise structure alignment may be slightly ambiguous. Using FD0 under these circumstances would result in the penalizing of good alignments. Allowing a shift of up to two residues in the FD2 score better accommodates the ambiguity inherent in remote homolog alignments.

The second change to this score was to limit it to SSEs in the template. This adjustment was also due to the nature of structural homologs with low sequence similarity. Though the alignment in SSEs is clear within one or two residues, there is often little or no structural similarity in loop regions. Regardless, structure alignment methods return a correspondence in these regions, such as that seen in Figure 4.2. If alignments in such regions are taken to be correct, however, they are not just meaningless; they are misleading. Penalizing alignments for not reporting the correct random correspondence of residues in non-homologous loop regions is erroneous. Similarly, considering an alignment to be more accurate if it correctly guesses this loop alignment is also a mistake. For these reasons, we implemented an FDS2 score — the 'S' stands for SSE — that reports the fraction of template SSE residues aligned within two of their position in the correct alignment.



```
T SSE   EEEEEEEECHHHHHHHHHHCCCCCCCCCHHHHHHCCCCCCCCCCCCCHHHHHHHHHHHHHHHHHHHH
Q SSE   CEEEEEE-EC--C--CC---------------------------------C-CHHHHHHHHHHHHH-
Templ   ISVILFLNKQDLLAEKVLAGKSKIEDYFPEFARYTTPEDATPEPGEDPRVTRAKYFIRDEFLRISTA
Query   KLASVFS-ST--G--TG---------------------------------G-GQEQTITSTWTTLA-
```

Figure 4.2: Aligning non-homologous loops - This excerpt from a structure alignment of template d1azsc2 (green) and query 2rg1b (red) shows structurally homologous SSEs and very non-homologous loops. The structure-based sequence alignment matches several query loop residues to dissimilar partners in the template.

4.2 The Inter-Alignment Distance

We have developed a new method of determining the similarity between two alignments which we term the inter-alignment distance (IAD). It is similar to the gALD measure in that it begins by visualizing the alignments as paths on the matrix, as shown in Figure 4.3.

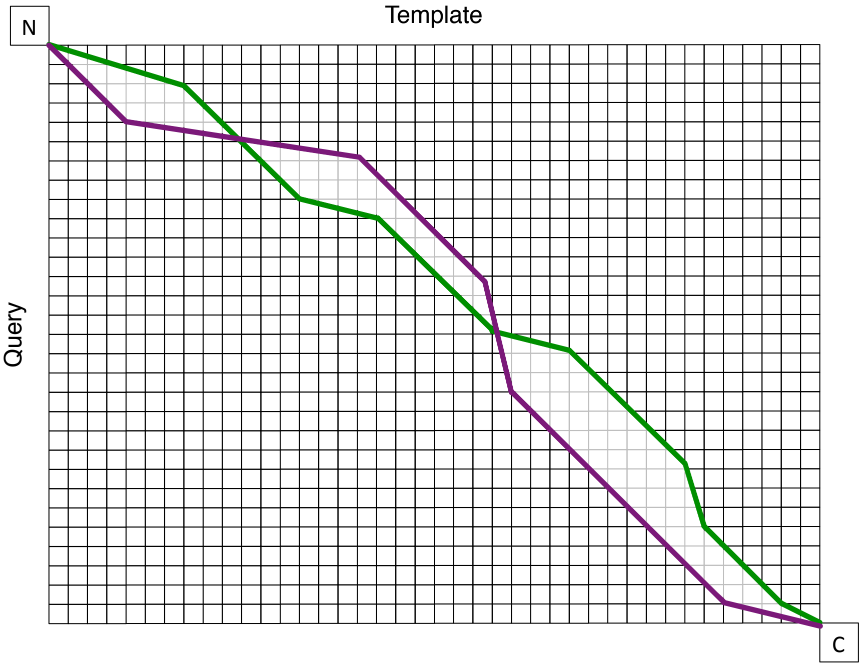


Figure 4.3: Calculating the inter-alignment distance - Two alignments are plotted as paths through the matrix. The total area between them (all three regions) can be calculated and divided by the template length to return the IAD.

The calculation of the IAD is very simple. The two alignments/paths trace out a closed shape in the matrix. The area of this closed shape is proportional to how 'far apart' the alignments are, with all areas contributing positively. An area of zero would mean that the alignments are identical. In agreement with our intuitive sense, the area between the alignments clearly increases as their similarity decreases. To normalize this measure between different pairs of alignments, we divide the area by the length of the template sequence to calculate the final value of the IAD. Dividing a shape by its width returns its average height, which in this case can be interpreted as the average shift (measured in residues) of the query sequence from its position in the other alignment.

There are several advantages to using the IAD. One is that it is hard to confuse this measure with examples like those in Figure 4.1. That is, a high IAD implies a significant dissimilarity exists between the two alignments, while a low IAD means the two must be close. In Figure 4.1, the light blue and light red alignments would have high IADs when compared to the other alignments and, with a low threshold set, would not be clustered together with the more similar alignments.

4.2.1 Mathematical properties of the IAD

It should be noted that the IAD score is symmetric. That is, for two alignments A and B, $IAD(A, B) = IAD(B, A)$. This is not the case for some other alignment methods. For example, in FDx scores, the denominator in the calculation is the number of pairs in one of the two alignments.

More interestingly, the IAD appears to satisfy the triangle inequality. A formal proof is beyond the scope of our work, but it has been tested on many thousands of alignments in many different sequence pairs and always found to be true. Figure 4.4 suggests why this inequality holds. The areas[1] they define are marked with Roman numerals I-V. If the triangle inequality is true, then for three alignments A, B and C, we have

$$IAD(A,B) \leq IAD(A,C) + IAD(B,C) \tag{4.2}$$

We can demonstrate that this is true for this case by simply filling in the areas from Figure 4.4. We then have

$$\begin{aligned} II + III + IV &\leq (I + II + IV + V) + (I + III + V) \\ 0 &\leq 2 \times (I + V) \end{aligned}$$

4.2.2 Measuring sampling with the IAD

Rearranging Equation 4.2 and renaming our alignments allows us to use the IAD to think about sampling alignment space in a quantitative manner. If we let alignment A represent the optimal alignment (opt), B the correct alignment

[1] In the equations that follow, we compare areas and ignore the fact that the IAD is actually the area divided by the length of the template. That factor is common to all areas and therefore would cancel out.

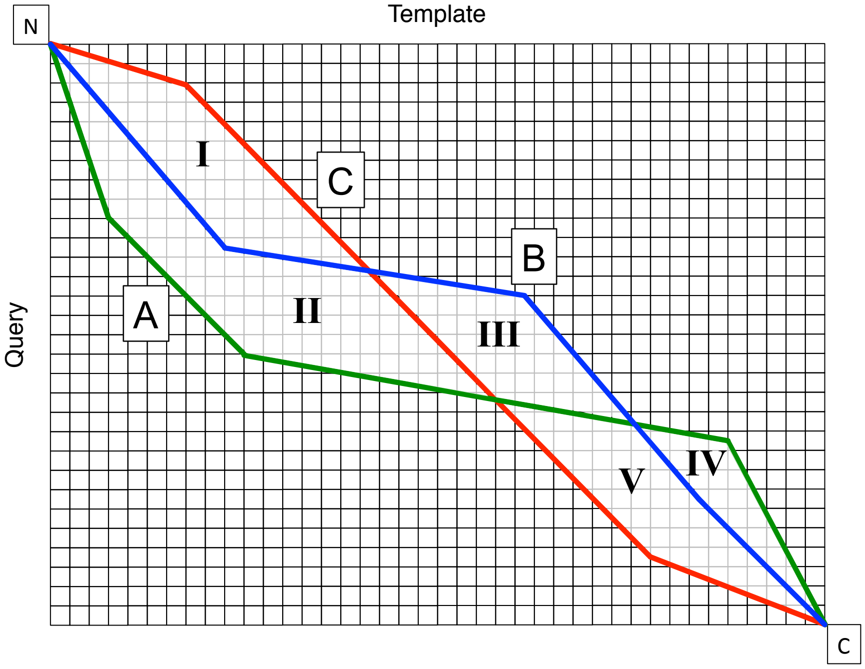


Figure 4.4: The IAD satisfies the triangle inequality - Three paths through the alignment matrix representing different alignments define the five regions separating them.

and C a suboptimal alignment from an ensemble (sub), we then obtain

$$IAD(correct, sub) \geq IAD(correct, opt) - IAD(opt, sub) \tag{4.3}$$

Equation 4.3 establishes a lower bound on how close we can expect one of our suboptimal alignments to be to native given the error of the initial, optimal alignment and the distance away from optimal that our ensemble of suboptimal alignments searches. For example, if our optimal alignment has an IAD from correct of 10 and the ensemble of suboptimal alignments are all within an IAD of 3 from the optimal, then it is impossible for any of these suboptimal alignments to be closer than an IAD of 7 from the correct alignment. The IAD was used in this type of quantitative analysis of alignment space sampling in Section 5.4.4.

It should be noted that the IAD is just a distance measure and has no indication of direction. Returning to the analysis of the hypothetical example above, it is possible to have an ensemble of suboptimal alignments that samples regions of alignment space with IADs of 10 or more from the optimal alignment and still not approach the native alignment. Equation 4.3 is only a lower bound, not a guarantee of success. The reason is, again, the density of alignment space (see Sections 2.1.1 and 2.1.2). In a different take on the redundancy problem outlined in Figure 2.2, we could attempt to find all alignments within an IAD of 10 from a given alignment, such as the optimal alignment in this example. One of the alignments in this set would have to be the correct alignment. But the enormous number of other possibilities means that while sampling alignment space broadly is a necessary condition for finding the correct alignment, it is clearly not sufficient. A suboptimal alignment method needs to carefully choose which areas to sample.

4.3 Conclusion

The complexity of alignment space means that no single method captures perfectly all the different ways two alignments can be similar. Our approach in this work has been to use several different methods for reporting results when practical. We use the FDS2 score for describing accuracy in SSE regions between remote homologs. The IAD measure describes the overall closeness of an alignment being tested for either assessing its quality against the correct alignment or clustering it with others in the ensemble to reduce redundancy. And, finally, since our interest is ultimately in structure prediction, we also build models and evaluate them against the native query structure to determine the quality of the underlying alignments.

5

Validation on CASP Targets

5.1 Abstract

For the initial test of the S4 algorithm, we selected a set of target proteins from past CASP experiments. Since S4 is intended to find the most remote homologies between sequences and structures, for each target we found structural homologs with very low sequence identity to serve as templates. The resulting test set is very challenging and outside the realm of most previous attempts at finding remote homologies. This set contains 2,898 query/template pairs and is heavily populated by those with low sequence identity: over 90% have less than 20% identity and more than 60% have less than 10% identity.

Our results indicate that S4 constitutes a significant improvement over traditional dynamic programming-based suboptimal and optimal alignment methods

using the same similarity score. Since the other methods share the same similarity score as S4, the improvement in alignment quality must arise from the algorithm's broader and more nuanced sampling of alignment space. Because of its ability to consistently find the correct alignment between remote homologs, S4 therefore offers the possibility of greatly expanding the set of sequences that can currently be modeled by homology.

5.2 Introduction

Most protein sequences do not have an experimentally determined structure and at least 40% do not even have a close sequence homologue with a known structure (76). Nevertheless, the current Protein Data Bank (PDB) (28) is thought to represent structure space nearly exhaustively (23, 24, 25). Therefore, for most proteins, a structural homolog that can serve as a 'template' for modeling at least part of its structure is likely to exist. However, the degree of sequence similarity will generally be too low to allow an accurate sequence alignment to be found (29). The central problem is that current alignment methods based on dynamic programming (DP) (77) generate the unique optimal alignment (the alignment producing the highest score based on a residue-residue similarity score and a gap penalty), while the correct alignment (producing the most accurate model) is not guaranteed to be optimal in terms of this score at low sequence identity ranges.

Because of this difficulty, it is generally necessary to consider an ensemble of alternative alignments in order to produce an accurate model when aligning to a remotely homologous template. Such ensembles are frequently called suboptimal since by necessity they have lower scores than the optimal alignment produced by DP. However, it is important to emphasize that they are only suboptimal in the context of a DP sequence alignment score; the expectation is that one of them will produce a more accurate model than the optimally scoring alignment when evaluated in structural terms.

A number of alternative alignment methods have been developed (see Section 2.3.1), but all of these methods represent variations on the standard dynamic programming algorithm that allow it to produce an ensemble of results. As described in Chapter 3, S4 takes a fundamentally different approach. In validation tests that we describe below, we show that S4 generally produces an accurate alignment within a set of 100 top-ranked alternatives (based on its scoring algorithm) and almost always within a set of 1000 alternative alignments. The utility of the S4 approach is most evident when the target/template sequence identities are low, but S4 also improves accuracy when the homology is clear. Our results are shown to constitute a significant improvement over DP-based alternative alignment methods, which we show is due to unique features of the algorithm that allow S4 to overcome the difficulties associated with keeping the set of alternative alignments manageably small while still broadly searching alignment space.

The ability to generate a small set of alignments likely to contain the correct one suggests that S4, when combined with structure- and function-based model evaluation procedures, offers the possibility of significantly improving the accuracy of homology models, extending the number of sequences that can currently be modeled based on existing structures in the PDB, and detecting remote relationships between sequences and structural templates.

5.3 Methods

5.3.1 Creation of the test set

S4 was tested on a set of targets from the CASP experiments (T0129-T0359) (36). To identify potential templates for each target sequence, the target was structurally aligned to other proteins in the PDB using the ska program (34, 38). Templates were then selected to ensure that a meaningful structural relationship existed within each target-template pair. Several conditions had to be met: the protein structural distance (PSD) (34) could not exceed 0.5 (corresponding to a maximum RMSD of about 3.5); the sequence identity was less than 50%; and a 'pseudomodel' of the target built from the aligned portions of the structure-based sequence alignment and based on the template structure had to return a TM-score (35) of 0.5 or greater compared against the native query structure. Also, proteins of length greater than 350 residues were not considered.

The resulting test set contained 2,898 query sequence/template pairs and was heavily populated by pairs with low sequence identity: over 90% of all pairs had less than 20% identity and more than 60% had less than 10% identity. Overall, there were 120 queries with an average of 24 templates per target.

5.3.2 Dynamic programming-based methods

The methods referred to as UW and CW in Results and Figures 5.2 and 5.4 are straight-forward implementations of Waterman's method (66) that incorporate the HMAP scoring function and gap penalty. In the UW approach, which is essentially identical to Waterman, alternate alignments are found by allowing the DP procedure to branch at any point where doing so will lead to an alignment with a score within δ of optimal. Branching to alternate alignments for the CW approach is allowed only when moving between SSE and loop regions. It is impossible to know which value of δ will generate an ensemble of a desired size. To generate the alignments for comparison, we started with very small values for δ and steadily increased it until the ensemble size exceeded 1000. We then sorted the alignments by their DP-based score and kept only the top 1000.

5.3.3 Building and evaluating models

Models were built with the program Nest (38) for all S4 alignments, the optimal HMAP alignment and the correct/structure-based alignment. TM-score (35) was used to evaluate the accuracy of the model compared to the native query structure.

5.4 Results

5.4.1 Improvement in alignment accuracy

We define the correct alignment to be the structure-based sequence alignment between the target and template. In the evaluation of S4 described below, it is important to recognize that an alignment can still be considered accurate even if it does not correspond to the correct alignment residue for residue. For example, consider a situation in which a template has a helix with an axis that is at an angle with respect to that of the topologically equivalent helix in the target. Because of this intrinsic dissimilarity of the template and query structures, no alignment in this region can be considered strictly correct even though there may be residues in the target and template that occupy roughly equivalent positions in space. The same difficulty occurs in the alignment of loop regions and also in β-strands, where β-bulges can affect alignment accuracy. However, it is clearly desirable — and still accurate, we would argue — for an alignment algorithm to pair residues in topologically equivalent SSEs, even if this pairing does not exactly correspond to the structure-based sequence alignment because of conformational differences.

Because of these issues, we evaluate S4 alignments using two different measures and also directly assess improvement in the quality of the models produced from the alignments. Figure 5.1 plots the inter-alignment distance (IAD) for the best alignment generated by S4 and the single optimal alignment produced by two DP-based profile-profile alignment methods, HMAP (32) and SP3 (33). As described in Methods, IAD corresponds to the average deviation of a given alignment from the correct alignment. Thus, an IAD of 2 implies that, on average, each residue is shifted by two away from its position in the correct alignment, and further implies that topologically equivalent SSEs have been correctly paired.

While Figure 5.1 is not a fair comparison in the sense that the best alignment out of an ensemble of 100 or 1000 is plotted against a single optimal alignment, the figure does demonstrate the importance of considering alternative alignments. The order of query/template pairs is determined by the IAD of the optimal alignment and moving left-to-right in the graph corresponds roughly with alignment pairs that range from high to low identity. With very few exceptions, S4 generates an alignment with improved accuracy at all sequence identity levels and the improvement is quite dramatic at lower identities when the optimal alignment is severely flawed.

Table 5.1 presents this explicitly, showing IAD values for the different methods averaged over all pairs in several ranges of sequence identity. At the 0-5% sequence identity range, the average IAD for the optimal alignment is over 13, implying that many topologically equivalent SSEs are not correctly paired, and highlighting the advantage of sampling alignment space with S4 in the low identity regime. We note that at the highest identity levels (over 30%) the single, top-ranked alignment produced by S4 is essentially identical to the alignment produced by DP-based optimal approaches: the number of correctly aligned

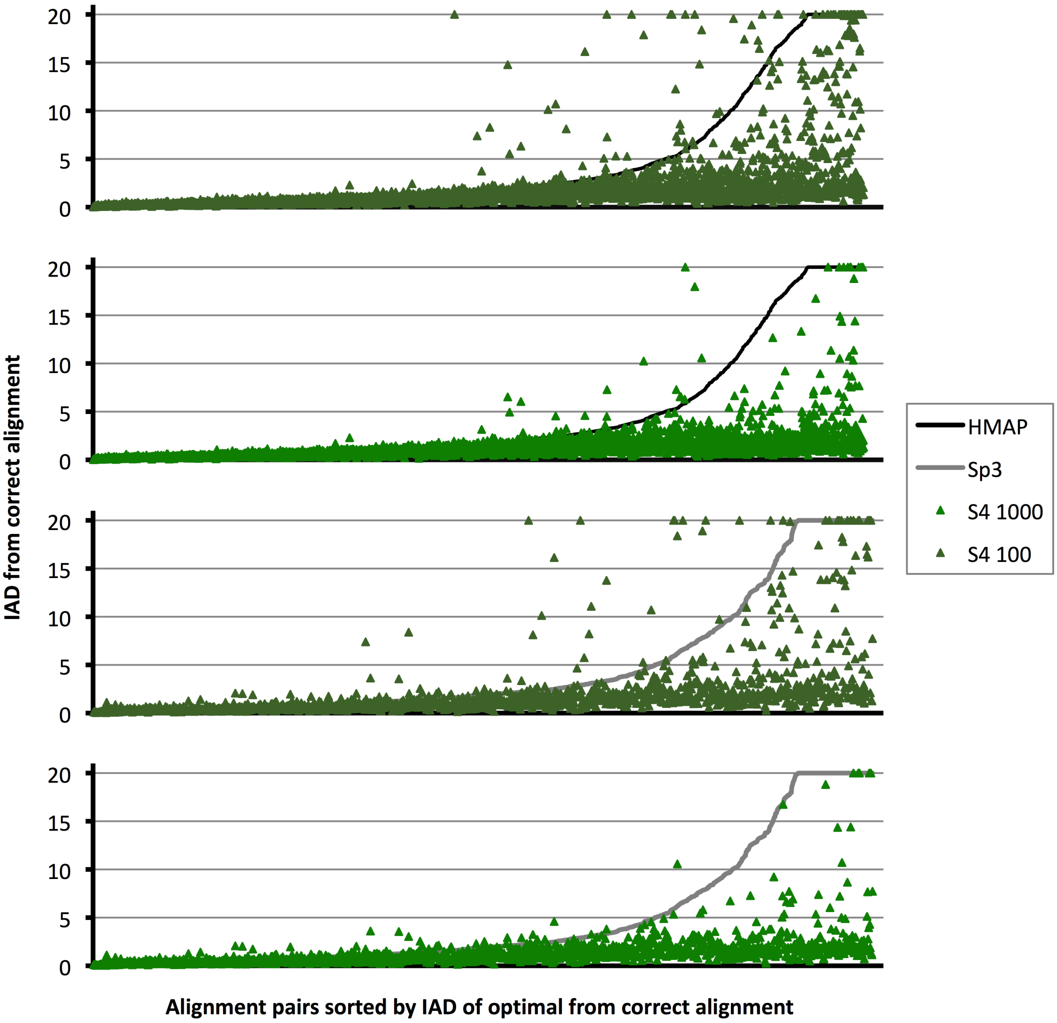


Figure 5.1: Alternate alignments are needed for remote homologs - The alignments are ordered by increasing IAD of the optimal alignments, either HMAP or Sp3. (See Section 4.2 for explanation of the IAD.) The triangles represent the lowest/best IAD out of the top 100 or 1000 S4 alignments. The HMAP graphs contain all 2,898 alignment pairs in the set, while the Sp3 graphs contain only a subset of 1430 pairs for which comparable alignments were obtainable. IADs are capped at 20 in order to show more detail for the majority of alignments with smaller IAD values.

residues in the top-ranked S4 alignment is within 2% of that of the optimal alignment (data not shown). Hence, S4 can be used exactly as DP-based approaches at this sequence identity level, but there is still a significant chance of generating an improved alignment if alternatives are considered. To determine the extent to which the improvement in alignment quality of S4 relative to the optimal alignment was due simply to the increased number of alignments generated, we also compared S4 to two versions of the conventional DP-based Waterman algorithm for generating alternate alignments (66), which have been implemented in-house as part of HMAP. A problem with a strict implementation of this algorithm (Unconstrained Waterman, UW) is that alignments can be generated that are not meaningfully different: variations in loop regions produce essentially equivalent models. Thus, we also implemented a modified algorithm (Constrained Waterman, CW), which is allowed to sample alignment variations at the transition point between template SSE and loop regions, but not within them. (See 5.3.2 for more detail.)

While the IAD is effective in measuring overall alignment accuracy, it does not give a specific fraction of how many residues are within a specified distance from their correct alignment. For this reason and for easier comparison with the sequence alignment literature, Figure 5.2 uses the more traditional FD2 measure (73), but restricts it to regions corresponding to template SSEs. This adapted measure, which we call FDS2, is simply the percentage of such residues that are within 2 residues of their position in the correct alignment. (The FDS2 measure is discussed in detail in Section 4.1.3.)

		Average IAD			Improvement	
ID(%)	# Pairs	S4 1000	S4 100	HMAP Opt	S4 1000	S4 100
20-50	252	0.323	0.335	0.371	12.9%	9.8%
15-20	275	0.504	0.526	0.657	23.3%	19.9%
10-15	592	0.881	1.051	1.750	49.7%	39.9%
5-10	1287	1.628	2.458	6.318	74.2%	61.1%
0-5	519	2.322	4.290	13.716	83.1%	68.7%

ID(%)	# Pairs	S4 1000	S4 100	Sp3 Opt	S4 1000	S4 100
20-50	124	0.306	0.316	0.321	4.8%	1.6%
15-20	138	0.500	0.525	0.582	14.1%	9.8%
10-15	308	0.993	1.214	2.258	58.7%	46.2%
5-10	610	1.627	2.523	7.500	78.3%	66.4%
0-5	250	2.349	4.769	16.049	85.4%	70.3%

Table 5.1: Improvement over optimal alignments - The top chart shows S4's performance compared to the DP-based optimal alignment program HMAP, with which it shares a scoring function. The bottom chart makes the same comparison with the optimal alignment program Sp3.

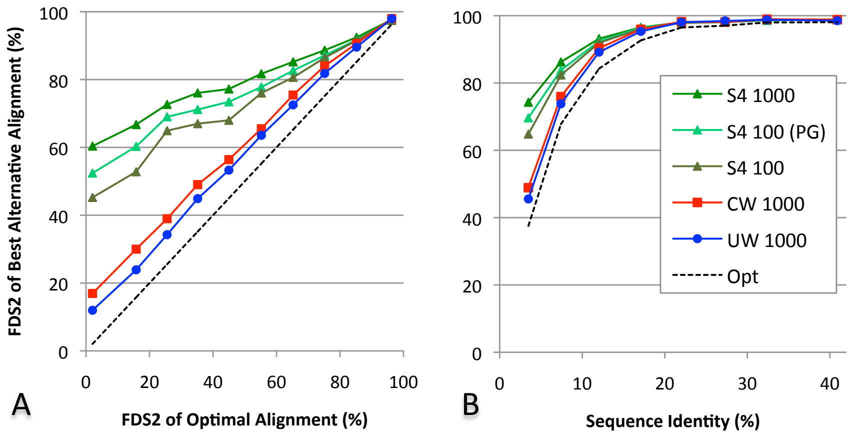


Figure 5.2: Evaluating the improvement in alignment accuracy - (a) The alignment pairs were put in groups along the horizontal axis based on the FDS2 of the optimal alignment. (i.e., 0-10%, 10-20%, etc.). Using the accuracy of the optimal alignment as a baseline, the improvement of the best suboptimal alignment is the distance above the dotted line. The data points for the alternative methods represent the average over all pairs in the group of the highest FDS2 in the ensemble. The number of alignments generated by each method is given in the inset legend. (b) Again using the FDS2 measure of accuracy, the alignment methods are compared versus sequence identity.

Figure 5.2a depicts the highest FDS2 in the ensemble from each method as a function of the FDS2 of the optimal alignment. Thus, the distance above the dotted line represents the improvement over optimal for the best alternative alignment generated. S4 is seen to significantly outperform the DP-based optimal and suboptimal algorithms, particularly when the optimal alignment is flawed. Indeed even the best alignment out of the top 100 S4 alignments is significantly better than the best out of 1000 from the DP-based methods. A further improvement in accuracy can be obtained by modeling the ensemble of 1000 alignments and using the pG score (50, 51), which is a normalized ProsaII score (49), to select the top 100. This is shown as "S4 100 (PG)" in the figure. Panel B shows the same data, but as a function of sequence identity. Again, we see a significant improvement compared to all DP-based methods for aligning remote homologs, even when using an ensemble one tenth as large.

5.4.2 Evaluation of Models from S4 alignments

The results shown in Figure 5.1 suggest that S4 generates alignments that are much improved over DP-based optimal methods, but since the IADs of the best S4 alignments are not 0 (i.e., the S4 alignments are not identical to the correct alignment) an important question is whether these alignments produce reasonable and accurate models. Using the TM-score (35) to compare models to the native query structure, Figure 5.3 shows that the models built from the best S4 alignment are significantly more accurate than models made from the optimal alignment.

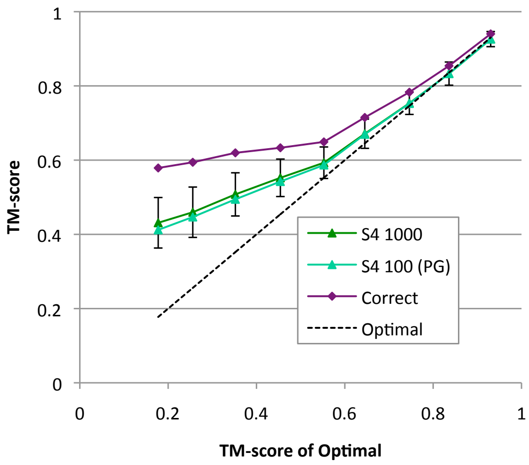


Figure 5.3: Model quality from alignments built from S4 - The test set was ranked according to the optimal TM-score and then divided into nine groups (0-0.2, 0.2-0.3, etc.). The dotted black line represents the TM-score obtained from a model of the optimal alignment and is thus a 45-degree line. The average TM-score for models made from the correct/structure-based alignment and best S4 alignment for all alignment pairs in each group are shown for comparison. As in Figure 5.2a, distance above the dotted line represents improvement over the optimal alignment.

The data appears to fall into two distinct sections on either side of an optimal TM-score of 0.5. If we understand the TM-score of the model made from the correct alignment to be the maximum achievable accuracy, there is evidently much less room for improvement above this point than below it. In fact, many of the optimal models fall well below the 0.4 threshold understood to be the minimum necessary for a meaningful prediction. The best models from S4 alignments, however, are not only consistently above this threshold, but often very close to the accuracy of the model from the correct alignment: 6% have a TM-score greater than or equal to that of the model built from the correct alignment and nearly half (48%) are within 0.05. Finally, we can see that the number of alignments returned can be reduced by an order of magnitude with very little loss in accuracy by calculating a pG score. The proximity of the S4 1000 and S4 100 (PG) lines demonstrates that the pG score consistently ranks in the top 100 the best model from the entire ensemble. In other words, the statistical potential that is embedded in Prosa/PG is able to discern native structures more effectively than the alignment scoring function alone.

5.4.3 Sampling of alignment space

S4 uses the same scoring function as the other DP-based methods used in Figure 5.2. It therefore owes its success not to greater scoring sophistication, but to its sampling of diverse regions of alignment space while also avoiding regions that would produce poor alignments. The latter feature is achieved with the rules and filters discussed in Section 3.1.2. The ability of S4 to sample broadly, however, should manifest itself in greater sampling at both the residue and whole alignment levels. Indeed, in Figure 5.4a, we see that S4 samples 3-5x as many different query residues at each template position as do the DP-based methods with the same ensemble size. Furthermore, we find that increased sampling per residue coincides with searching a broader region of alignment space. In Figure 5.4b, we choose the structure-based sequence alignment as a reference and report the standard deviation of the IAD for all alignments in an ensemble. A low standard deviation indicates that many of the alignments in the ensemble are clustered around a particular distance from the correct alignment, which implies that they are in a narrow region of alignment space. For DP-based methods that region will be centered on the optimal alignment. Combined with the results from Figure 5.2, we see in Figure 5.4b that not only does S4 return an alignment closer to the correct alignment than the DP-based methods, it samples much further away from it as well in a broad search of alignment space.

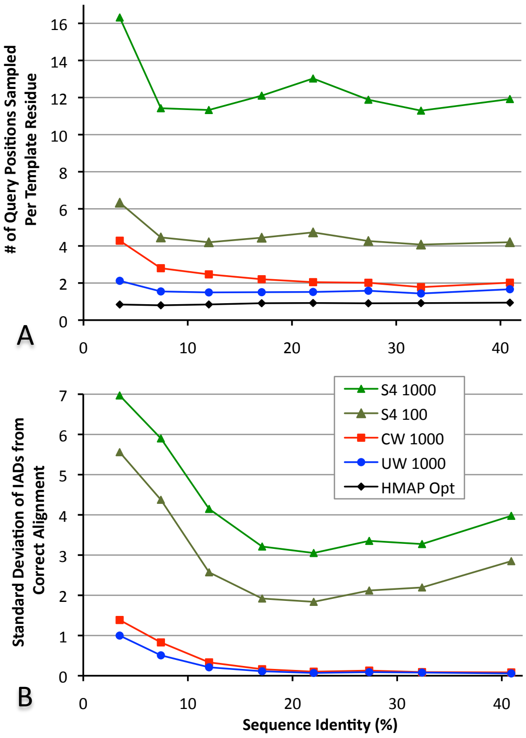


Figure 5.4: Generating diversity in the alignment ensemble - (a) The vertical axis describes the number of unique query residues sampled at each template residue position in the top 1000 alignments for all three suboptimal alignment methods as well as for the top 100 alignments for S4. For comparison, the optimal alignment sampling, which is necessarily at most one query position per template residue, is also shown. The horizontal axis represents pairs grouped by sequence identity and is the same for both graphs. (b) The standard deviation of the IAD from native of the ensemble generated by each suboptimal alignment method is plotted on the vertical axis. A greater standard deviation implies a larger portion of alignment space sampled.

Interestingly, both panels of Figure 5.4 show that all methods, including

S4, return more diverse alignments when the sequence identity is low. This is likely due to the fact that the scoring landscape becomes much flatter as sequence identity decreases. Another trend seen in Figure 5.4 is that S4 samples fairly broadly even at high levels of sequence identity. In contrast, DP-based suboptimal methods seem to converge on the optimal alignment at high levels of identity. The difference is due to the fact that S4 begins by sampling fragments from across the alignment matrix instead of simply proceeding from the optimal alignment

5.4.4 An example of aligning a remotely homologous pair

Combining broader sampling with thorough filtering of unmodelable regions of alignment space allows S4 to find near-native alignments when DP-based methods cannot. We present in Figure 5.5 a typical example of a difficult query sequence/template structure pair and illustrate the success of S4 in aligning remote homologs. The query sequence is the N-terminal domain of KaiA, a non-enzyme circadian clock protein (78). The template is a single domain of DXR, which is a reductoisomerase inhibited by compounds with anti-malarial activity (79). These two proteins have high overall structural similarity despite being classified as different folds in SCOP (80) and having less than 2% sequence identity.

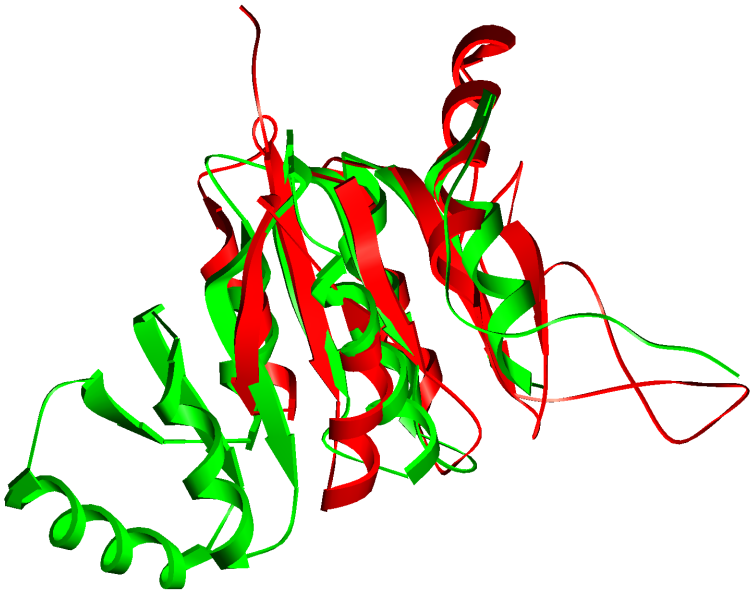
Figure 5.5 shows the sequence of the template, DXR, and several alignments of KaiA, including the correct alignment, the optimal alignment, and the best alignments out of 1000 from S4 and CW. The low sequence identity of this pair belies significant sequence similarity in the correct alignment. In fact, the shared HMAP scoring function is able to find the correct homology, for both S4 and the DP-based methods, in four out of the eight homologous SSEs (T1-T3, T12). Furthermore, S4 was able to build the correct alignment with only the high-similarity fragments it finds in the first step of the algorithm (the primary fragments described in Section 3.1.1). It did not have to resort to filling in gaps with secondary fragments of lower similarity, in this case. The problem, therefore, is not insufficient sequence homology to detect promising regions of alignment space, but rather a sampling of this space that is too narrow to include the correct alignment. In particular, the possibility of large deletions, like T4-T7 in this example, create ambiguity that must be sampled to find the correct alignment.

To quantify this problem, the optimal alignment has an IAD from the correct alignment of 15.4. Of the 1000 alignments in the CW ensemble, the one that ranges furthest from the optimal alignment has an IAD from optimal of 9.1. Therefore, even if this exploration of alignment space proceeded directly toward the correct alignment, which it did not, the closest CW alignment would have at best an IAD from correct of 6.3 (= 15.4 - 9.1), since the IAD measure satisfies the triangle inequality. (See Section 4.2.1.) In fact, the CW alignment closest to the correct alignment ('CW best' in Figure 5.5) improves upon the optimal alignment only very slightly, with an IAD from correct of 15.1. When held to a small, finite ensemble size, the Waterman method simply does not explore far enough from the optimal alignment to correct significant errors.

While the CW ensemble ranges as far from optimal as an IAD of 9.1, the average IAD from optimal for the ensemble is only 0.31 with a standard deviation of 0.77. Though S4 does not proceed from the optimal alignment in the same manner as the DP-based methods, since it shares the same scoring function, it often samples the region of the optimal alignment. In this example, the alignment in Figure 5.5 labeled 'S4 opt' is from the S4 ensemble and has an IAD of 0.12 from the optimal alignment. From there, however, S4 samples much more broadly, with an average ensemble IAD from optimal of 9.48, a standard deviation of 4.61 and a maximum of 25.43. This ensemble includes the alignment labeled 'S4 Best', which has an IAD from the correct alignment of 0.56 and an FDS2 of 97%.

5.5 Discussion

Like the original DP algorithms, S4 consists of a buildup of ungapped alignments. S4 can be thought of as having two separate steps: finding promising regions of alignment space and then using a statistical potential to select a single representative for that region. The method is thus not biased toward any one particular optimal alignment. One might presume that alignment space is too large to be sampled in this manner, but we found that this method is successful with sequences up to 350 residues in length. S4 returns alignments within



```
SSE Index        T1,Q1          T2,Q2         T3,Q3         T4          T5        T6
DXR              EEEEEE      HHHHHHHHHHHH   EEEEEEEEE   HHHHHHHHHHH    EEEEE  HHHHHHHHH
KaiA             EEEEEE   HHH  HHHHHHHH    E EEE EEE
DXR       ---M---KQLTILG--STGSIGCSTLDVVRHNPEHFRVVALVAGKNVTRMVEQCLEFSP--RYAVMDDEASAKLLKT
KaiA      MLSQ---IAICIWVESTAI--LQDCQRALS-ADR-YQL-QVC-----------------------E-----------
S4 Best   ---MLSQIAICIWV--ESTAILQDCQRALS--ADRYQLQVC------------------------------------
CW Best   ---MLSQIAICIWV--ESTAILQDCQRALS--ADRYQLQVCES--GEMLLEYAQTHRDQIDCLILVAANPSFRAVVQ
Optimal   ---MLSQIAICIWV--ESTAILQDCQRALS--ADRYQLQVC--ESGEMLLEYAQTHRDQIDCLILVAANPSFRAVVQ
S4 “Opt”  ---MLSQIAICIWV--ESTAILQDCQRALS--ADRYQLQVC--ESGEMLLEYAQTHRDQIDCLILVAANPSFRAVVQ

SSE Index   T6     T7     T8,Q4          T9,Q5          T10,Q6          T11,Q7
DXR       HHHHH    EEE  HHHHHHHHH        EEEEE        HHHHHHHHHH  H      EEE
Kaia                    HHHHHHHHH         EEEE        HHHHHHHHHHH        EEEEE
DXR       MLQQQGSRTEVLSGQQAACDMAA--LE--DVDQVMAAI-VG-AAGLLPTLAAIR--AGK----TILLAN-------
KaiA      -------------SGEMLLEYAQ--THRDQIDCLILV-AA-NPSFRAVVQQLCFEGVVV----PAIVVGDRDSEDPD
S4 Best   ------------ESGEMLLEYAQTHRD--QIDCLILVA-AN-PSFRAVVQQLCF--EGVVVPAIVVGDR--------
CW Best   QLCFE--------G-------VV--VP----AIVVGDR-DS-EDPDEPAKEQLY--HSA----ELHLGI--------
Optimal   QLCFE--------G-------VV--VP----AIVVGDR-DS-EDPDEPAKEQLY--HSA----ELHLGI--------
S4 “Opt”  QL------CFE--G-------VV--VP----AIVVGDR-DS-EDPDEPAKEQLY--HSA----ELHLGI--------

SSE Index             Q8                        T12,Q9                   Q10
DXR                                       HHHHHHHHHHHH
KaiA                  EEE                   HH HHHHHHH HHH           HHHHHHH
DXR       --K------------D------------------MRTPIAHTMAWP-NRVNSGVKPLDFC----------------
KaiA      EPAKEQLYHSAELHLGIHQ---------------LEQ-LPYQVDA-ALA-----------EFLRLAPVETMA----
S4 Best   --DSEDPD-------EPAKEQLYHSAELHLGIHQLEQLPYQVDAAL-AEF-LRLAPVETM------------A---
CW Best   --H------------Q------------------LEQLPYQVDAAL-AEF-LRLAPVETM-------------A--
Optimal   --H------------Q------------------LEQLPYQVDAAL-AEF-LRLAPVETM--------------A-
S4 “Opt”  --H------------Q------------------LEQLPYQVDAAL-AEF-LRLAPVETM---------------A
```

Figure 5.5: Finding the correct alignment between remote homologs - Top: The structural alignment of template DXR (green) and query KaiA (red). Close structural homology clearly exists among the common portion (four strands and four helices) despite a significant deletion of 4 SSEs in DXR. Bottom: This multiple sequence alignment depicts: (Lines 1-5) the native alignment of DXR and KaiA and the pattern of their SSEs along with their indices; (6) the best of 1000 S4 alignments; (7) the best of 1000 CW alignments; (8) the optimal alignment; and (9) the S4 alignment closest to the optimal alignment.

seconds or minutes for sequences up to 250 residues, while possibly requiring several hours to align sequences close to the maximum length.

The focus at all points of the algorithm is on finding alignments that make sense in structural terms. Alignments that led to biologically or geometrically flawed structures inspired new rules to recognize and eliminate such fragment combinations. The method therefore does not suffer from a particular drawback of those based on DP, which is a focus on local residue-residue similarity to the exclusion of global considerations. The result is a method that does not rely on DP until we have identified narrow regions of alignment space that are non-redundant, contain high similarity and lead to biologically plausible models.

Another key feature of the algorithm is the implicit assumption that for remotely related proteins, the degree of similarity will vary over the sequence. Thus, in addition to portions of the sequences that are related closely enough to be easily recognizable using a pairwise residue similarity score such as that implemented in HMAP, there will also be fragments that have diverged to the extent that no detectable similarity remains. This is again illustrated in Figure 5.5. Much of the optimal DP-based HMAP alignment is correct (the N-terminal strand-helix-strand and the final C-terminal helix), but these regions of more obvious homology are separated by a long deletion and fragments of lower similarity. S4 is successful in these cases by sampling alternative paths through the region that meet our requirement for basic modelability.

It should also be noted that S4 is independent of the underlying scoring function and gap penalty. In the work presented here, the in-house HMAP scoring function was used for ease of development, but S4 can be adapted to use any other residue-residue scoring function. Looking again at Figure 5.4, we emphasize that what is new about S4 is not the way similarity within the alignment space is detected, but how the alignment space itself is sampled. Therefore, if future improvements to dynamic programming scoring functions are able to raise the level of accuracy of the DP methods as depicted in Figure 5.2, then S4's performance should improve as well.

There are two potential drawbacks to the use of S4. The first is that the template must be supplied to the algorithm instead of being selected by it. As shown in Table 5.2, however, S4 can improve alignment accuracy in situations where a template has been selected by other methods, such as PSI-BLAST. Not only does S4 improve the alignment accuracy overall, but for templates where the sequence relationship is weak (E-value $> 10^{-6}$) S4 typically generates alignments that are nearly twice as accurate as those generated by PSI-BLAST. Perhaps the most interesting item in Table 5.2 is the majority of pairs in the set for which PSI-BLAST did not detect any homology. S4 is nearly as accurate in these cases as it is for the low and high homology pairs. This suggests an interesting application of S4 that would obviate the need for template selection: use the algorithm to scan a genome to find sequences that can be aligned to a particular template of interest. Table 5.2 implies that many such sequences could be found, thus greatly enlarging the number of proteins that can be modeled accurately given currently known structures. In fact, the results of carrying out

		PSI-BLAST E-value		
		$< 10^{-6}$	$> 10^{-6}$	No hit
	Pairs	902	480	1516
	ID%	17.1	9.1	6.8
FD0	S4 1000	81.0	70.5	56.0
	S4 100	80.3	68.7	52.1
	PSI	73.0	39.4	N/A
FDS2	S4 1000	95.6	89.9	81.9
	S4 100	94.9	87.9	76.0
	PSI	83.4	48.8	N/A

Table 5.2: Comparison with PSI-BLAST - A comparison of the best S4 alignment from ensembles of 100 and 1000 alignments versus the single alignment from PSI-BLAST over regions of high, low and no (detectable) homology.

such an experiment are described in Chapter 6.

The difficulty of recognizing an accurately modeled sequence, however, is the second drawback of using S4. Though we have shown that the output ensemble of 100 or 1000 alignments is very likely to contain an accurate alignment, there is no selection method in S4 for identifying it. This is not so much a problem with S4 in particular as it is a difficulty with homology modeling in general. Methods such as Prosa and DFIRE can help by giving an indication of model quality or rank a set of output models without knowledge of the native query structure. In addition, studies such as that done by Sánchez and Sali (50) have developed cutoffs for these model evaluation tools that denote a high probability of model accuracy. In the end, the difficulty of selecting a single alignment/model from S4's ensemble can be approached with the same tools that determine the accuracy of the single result from an optimal alignment method. The advantage of using S4, however, is that there is a much better chance of a good alignment or model being found. Using model evaluation tools to validate the models built from S4 alignments is also performed in the experiments described in Chapter 6.

Even if a tool existed to consistently identify the best model in an ensemble, the fact remains that the imperfect correspondence of the template and query structures will always lead to errors in models. We see in Figure 5.3 that even models made from the structure alignment can be quite different from the native query structures. Of course, this is the problem addressed by model refinement techniques. But since refinement methods cannot typically fix all errors, starting with better 'raw' models built from S4 alignments should result in refined ones that are closer to native. Being able to generate a more accurate alignment in a small set of alternatives is an important step toward building improved homology models and expanding the number of sequences that can be modeled with the current set of known structures.

6

Results: Expanding the search for homology

6.1 Abstract

As we push toward the alignment of increasingly remote homologs, the assumption of the underlying homology of the query/template pair becomes doubtful. In Chapter 5, we tested S4 only on pairs known to be structurally homologous. In a real prediction context there is no such certainty. In this chapter we pair S4 with the model evaluation tool ProsaII in order to assess the presence or absence of homology by judging the 'nativeness' of the models produced. In order to simulate a search for a homolog within the genome of a model organism of interest, either the query or template of each pair tested is a protein from *E.*

coli. Thresholds were established to separate homologous from non-homologous pairs, which are then used to determine precision and recall values for three separate experiments.

Experiment 1 paired a set of *E. coli* query sequences with a representative template from each SCOP fold within the same class as the query. In Experiment 2 we chose a structurally diverse set of templates and paired each with every sequence in *E. coli* with a known structure. Lastly, in Experiment 3 we used PSI-BLAST to find remote homologs from the PDB for a set of queries from *E. coli* with the goal of discriminating homologs from non-homologs. Experiments 1 and 2 represent blind searches of structural and sequence databases, respectively. Experiment 3 seeks to extend the use of a current template selection technique into the detection of more remote homologs.

In all experiments, there were two goals: (1) building accurate alignments with S4 for homologous pairs and (2) using model building and evaluation to discriminate homologs from non-homologs. Our results show that S4 was able to produce alignments which led to good models in many cases, even for very remote homologs. And the discrimination ability of the homology modeling procedure allowed the identification of many structurally similar proteins in our blind searches. The better models and better discrimination of Experiment 3, however, demonstrated that the most immediately useful application of this methodology is in extending template selection techniques into regions of more remote homology.

6.2 Introduction

In Chapter 5 we demonstrated S4's ability to find a quality alignment between remotely homologous sequence/structure pairs. Insofar as we restrict the scope of our interest to producing sequence alignments, that result may be sufficient to demonstrate the success of this method. Upon adopting a broader perspective, however, several complications emerge.

The first issue involves the process of template selection. In Chapter 5, the templates were known to be structural homologs to the query sequences. While it is necessary to know the query structure for validation purposes, the question remains of whether these remote homologs would have been detectable by a template selection method. In other words, even though S4 can produce a good alignment to a template, in a real prediction context it may never get the opportunity if that template cannot be identified as a likely homolog for the query.

Another aspect of the test in Chapter 5 seems peculiar from the perspective of true structure prediction. All the pairs in the test set were homologous. While it is impossible to assess the success of an alignment method on cases where there is no homology and therefore no correct answer to be found, the addition of non-homologous pairs to the test set is more realistic. In a real prediction context in which the query structure is unknown and no closely homologous template structures are available, it is not sufficient to produce good alignments and models from those templates that are homologous. It is also necessary to recognize bad templates, alignments or models.

Evaluation methods exist to accomplish this task. Recognition of bad templates or alignments can be done by alignment methods that report a probability measure, such as PSI-BLAST (31). This is not well-suited to our purposes, however, since we desire an alignment method that can recognize homology beyond what these methods can detect reliably. Downstream of the alignment stage, we can use model evaluation techniques, such as those described in Section 1.3.5, to determine if models built from alignments to our chosen templates are likely to be native. In this way, we can use the final stage of the homology modeling process to check the accuracy of all preceding stages.

This is essentially the approach we adopted in performing three separate experiments based on proteins from the *E. coli* genome. All three tests use the model evaluation tool ProsaII (49) to determine whether the template was homologous and, if so, whether S4 produced a quality alignment. Of course, these experiments also test the accuracy of ProsaII itself, since it is possible that quality alignments to a homologous template will lead to good models that are misclassified at the final step.

The first issue raised above, regarding how to recognize a remote template in the first place, is essentially avoided in Experiments 1 and 2. Experiment 1 pairs query sequences from *E. coli* with a representative sample of structures from the PDB. Several of these structures will be structurally similar to the query, but the vast majority will not. In Experiment 2, we selected several template structures from outside the *E. coli* genome and paired them with query sequences representing all *E. coli* proteins with known structures. Again, several of the sequences in the genome will be structurally homologous to each of our templates, but the vast majority will not be. In both Experiments 1 and 2, we would like to answer two questions. First, can S4 produce an alignment which leads to a good model when the query and template are homologous? And secondly, can ProsaII distinguish between good models arising from quality alignments to homologous templates and bad models arising from either non-homologous templates or bad alignments to good templates.

In Experiment 3, we use PSI-BLAST, a standard method of template selection, to determine the set of structures to which to align sequences from *E. coli*. PSI-BLAST does not use the DP algorithm, but instead calculates an E-Value based on the likelihood of finding segments similar residues consecutively aligned. In order to focus on remote homologs, we remove all close homologs from the set and include only those with E-values from 0.001 - 10. Templates in this range are typically ignored as being too unreliable for homology modeling. Altschul *et al* considered an E-value of 0.05 to be “marginally significant”. Other work has shown shown large increases in the false positive rate for PSI-BLAST at E-values between 0.001 - 10 (81, 82, 83). We therefore chose this range of E-values as a test of our ability to find good models and discriminate true from false homologs.

The motivation behind these experiments was to move beyond the assessment of alignment quality between remote homologs and instead begin to determine the extent of our ability to perform structure prediction in this region. Several studies (51, 84, 85) have been done that have attempted to increase the 'leverage' of the PDB by demonstrating that each structure can be used as a template for a wider range of homologous sequences. In other words, as we become more successful in producing quality models of query sequences from remotely homologous templates, we increase the leverage of each known protein in the PDB. This leverage aids structural genomics initiatives in selecting structures to solve that will provide suitable templates for the widest range of sequences (86, 87).

6.3 Methods

These three experiments examine different ways of finding templates for S4. After the template selection stage, the homology modeling process is identical. Alternative alignments are produced with S4, which are then modeled without refinement[1] and evaluated by ProsaII. We can assess these results as we did in Chapter 5 by asking if S4 produced an alignment that led to a quality model. However, we can also assess whether the model evaluation technique was able to recognize good from bad models.

[1] Model refinement was not performed for two reasons. First, it can be quite computationally intensive. And second, though a model refinement technique is unlikely to fix errors caused by a faulty alignment, it is best not to introduce a potentially confusing extra variable and instead simply allow the alignment method to determine the model quality.

The S4 algorithm was not changed from the version that produced the results in Chapter 5. Also, the alignment validation methods used in the set of experiments described in this chapter are the same FDS2 and IAD scores described in Sections 4.1.3 and 4.2, respectively. Also as in Chapter 5, the TM-score was used to assess model quality through comparison to the native query structure. This chapter focuses mainly on the accuracy of the model resulting from the alignment, which is also a measure of alignment quality.

6.3.1 Creation of the test sets

The test sets for these experiments were created separately, though there is a small degree of overlap since all are based on the *E. coli* genome. All three use the same definition of what constitutes a homologous or 'good' template. Such a template is defined as one for which a model can be made from the structure-based sequence alignment with a TM-score of at least 0.5 when compared to the native query structure. That is, if a quality model can be made from the structure-based sequence alignment, then it should be possible, in principle, to build a quality model from a similar alignment generated from S4 or another method.

Similar to the constraints used in the CASP set discussed in Chapter 5, both query and template sequences were limited to a maximum of 300 residues. An additional minimum of 75 residues was added for all three experiments to exclude short sequences.

Experiment 1: Template recognition

Experiment 1 sought to determine if homologous templates could be found for a query sequence from *E. coli* through a naive scan of the PDB. To minimize the number of templates for each query, we only used one template from each SCOP fold within the same SCOP classification as the query. So that the accuracy of these experiments could be assessed, only those proteins in the *E. coli* genome with solved structures were used. The classification level of the SCOP hierarchy is a very broad designation grouping together proteins that are entirely composed of alpha helices, for example. We chose at random a single structure to represent each SCOP fold within that class. The vast majority of these structures would not be suitable templates for the *E. coli* sequence. To ensure that each sequence had at least one homologous template in the test set, a structural alignment method was used to find remote homologs for each sequence.

In order to obtain a broad sense of the ability of S4 and ProsaII to build and recognize good models, we chose queries from *E. coli* from all four of the primary SCOP classifications. Of the fifteen queries used in this experiment, 3 are from class A (all α proteins) and 4 are from each of classes B (all β proteins), C (proteins with integrated α and β units) and D (proteins with segregated α and β units).

Finally, since practically all members in the same SCOP fold may be considered good templates, we needed to reduce the number of templates returned by the structural alignment search. This was done by making the test set more difficult. Templates were not used if the constrained Waterman (CW) method from HMAP (see Section 5.3.2) returned an alignment with an IAD below 4 or an FDS2 above 50% within its top 1000 alignments. Since this method shares its similarity score with S4, by removing all pairs with clear homology as defined by the HMAP scoring function, we are increasing the difficulty for S4.

The resulting set contains 15 query sequences aligned to an average of 112 templates each for a total of 1684 pairs. On average, each query is paired with 17 homologous templates, with which it has a sequence identity of 4.9 ± 2.0%.

Experiment 2: Query recognition

For Experiment 2, we began by selecting selecting 38 templates from the PDB that were structural homologs of proteins from *E. coli*. As in Experiment 1, the templates were chosen from across the four main SCOP classifications. 11 from class A, 7 from class B, 10 from class C and 10 from class D. These 38 templates were then paired with each of the 217 sequences in *E. coli* with a known structure.[1]

Unlike Experiment 1, no effort was made to exclude pairs for which a good alignment could be generated within 1000 CW alignments. Even so, the set is of an equally low sequence identity. Homologous pairs had an average sequence

[1] There are more than 217 proteins in *E. coli* with known structures, but we eliminated those that were shorter than 75 residues or longer than 300.

identity of 4.6 ± 2.1%. Out of 8,457 pairs of queries and templates, only 132 were homologous. The average template thus had only 3-4 homologous queries among 217 total.

Experiment 3: PSI-BLAST Validation

In this experiment, we began by running PSI-BLAST on the sequences of all *E. coli* proteins with structures. Many hits to templates in the PDB were returned, but we discarded all but those with E-values in the range 0.001-10. The resulting set represented all four major SCOP classes, but not as evenly as in the previous experiments when the set was designed with this goal in mind. For this experiment, 3 queries were in class A, 3 in class B, 15 in class C, and 8 in class D. These 29 queries were aligned to an average of 15 templates each for 433 total pairs. Of these, 257 pairs were homologous and 176 were not.

6.3.2 Homology modeling with S4, Nest and ProsaII

For all pairs in all experiments, alignments were made with not only S4, but also the optimal HMAP method as well as the constrained and unconstrained Smith-Waterman suboptimal methods. Models were made from the optimal HMAP alignments, but not the 1000 alignments from the Smith-Waterman suboptimal methods. All models made were evaluated by ProsaII, which uses a statistical potential to check the fit of a sequence for the conformation of the model. It compares this calculation for many other unrelated folds to calculate a z-score for the model.

More helpfully, we can use this z-score to determine the likelihood that the model is near its native conformation (51). The z-score is normalized for differences in protein size by dividing by the natural logarithm of the length of the model. This value is termed the pG, which stands for 'probability of good', but should really be thought of as a measure that can be compared against a threshold to separate good and bad models with a certain level of confidence. It was found that when this score was over 0.5, it resulted in fewer than 5% false positives and 8% false negatives (50). Higher values can be used if greater precision and sensitivity are desired.

6.3.3 Establishing thresholds for homology detection

As this work is still on-going, a full analysis will require a k-fold cross validation approach to avoid over-fitting to the data. Since that work has not yet been completed, the results given here are only preliminary.

The key to interpreting the mass of model evaluation results in this data set is a threshold that separates query/template pairs into those expected to be homologous and those that are not. We combined pairs from all three experiments together and attempted to find an effective means of detecting homology. As described in Section 6.3.2, we first used ProsaII to calculate a pG score for each model. Unfortunately, we found that the pG score returned a high number of false positives for our purposes. For example, for homologous pairs for which S4 found a good alignment and model, several of the 1000 pG scores may be high, but not necessarily for the models which are close to native. Furthermore, for non-homologous query/template pairs in which there was no good alignment to be found or model to be made, one or several out of 1000 models would have a high pG score nonetheless.

Our observed false positive rates were roughly in line with the values from other studies cited in Section 6.3.2, though we had to use higher pG thresholds to achieve them. In contrast to these studies, which used sequence homology-based template selection methods, our data set contains many more non-homologous pairs due to the fact that we attempt to align sequences from the genome and representatives from the PDB in an essentially blind search for homology. Since it too uses PSI-BLAST, Experiment 3 is far closer to these previous studies than are Experiments 1 and 2.

For these reasons, in future refinements of this methodology, it will be wise to establish separate thresholds depending on one's prior knowledge of the likely fraction of homologous pairs in the data set. However, in an effort not to further over-fit our selection criteria to our data, all pairs from all three experiments were used to determine two threshold values: (1) the pG score which denotes a model with 'positive' homology and (2) the minimum number of such models in an ensemble needed to consider the query/template pair homologous. A receiver operating characteristic (ROC) curve was determined for different levels of this second threshold, while the pG score threshold varied along the curve. The area under the curve (AUC) was calculated for each and a maximum value of 0.8486 occurred at 5. That is, the best discrimination between homologous and non-homologous pairs occurred when 5 or more models had a pG score above the threshold. It should be noted that, though the maximum occurred at 5, a range of thresholds from 1 to 12 had AUC values within 1% of the maximum.

Since the data set is highly skewed toward non-homologous pairs, precision and recall values are more useful than true positive and false positive rates given in a ROC curve. Since the ROC curve with the best discrimination (i.e., highest AUC) is equivalent to the dominant precision vs. recall curve (88), the same threshold of 5 for the number of models over the pG threshold was kept. A plot of this curve is given in Figure 6.1.

Though we could choose any level of recall and precision along the blue line, for the purposes of evaluating the results of these three experiments we will use the third point from the left, which represents 29% recall at 74% precision. This point on the graph corresponds to a pG minimum of 0.9. Therefore, in the results described below, a pair will be considered homologous if out of 1000 S4 alignments, 5 or more have a pG score of 0.9 or greater. This is quite a bit higher than the 0.5 and 0.7 figures quoted in previous studies (50, 51). But the higher threshold is necessary to account for the greater chance of returning a false positive due to the far smaller fraction of homologous pairs in this test and

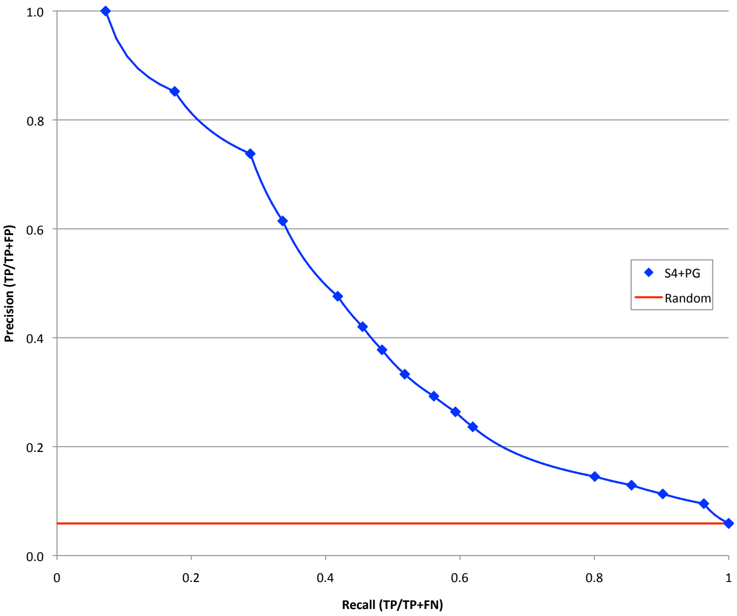


Figure 6.1: Precision vs. Recall - The blue points represent the actual calculated data with the line interpolating between them. The scalloped shape is due to the non-linear interpolation on precision-recall graphs. The red line represents the discrimination of a naive or random classifier.

the larger number of models (including many incorrect ones) per pair.

6.4 Results

6.4.1 Experiment 1: Template recognition

In Experiment 1 we aligned query sequences from *E. coli* to a single representative of each SCOP fold in the same SCOP class as the query. Of the 1,684 pairs that resulted from this process, 249 (15%) represented homologous pairs. Using the cutoffs determined in Section 6.3.3, we can prepare Table 6.1. The precision and recall values for these data are 48.4% and 6.0%, respectively. These are

		Homologous		
		Yes	No	
Predicted	Yes	15 TP	16 FP	31 (1.8%)
	No	234 FN	1419 TN	1653 (98.2%)
		249 (14.8%)	1435 (85.2%)	

Table 6.1: Homology prediction in Experiment 1

considerably lower than the 74% precision and 29% recall expected from Figure 6.1. Nevertheless, the combination of S4 and ProsaII was able to correctly identify 15 structurally homologous templates in a random selection of templates from the same SCOP class. Though recall of 6% may seem fairly low, we should remember that we are not filtering through likely templates in this experiment. Instead, this experiment is similar to a blind search of known structures. Furthermore, the fact that all pairs with a good alignment within the top 1000 from the constrained Waterman (CW) method were removed made this a truly difficult test.

If we evaluate S4's performance apart from that of ProsaII, we find that there were many pairs for which good models were made and yet did not pass our thresholds for homology prediction. Among the 249 homologous pairs in this set, the average of the best TM-scores made from the top 1000 S4 alignments was 0.48 ± 0.08. The average best TM-score for the top 100 was 0.45 ± 0.09. The developers of TM-score, cite 0.4 as the minimum for a model to be in the same basic shape as the native structure (35). Table 6.2 lists the number and percentage of homologous pairs for which a model was made with a TM-score

Ensemble Size	Best TM > 0.4	Best TM > 0.5
1000	203 (82%)	95 (38%)
100	183 (73%)	64 (26%)

Table 6.2: Success of S4 alignment method in Experiment 1 - Of 249 homologous pairs, the number (and percentage) of pairs for which S4 returned an alignment that led to a model with a TM-score above 0.4 and 0.5 is given above.

above cutoffs of 0.4 and 0.5 for different ensemble sizes.

Since the homology prediction was performed on the top 1000 models, we can compare the first row of Table 6.2 to the first column, under 'Yes', in Table 6.1. Even using the higher definition of model quality of TM > 0.5, we can see that the difficulty in detecting good models is a major hindrance to correctly predicting homology. Though S4 produced an alignment that led to a good model for 38% of all homologous pairs, only 6% (15/249) were able to pass the model evaluation thresholds.

Though an in-depth exploration of the reasons for errors in ProsaII is beyond the scope of this work, Figure 6.2 highlights the difficulty in using this method. All 1,684 pairs in the test set are represented here by the TM-score and pG calculated from a model made from the correct, structure-based sequence alignment. Though the non-homologous pairs are found more frequently with low pG scores, the homologous pairs are also quite low in this regard. The average pG score for a model made from the correct alignment for a *homologous* pair is only 0.292 ± 0.291. In other words, even if S4 returned the correct alignment, it is unlikely a model made from it would pass the homology prediction thresholds.

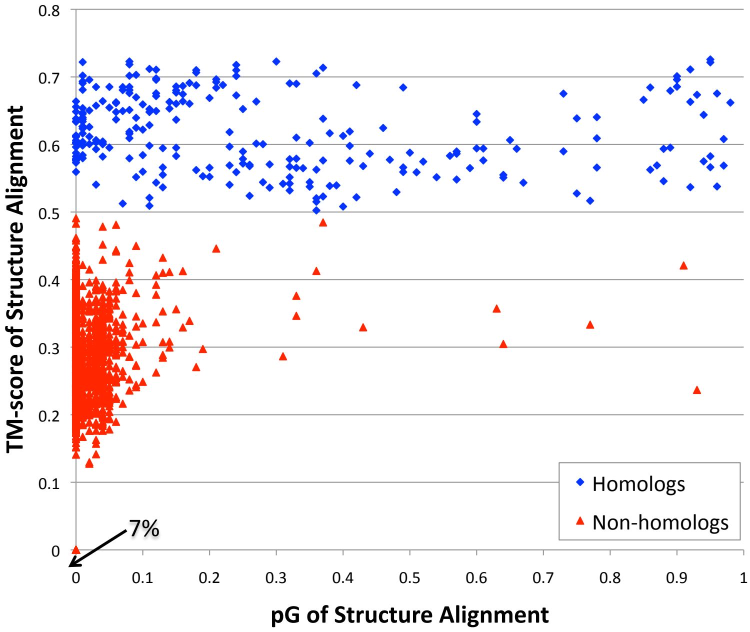


Figure 6.2: pG scores of correct models - Models made from the structure alignment were evaluated by TM-score and ProsaII. For many non-homologs, a structure alignment could not be made. This was true for 7% of the entire data set, which is shown as a single point at the origin.

6.4.2 Experiment 2: Query recognition

In the previous experiment, we scanned representatives of SCOP folds for a homologous template for a particular query in *E. coli*. Experiment 2 is essentially the reverse of this. We aligned a particular template to all sequences in *E. coli* with a known structure in search of those that are homologous. Once the queries were selected for the templates in this manner, the homology modeling process was the same as that described in 6.3.

Overall, the size of the set was much larger than that of Experiment 1 and contained considerably fewer homologous pairs, as can be seen by comparing the first columns of Tables 6.1 and 6.3. As in Experiment 1, the set is overwhelmingly

		Homologous		
		Yes	No	
Predicted	Yes	14 TP	41 FP	55 (0.7%)
	No	118 FN	8284 TN	8402 (99.3%)
		132 (1.6%)	8325 (98.4%)	

Table 6.3: Homology prediction in Experiment 2

Ensemble Size	Best TM > 0.4	Best TM > 0.5
1000	100 (76%)	24 (18%)
100	71 (54%)	17 (13%)

Table 6.4: Success of S4 alignment method in Experiment 2 - Of 132 homologous pairs, the number (and percentage) of pairs for which S4 returned an alignment that led to a model with a TM-score above 0.4 and 0.5 is given above.

full of non-homologous pairs and thus precision and recall, 25.5% and 10.6% in Experiment 2, respectively, are of particular interest. The precision is lower than in Experiment 1, but not so low that a user of this method would be overwhelmed with non-homologous queries that are falsely predicted to be structurally related. The recall is slightly higher than in Experiment 1. Though only 1 in 10 structural homologues were correctly identified, one should consider that this is a blind scan of a genome with no guarantee of any significant sequence similarity between the query and template.

Even with the difficulty of this set, Table 6.4 demonstrates that S4 succeeded in creating alignments that led to good models for a large percentage of the homologous pairs.

If we take the perspective of a researcher who has a set of templates for

		Homologous		
		Yes	No	
Predicted	Yes	151 TP	7 FP	158 (36.5%)
	No	106 FN	169 TN	275 (63.5%)
		257 (59.4%)	176 (40.6%)	

Table 6.5: Homology prediction in Experiment 3

which he would like to know if there exist structural homologs in *E. coli*, we would have identified a homologous query for 11 of the 38 templates. (One template was correctly identified with four queries, while the others only one each.) Considering we were limited to the ∼10% of the *E. coli* genome for which a structure is known, it is very likely that more templates would have found homologous queries if a scan of the whole genome was undertaken.

6.4.3 Experiment 3: PSI-BLAST Validation

In Experiments 1 and 2 the precision and recall values were well below the levels that might have been expected based on the analysis of the entire set (all three experiments together) described in 6.3.3. The results of Experiment 3 shown in Table 6.5, however, are more in line with our initial expectations. Our discrimination thresholds resulted in a precision of 95.6% and recall of 58.8%. While not every homologous pair was captured, the high precision would give a user confidence in using a PSI-BLAST template validated by this method.

The 106 false negatives in Table 6.5 are the obvious area for improvement.

Recognizing some of these pairs as homologs would increase the recall of this method. There are two ways a pair can become a false negative: either S4 did not find alignments that led to good models, or good models were not recognized as such by the pG score. The latter appears to have been the primary source of these errors. Of the 106 false negatives, the best S4 model had an average TM score of 0.572 ± 0.616. But these same pairs had on average less than one (0.717 ± 1.170) model with a pG score above the 0.90 threshold.

It is also possible that the single best S4 model may have had a good TM-score, but there were not enough good models made for the pG to rate at least five with the score of 0.90 or better needed to be considered a homologous pair. However, Table 6.6 shows that many good models were made for these false negative pairs. For example, the value in the upper-right corner signifies that of the 106 false negative pairs, 24 had ensembles that contained 5 or more models with TM-scores greater than 0.60. The problem is that even when good models are made, not enough of them are recognized as such. Similar to what we saw in Figure 6.2 in Experiment 1, the problem with model evaluation becomes clear when we examine the scores of the models built from the structure alignment, which are essentially the best homology models we can hope to build without refinement. Though the average TM-score of this best model from each of the 106 false negative pairs was 0.62 ± 0.06, the average pG was only 0.38 ± 0.32.

As a final comparison, we can look at S4's results alongside those from the HMAP optimal and suboptimal alignment methods, with which it shares a scor-

Number of models	Minimum TM-score				
	0.40	0.45	0.50	0.55	0.60
≥ 5	105	99	85	59	24
≥ 25	100	93	69	40	15
≥ 50	100	88	60	28	7
≥ 100	96	74	43	19	1

Table 6.6: Analysis of model quality of false negatives - The left-hand column holds minimum values for the number of models in the ensemble that exceeded the minimum TM-scores. (e.g., The upper-left entry of 105 means that 105 of the 106 false negative pairs had ensembles with 5 or more models with TM-scores of at least 0.40.)

ing function. Similar to Figure 5.1, we can plot the best out of 1000 CW alignments in ascending order of IAD to show the improvement of S4 for ensembles of either 1000 or 100, particularly for pairs for which no good alignments were returned by the CW method. In the final 18 pairs in Figure 6.3, the best CW alignment is at an IAD of 5 or more from the correct alignment. In 15 of these 18 cases, S4 1000 returns an alignment that is less than an IAD of 2 from correct. For these same pairs, S4 100 returned 9 ensembles containing an alignment less than 2 from correct and 5 more that were within 3.

We can also look at this data sorted by the E-value from PSI-BLAST, where an increasing E-value denotes decreasing certainty of homology between the query and template. Figure 6.4 shows that while CW is already failing to find a good alignment in many of these cases, the rate of failure seems to be increasing with the E-value. Though S4 1000 also contains two cases where no good alignment was found as the E-value neared the maximum value of 10 used in this

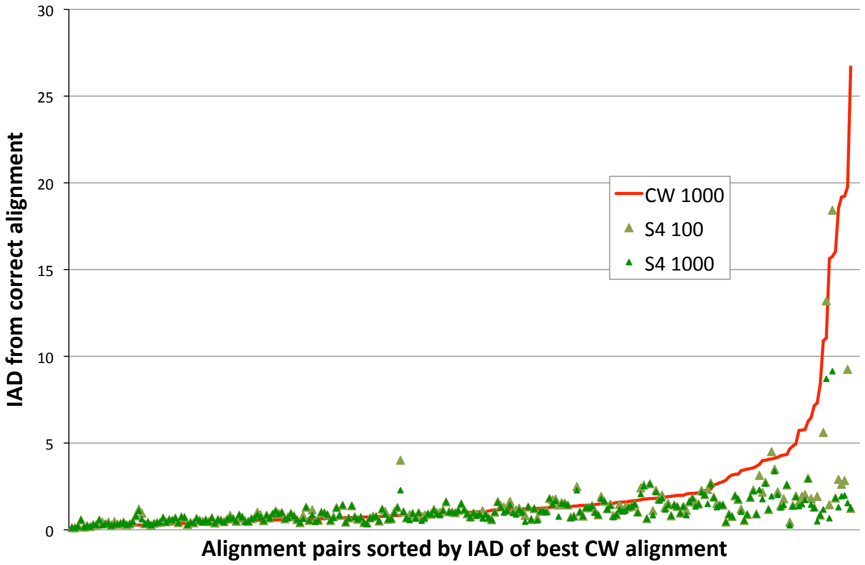


Figure 6.3: Improvement over Smith-Waterman alignments - The red line represents the IAD of the best out of 1000 CW alignments, sorted in ascending order, for all 257 homologous pairs in Experiment 3. The best of 100 and 1000 S4 alignments (green triangles) show a large improvement, especially for the most difficult alignments.

study, it still succeeded in producing a good alignment for the vast majority of pairs.

6.5 Discussion

This chapter described the setup and results of three related experiments. All three simulated different possible scenarios in which a researcher would be interested in finding structural homologs to a protein of interest. In Experiment 1 we began with a set of query sequences from *E. coli* and searched for homologs among representatives of each SCOP fold in the same class. We reversed this perspective in Experiment 2 by beginning with a known structure of interest

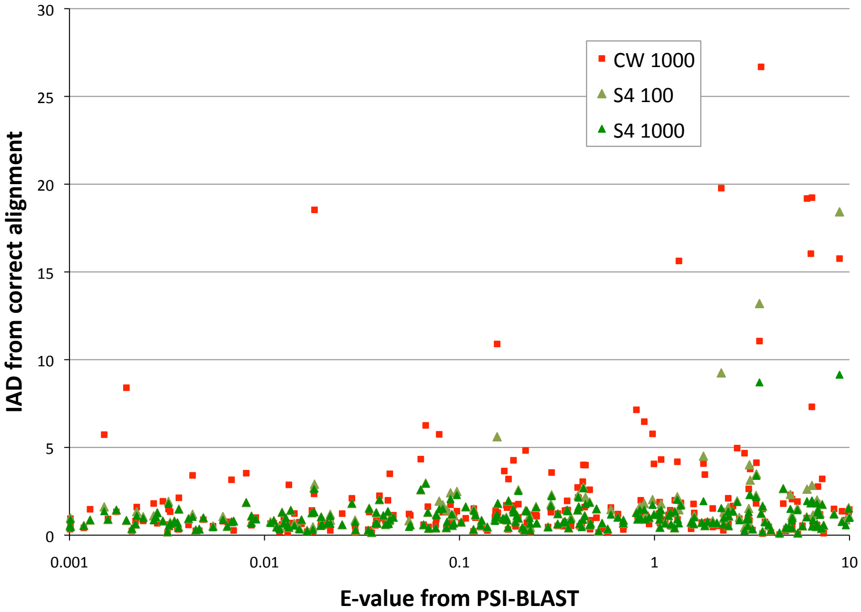


Figure 6.4: Alignment quality vs. PSI-BLAST E-value - Similar to Figure 6.3, we plot the same points, but as a function of the PSI-BLAST E-value for the particular query/template pair. The horizontal axis is a logarithmic scale.

and looking for homologs among the sequences of the *E. coli* genome. Finally, in Experiment 3 we tested our ability to validate the homology of templates selected by PSI-BLAST for *E. coli* query sequences, but at a low level of similarity typically ignored by homology modelers as being either too uncertain in its structural correspondence to the query or too remote for a good alignment to be made.

These experiments are grouped together here because they all shared two common goals. First, these tests gave us another opportunity to test S4's ability to find good alignments between remote homologs. And second, by adding a final model evaluation step, we could assess the difficulty in selecting which of those 100 or 1000 alignments were closest to the native structure. By doing so, we were also able to successfully identify those protein pairs for which no clear structural relationship existed. The specificity, or true negative rate, was quite high: 98.9%, 99.5% and 96.0% for Experiments 1-3, respectively.

Both the composition of the individual test sets and the level of success in the discrimination of good and bad models suggest strongly that Experiments 1 and 2 are quite different than Experiment 3. These first two are searches for very-remote homologs among a background of many non-homologs, while the last is essentially a validation of a current template selection tool in a range in which it is no longer consistently accurate. While it is possible that our homology prediction had better precision in Experiment 3 because there were many fewer non-homologs in its set and thus less opportunity to have a false positive, it is also true that Experiment 3 was simply easier than the other two. Table 6.7, which is divided between the results from all homologous pairs and the subset of those pairs that were correctly predicted by S4/pG, allows us to see this quantitatively.

Focusing first on the left-half of Table 6.7, the most obvious difference between the experiments is the greater sequence identity in Experiment 3. The TM-score of the model built from the structure alignment, which we use to represent the best model that is possible to build for a query/template pair, is also slightly higher for Experiment 3. Taken together, this implies that for the pairs in Experiment 3, an alignment that leads to a better model exists and that it will be easier to find. The fact that the pG score was better able to discriminate

		All Homologs			True Positives		
	Experiment	1	2	3	1	2	3
	Pairs	249	132	257	15	14	151
	ID%	4.9	4.6	9.2	5.1	6.5	10.1
TM-score	Str	0.62	0.55	0.67	0.61	0.57	0.70
	S4 1000	0.48	0.45	0.62	0.50	0.52	0.66
	S4 100	0.45	0.42	0.61	0.46	0.49	0.65
	Opt	0.29	0.32	0.52	0.28	0.37	0.56
IAD	S4 1000	3.2	2.4	1.1	4.1	1.5	0.8
	S4 100	5.3	3.9	1.3	6.6	1.8	0.9
	CW 1000	15.9	8.3	2.0	13.9	5.7	1.3
	Opt	19.8	11.8	3.0	18.0	7.9	1.9
FDS2	S4 1000	65.4	73.4	88.6	65.9	74.8	89.6
	S4 100	55.7	63.1	86.5	55.4	67.4	87.8
	CW 1000	25.5	48.5	83.0	32.9	50.1	85.4
	Opt	17.1	32.2	76.5	21.6	35.9	78.8

Table 6.7: Comparison of results across experiments - This table divides the results for several categories of comparison for each experiment between all homologous pairs and those for which S4/pG detected the homology. ID% is the percent identity between the query and template in the structure-based sequence alignment. Str refers to the correct or structure alignment. CW and Opt denote the suboptimal alignments from the constrained Waterman method and the optimal alignment, respectively. All scores for suboptimal methods reflect the best value for the entire ensemble. The numbers following the suboptimal methods denote the size of the ensemble.

in Experiment 3 is likely due to the fact that the good models we wanted it to rate highly were actually closer to native. The best S4 model from each ensemble had an average TM-score of 0.62 in Experiment 3 versus 0.48 and 0.45 for Experiments 1 and 2, respectively. The relationship between better alignments (as measured by IAD and FDS2) and better models is also evident in Table 6.7.

The right-half of Table 6.7 reports the same results, but for the subset of homologous pairs that were correctly predicted by S4/pG. With few exceptions, the alignment and model quality measures are all improved for this group. While that may be expected, it is further evidence that better models increase the discrimination ability of model evaluation. In fact, model evaluation methods often report their ability to detect the native structure from a set of decoys, which demonstrates the ability of their energy functions to recognize structures that are native or very nearly so (89, 90, 91, 92). However, what is arguably more important for the purposes of homology modeling, especially with remote homologs, is an evaluation method which can detect models that, while not highly similar to native, still maintain the correct overall shape and topology of SSEs. This is certainly a more difficult challenge.

It is possible that the results for Experiment 1 could be improved through an iterative process of building models and filtering the results. We would initially set low thresholds and eliminate only those folds that produced the worst models. Next, we could select several templates from within the remaining folds and repeat the process. This would allow us to find a representative of the fold which is a better partner for alignment than the original choice. If possible, the representatives of each fold should have little sequence similarity to each other to provide the query with a maximum diversity of alignment partners. If a fold produces poor models for all or most templates upon this second examination, it can be eliminated. This could be iterated as time permits. Such a process might reduce the number of false positives by re-checking other templates in the same fold. Also, it could reduce false negatives by possibly finding a better alignment partner than the originally chosen representative of the fold.

All experiments could be improved if some knowledge of the template or query were available. For example, Lee and colleagues extended the work done in the SkyLine homology modeling pipeline tool (51), by demonstrating that START domains could be identified through a similar process (93) that also utilized specific knowledge of the template. Starting with a known structure, they used PSI-BLAST to select and align sequences from the NCBI non-redundant database (94). Knowledge of the characteristics of START domains, such as the pattern of positive and negative surface electric potentials and locations of cavities, gives the modeler another way of assessing quality beyond model evaluation tools like ProsaII. In a large-scale test such as the one described here, this level of detailed analysis is not feasible. But for a smaller study focused on a single protein or family of proteins, one can imagine using such information to further reduce the number of false positives at all levels of recall.

Based on the results described in Section 6.4 and Table 6.7, the pG score does appear to better discriminate good from bad models when the good models are closer to native. This is not a surprising result, but it does suggest that if our models in Experiments 1 and 2 were slightly better, the precision and recall would have been higher as well. Aside from making better alignments — and there was room for improvement — it might be possible to make small improvements to the models with a small amount of refinement before they are evaluated by ProsaII. Refinement was not done in this case due to both computational restraints and the desire to see as clearly and directly as possible the relationship between alignment quality and homology detection. However, to the degree refinement methods can make good models better, they are unlikely to improve models arising from non-homologous query/template pairs since there is no correct model to move toward. In this case, a small amount of refinement applied to all models might increase our discrimination power.

Another goal of these three experiments was to determine whether a model evaluation tool could reliably select the better models from the ensemble. That is, if using S4 results in 100 or 1000 models, many of which are quite different from each other, how will a user be able to tell which of these is closest to native? Of course, Figure 6.2 demonstrates that the best models are not often rated highly by model evaluation techniques. Table 6.8 shows the difficulty of using the pG score alone to select a good model from the ensemble. In the context of a real prediction, a researcher would like to select from the ensemble the model with the highest pG score and be confident that it is a good model. The top row demonstrates that the TM-score of the model thus obtained would

	Experiment		
	1	2	3
Best pG	0.37 ± 0.11	0.34 ± 0.09	0.53 ± 0.11
Best of top 10 by pG	0.44 ± 0.10	0.40 ± 0.09	0.59 ± 0.09

Table 6.8: Selecting the best models from the ensemble - Looking only at the homologous pairs within each experiment, the first row represents the TM-score obtained from selecting from the ensemble the model with the highest pG. The second row shows the best TM-score among the 10 models with the highest pGs.

be quite low for all but Experiment 3. If the researcher were willing to look at the top 10 models as determined by pG, the average TM score of the best model among the 10 is given in the bottom row. The averages in Table 6.8 are calculated over all homologous pairs in each experiment. Therefore, the larger fraction of homologous pairs for which a good model was not found in Experiments 1 and 2 will lower these values.

As stated at the outset of this chapter, the use of the S4 alternative alignment procedure as part of a homology detection method is still a work in progress. A more sophisticated approach would establish more precise thresholds according to the expected level of homology one is attempting to discern. From this perspective, the work described here is simply a proof of concept. What was proven was our ability to find homology, albeit at low recall, in a blind search of structure and sequence databases. We also showed that S4 was able to build good alignments between templates and queries for the upper portion of the spectrum of PSI-BLAST results that are typically ignored. More importantly, we found that we could discern which pairs were homologous based on the models built from S4's alignments. There is certainly more room for improvement, but this work represents a real advance in our ability to perform accurate structure prediction from increasingly remote templates.

7

Conclusions

7.1 Homology modeling with alternative alignments

Extending the work described in Chapter 5 by using S4 in a full homology modeling pipeline in Chapter 6 allowed us to evaluate the overall value gained from the additional expense of producing extra alignments and models. We can use the data from Tables 6.7 and 6.8 to make a comparison between the TM-scores of models made from the optimal alignment and that of the single model from the ensemble created by S4 with the best pG score. That is, if a researcher was given no extra information and had to select a single best model based only on the prediction of ProsaII, how would the quality of the model compare to that of a single model resulting from the optimal alignment? We find that though the best model created by S4 had a significant average improvement in terms of the TM-score of 0.19, 0.13 and 0.10 over the optimal model for Experiments 1-3, respectively, the improvement over optimal for the single model with the best pG was only 0.08, 0.02 and 0.01.

The differences between these sets of numbers demonstrates both the advantage of making alternative alignments and models, but also the difficulty of judging the nativeness of those models. It appears that our ability to detect homology and produce correct alignments may currently be ahead of our ability to recognize near-native structures. This discrepancy may be partly explained by the possibility that model evaluation, which is a 3D problem, is simply more difficult than sequence alignment, which is essentially a 2D problem. As a consequence, the landscape of a model evaluation scoring function may be more complex, with more 'local minima' that allow non-native structures to receive high scores. Another potential explanation is that the scoring functions for alignment methods, which have benefitted from a wealth of sequence data, are more sensitive than the statistical potentials used in model evaluation, which are based on a relatively much smaller database of structures. Lastly, since the number of different structural forms appears to be limited, the great number of different sequences with similar structures make the native conformations difficult to recognize.

Table 6.8 suggests, however, that only a small improvement in the accuracy of model evaluation will result in large gains in returned model quality. The bottom row lists the TM-score of the best model among the top 10 as scored by ProsaII. There is clearly a significant improvement over the quality of the single model with the highest pG, nearly equaling the quality of the best model of the entire ensemble. Furthermore, as noted in Section 6.5, outside information can be brought to bear on the model evaluation problem to remove those models that lack a key feature of the query.

7.2 The lower limit of sequence similarity

While it is clear that the accuracy of any alignment method decreases as sequence identity moves from high to low, this relationship is less clear at the lowest levels of identity. Of course, what is retained between the most remote structural homologs is sequence similarity. It is this similarity that allows scoring functions to detect corresponding regions of the query and template. One would expect that for two proteins to be structurally homologous, they must retain at minimum some small level of similarity no matter how divergent their sequences. For example, a pair of structural homologs must at the very least have hydrophobic residues in corresponding sequence positions so that they are buried in the core of both proteins. Also, the preference of certain amino acid types for either α-helices or β-strands makes it likely that some additional sequence similarity will be found between proteins with secondary structure elements in a topologically equivalent arrangement.

It is plausible that there could be a difference in terms of ease of alignment between proteins that share only this level of incidental similarity and those with an actual, though very distant, evolutionary relationship. One would expect homologs that evolved by convergent evolution to share only this incidental similarity, while those that diverged from a common ancestor to retain somewhat more. Though research has suggested that, given enough evolutionary distance, there is little difference in terms of *identity* between homologs that evolved through convergent or divergent paths (22), it seems plausible that more *similarity* would be retained during divergent evolution. Since mutations in diverging sequences must accomodate themselves into the existing structure of the protein, there is a bias toward replacement with a similar residue that is absent for converging sequences. In regards to S4, perhaps this difference in similarity partially accounts for the difference between pairs in this region of very low sequence identity that were and were not aligned correctly.

Digging a bit deeper, we can use detection by PSI-BLAST in Experiment 3 as an indication of similarity, but then remove the pairs with the highest sequence identity until the average is in the range of Experiments 1 and 2. Table 7.1 is similar to Table 6.7, but with Experiment 3* denoting this low identity segment of the homologs detected by PSI-BLAST. In every measure, the S4 alignments and models from Experiment 3* are better than those produced from the other experiments. Since the sets from Experiments 1 and 2 were found by structure alignment techniques, their 4-5% identity may approximate the incidental similarity of convergent homologs, while this low identity segment of PSI-BLAST hits may be more representative of divergent proteins.

		All Homologs		
	Experiment	1	2	3*
	ID%	4.9	4.6	4.6
TM-score	Str	0.62	0.55	0.60
	S4 1000	0.48	0.45	0.54
	S4 100	0.45	0.42	0.51
IAD	S4 1000	3.2	2.4	1.9
	S4 100	5.3	3.9	2.9
FDS2	S4 1000	65.4	73.4	80.3
	S4 100	55.7	63.1	72.5

Table 7.1: Comparison of low identity pairs across experiments - This table presents a comparison of S4's results in the experiments with the higher identity pairs from Experiment 3 removed to make a new set, 3*.

7.3 Future work

Several parts of the S4 methodology seem adaptable to other uses. Perhaps most obviously, the inability of ProsaII to consistently distinguish models from homologous and non-homologous pairs suggests that S4 might be a useful technique for generating decoy sets for the development and testing of model evaluation methods. The goal of decoy generation is to create a set of biologically plausible, but non-native structures (89). With its emphasis on creating modelable alignments and its internal use of the DFIRE statistical potential, S4 would seem well-suited to this task.

Figure 6.3 suggests another interesting application of S4. Since the method was successful at returning good alignments to templates identified by PSI-

BLAST throughout the range of e-values from 0.001 - 10, it appears likely that this upper limit could be extended further before the quality of alignments decreases to an unacceptable level. This could result in a great number of newly identified remote structural homologs. Though the proportion of true homologs among PSI-BLAST hits will decrease as the e-value increases, the number of templates returned should increase and offset this effect, as is seen in the horizontally consistent distribution in Figure 6.3. Table 7.1 suggests that extending PSI-BLAST may be a more effective means of identifying remote homologs than blind searches of sequence and structure databases.

Extending our ability to model proteins based on remote homology is an important goal. These tools allow us to know the structure of an increasingly large fraction of all sequences, thereby granting us a better understanding of possible functions and interactions. The study of how remote homologs can share a common three-dimensional shape may also lead to a better understanding of the complex relationship between sequence and structure.

References

[1] C. Anfinsen, E. Haber, M. Sela, and F. White, Jr. **The Kinetics of Formation of Native Ribonuclease During Oxidation of the Reduced Polypeptide Chain**. *Proc Natl Acad Sci USA*, **47**(9):1309–14, 1961. 2

[2] C. Levinthal. **How to Fold Graciously**. *Mossbauer Spectroscopy in Biological Systems*, pages 22–24, 1969. 3

[3] C. Anfinsen. **Principles that Govern the Folding of Protein Chains**. *Science*, **181**(4096):223–30, 1973. 4

[4] J. Bowie, R. Lüthy, and D. Eisenberg. **A Method to Identify Protein Sequences That Fold into a Known Three-Dimensional Structure**. *Science*, **253**:164–70, 1991. 5, 20, 33

[5] K. Yue and K. Dill. **Inverse protein folding problem: Designing polymer sequences**. *Proc Natl Acad Sci USA*, **89**(9):4163–67, 1992. 5

[6] A. Street and S. Mayo. **Computational Protein Design**. *Structure*, **7**(5):R105–9, 1999. 5

[7] C. Levinthal. **Are there pathways for protein folding?** *J. Chim. Phys.*, **65**:44–45, 1968. 6

[8] K. Dill and H. Chan. **From Levinthal to pathways to funnels**. *Nature Structural Biology*, **4**(1):10–19, January 1997. 6

[9] R. Zwanzig, A. Szabo, and B. Bagchi. **Levinthal's Paradox**. *Proc Natl Acad Sci USA*, **89**:20–22, January 1992. 7

[10] A. Sali, E. Shakhnovich, and M. Karplus. **How does a protein fold?** *Nature*, **369**:248–51, May 1994. 7

[11] R. Baldwin and G. Rose. **Is protein folding hierarchic? I. Local structure and peptide folding**. *Trends Biochem Sci*, **24**(1):26–33, January 1999. 7

[12] R. Baldwin and G. Rose. **Is protein folding hierarchic? II. Folding intermediates and transition states**. *Trends Biochem Sci*, **24**(2):77–93, February 1999. 7

[13] H. Scheraga, M. Khalili, and A. Liwo. **Protein-Folding Dynamics: Overview of Molecular Simulation Techniques**. *Annu Rev Phys Chem*, **58**:57–83, 2007. 8

[14] W. Cornell, P. Cieplak, C. Bayly, I. Gould, K. Merz, D. Ferguson, D. Spellmeyer, T. Fox, J. Caldwell, and P. Kollman. **A Second Generation Force Field for the Simulation of Proteins, Nucleic Acids, and Organic Molecules**. *JACS*, **117**(19):5179–5197, 1995. 8

[15] A. MacKerell, D. Bashford, M. Belott, R.L. Dunbrack, Jr., J. Evanseck, and M. Karplus. **All-Atom Empirical Potential for Molecular Modeling and Dynamic Studies of Proteins**. *J Phys Chem B*, **102**(18):3586–3616, 1998. 8

[16] Y. Duan and P. Kollman. **Pathways to a Protein Folding Intermediate Observed in a 1-Microsecond Simulation in Aqueous Solution**. *Science*, **282**:740–44, October 1998. 8

[17] V. Pande and D Rokhsar. **Molecular dynamics simulations of unfolding and refolding of a beta-hairpin fragment of protein G**. *Proc Natl Acad Sci USA*, **96**:9062–67, August 1999. 8

[18] S. Jang, E. Kim, S. Shin, and Y. Pak. **Ab Initio Folding of Helix Bundle Proteins Using Molecular Dynamics Simulations**. *JACS*, **125**(48):14841–46, 2003. 8

[19] A. Liwo, M. Khalili, and H. Scheraga. **Ab initio simulations of protein-folding pathways by molecular dynamics with the united-residue model of polypeptide chains**. *Proc Natl Acad Sci USA*, **102**(7):2362–2367, February 2005. 8

[20] R. Das, B. Qian, S. Raman, R. Vernon, J. Thompson, P. Bradley, S. Khare, M. Tyka, D. Bhat, D. Chivian, D. Kim, W. Sheffler, L. Malmstro, A. Wollacott, W. Chu, A. Ingemar, and David B. **Structure prediction for CASP7 targets using extensive all-atom refinement with Rosetta@home**. *Proteins*, **69 Suppl**(8):118–28, 2007. 8

[21] P. Alexander, Y. He, Y. Chen, J. Orban, and P. Bryan. **The design and characterization of two proteins with 88% sequence identity but different structure and function**. *Proc Natl Acad Sci USA*, **104**(29):11963–8, 2007. 9

[22] B. Rost. **Protein structures sustain evolutionary drift**. *Folding & Design*, **2**(3):S19–24, 1997. 10, 121

[23] D. Kihara and J. Skolnick. **The PDB is a Covering Set of Small Protein Structures**. *J Mol Biol*, **334**:793–802, 2003. 10, 67

[24] Y. Zhang and J. Skolnick. **The protein structure prediction problem could be solved using the current PDB library**. *Proc Natl Acad Sci USA*, **102**(4):1029–34, January 2005. 10, 67

[25] Y. Zhang, I. Hubner, A. Arakaki, E. Shakhnovich, and J. Skolnick. **On the origin and highly likely completeness of single-domain protein structures**. *Proc Natl Acad Sci USA*, **103**(8):2605–2610, February 2005. 10, 67

[26] A. Fiser and A. Sali. **Modeller: generation and refinement of homology-based protein structure models**. *Meth Enzymol*, **374**:461–91, 2003. 11

[27] R. Norel, D. Petrey, and B. Honig. **PUDGE: a flexible, interactive server for protein structure prediction**. *Nucleic Acids Research*, **38**(Suppl):W550–4, 2010. 12

[28] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. Shindyalov, and P. Bourne. **The Protein Data Bank**. *Nucleic Acids Research*, **28**(1):235–42, 2000. 13, 67

[29] B. Rost. **Twilight zone of protein sequence alignments**. *Protein Eng.*, **12**(2):85–94, 1999. 13, 34, 67

[30] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman. **Basic Local Alignment Search Tool**. *J Mol Biol*, **215**:403–410, 1990. 13

[31] S. Altshcul, T. Madden, A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. Lipman. **Gapped BLAST and PSI-BLAST: a new generation of protein database search programs**. *Nucleic Acids Research*, **25**(17):3389–3402, 1997. 13, 91

[32] Christopher L. Tang, Lei Xie, Ingrid Y. Y. Koh, Shoshana Posy, Emil Alexov, and Barry Honig. **On the Role of Structural Information in Remote Homology Detection and Sequence Alignment: New Methods Using Hybrid Sequence Profiles**. *J Mol Biol*, **334**(5):1043–1062, 2003. 13, 33, 42, 72

[33] H. Zhou and Y. Zhou. **Fold recognition by combining sequence profiles derived from evolution and from depth-dependent structural alignment of fragments**. *Proteins*, **58**(2):321–8, 2005. 13, 72

[34] A. S. Yang and B. Honig. **An integrated approach to the analysis and modeling of protein sequences and structures. I. Protein structural alignment and a quantitative measure for protein structural distance**. *J Mol Biol*, **301**(3):665–78, 2000. 14, 28, 69

[35] Y. Zhang and J. Skolnick. **Scoring function for automated assessment of protein structure template quality**. *Proteins*, **57**(4):702–10, 2004. 14, 57, 69, 70, 76, 102

[36] J. Moult, K. Fidelis, A. Kryshtafovych, B. Rost, T. Hubbard, and A. Tramontano. **Critical assessment of methods of protein structure prediction-Round VII**. *Proteins*, **69**(Suppl 8):3–9, 2007. 14, 19, 69

[37] Z. Xiang and B. Honig. **Extending the Accuracy Limits of Prediction for Sidechain Conformations**. *J Mol Biol*, **311**:421–30, 2001. 16, 18

[38] D. Petrey, Z. Xiang, C. L. Tang, L. Xie, M. Gimpelev, T. Mitros, C. S. Soto, S. Goldsmith-Fischman, A. Kernytsky, A. Schlessinger, I. Y. Koh, E. Alexov, and B. Honig. **Using multiple structure alignments, fast model building, and energetic analysis in fold recognition and homology modeling**. *Proteins*, **53**(Suppl 6):430–5, 2003. 16, 28, 69, 70

[39] A. Sali and T. Blundell. **Comparative Protein Modelling by Satisfaction of Spatial Restraints**. *J Mol Biol*, **234**:779–815, 1993. 16

[40] H. van Vlijmen and M. Karplus. **PDB-based protein loop prediction: parameters for selection and methods for optimization**. *J Mol Biol*, **267**:975–1001, 1997. 17

[41] J. Wojcik, J. Mornon, and J. Chomilier. **New efficient statistical sequence-dependent structure prediction of short to medium-sized protein loops based on an exhaustive loop classification**. *J Mol Biol*, **289**:1469–1490, 1999. 17

[42] A. Fiser, R. Do, and A. Sali. **Modeling of loops in protein structures**. *Protein Sci*, **9**:1753–1773, 2000. 17

[43] A. Canutescu and R.L. Dunbrack, Jr. **Cyclic coordinate descent: A robotics algorithm for protein loop closure**. *Protein Sci*, **12**:963–72, 2003. 17

[44] A. Canutescu, A. Shelenkov, and R.L. Dunbrack, Jr. **A graph-theory algorithm for rapid protein side-chain prediction**. *Protein Sci*, **12**:2001–14, 2003. 18

[45] X. Li, M. Jacobson, and R. Friesner. **High-resolution prediction of protein helix positions and orientations**. *Proteins*, **55**(2):368–82, 2004. 18

[46] C. Rohl, C. Strauss, D. Chivian, and D. Baker. **Modeling structurally variable regions in homologous proteins with rosetta**. *Proteins*, **55**(3):656–77, 2004. 18

[47] J. Zhu, L. Xie, and B. Honig. **Structural refinement of protein segments containing secondary structure elements: Local sampling, knowledge-based potentials, and clustering**. *Proteins*, **65**(2):463–79, 2006. 18

[48] H. Zhou and Y. Zhou. **Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction**. *Protein Sci*, **11**(11):2714–26, 2002. 20, 47

[49] M.J. Sippl. **Recognition of errors in three-dimensional structures of proteins**. *Proteins*, **17**(4):355–62, 1993. 20, 76, 91

[50] R. Sánchez and A. Sali. **Large-scale protein structure modeling of the Saccharomyces cerevisiae genome**. *Proc Natl Acad Sci USA*, **95**(23):13597–602, 1998. 20, 76, 86, 98, 100

[51] N. Mirkovic, Z. Li, A. Parnassa, and D. Murray. **Strategies for high-throughput comparative modeling: applications to leverage analysis in structural genomics and protein family organization**. *Proteins*, **66**(4):766–77, 2007. 20, 76, 93, 98, 100, 114

[52] R. Lüthy, J. Bowie, and D. Eisenberg. **Assessment of protein models with three-dimensional profiles**. *Nature*, **356**:83–5, 1992. 21

[53] I. Shindyalov and P. Bourne. **Protein structure alignment by incremental combinatorial extension (CE) of the optimal path**. *Protein Eng.*, **11**(9):739–47, 1998. 28

[54] L. Holm and C. Sander. **Protein Structure Comparison by Alignment of Distance Matrices**. *J Mol Biol*, **233**:123–38, 1993. 28

[55] S. Needleman and C. Wunsch. **A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins**. *J Mol Biol*, **48**:443–53, 1970. 29

[56] R. Edgar and K. Sjolander. **A comparison of scoring functions for protein sequence profile alignment**. *Bioinformatics*, **20**(8):1301–8, 2004. 33

[57] M. Marti-Renom, M. Madhusudhan, and A. Sali. **Alignment of protein sequences by their profiles**. *Protein Sci*, **13**(4):1071–87, 2004. 33

[58] S. Liu, C. Zhang, S. Liang, and Y. Zhou. **Fold recognition by concurrent use of solvent accessibility and residue depth**. *Proteins*, **68**(3):636–645, 2007. 33

[59] J. Shi, T. Blundell, and K. Mizuguchi. **FUGUE: sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties**. *J Mol Biol*, **310**(1):243–57, 2001. 33

[60] G.J. Barton and M. Sternberg. **Evaluation and improvements in the automatic alignment of protein sequences**. *Protein Eng.*, **1**(2):89–94, 1987. 33

[61] A. Lesk, M. Levitt, and C. Chothia. **Alignment of the amino acid sequences of distantly related proteins using variable gap penalties**. *Protein Eng.*, **1**(1):77–78, 1986. 33

[62] M. Madhusudhan, M. Marti-Renom, R. Sanchez, and A. Sali. **Variable gap penalty for protein sequence-structure alignment**. *Protein Eng.*, **19**(3):129–33, 2006. 33

[63] S. Benner, M. Cohen, and G. Gonnet. **Empirical and structural models for insertions and deletions in the divergent evolution of proteins**. *J Mol Biol*, **229**(4):1065–82, 1993. 33

[64] B. Qian and R. Goldstein. **Distribution of Indel Lengths**. *Proteins*, **45**(1):102–4, 2001. 33

[65] N. Goonesekere and B. Lee. **Frequency of gaps observed in a structurally aligned protein pair database suggests a simple gap penalty function**. *Nucleic Acids Research*, **32**(9):2838–43, 2004. 33

[66] M. S. Waterman. **Sequence alignments in the neighborhood of the optimum with general application to dynamic programming**. *Proc Natl Acad Sci USA*, **80**(10):3123–3124, 1983. 36, 70, 74

[67] M. Zuker. **Suboptimal sequence alignment in molecular biology. Alignment with error analysis**. *J Mol Biol*, **221**(2):403–20, 1991. 36

[68] M. Saqi and M. Sternberg. **A simple method to generate non-trivial alternate alignments of protein sequences**. *J Mol Biol*, **219**(4):727–32, 1991. 36

[69] B. John and A. Sali. **Comparative protein structure modeling by iterative alignment, model building and model assessment**. *Nucleic Acids Research*, **31**(14):3982–92, 2003. 37

[70] D. Chivian and D. Baker. **Homology modeling using parametric alignment ensemble generation with consensus and energy-based model selection**. *Nucleic Acids Research*, **34**(17):e112, 2006. 37

[71] L. Jaroszewski, W. Li, and A. Godzik. **In search for more accurate alignments in the twilight zone**. *Protein Sci*, **11**(7):1702–1713, 2002. 37

[72] R.B. Russel, R.R. Copley, and G.J. Barton. **Protein fold recognition by mapping predicted secondary structures**. *J Mol Biol*, **259**(3):349–365, 1996. 46

[73] J.M. Sauder, J.W. Arthur, and R.L. Dunbrack, Jr. **Large-scale comparison of protein sequence alignment algorithms with structure alignments**. *Proteins*, **40**(1):6–22, 2000. 54, 74

[74] M. Cline, R. Hughey, and K. Karplus. **Predicting reliable regions in protein sequence alignments**. *Bioinformatics*, **18**(2):306–14, 2002. 56

[75] H. Chen and D. Kihara. **Estimating quality of template-based protein models by alignment stability**. *Proteins*, **71**(3):1255–74, 2008. 56

[76] P. Bork. **Powers and Pitfalls in Sequence Analysis: The 70% Hurdle**. *Genome Res*, **10**:398–400, 2000. 67

[77] M. Waterman, T. Smith, and W. Beyer. **Some Biological Sequence Metrics**. *Advances in Mathematics*, **20**(3):367–87, 1976. 67

[78] S. Williams, I. Vakonakis, S. Golden, and A LiWang. **Structure and function from the circadian clock protein KaiA of *Synechococcus elongatus*: A potential clock input mechanism**. *PNAS*, **99**(24):15357–62, 2002. 80

[79] A. Mac Sweeney, R. Lange, R. Fernandes, H. Schulz, G. Dale, A. Douanganmath, P. Proteau, and C. Oefner. **The Crystal Structure of *E. Coli* 1-Deoxy-D-xylulose-5-phosphate Reductoisomerase in a Ternary Complex with the Antimalarial Compound Fosmidomycin and NADPH Reveals a Tight-binding Closed Enzyme Conformation**. *J Mol Biol*, **345**(1):115–27, 2005. 80

[80] A. Murzin, S. Brenner, T. Hubbard, and C. Chothia. **SCOP: A Structural Classification of Proteins Database for the Investigation of Sequences and Structures**. *J Mol Biol*, **247**:536–40, 1995. 80

[81] K. Ginalski, J. Pas, L. Wyrwicz, M. von Grotthuss, J. Bujnicki, and L Rychlewski. **ORFeus: detection of distant homology using sequence profiles and predicted secondary structure**. *Nucleic Acids Research*, **31**(13):3804–07, 2003. 92

[82] L. Rychlewski, L. Jaroszewski, W. Li, and A. Godzik. **Comparison of sequence profiles. Strategies for structural predictions using sequence information**. *Protein Science*, **9**:232–41, 2000. 92

[83] J. Park, K. Karplus, C. Barrett, R. Hughey, D. Haussler, T. Hubbard, and C. Chothia. **Sequence Comparisons Using Multiple Sequences Detect Three Times as Many Remote Homologues as Pairwise Methods**. *J Mol Biol*, **284**:1201–10, 1998. 92

[84] R. Sánchez, U. Pieper, F. Melo, N. Eswar, M. Marti-Renom, M. Madhusudhan, N. Mirkovic, and A. Sali. **Protein structure modeling for structural genomics.** *Nature Structural Biology*, **7**:986–990, 2000. 93

[85] M. Chance, A. Fiser, A. Sali, U. Pieper, N. Eashwar, G. Xu, T. Radahakanan, and N. Marinkovic. **High-throughput computational and experimental techniques in structural genomics**. *Genome Res*, **14**:2145–54, 2004. 93

[86] A. Sali. **100,000 protein structures for the biologist**. *Nature Structural Biology*, **5**:1029–32, 1998. 93

[87] D. Vitkup, E. Melamud, J. Moult, and C. Sander. **Completeness in structural genomics**. *Nature Structural Biology*, **8**:559–66, 2001. 93

[88] J. Davis and M. Goadrich. **The Relationship Between Precision-Recall and ROC Curves**. *Proceedings of the 23rd Annual Conference on Machine Learning*, 2006. 100

[89] B. Park and M. Levitt. **Energy Functions that Discriminate X-ray and Near-native Folds from Well-constructed Decoys**. *J Mol Biol*, **258**(367-92), 1996. 113, 122

[90] T. Lazaridis and M. Karplus. **Discrimination of the Native from Misfolded Protein Models with an Energy Function Including Implicit Solvation**. *J Mol Biol*, **288**:477–87, 1998. 113

[91] D. Petrey and B. Honig. **Free energy determinants of tertiary structure and the evaluation of protein models**. *Protein Science*, **9**:2181–91, 2000. 113

[92] B. Wallner and A. Elofsson. **Can correct protein models be identified?** *Protein Sci*, **12**:1073–86, 2003. 113

[93] H. Lee, Z. Li, A. Silkov, M. Fischer, D. Petrey, B. Honig, and D. Murray. **High-throughput computational structure-based characterization of protein families: START domains and implications for structural genomics**. *J Struct Funct Genomics*, **11**:51–59, 2010. 114

[94] D. Benson, I. Karsch-Mizrachi, D. Lipman, J. Ostell, and D. Wheeler. **GenBank: update**. *Nucleic Acids Research*, **32**:D23–D26, 2004. 114