K2: Protein Structure Comparisons and Their Statistical Significance

Joseph D. Szustakowski and Zhiping Weng
Boston University, Department of Biomedical Engineering, Boston, MA 02215
Corresponding author: Zhiping Weng (zhiping@bu.edu, phone 617-353-3509, fax 617-353-6766).

ABSTRACT

We previously developed a protein structure alignment algorithm that uses a genetic algorithm (KENOBI, Szustakowski and Weng, Proteins 2000; 38:428-440) and generates high-quality alignments judged using manually curated structure alignments. We have improved the algorithm’s performance with the use of a rapid first stage vector-based SSE alignment. Here we analyze the importance of three key genetic algorithm operators in the new version of our algorithm (K2, available at http://sullivan.bu.edu/k2). We also present the formulation of statistical significance calculations and the results of a large-scale database comparison. K2 can detect 82% of homologous protein pairs, compared to 10% by the most widely used pairwise sequence algorithm BLAST and 39% by another structure alignment algorithm.

Abbreviations

AA amino acid; GA genetic algorithm; PDF probability density function; RMSD root mean square deviation; SSE secondary structure element; 3D three-dimensional

1 INTRODUCTION

The past decade has witnessed a rapid accumulation of protein three-dimensional (3D) structure information. As of June 19, 2001, there were 15,435 entries in the Protein Data Bank (PDB, (Berman, Westbrook et al. 2000)). Furthermore, structural genomics (Burley, Almo et al. 1999) aims to determine a large number of protein structures with broad architecture coverage. The considerable amount of structure data has established structure comparison as an essential technique for understanding protein sequence, structure, function, and evolution.

A structure comparison algorithm identifies a set of residue equivalencies between two proteins based on their 3D coordinates, called the structure alignment. This set of equivalencies allows the superposition of one protein onto the other after rigid rotation and/or translation. Structure alignments find applications in two areas. First, they can indicate if two proteins share the same fold. Strong structural similarity is often the result of functional similarity and evolutionary relatedness (Rozwarski, Gronenborn et al. 1994). Since structures are more conserved than sequences (Chothia and Lesk 1986; Murzin 1996; Murzin 1998), structure comparison can identify more distant homologs than sequence comparison. Second, structure comparison is the most accurate computational technique to delineate the functionally equivalent residue pairs in two proteins. As a result, structure comparison is an integral part of many structure prediction methods such as homology modeling and threading. It is also used as the gold standard for testing such algorithms.
A large number of structure alignment algorithms exist (see review (Eidhammer, Jonassen et al. 2000) and the references therein). We have recently developed a structure alignment algorithm called KENOBI, with the goal of generating detailed and biologically meaningful alignments (Szustakowski and Weng 2000). Since protein structures are mostly conserved in cores, KENOBI first aligns protein cores, as represented by secondary structure elements (SSEs). It then uses a genetic algorithm (GA) to optimize the alignment according to an elastic similarity score. Subsequently, it extends the SSE alignment to include any positions in loops or turns deemed equivalent in a convergent process. KENOBI was tested on eight highly representative protein families and proven to be robust. Specifically, KENOBI is able to generate high quality alignments that are in complete agreement with manually curated alignments, as well as with experimental results (Szustakowski and Weng 2000).

We have enhanced KENOBI with the addition of two important features that are included in a C++ program named K2, available for use over the internet (http://sullivan.bu.edu/k2). K2’s improvements include a rapid vector-based SSE alignment that prefaces the GA, and statistical significance calculations for the resulting alignments. Our aim in this chapter is to describe these enhancements in detail. We will also revisit the genetic algorithm and try to gauge the impact its various operators and their parameters have on its performance.

The use of vector-based SSE alignments is not new. The basic idea is to represent two proteins’ SSEs with vectors and then identify a set of equivalent vectors. Several previous efforts used graph-theoretic and/or clustering techniques (Grindley, Artyymiuk et al. 1993; Rufino and Blundell 1994; Madej, Gibrat et al. 1995; Mizuguchi and Go 1995; Alexandrov and Fischer 1996) to identify equivalent SSEs, while others employed a geometric hashing method (Holm and Sander 1995; Alesker, Nussinov et al. 1996) borrowed from the computer vision field. The vector representation of SSEs is popular because it greatly reduces the computational complexity of the structure alignment problem. This method is not without drawbacks, chief among them the inability to identify specific amino acid (AA) equivalencies. Several methods refine the SSE alignments at the amino acid level using a variety of techniques that include rigid body transformations, least squares fitting, Monte Carlo sampling, and dynamic programming. (Holm and Sander 1995; Madej, Gibrat et al. 1995; Alesker, Nussinov et al. 1996; Alexandrov and Fischer 1996) In contrast, other methods avoid the vector abstraction altogether and aim to align the specific amino acid positions directly. (Holm and Sander 1993; Gerstein and Levitt 1998; Szustakowski and Weng 2000)

K2 successfully marries a fast vector-based SSE alignment technique with a slower but reliable genetic algorithm that aligns the amino acid positions. This combination of methods leverages their distinct advantages. Vector alignments can be computed very quickly and efficiently. In addition to correctly identifying equivalent SSEs they can exclude dissimilar proteins from the more time consuming AA refinement. The GA was previously proven capable of correctly aligning protein structures from a population of randomly generated SSE alignments. (Szustakowski and Weng 2000) It follows the “SSE first” philosophy by searching for the optimal alignment of amino acids within SSEs. The introduction of the vector-based alignments provides an intelligent, directed method for selecting an initial population of alignments worthy of detailed refinement.
In contrast to the realm of sequence alignment, where the basic theory exists (Karlin and Altschul 1990) and its careful implementation has become an indispensable component of the widely used programs BLAST and PSI-BLAST (Altschul, Madden et al. 1997), little work has been done on the statistical significance of structure alignments. Notably, VAST computes a P-value based on the number of aligned SSEs and the probability of aligning the same number of SSEs by chance (Gibrat, Madej et al. 1996); Holm and Sander quantify the significance of a structure alignment using the number of standard deviations it scores above the mean (Z-score) of a distribution of scores generated from an all-against-all comparison of a nonredundant structure database (Holm and Sander 1993; Holm and Sander 1996). Levitt and Gerstein convincingly showed that structure comparison scores follow an extreme-value distribution in a manner analogous to sequence comparison scores. They derived a P-value measure for the alignment score empirically from a set of nonredundant structure alignments (Levitt and Gerstein 1998).

Here we have applied Levitt and Gerstein’s approach and extended it in several ways. Their alignment score is a quadratic function of the distances between equivalent residues in the two proteins. We employ a different alignment score - the elastic similarity score developed by Holm and Sander (Holm and Sander 1993). Levitt and Gerstein’s program can only perform sequentially constrained alignments. K2 can perform both constrained and non-constrained alignments; we have developed statistical significance calculations for non-constrained alignments. Finally, K2’s statistical significance calculations take into account the protein’s size in a simpler fashion than theirs; moreover, K2’s calculations also account for the SSE composition of the proteins.

Equipped with the newly developed statistical significance measure, we tested K2’s ability to identify similar proteins by aligning all pairs of proteins assigned to the same SCOP fold (Hubbard, Ailey et al. 1999), a total of 199,789 alignments. K2’s sensitivity ranged from 75% for the loosest definition of similar proteins, to 89% for the strictest definition. BLAST, the most popular pairwise sequence alignment algorithm, exhibited sensitivities from 5% to 22% for the same datasets. Levitt and Gerstein, using an older version of SCOP (v32), reported a sensitivity of 39%; for the same similarity criteria and SCOP (v53) K2’s sensitivity was 82%.

As a test of specificity, we aligned 10,000 pairs of unrelated proteins. At the same statistical significance threshold used in the sensitivity tests, only 30 alignments were deemed similar for a specificity of 99.7%.

2 METHODS

The K2 algorithm employs three basic stages. K2 first searches for the best alignment of the proteins’ SSEs. In the second stage, a genetic algorithm manipulates the initial SSE alignments to optimize the alignment of the amino acid positions within SSEs. Finally, the protein backbones are superposed based on the position equivalencies determined in stages one and two, and then searched for additional equivalent positions in non-SSE regions.
This hierarchical approach to alignment reflects the nature of protein structures. In general, it is known that protein cores are more conserved than surfaces, loops, and turns. By focusing first on aligning the SSEs, K2 reduces the overall search space to that portion most likely to yield meaningful results. Moreover, the first stage SSE alignments can be accomplished in a rapid and efficient fashion that provides the second stage genetic algorithm an initial population of optimal SSE alignments.

2.1 Stage 1: SSE Alignments

K2’s first stage seeks to properly align the SSEs from the two proteins. Protein SSEs are computed by DSSP (Kabsch and Sander 1983) and then smoothed (http://bmerc-www.bu.edu/needle-doc/latest/dssp-progs.html). Each SSE is modeled as if it were a vector by first determining the SSE’s principal axis with the smallest moment, and then projecting the first and last α-carbon onto this axis to assign endpoints. The task then is to find a set of equivalent vectors from the two proteins.

Three non-collinear points uniquely define a 3D coordinate system. Suppose we knew a priori of three equivalent points in both proteins. We could use these points to define internal coordinate systems for the two proteins that would place equivalent SSEs in close proximity in 3D space, making the task of identifying them straightforward. In reality, we have no such prior knowledge and instead must search for such sets of points that will transform the protein structures into equivalent coordinate systems.

Two non-collinear SSEs can provide four points. Two pairs of equivalent SSE vectors should define internal coordinate systems that correctly orient the two proteins. One plausible strategy would be to use all possible pairs of SSEs as basis for transformations. For two proteins of N and M SSEs respectively, we would have to consider \( N(N-1)M(M-1) \) transformation pairs\(^1\). We can reduce this search space with a simple heuristic. Instead of using all pairs of SSEs, it is sufficient to choose only those in physical proximity. If on average each SSE has \( Q \) neighboring SSEs, the number of transformation pairs we must consider is \( Q^2NM \). \( Q \) is largely independent of protein size – SSEs can only accommodate a limited number of neighbors – and is typically less than six.

Given two SSE vectors, K2 defines a coordinate system by first placing the origin at the center of one with the x-axis aligned with the vector and the positive direction toward it carboxyl end. The orientation of the y-axis is determined by the cross product of the two SSE vectors. Finally, the orientation of the z-axis is determined by the cross product of the x- and y-axes.

Once we have transformed the proteins, it is still necessary to determine which SSEs in protein A are equivalent to which SSEs in protein B. By computing the Euclidean distances between all SSEs in protein A and B we can generate a matrix \( D_{ij} \) whose entries are the distances between SSEs \( A_i \) and \( B_j \). This distance matrix can be easily converted to a similarity matrix \( S_{ij} \), which we can use to assess the goodness of potential SSE alignments.

---

\(^1\) It is important to note that the actual number of transformations may be less than \( N(N-1)M(M-1) \), depending on the precise SSE composition of the proteins. It is safe to assume that α-helices and β-strands cannot be equivalent. K2 considers only those transformation pairs based on equivalent SSE types. For the remainder of this discussion we will ignore this detail to simplify our analysis.
When constructing an alignment, we wish to consider two cases. One is the sequentially constrained alignment, in which the SSEs must maintain the same ordering in the alignment as in the proteins’ primary sequences. That is to say, if SSE $A_i$ is aligned to $B_m$, and $A_j$ is aligned to $B_n$, and if $i < j$, then $m < n$ must also be true. This sequentially constrained case is analogous to the sequence alignment problem; for such cases K2 calculates optimal alignments using the semi-global version of the Needleman-Wunsch (Needleman and Wunsch 1970) dynamic programming algorithm.

For non-sequentially constrained alignments we turn our attention to the well-studied matching or assignment problem, a classic example of which is the marriage problem. Imagine we are presented with a set of boys, a set of girls, and a matrix $L_{ij}$ that describes how well each girl $i$ likes each boy $j$, and that we are charged with the task of marrying the boys and girls and making them as happy as possible. Our goal is to assign the boys to girls in a way that maximizes the sum of the $L_{ij}$ scores over all couples. This problem is commonly modeled as a bipartite graph with weighted edges, and solved by identifying the matching with the maximum sum of weights. It should be clear that our problem is entirely analogous to the marriage problem, with SSEs from proteins $A$ and $B$ replacing the sets of boys and girls, respectively, and our $S_{ij}$ matrix taking the place of the $L_{ij}$ matrix. There are several algorithms that solve this problem exactly. K2 uses the fastest currently known which runs in $O(nm+n\log m)$ time. (Fredman and Tarjan 1984) This is only slightly worse than the $O(nm)$ time dynamic programming used for sequentially constrained alignments.

Once the SSEs have been aligned, either by dynamic programming or by maximal matching, K2 inspects the alignments in 3D space to ensure they make sense physically. All matched vector pairs are subjected to two constraints. First, the closest approach between two aligned SSEs must be less than 10 Å. Second, the angle formed by the two vectors in 3D space must be less than ninety degrees. Any SSE pair that fails either of these tests is removed from the alignment by matching the SSEs with gaps. The default distance and angle cutoffs are lenient and meant only to eliminate pairings that are clearly incorrect; these values can be adjusted for stricter or looser filtering of the SSE alignments.

### 2.2 Stage 2: Detailed Alignments Generated Using Genetic Algorithm

Once we have decided upon a SSE alignment we are still left with the difficult task of determining the specific AA assignments. To select an optimal alignment we must decide on two things: How do we evaluate the goodness of an alignment? and How do we search for good alignments?

For evaluating alignments K2 uses the elastic similarity score developed by Holm and Sander (Holm and Sander 1993):

$$S = \begin{cases} \sum_{i=1}^{L} \sum_{j=1}^{L} \left( \theta - \frac{d_{ij}}{d_{ij}} \right)^2 \left( \frac{\theta_{ij}}{\theta_{ij}} \right)^2, & i \neq j \\ \theta_{ij}, & i = j \end{cases}$$

(1)
where $d^A_{ij}$ and $d^B_{ij}$ are the distances between equivalent positions $i$ and $j$ in protein $A$ and $B$; $\overline{d}_{ij}$ is the average of $d^A_{ij}$ and $d^B_{ij}$; $\theta$ is a constant set to 0.2; $a$ is a constant set to 20 Å. The principal behind this score is rather simple – equivalent positions in two proteins should have similar distances to other equivalent positions. Consider two simple proteins, $A$ and $B$, with positions 1, 2, and 3 in protein $A$ equivalent to positions 1’, 2’, and 3’ in $B$, respectively. It stands to reason that since 1 and 1’ are equivalent, and 2 and 2’ are equivalent, the distance between 1 and 2 ($d^A_{12}$) should be nearly equal to the distance between 1’ and 2’ ($d^B_{1’2’}$). The first term of equation (1) quantifies this by rewarding pairs of aligned positions whose distances deviate less than 20% from their average. The exponential term is a scaling factor that down weights the contribution of distant pairs to the overall score. It should be noted that this scoring function is applied only to aligned positions – unaligned positions do not contribute to the alignment score.

We have previously developed a genetic algorithm for aligning protein structures (Szustakowski and Weng 2000). That algorithm randomly generated an initial population of SSE alignments and optimized them according to equation (1) to arrive at correct AA position alignments. Here we apply the same GA to the SSE vector alignments described in section 2.1. These vector alignments are near-optimal and thus give the GA a tremendous head start, enabling the GA to find optimal AA alignments with greater frequency and speed.

The first stage of our algorithm leaves us with one optimal SSE alignment for every pair of transformations; for moderate sized proteins, we will easily have one hundred or more such alignments. The GA requires a diverse initial population of SSE alignments. Rather than simply selecting the single highest scoring SSE alignment for optimization, we have found the GA performs best on a broader sampling of the high scoring alignments. K2’s default behavior is to generate an initial population of 100 alignments from the 50 highest scoring SSE alignments. This population is biased to include more copies of the higher scoring SSE alignments, imparting on them a competitive advantage in comparison to the other alignments.

The basic unit of the alignment is the SSE element. Every SSE has a stop and start position that are fixed and assigned by DSSP. (Kabsch and Sander 1983) SSEs also have variable boundaries that are adjusted by the GA as it optimizes the alignment (see Figure 1a). These variable boundaries are initially assigned randomly. Every SSE is aligned either with an SSE from the other protein or with a null element (Figure 1b). SSEs aligned with other SSEs contribute to the scoring function; those aligned with null elements are effectively unaligned and do not contribute to the scoring function. SSEs may only be aligned with other SSEs of the same type – $\alpha$-helices with $\alpha$-helices, and $\beta$-strands with $\beta$-strands.

The genetic algorithm employs several stochastic operators that make both small and large modifications to the alignments in an attempt to improve their scores. The GA is set to run for a specified number of rounds (by default 200). All alignments are evaluated according to equation (1) at the conclusion of every GA round. Alignments with improved scores keep all modifications made to them. Those with diminished scores are reverted to their previous state before proceeding to the next round.
The mutation operator is responsible for fine-tuning the precise AA pairings. This operator modifies the adjustable boundaries of aligned SSEs (see Figure 1a) in one of several ways. It can change the lengths of the aligned regions by extending or shrinking the adjustable borders. It can also shift the adjustable borders of one SSE in either direction. By default, the mutation operator adjusts SSE pairs with 10% chance. The exact nature of the mutation is determined randomly.

The stage one alignment procedure typically identifies optimal SSE alignments, but may in some cases miss a pair of equivalent SSEs or include a pair of SSEs that are not in fact equivalent. Moreover, even if the vector alignment procedure identifies the correct SSE alignment, the initial population (taken from the 50 highest scoring SSE alignments) is likely to include both optimal and near-optimal alignments. The hop operator was designed to target alignments with misaligned SSEs. Alignments are subject to this operator at a rate of 5%. The hop operator essentially switches the positions of two SSEs in an alignment (see Figure 1c). For non-sequentially constrained alignments, the selection of one SSE is biased towards those which make small or negative contributions to the overall alignment’s score, and the second is chosen blindly from the remaining SSEs. For constrained alignments it is imperative that the relative ordering of SSEs be maintained. To guarantee this, the operator selects a null element as the first SSE and one of its immediate neighbors as the second.

One of the advantages of genetic algorithms is the use of a population of solutions; this allows GAs to simultaneously optimize many solutions. Moreover, GAs routinely allow the solutions to recombine and form improved solutions from parts of existing ones. K2’s GA has two operators that affect such behavior.

As noted previously, K2 allows SSEs to align only with other SSEs of the same type. The GA leverages this distinction with the swap operator. The swap operator randomly picks (5% chance by default) a pair of alignments and exchanges all of the aligned helices in one for all of the aligned helices in the other. This alteration can be extremely effective when one alignment has well aligned helices and the other well aligned strands.

We are left with the coarsest-grained operator employed by the GA, the crossover (See Figure 1d). The crossover is similar in spirit to the swap operator, but its actions are not limited to a single SSE type. The crossover operator is applied to every alignment every round. Each alignment is randomly paired with another alignment. Each pair of alignments is then subjected to several manipulations. First, the strand SSE pairs are sorted, and a crossover point randomly selected. The two alignments then exchange all SSE pairs on one side of the crossover point. Next, the helix pairs are crossed over in like fashion. Finally the alignments are modified to ensure they each contain exactly one copy of each SSE from each protein.

### 2.3 Stage 3: 3D Superposition

Once the GA has completed, the highest scoring alignment is subjected to a simple refinement protocol. The refinement first eliminates SSE pairs with negative scores from the alignment. It then attempts to improve the alignment’s score by extending or shrinking the aligned SSEs, and shifting the aligned SSEs relative to each other.

---

2. SSEs are selected with a probability proportional to \( e^{-s} \) where \( s \) is the SSE’s contribution to the alignment’s score.

3. Equivalently, K2 could swap strands rather than helices.
The proteins are then superposed in 3D space based on the alignment’s equivalent AA positions so that K2 can search their backbones for additional equivalent positions. Two positions are deemed equivalent if they are each other’s nearest neighbors and if the distance between them is less than some threshold (5 Å by default). Following this recruitment the alignment is pruned by discarding any pairs that are too distant from each other (> 5Å) or that do not occur in a run of four or more aligned residues. This process of superposition, recruitment, and pruning is repeated until the alignment converges.

2.4 Evaluation of Statistical Significance

To determine if two protein structures are similar, it is necessary to first align the structures and then quantify the goodness of fit between the structures. The elastic similarity score defined by Equation (1) has proven to be a very good target function, capable of discriminating between optimal and sub-optimal alignments of two structures. It is not, however as well suited for comparing alignments of different structures for several reasons. It is an additive function that scales roughly with the square of the length of the alignment. While this property is useful when trying to find the optimal alignment of two structures (every correctly aligned pair of residues increases the value of the target function), it makes comparing alignments of different structures difficult. For example, is an alignment of two large structures with 100 aligned residues and a raw score of 150 better or worse than an alignment of two small structures with 90 aligned residues and a score of 120?

Confounding this length dependency is the importance of the SSE composition of the aligned structures on the alignment score. In general, alignments of all α-proteins have higher scores than alignments of all β-proteins, while scores for mixed proteins fall between these two. Because of these complications, it is impossible to compare alignments of different proteins in a meaningful fashion based on raw elastic similarity scores alone.

What is needed is a measurement of the statistical significance of the alignment scores. Specifically, given two protein structures and an alignment score, we would like to know the probability of obtaining an equal or better score by chance alone. An excellent example of such a measure is the BLAST P-value. BLAST is a popular tool used for searching large protein or DNA sequence databases for sequences similar to a query sequence. BLAST’s popularity is in part due to its rigorous statistical significance calculations (Karlin and Altschul 1990). For our purposes, the key feature of these calculations is the use of an extreme value distribution (Gumbel 1958) to model the background probabilities of sequence alignment scores. Extreme value distributions often result from optimization procedures and have a characteristic “slow” or “heavy” tail that decays in exponential fashion. It has previously been shown that, like sequence alignments scores, structure alignment scores follow an extreme value distribution. (Levitt and Gerstein 1998)

The extreme value distribution probability density (PDF) function is of the form:

\[ \rho(Z) = e^{-Z}e^{-e^{-Z}} \]  

(2)

where Z is a normalized score. For BLAST sequence alignments Z is defined as:
\[ Z = \lambda S - \ln(\kappa L) \]  

(3)

where \( S \) is the alignment score, \( \lambda \) and \( \kappa \) are parameters, and \( L \) is the product of the length of the two sequences. Once \( Z \) is calculated, we would like to know the probability of obtaining an alignment with equal or greater score by chance alone. This quantity is known as a P-value and is calculated by integrating the PDF (2) from \( Z \) to \( +\infty \). In general this value is:

\[ P(Z) = 1 - e^{-e^Z} \]  

(4)

Karlin and Altschul were able to solve the BLAST statistics for non-gapped alignments analytically (Karlin and Altschul 1990). The complex nature of the structure alignment problem and elastic similarity score make an analytic solution for K2’s statistics infeasible. Instead, we approached the problem empirically, using a dataset of roughly two million pairwise structure alignments to estimate the parameter values for Equation (3). These alignments were generated from approximately two thousand unrelated non-redundant protein domains and protein substructures.

In adapting Equation (3) for structure alignments the structure similarity score is clearly analogous to the sequence alignment score \( S \). For sequence alignments, \( L \) is equal to the product of the lengths of the sequences, and is used as a measurement of the sequence alignment search space. K2 operates primarily on SSEs, of which there are two distinct types, so we must use a different length measurement. We have found that the product of the number of alignable SSEs\(^4\) in the two proteins makes an effective length measurement.

Each of the 2 million alignments was assigned to one of three categories based on the structures’ alignable SSE compositions: all-\( \alpha \), all-\( \beta \), and mixed. These datasets were binned according to alignment length and score. All bins measured 5 length units by 5 score units. The data was normalized such that for any given length, the frequencies over all scores summed to one. In this manner, the distribution of alignments was segmented into a family of curves with unit area, one curve for each length bin. We then used this data to estimate values for \( \lambda \) and \( \kappa \) for each length curve. Simple logarithmic plots of the results demonstrated that \( \lambda \) and \( \kappa \) were functions of \( L \) of the general form:

\[ a_1 e^{-a_1 L} + b_1 e^{-b_1 L} \]  

(5)

By fitting Equation (5) to the binned data of various lengths, it was possible to estimate the values of \( a_1, a_2, b_1, \) and \( b_2 \) (see Table 1). Given these parameters and \( L \) it is then possible to compute \( \lambda \) and \( \kappa \), which in turn are used to compute \( Z \).

---

\(^4\)An SSE is deemed alignable if and only if the other protein contains an SSE of the same type. For example, if protein \( A \) contained 2 strands and 3 helices, and protein \( B \) contained 4 strands and 5 helices, then both the strands and helices would be alignable and the length would be \((2*4)*(3*5) = 120\). If protein \( B \) had 4 strands and no helices then only the strands would be alignable for a length of \((2*4) = 8\).
3 RESULTS AND DISCUSSION

To assess the accuracy of a structure alignment it is necessary first to establish a gold standard for comparison. Gerstein and Levitt (Gerstein and Levitt 1998) argued convincingly that the best basis for comparisons are manually curated structure alignments. We have previously shown (Szustakowski and Weng 2000) that the genetic algorithm described above can consistently produce structure alignments that agree with such standards. Here, we revisit some of the more difficult alignments we’ve encountered to highlight K2’s improved performance.

3.1 Difficult Cases

One deceptively difficult alignment involves 7FABL2 and 1REIA, an immunoglobulin constant domain and variable domain, respectively. While both proteins exhibit the characteristic immunoglobulin β-sandwich fold, the repetitive nature of these structures presents alignment algorithms with many locally optimal alignments in which they may easily become trapped.

1TRNA and 1KXF are both serine proteases; as such, they each contain a catalytic triad of three amino acids: Asp-His-Ser. These proteins are very distantly related – 1TRNA is human trypsin, while 1KXF is a viral coat protein – and share effectively no sequence identity. Moreover, the strands in 1KXF tend to be distorted, shifted, and broken when compared to their counterparts in other serine proteases. While there is no manually curated structure alignment for this pair, we can still verify the correctness of automated structure alignments by checking for proper alignment of the conserved catalytic triad positions (Figure 2).

Streptavidin (1STP) and avidin (1AVE) are both β-barrel structures. Much like the immunoglobulins, these structures are highly repetitive and present algorithms with many enticing local optima.

We have found that two DNA-metyltransferases, 1BOO and 2ADMA, make an excellent test case for the non-sequentially constrained version of the algorithm. The catalytic domains of these proteins share a similar overall structure that consists of a seven-stranded β-sheet surrounded by three α-helices on either side (see Figure 3). The primary sequences of these proteins have been circularly permuted, resulting in a rearrangement of the order of the equivalent SSEs in the two proteins.

We can see from Table 2 that the current version of our algorithm has much improved convergence properties compared to the old version. In fact, even when faced with these most difficult test cases, K2 converges to the correct alignments for 100 of 100 different random seeds. This improved performance must be a result of the stage one SSE alignments, because the other procedures (GA, refinement, superposition) are unchanged from the previous version. The stage one alignments hand the GA near-optimal alignments, so the GA needs only to perform a local search to optimize the AA pairings and perhaps eliminate or recover a mis- or un-aligned SSE pair.
3.2 GA Performance Characteristics

To characterize the genetic algorithm’s performance, we aligned the serine proteases and methyltransferases (see above) both with and without the crossover operator, and with several different mutation and hop rates.

We have observed that the crossover operator is the GA’s driving force. This phenomenon is clearly illustrated in Figures 4a-b. Here we see the effects on performance of turning off the crossover operator for the serine protease and methyltransferase alignments. With the crossover, K2 quickly converges to an optimal alignment in both cases; without the crossover K2 is unable to reach a similar score within 350 GA rounds.

The crossover’s power is largely a result of its flexibility. When two alignments are recombined the changes may be drastic, subtle, or somewhere in between. Two dissimilar alignments may create offspring that bear little resemblance to them and include many different aligned SSE pairs. As the GA progresses and the population becomes more optimal and homogenous recombinations tend to fine-tune rather than rearrange alignments. Recombining two alignments with correctly aligned SSEs typically adjusts the specific AA alignments of several SSEs at once.

To gauge the effects of the mutation operator we aligned the same two pairs of proteins with mutation rates ranging from 0 to 1. K2 performs very poorly when the mutation rate is set to zero; this setting effectively turns off the mutation operator and makes it nearly impossible for K2 to fine-tune the specific AA alignments. As the mutation rate increases from 0 to 0.10, K2 can once again search for correct AA alignments, and performance improves markedly (see Figures 4c-d). We observed no further improvements in performance as the mutation rate increased from 0.10 (not shown). As the mutation rate approached 1.0 we found that performance actually worsened. A mutation rate of 0.75 substantially hindered the GA’s performance for the serine proteases, and had a more moderate but noticeable effect for the methyltransferases. For such high mutation rates, it is likely that the alignments will be subjected to one or more mutations. We have found that most mutations actually hurt the alignment and are eventually rejected. When the mutation rate is low, alignments occasionally receive a beneficial mutation that is then carried into subsequent GA rounds. When the mutation rate is high, such good mutations tend to be countered by bad mutations elsewhere in the alignment, and are therefore rejected. This again makes it nearly impossible for the GA to fine-tune the alignments.

Results for the hop rate for the serine proteases are largely analogous to those of the mutation rate. Once again we see that small and large hop rates under perform moderate settings. The highest-scoring initial SSE alignments included two pairs of SSEs that were not in fact equivalent. The hop operator corrected these errors by dutifully realigning the SSEs with null elements.

Varying the hop rate had no effect on performance for the methyltransferases. As in the serine protease case, the highest-scoring initial SSE alignments also contained two incorrectly aligned pairs of SSEs. This leads us to ask: How was the GA able to correct these errors when the hop operator was turned off? The answer is quite simple. The initial population of SSE alignments contained several lower-scoring alignments that were missing some of the correct SSE pairs as
well as the incorrect SSE pairs. At various stages of the GA the crossover operator recombined higher- and lower-scoring alignments and produced new alignments with all correctly aligned SSEs.

### 3.3 Statistical Significance Calculations

To characterize the behavior of K2’s P-value calculations we applied several tests in a manner similar to that described by Levitt and Gerstein. Specifically, we used two bases for comparison: BLAST (Altschul, Madden et al. 1997) sequence alignments, and the SCOP database. (Hubbard, Ailey et al. 1999) The SCOP database classifies protein domain structures using a combination of automatic and manual methods. Each domain is placed in a hierarchy that describes its class, fold, superfamily, and family. We need only concern ourselves with the fold and superfamily designations. The SCOP fold describes a protein’s major SSEs and their topological connectivity and can be thought of as a general characterization of what a protein “looks like”. Membership in the same SCOP fold does not necessarily imply evolutionary relatedness. Proteins are designated as members of the same superfamily if they are thought to be related; that is to say if structural and functional similarities suggest a common ancestor.

We aligned 3839 SCOP protein domains\(^5\) for a total of 199,789 alignments. Each protein was aligned with the other domains in the same SCOP fold. We also aligned these proteins’ sequences with BLAST. (Altschul, Madden et al. 1997) Results are shown in Figure 5a. For each alignment we have plotted its BLAST and K2 P-values. The general behavior of the data is consistent with what is known about the relationship between protein sequence and structure. Proteins with little or no sequence similarity (e.g. BLAST P-value > 10\(^{-5}\)) can exhibit a range of structural similarities – some may be very similar while others have little or no structural similarity. As sequence similarity increases, say from BLAST P-values of 10\(^{-15}\) to 10\(^{-60}\), we observe that the number of proteins with poor or moderate structural similarity decreases. In other words, as sequence similarity increases across this range, so does structural similarity. Finally, we note that those proteins with very similar sequences (P-value < 10\(^{-60}\)) nearly always have very similar structures.

While Figure 5a offers us a useful glimpse into this behavior, the sheer density of points on the lower and right edges of the graph makes it a bit misleading. In Figure 5b we have extended this graph into the third dimension, using the height of each bin to represent the number of data points contained therein. We now see that the overwhelming majority of data points follow the sequence-structure relationship we just described above.

For a more detailed analysis, it is useful to segregate the data into several categories. We shall use SCOP’s fold and superfamily classifications to define our positive test set, i.e. proteins with structures that truly are similar. We can pick P-value thresholds for each method that differentiates “similar” proteins from “dissimilar” proteins. We have chosen a P-value of 10\(^{-2}\) for BLAST and K2. These two thresholds effectively divide the aligned proteins into four categories: those identified as similar by K2 and BLAST (lower left quadrant), those identified

---

\(^5\) Structures were taken from the SCOP(v53) list of proteins with less than 95% pairwise identity as defined by ASTRAL. (Brenner, Koehl et al. 2000) The list was pruned to exclude structures solved by NMR and include only those from the SCOP all-\(\alpha\), all-\(\beta\), \(\alpha/\beta\), and \(\alpha+\beta\) classes.
by K2 but not by BLAST (lower right), those identified by BLAST but not by K2 (upper left), and those identified by neither method (upper right).

It is useful to consider several definitions for our positive test set. We’ve chosen four separate criteria: pairs of proteins within the same fold with sequence identity < 95%; pairs within the same fold with < 40% identity (the most difficult case); pairs within the same superfamily with < 95% identity (the easiest case); and pairs within the same superfamily with < 40% identity.

A summary of these results is shown in Table 3. With P-value thresholds of $10^{-2}$ we observed that K2’s sensitivity ranges from 75% for the most difficult positive test set to 89% for the easiest, while BLAST’s sensitivity ranges from 5% to 22%. Levitt and Gerstein performed a similar analysis with an older version of SCOP (v32) (Levitt and Gerstein 1998). Using the same-superfamily, < 40% identity criteria, they calculated 2,107 pairwise alignments of 941 different structures and reported a sensitivity of 39%. For the same criteria and a newer version of SCOP (v53), K2’s sensitivity was 82%. The composition and organization of the SCOP database has changed significantly from v32 to v53 making a direct comparison of these values infeasible.

Alignments in the lower left quadrant are similar enough to be identified by both sequence and structure.

Those in the lower right quadrant are identifiable by structure, but not by sequence. Such proteins generally fall into two categories: related proteins whose sequences have diverged beyond recognition, and unrelated proteins whose structures have converged to the same fold. Ark-clam hemoglobin (d3sdha_) and human-hemoglobin (d1babb_) are examples of the former. These proteins are certainly related – they have highly similar structures (K2 P = 7*10^{-10}), and both use a heme group to bind oxygen. Their sequences however have diverged greatly, sharing only 21% sequence identity at a BLAST P-value of 0.40.

Two proteins from the SCOP immunoglobulin-like fold provide an example of structure convergence. β-galactosidase (d1bgl1) is a sugar-cleaving enzyme, while the fibronectin type-III module (d1fna__) is a cell-adhesion protein. There is no evidence to suggest these proteins are related; they carry out very different molecular functions and have no significant sequence similarity (BLAST P = 1). Their structures, however are very similar, and can be aligned over 56 residues at 1.67 Å RMSD for a K2 P-value of 2.3*10^{-5}.

Of the 199,789 alignments, 79 were identified as similar by BLAST but not by K2. They are found in the upper left quadrant. 76 of these 79 alignments are from five folds: EF hand-like (43), immunoglobulin-like β-sandwich (15), long α-hairpin (8), sh3-like barrel (7), and phospholipase (3). These structures fool K2 because they contain shifted, twisted, or broken SSEs, exist in bound and unbound states, are bound to different ligands, or belong to SCOP folds defined by three or fewer core SSEs.

Data points in the upper right quadrant represent levels of structural similarity undetectable by automatic methods. In addition to the characteristics described above for the upper left quadrant alignments, many of these structures simply do not contain large regions that can be rigidly
superposed. Specifically, while they do contain the same major SSEs, their sizes and relative orientations vary greatly from protein to protein, making it impossible to superpose more than a small subset of them at once.

We must not forget that to be useful the K2 P-value calculations must not only be sensitive, but specific as well. As a test of specificity, we aligned 10,000 pairs of proteins randomly selected from different SCOP folds; such pairs should share little or no structural similarity and provide a suitable negative test set. Only 30 of the 10,000 alignments had P-values less than $10^{-2}$, and only 9 had P-values less than $10^{-3}$. Visual inspection of these 30 alignments confirmed they are not mistakes; in fact, they all exhibit fair amounts of structural similarity despite their classifications in different folds. These alignments are not errors by K2 or SCOP’s curators, but rather examples of nature’s knack for reusing successful protein substructures. The 30 alignments each belong to one of four categories: β-sheets (18), α-β-α sandwiches (7), α-β structures (4), and three-helix-bundles (1). In Figure 6 we see two such examples. d1vid__ and d1qq5a (Figure 6a) both contain a large, α-β-α sandwich that aligns 70 residues with 2.59 Å RMSD and a P-value of $8.72 \times 10^{-5}$. d1ec7a1 and d2vaaa1 (Figure 6b) have smaller common substructures, three-stranded β-sheets that align at 1.29 Å RMSD over 27 residues with a P-value of $6.52 \times 10^{-4}$.

What does this all mean? Our sensitivity analysis indicates we can use K2 and a P-value threshold of $10^{-2}$ to identify similar proteins structures with a sensitivity of between 75 and 89%, depending on our positive test set criteria. In general, we would expect to find between 4 and 15 times as many similar proteins using K2 rather than BLAST. The specificity calculations indicate that at a P-value of $10^{-2}$ we expect a false positive rate of 30 per 10,000 alignments, or a specificity of 99.7%. Moreover, we have seen that these false positives are not errors, but rather are unrelated proteins that share common substructures.

4 CONCLUSIONS

We have presented a method (K2) to compare protein structures that utilizes a wide variety of techniques. K2 is largely based on a method (KENOBI) we previously developed that uses a genetic algorithm to align protein structures. K2’s major improvement is the introduction of a fast vector-based SSE alignment stage. This step provides the genetic algorithm with intelligently selected, nearly optimal alignments that greatly enhances its performance.

Analysis of the various genetic algorithm operators revealed their flexibility in determining optimal amino acid equivalencies between two proteins. The operators function at a variety of levels, from specific amino acid positions (mutate) to entire classes of SSEs (swap). We observed that the GA is largely driven by the crossover. Moreover, the crossover operator is flexible enough to also perform the duties of other operators when they are turned off.

Finally, we developed a framework for calculating the statistical significance of the alignments generated by K2. Comparisons to a structure classification database and sequence alignments revealed that it is both highly sensitive and specific and can be used to identify proteins that share a common fold.
Acknowledgements

This work was supported by NSF grant DBI#0078194. The authors thank Compaq Computer Corporation’s High Performance Technical Computing group for use of the Compaq BioCluster, a configuration of twenty-six AlphaServer ES40 compute nodes (104 CPUs, 120 Gb total memory) and one ES40 1-terabyte fileserver.
<table>
<thead>
<tr>
<th>Protein Type</th>
<th>( a_1 )</th>
<th>( a_2 )</th>
<th>( b_1 )</th>
<th>( b_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \lambda ), all-( \alpha )</td>
<td>0.089</td>
<td>0.056</td>
<td>0.025</td>
<td>0.00005</td>
</tr>
<tr>
<td>( \kappa ), all-( \alpha )</td>
<td>2.3</td>
<td>0.076</td>
<td>0.035</td>
<td>( 10^{-7} )</td>
</tr>
<tr>
<td>( \lambda ), all-( \beta )</td>
<td>0.32</td>
<td>0.095</td>
<td>0.05</td>
<td>0.0006</td>
</tr>
<tr>
<td>( \kappa ), all-( \beta )</td>
<td>1.7</td>
<td>0.21</td>
<td>0.19</td>
<td>0.013</td>
</tr>
<tr>
<td>( \lambda ), mixed</td>
<td>0.062</td>
<td>0.04</td>
<td>0.059</td>
<td>0.0018</td>
</tr>
<tr>
<td>( \kappa ), mixed</td>
<td>0.70</td>
<td>0.065</td>
<td>0.14</td>
<td>0.0095</td>
</tr>
</tbody>
</table>
Table 2 K2 Convergence Results

<table>
<thead>
<tr>
<th>Alignment</th>
<th>Old algorithm</th>
<th>New algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1REIA, 7FABL2</td>
<td>85</td>
<td>100</td>
</tr>
<tr>
<td>1STP, 1AVE</td>
<td>80</td>
<td>100</td>
</tr>
<tr>
<td>1TRNA, 1KXF</td>
<td>98</td>
<td>100</td>
</tr>
<tr>
<td>1BOO, 2ADMA</td>
<td>80</td>
<td>100</td>
</tr>
</tbody>
</table>

Convergence data for four alignments. All new alignments were performed without sequential constraints. For the old alignments, 1REIA-7FABL2 and 1TRNA-1KXF were aligned with sequential constraints, while 1STP-1AVE and 1BOO-2ADMA were aligned without sequential constraints.
<table>
<thead>
<tr>
<th>Positive Set</th>
<th>LL</th>
<th>LR</th>
<th>UL</th>
<th>UR</th>
<th>Total</th>
<th>K2 Sensitivity</th>
<th>BLAST Sensitivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>same-fold, 95% identity</td>
<td>31909</td>
<td>133709</td>
<td>79</td>
<td>34092</td>
<td>199789</td>
<td>83%</td>
<td>16%</td>
</tr>
<tr>
<td>same-fold, 40% identity</td>
<td>1582</td>
<td>22237</td>
<td>8</td>
<td>7927</td>
<td>31754</td>
<td>75%</td>
<td>5%</td>
</tr>
<tr>
<td>same-superfamily, 95% identity</td>
<td>31909</td>
<td>95722</td>
<td>79</td>
<td>16541</td>
<td>144251</td>
<td>89%</td>
<td>22%</td>
</tr>
<tr>
<td>same-superfamily, 40% identity</td>
<td>1582</td>
<td>11519</td>
<td>8</td>
<td>2873</td>
<td>15982</td>
<td>82%</td>
<td>10%</td>
</tr>
</tbody>
</table>

Positive set indicates if the protein pairs are from the same fold or same superfamily (defined by SCOP v53), and the maximum pairwise sequence identity. LL: the lower left quadrant of Figure 5a; protein pairs predicted to be homologous by both K2 and BLAST. LR: the lower right quadrant; protein pairs predicts to be homologous by K2 but not by BLAST. UL: the upper left quadrant; protein pairs predicted to be homologous by BLAST but not by K2. UR: the upper right quadrant; protein pairs predicted to be nonhomologous by both K2 and BLAST. Total indicates the total number of protein pairs and is equal to LL+LR+UL+UR. K2 sensitivity is equal to (LL+LR)/Total; BLAST sensitivity is equal to (LL+UL)/Total.


Figure Captions

**Figure 1.** Aligning two protein structures using a Genetic Algorithm. Each protein structure is represented as a set of Secondary Structure Elements (SSEs), shown here as boxes.
(a) An aligned pair of SSEs: positions 5-15 from SSE 1 are aligned with positions 12’-22’ from SSE 3’.
(b) An example SSE alignment with SSE 1 aligned with 1’, 2 aligned with 2’, and a null element aligned with 3’.
(c) The hop operator. Here, the null element and SSE 2 trade places.
(d) The crossover operator has selected two alignments (one in white, the other in gray) to manipulate. First it randomly selects a crossover point, illustrated here as a dashed line. It then recombines the alignments around this point to produce two new alignments.

**Figure 2.** Alignment of a human protease (1TRNA, gray) and a viral protease (1KXF, black). The residues composing the conserved Asp-His-Ser catalytic triad are shown in ball-and-stick form. Thick backbone regions denote alignable regions; thin backbone segments denote unaligned regions.

**Figure 3.** Alignment of two DNA methyltransferases with circularly permuted primary sequences: 1BOO in black and 2ADMA in gray.

**Figure 4.** K2’s performance over a variety of parameter settings. For all figures, the ordinate represents the population’s best score after each round of the Genetic Algorithm. Results were averaged over 5 runs with different random number generator seeds. (a) Methyltransferases and (b) serine proteases with and without the crossover operator; (c) Methyltransferases and (d) serine proteases with various mutation rates; (e) Methyltransferases and (f) serine proteases with various hop rates.

**Figure 5.** A comparison of the P-values of sequence alignments (calculated using BLAST) and structural alignments (calculated using K2). A P-value is the probability of obtaining an alignment with equal or better score by chance. 199,789 alignments have been included, representing all possible pairs of structures in the same fold according to the SCOP database (version 53).
(a) A two-dimensional plot with each dot representing one alignment. The dashed lines indicate P-values of $10^{-2}$ and separate the pairs into four categories: those identified by K2 and BLAST (lower left quadrant on the graph), those identified by K2 but not by BLAST (lower right), those identified by BLAST but not by K2 (upper left), and those identified by neither method (upper right).
(b) A three-dimensional histogram. The alignments have been binned according to their p-values; the heights of the bins correspond to the base-10 logarithm of the number of counts in the bin. Not shown in these plots are 400 points with BLAST P-value < $10^{-100}$ and K2 P-value < $10^{-16}$. 
Figure 6. Two examples of structures from different SCOP folds with significant K2 P-values ($< 10^{-3}$).
(a) d1vid__ (black) and d1qq5a (gray) both share a similar $\alpha$-$\beta$-$\alpha$ sandwich motif.
(b) d1ec7a2 (black) and d2vaal (gray) contain similar $\beta$-sheets.
Figure 1
Figure 1
Figure 4a
Figure 4b
Figure 4c
Figure 4d
Figure 4e
Serine Proteases

Figure 4f
Figure 5a
Figure 5b