

Prospect II: protein structure prediction program for the genome-scale application

Dongsup Kim, Dong Xu, Juntao Guo, Kyle Ellrott, and Ying Xu

*Protein Informatics Group, Life Science Division
Oak Ridge National Laboratory, Oak Ridge, TN 37831-6480, USA*

May 18, 2002

Abstract

A new efficient and reliable fold recognition component is added to the protein structure prediction package PROSPECT. There are four key features. (1) The evolutionary information is fully utilized. In addition to the new profile-profile alignment method, we introduce the efficient way to utilize the evolutionary information when we calculate threading potentials including singleton and pairwise energies. (2) We obtain the sequence-structure alignment using dynamic programming by optimizing the combined score of several scores that measure sequential and structural compatibility between sequence and structure. (3) From a given sequence-structure alignment, the alignment model is derived that represents characteristics of the alignment. By using these alignment models, the support vector machine (SVM) is trained to recognize true folds. (4) The confidence score that measures the reliability of prediction is introduced. It is found that the evolutionary information and other new features in PROSPECT greatly improve not only the alignment accuracy but also the fold recognition performance. The tests on several benchmarks indicate that the alignment accuracy of PROSPECT is significantly better than previous methods. We also demonstrate that the performance of PROSPECT on homology recognition is comparable or superior to any other method available at all levels of similarity.

1 Introduction

Due to the rapidly growing list of completed genome sequences, the need for fast, reliable, and automated computational tools for inferring structures and functions of the protein sequences is increasing. Recognizing known protein structures that are closely related to protein sequences with unknown structures and functions is the first step towards understanding their biological functions and predicting three-dimensional structures by comparative modelling. As defined by the popular SCOP hierarchical protein classification system, evolutionary relationships among related proteins can be classified into three categories: family, superfamily, and fold. The aim of current work is to develop a fast and reliable sequence-structure homology recognition method at family and superfamily levels to complement a more general fold-recognition software PROSPECT [Xu and Xu, 2000] that has been developed by our group. The objective of PROSPECT is to recognize the evolutionary relationship not only at the family and superfamily levels but also the fold level.

Generally, the fold recognition methods adopt several distinctive approaches. The first approach is solely based on sequence information. The hidden Markov model (HMM) methods and PSI-BLAST can be classified into this category. The second approach is to utilize structural information in a variety of ways. In profile method introduced by Bowie *et al.* [Bowie *et al.*, 1991], structural information is coded into each residue of the structural templates, and various dynamic programming alignment schemes (global, semi-global, local, and global-local) are used to recognize the homologous pairs. In threading method, the compatibilities between a query sequence and known protein structures are calculated based on the optimal sequence-structure alignments that minimize the knowledge-based potential functions including hydrophobicity function and the pairwise interaction. It has been demonstrated in many previous studies that each approach has its own strengths and shortcomings. For example, threading-based method such as THREADER performs worse in homology recognition at family and superfamily levels than sequence-based methods, while it shows the best performance at fold level recognition [Lindahl and Elofsson, 2000]. Motivated by these observations, many groups have attempted to combine both approaches [Jones, 1999, Panchenko *et al.*, 2000, Kelly *et al.*, 2000, Shi *et al.*, 2001], although optimal way to fully take advantage of both approaches were found to be nontrivial [Lindahl and Elofsson, 2000].

Despite their powerful features, the biggest disadvantage of the threading-based methods is that they are computationally expensive. It has been proven that the threading problem is an NP-hard problem. Unlike many heuristic algorithms including double dynamic programming [Jones *et al.*, 1992], frozen approximation [Godzik *et al.*, 1992], and Monte Carlo sampling algorithm [Bryant, 1996], PROSPECT [Xu and Xu, 2000] guarantees to find a globally optimal threading alignment in computationally efficient way. However, the computational cost of PROSPECT is still prohibitively large to be applied in genomic scale. Therefore, developing a computationally effective sequence-structure homology recognition method is highly desirable. The objective of this work is to develop a fast, reliable, and automated method to recognize homologs at family/superfamily level. There are several methods available that have been developed for the similar purpose. For example, GenTHREADER [Jones, 1999] uses a traditional sequence alignment algorithm to generate alignments, and then evaluates the alignments by a threading potentials, and finally produces a single measure of confidence in the proposed prediction by neural network. The program 3D-PSSM [Kelly *et al.*, 2000] generates large superfamily-based multiple sequence alignments for each protein in their structural library, and then combines these information with structural information and the sequence profile from PSI-

BLAST to encode the 1D- and 3D-profiles into each residue, which are matched against query sequence or profile to produce the alignment score. The unique feature of the program FUGUE [Shi *et al.*, 2001] is that it utilizes the environment-specific substitution tables and structure-dependent gap penalties. Some groups [Rychlewski *et al.*, 2000, Yona and Levitt, 2002] have developed the profile-profile alignment algorithms instead of the usual sequence-sequence or sequence-profile alignment algorithms.

Our approach is similar to the previous works in many aspects. We use all of sequential, structural, and evolutionary information and try to combine these information in an optimal way to maximize the performance. There are many innovative features in our method. First, the evolutionary information is fully utilized. In addition to the new profile-profile alignment method, we introduce an efficient way to utilize the evolutionary information in estimating the threading potentials including singleton and pairwise energies. Second, we obtain the sequence-structure alignment using dynamic programming by optimizing the combined score of several scores that measure sequential and structural compatibility between sequence and structure. At this stage, the pairwise interaction energies are not included in the combined score, and the optimal alignment is obtained by the usual dynamic programming algorithm in order to achieve high computational efficiency. The weighting factors for each score are optimized systematically to give the optimal alignment accuracy for the training set. Third, for a given alignment, “alignment model” is derived that represents characteristics of the alignment. The features in the alignment model are carefully chosen in order to maximize the recognition accuracy. They include the size of query sequence and template, total alignment score, contributions from all energy components including pairwise energy, the alignment length, and other features. The model is evaluated by the general pattern recognition technique known as the support vector machine (SVM) [Vapnik, 1995, Vapnik, 1998, Burges, 1998], to produce a single score that reflects the confidence level of the query sequence being related to the template structure.

The SVM draws a hyperplane in a multidimensional feature space to separate a set of binary labeled (positive and negative) training data. In this work, the SVM has been trained to primarily detect family/superfamily level relationship. If two proteins belong to the same superfamily according to the SCOP classification, the alignment model of these proteins is labeled as a positive model. The reason for this procedure is that in order to detect fold level relationship it requires a more computationally demanding alignment scheme that optimizes the threading potential that includes the pairwise interaction potential. It should be noted that all the automated fold recognition methods such as GenTHREADER, 3D-PSSM and FUGUE that use dynamic programming have been designed to perform family/superfamily level homology recognition. Therefore, our strategy is two-stage procedure. For a given query sequence we first try to find family/superfamily level relationships in template library. If there is no template whose SVM confidence score is greater than threshold value, we assume that there is no structure in our template library that belongs to the same superfamily with a query sequence, and apply a conventional threading method to detect fold level similarity. As a result, we have two alignment algorithms in PROSECT: a dynamic programming algorithm when we ignore the pairwise interactions and “divide-and-conquer” algorithm when the pairwise interactions are included during optimizing sequence-structure alignments.

In this paper, we describe our effort to develop a method to recognize family/superfamily level relationship and add a new capability to our protein structure prediction package PROSPECT. The main focus of this work is to make the method fast, so that it can be applied to a whole genome, and reliable, *i.e.*, low false positive rate. The novel ideas such as homolog-averaged

potential energy and confidence score by SVM are the key components of the method. In addition, more accurate potential energy parameters were developed by both improving potential energy models and using bigger and more complete protein structure database. These ideas and new parameters can be applied to the conventional threading method.

2 Overview of Prospect

In this section, we describe a brief overview of PROSPECT. The PROSPECT consists of four components:

- (1) non-redundant set of 3D protein structures to be used as threading templates,
- (2) knowledge-based energy functions to measure the fitness between a query sequence and a structural template,
- (3) threading algorithm to find the optimal alignment between a query sequence and a template,
- (4) and algorithm to estimate the confidence level of the predicted structure.

PROSPECT can use as the template library both the protein chains from the FSSP database and the protein domains from the DALI domain library. Currently, the FSSP database is default in PROSPECT. Each template was derived from a chain (FSSP) or a domain segment (DALI) in a PDB file, and contains four pieces of information; (1) sequence, (2) secondary structure types (α -helix, β -strand, and loop) assigned by the DSSP package, (3) solvent accessibilities (buried, intermediate, and exposed states) determined from the percentage of exposed solvent accessible surface area of a residue’s side chain according to the DSSP package, (4) C_β atom coordinates. The template file also contains the information on whether the residues are in the core region (α -helix and β -strand) or in the loop region. It is generally believed that the core region is more conserved among the structures of the same fold than the loop region. Some previous works have utilized this property in the form of structure-dependent gap penalty [Shi *et al.*, 2001] or by allowing no gaps within a core region [Xu and Xu, 2000]. In PROSPECT, the information on core region is utilized in two ways. First, only the pairwise interactions between the core region is taken into account. Second, in “divide-and-conquer” algorithm, we assume that no gap can occur within a core region. Not all the α -helix and β -strand regions are assigned to be a core region. They should meet the minimum length (currently 5) requirement.

The knowledge-based energy function, which will be described in later section in details, has the following form:

$$E_{total} = \omega_m E_{mutation} + \omega_s E_{singleton} + \omega_p E_{pairwise} + \omega_g E_{gap} + \omega_{ss} E_{ss}. \quad (1)$$

The mutation energy $E_{mutation}$ describes the compatibility between the template sequence and the query sequence. In the old PROSPECT, we used the conventional sequence-sequence alignment score using the PAM250 matrix [Gonnet *et al.*, 1992]. In the current version of PROSPECT, we employed the profile-profile alignment energy using the profile information generated by PSI-BLAST. The singleton energy $E_{singleton}$ represents the sum of the preferences $e_{single}(a, ss, sa)$ for aligning amino acid a of the target sequence onto a template position with a structural environment defined by secondary structure ss and solvent accessibility sa . $E_{pairwise}$ is the sum of pair-contact potentials $e_{pair}(a_i, a_j)$ between amino acids a_i and a_j of the target

sequence when they are aligned to template positions that are spatially close. There are two pairwise potentials: distance-dependent and distance-independent. For distance-independent pairwise potential, the cutoff distance is set to 7Å between the C_β atoms of a_i and a_j . E_{ss} is the secondary structure prediction score. In addition, a new method to optimally utilize the profile information was introduced to the new version. We found that our new energy scheme that utilizes the evolutionary information embedded in the profile generated by PSI-BLAST greatly increases the performance of PROSPECT in both aspects of the alignment accuracy and fold recognition. E_{gap} is the sum of the penalties $e_{gap}(g) = 10.8 + 0.6 * (g - 1)$ for an alignment gap of length g [Fitch and Smith, 1983, Gonnet *et al.*, 1992]. All statistics for estimating these terms are collected from the FSSP database (released in March 1998) [Holm and Sander, 1996]. More detailed description on the energy functions will be given in the later section.

The alignment method of PROSPECT consists of two algorithms: dynamic programming algorithm and divide-and-conquer algorithm. For dynamic programming algorithm, we employed the local, global, and global-local alignment schemes. The global alignment is default for the FSSP chain library, and the global-local alignment, where start and end gaps for a query sequence are not penalized, is set default for the domain library. If the pairwise interaction terms are included in alignment, the divide-and-conquer algorithm is used [Xu and Xu, 2000]. The algorithm solves the alignment problem by recursively solving a series of sub-alignment problems between sub-structures and sub-sequences, and then combining these sub-alignments in a consistent and optimal way. For more details, see Ref. [Xu and Xu, 2000]. Finally, the confidence level of the prediction is estimated by the SVM.

3 Energy Functions

In PROSPECT, there are three knowledge-based energy functions: mutation, singleton, and pairwise energies. In addition, affine gap penalty with the open gap penalty and the gap elongation penalty, and the predicted secondary structure information are used.

The mutation energy $E_{mutation}$ describes the compatibility between the template amino acid type, a_t , and the query protein amino acid type, a_q . In old PROSPECT, we used the PAM250 matrix [Gonnet *et al.*, 1992], which has been shown to be one of the best substitution matrix for the distant homology recognition [Fischer *et al.*, 1996, Abagyan and Batalov, 1997],

$$E_{mutation} = - \sum_{(t,q)} M(a_t, a_q), \quad (2)$$

where (t, q) is the aligned amino acid pair of the template and query sequences, and $M(a_t, a_q)$ the PAM250 matrix (the minus sign is to convert score to energy and to follow the convention that the lower energy is more preferable). In present version of PROSPECT, the matrix $M(a_t, a_q)$ is replaced by the profile-profile alignment score, which will be described in the next section.

The singleton energy $E_{singleton}$ is the measure of the query protein's preference to a certain structural environment characterized by the secondary structure (ss) and the solvent accessibility (sa) [Bowie *et al.*, 1991]. It is the sum of the singleton energy parameters, $e_s(a_i, ss_i, sa_i)$, over the all query amino acids aligned to the template sequence,

$$E_{singleton} = \sum_i e_s(a_i, ss_i, sa_i), \quad (3)$$

where the sum is taken over all the aligned amino acids of query protein, and ss_i and sa_i the secondary structure and the solvent accessibility of the template amino acid aligned to the query amino acid type, aa_i , respectively.

In this version of PROSPECT, the singleton energy is different from the old PROSPECT in two aspects. First, instead of using simple summation of parameters as in Eq. (3), we use homologue-averaged singleton energy given by

$$E_{singleton}^{ha} = \sum_i e_s^i(ss_i, sa_i). \quad (4)$$

The detailed description is given in the next section. Second, in deriving the singleton energy parameters $e_s(a_i, ss_i, sa_i)$ from the protein structure library, we employ a slightly different form,

$$e_s(a_i, ss_i, sa_i) = -\log \frac{p(a_i, ss_i, sa_i)}{p^o(a_i, ss_i, sa_i)} \quad (5)$$

$$p^o(a_i, ss_i, sa_i) = p(a_i)p(ss_i, sa_i), \quad (6)$$

where $p(a_i, ss_i, sa_i)$ is the probability of finding an amino acid type a_i at the structural environment characterized by the secondary structure type ss_i and the solvent accessibility type sa_i , and $p^o(a_i, ss_i, sa_i)$ is the similar probability under the assumption that an amino acid a_i has no structural preference. We used nine structural categories with three secondary structure states (α helix, β sheet, loop) and three solvent accessibility states (buried, intermediate, exposed). Following the widely-used convention, H (α -helix), G (3_{10} -helix), and I (π -helix) are classified as α helix, and E (extended strand) and B (residue in isolated β -bridge) states are classified as β sheet. All the other states are considered as loop. The boundary between different solvent accessibility levels were decided such that the number of residues in the database are equally distributed in each level. The results are $sa_i \leq 7\%$ for the buried state, $7\% \leq sa_i \leq 37\%$ for the intermediate state, and $sa_i \geq 37\%$ for the exposed state. It should be noted that the probability ratio $p(a_i, ss_i, sa_i)/p(a_i)p(ss_i, sa_i)$ can be rewritten as $p(a_i|ss_i, sa_i)/p(a_i)$, where $p(a_i|ss_i, sa_i)$ is the conditional probability of finding an amino acid type a_i at the site with the secondary structure type ss_i and the solvent accessibility type sa_i . In old PROSPECT [Xu and Xu, 2000], $p^o(a_i, ss_i, sa_i)$ is given by $p(a_i)p(ss_i)p(sa_i)$. This form ignores the correlation between the secondary structure and the solvent accessibility. For example, a loop region tends to have a high solvent accessibility. Although the old form has no adverse effect on the alignment accuracy, one consequence is that the old form of the singleton energy terms gives the biased scores depending on the spatial arrangement of the secondary structure units of the templates. As we can see in Figure 1, where the ss_i and sa_i dependent expectation values of the singleton energy given by $\sum_{i=1}^{20} p(a_i)e_s(a_i, ss_i, sa_i)$ are plotted, a template with an exposed beta sheet or buried loop region will always have higher singleton energy regardless of a query sequence, while a template with an buried beta sheet or a exposed loop will always have greater chance to be recognized as a correct template regardless of a query sequence. On the other hand, the new singleton energy terms have more or less constant expectation values. The probabilities, $p(a_i, ss_i, sa_i)$, $p(a_i)$, and $p(ss_i, sa_i)$, were obtained from a non-redundant set of known protein structures. We used FSSP database [Holm and Sander, 1996] for this purpose. Among 2689 protein structures in the database, we only used 2145 monomeric proteins that exclude the protein structures that were obtained by NMR experiments.

The pairwise energy $E_{pairwise}$ is used to describe the mutual preference between spatially close amino acids,

$$E_{pairwise} = \sum_{(i,j)} e_p(a_i, a_j; r), \quad (7)$$

where the sum is taken over the pairs of residues, (i, j) , in the region of the core secondary structures. Only the pairs that are separated by at least 3 amino acids are considered. The distance r between residues is estimated by the the distance between C_β atoms. Unlike the previous version of PROSECT where a single cutoff distance $r_c = 7\text{\AA}$ is used, current pairwise energy is distance-dependent. The distance is divided into 4 intervals, 5, 7, 9, and 11 \AA , and any pair with $r > 11\text{\AA}$ is ignored. In Figure 2, several pairs of amino acids are plotted. As expected, as the distance gets larger, the pairwise potentials converge to zero regardless of amino acid type. The pairwise energy parameters, $e_p(i, j)$, is the log ratio of two probabilities,

$$e_p(i, j) = -k_B T \log \frac{p(i, j; r_c)}{p^o(i, j; r_c)}, \quad (8)$$

where $p(i, j; r_c)$ is the probability to find a pair of amino acids, i and j , within the cutoff distance, r_c , and $p^o(i, j; r_c)$ is the similar probability but under the assumption that two amino acids, i and j , are mutually independent.

Given the protein structure database and the cutoff distance, deriving $p(i, j; r_c)$ is straintforward; by counting the number of amino acid pair i and j within the cutoff distance, $M(i, j; r_c)$, and the total number of pairs, M_{total} , in the database. Then the probability is simply given by the ratio, $M(i, j; r_c)/M_{total}$. However, to estimate the ‘‘background probability’’ $p^o(i, j; r_c)$, care should be taken. Let n_i, n_j , and N be the number of amino acid type i, j , and the total number of amino acids in the database, respectively. Then, the number of pairs of i and j is $n_i n_j$ if $i \neq j$, $n_i n_j / 2$ if $i = j$, and the total number of pairs $N(N - 1) / 2 \simeq N^2 / 2$, therefore, the probability $p(i, j) = 2p(i)p(j)$ if $i \neq j$, $p(i)p(j)$ if $i = j$, where $p(i) = n_i / N$. Assuming that the residues are uniformly distributed in a protein, $p^o(i, j; r_c)$ is given by

$$p^o(i, j; r_c) = \begin{cases} 2p(i)p(j)M(r_c)/M_{total}, & \text{if } i \neq j; \\ p(i)p(j)M(r_c)/M_{total}, & \text{if } i = j, \end{cases} \quad (9)$$

where $M(r_c)$ is the total number of pairs *within* the cutoff distance. The same FSSP database that we used for the singleton energy parameter was also used for estimating the pairwise energy parameters.

As some previous works suggested [Lu and Skolnick, 2001], the distance-dependent pairwise potential is better in recognizing a correct fold than the distance-independent one. However, it is somewhat unclear how much improvement we can get from the distance-depedent pairwise potetial. In Figure 3, the pairwise energies estimated by the distance-dependent and distance-independent with cutoff distance of 7 \AA are shown for the alignments between a query sequence 1cpc and FSSP templates. The correlation between two data seem reasonably good. This finding is consistent the fact that the distance-independent pairwise potential with 7 \AA cutoff describe the pairwise interaction reasonably well [Xu and Xu, 2000]. Nonetheless, we use the distance-dependent pairwise potential because a recent work on the accuracy of statistical potentials for fold assessment found that the performance of the distance-independent potentials depend on the size of proteins [Melo *et al.*, 2002].

It is known that the secondary structures are often well conserved among the homologs. The secondary structure information predicted by the program PHD is compared with the secondary

structures of the aligned portion of the template, and their compatibility is measured by the scoring scheme that gives a reward or penalty depending the predicted probability of being a particular secondary structure type.

4 Utilizing Evolutionary Information

It is generally accepted that homology detection can be improved by utilizing multiple sequence alignment information [Gribskov *et al.*, 1987, Henikoff and Henikoff, 1997, Karplus *et al.*, 1999, Panchenko *et al.*, 2000, Rychlewski *et al.*, 2000, Yona and Levitt, 2002]. It can provide wealthy information about structural and functional relationships within a group of related proteins, such as a evolutionary conservative region or conserved hydrophobicity patterns. Those information can be represented by a variety of formulations, such as motifs, profiles [Gribskov *et al.*, 1987], position-specific score matrices (PSSM) [Altschul *et al.*, 1997], and hidden Markov models [Sjlander *et al.*, 1996]. Regardless of the formulation, the essential idea is that it should represent a protein family, rather than a protein itself, expressed in the form of position-dependent scores. If a certain amino acid is highly conserved at a particular position, that amino acid is assigned a high positive score, others are assigned high negatives. On the other hand, for a weakly conserved positions, near zero value is assigned to all the amino acid types.

One of the most popular algorithms is the Position-Specific Iterated BLAST (PSI-BLAST). The PSI-BLAST compares a query sequence against protein database using the gapped BLAST program to construct a multiple sequence alignment, and then generates a PSSM. The PSSM is then used to identify additional similar sequences, and these are again used to update the PSSM. This process is iterated user-specified number of times or until it converges. There are several issues associated with PSI-BLAST: (1) how to construct multiple sequence alignments, (2) how to convert the multiple sequence alignments to the position-specific scores, and (3) how to search a database using the scores. Among those, relevant to the present work is how to convert the multiple alignments to the score matrix.

Given the multiple sequence alignments, PSI-BLAST constructs the $L \times 20$ position-specific score matrix by taking the logarithms of the ratio between the estimated probability for residue j to be found at the position i (p_{ij}) and the probability with which it would be found by chance (p_j , position-independent),

$$s_{ij} = \log \frac{p_{ij}}{p_j}, \quad (10)$$

where $1 \leq i \leq L$ is the residue position, and $1 \leq j \leq 20$ the amino acid type. When deriving p_{ij} , the original PSI-BLAST [Altschul *et al.*, 1997] and a newly improved version [Schäffer *et al.*, 2001] take into account not only the actual observed weighted frequency of amino acid j at position i but also the “pseudocount frequency” which is constructed using the prior information on amino acid mutation propensities implicit in the mutation matrix [Tatusov *et al.*, 1994]. A collection of these probabilities in the form of $L \times 20$ matrix, here named the frequency matrix, is an efficient representation of the protein family to which a query protein belong. The frequency matrices, along with the PSSM, for both query protein and templates are created by running PSI-BLAST to be used to implement the idea of profile-profile alignment and the threading potential energies averaged over homologues.

A straightforward way to utilize the evolutionary information is to calculate the mutation energy in our scoring scheme as the same way as in sequence-profile alignment: a simple lookup of the matrix element in the PSSM corresponding to the amino acid at the aligned position in

the sequence. It can be done either by constructing the PSSM for the templates and aligning them against a query sequence or vice versa. Another way, as demonstrated in other work [Rychlewski *et al.*, 2000], is to compare two protein families by constructing the PSSM’s for both templates and query proteins and performing profile-profile alignment. In their work [Rychlewski *et al.*, 2000], Rychlewski and coworkers defined the score for aligning two profile positions as the dot product between the profile vectors corresponding to those positions.

In present work, we define the score as the dot product between the profile vector (column vector of PSSM corresponding to the aligned position) of the template and the frequency vector (column vector of the frequency matrix corresponding to the aligned position) of a query protein,

$$m_{ij} = \sum_{k=1}^{20} s_{ik} p_{jk}, \quad (11)$$

where m_{ij} is the mutation score for the alignment between the position i of the template protein and the position j of a query protein, s_{ik} the PSSM of a template, and p_{jk} the frequency matrix of a query protein. It should be understood that Eq.(11) is a natural extension of the score for the sequence-profile alignment. As stated earlier, the frequency vector contains the weighted frequencies of occurrence of 20 amino acids at a particular position among the protein family members. Therefore, m_{ij} in Eq.(11) is the averaged score over 20 amino acids weighted by their probability of occurrence among the protein family members. In other words, the profile-profile alignment score is the mutation score averaged over homologs.

The same idea of “average over homologs” can be applied to the singleton and pairwise energies. Recent theory [Finkelstein, 1998] and computational studies [Reva *et al.*, 1999, Cui and Wong, 2000] suggest that the energy averaged over a set of homologous sequences can improve protein fold recognition. The reason for this improvement is that the potential energy parameters are inevitably noisy, and the fact that the homologs share a common fold implies that for a correct fold, each homolog tends to have a similar (reasonably good) energy, and for an incorrect fold, the energies for homologs tend to be random. Therefore, the sum of the energies tend to be canceled out for an incorrect fold, while they are “constructively interfered” for a correct fold. One of the strengths of Eq.(11) is that it removes the need to run many threadings for arbitrarily many homologous sequences, therefore computationally efficient. Moreover, it is superior in quality because the frequency matrix is constructed in such a way that it optimally represents the protein family by not only counting the actual frequencies in multiple sequence alignments but also considering the prior information on amino acid mutation propensities.

The singleton energy averaged over homologs is given by

$$\begin{aligned} E_{singleton}^{avg} &= \sum_i \sum_{j=1}^{20} p_{ij} e_s(j, ss_i, sa_i) \\ &= \sum_i e_s^i(ss_i, sa_i), \end{aligned} \quad (12)$$

where the summation index i includes all the aligned sequences of a query protein, and ss_i and sa_i are the secondary structure and the solvent accessibility of the template sequence aligned to the query sequence position i , respectively. If the *position-dependent* singleton energy parameters $e_s^i(ss_i, sa_i) = \sum_{j=1}^{20} p_{ij} e_s(j, ss_i, sa_i)$ are pre-calculated, no additional computational cost is introduced in this formulation. The expression for the pairwise energy averaged over

homologs can be derived in a similar way,

$$\begin{aligned}
 E_{pairwise}^{avg} &= \sum_{(i,j)} \sum_{k,l=1}^{20} p_{ik} p_{jl} e_p(k,l) \\
 &= \sum_{(i,j)} e_p^{ij}.
 \end{aligned}
 \tag{13}$$

The *position-dependent* pairwise energy parameters $e_p^{ij} = \sum_{k,l=1}^{20} p_{ik} p_{jl} e_p(k,l)$ are the pairwise energies for the pair of query sequences (i, j) when they are aligned to the template sequences that are closer than cutoff distance. These parameters can be also pre-calculated. As discussed in the later Section, these homologs-averaged potential energies not only improve the fold recognition but also greatly increase the alignment accuracy.

5 Support Vector Machine

The support vector machine (SVM) [Vapnik, 1998, Burges, 1998] a powerful method for the general pattern recognition problems founded on the well-developed statistical learning theory [Vapnik, 1995, Vapnik, 1998]. Recently, there have been growing interest and applications to the computational biology area, such as micorarray data analysis [Furey *et al.*, 200], translation initiation sites recognition [Zien *et al.*, 2000], protein secondary structure prediction [Hua and Sun, 2001], and fold recognition [Ding and Dubchak, 2001]. In most cases, the performance of SVM is either comparable or significantly superior to the conventional machine learning methods such as the neural networks (NN). The SVM has many attractive features when it is compared to NN. For example, it is less prone to the overtraining, and has an effective way to avoid the the overtraining. It also can hand large data set efficiently. The biggest advantage from a practical point of view is that unlike NN, it is guaranteed to find the global optimum when we train the SVM. Another advantage over NN is that it is fairly easy to implement SVM in the real applications, compared to NN where it often times requires skill of an experienced expert.

The SVM draws a hyperplane in a multidimensional feature space to separate a set of binary labeled (positive and negative) training data . The hyperplane is chosen to be maximally distant from the closest positive and negative data. When it is not possible to find a hyperplane to separate positive and negative data, one can use the techniques of “kernels” to map the given data to a higher dimensional feature space. Suppose we try to estimate a function $f : R^N \rightarrow \{\pm 1\}$ using a given set of N -dimensional training data, $(\mathbf{x}_1, \dots, \mathbf{x}_l)$, with class labels $y_i = \{\pm 1\}$, such that f will correctly classify a new example (\mathbf{x}, y) which was generated from the same probability distribution $P(\mathbf{x}, y)$ as the training data. If we put no restriction on the function f , even a function that satisfies $f(\mathbf{x}_i) = y_i$ for all i 's do not generalize well to new examples; simply minimizing the training error does not imply good learning. Statistical learning theory [Vapnik, 1995, Vapnik, 1998] shows that it is crucial to restrict the class of functions. The SVM is based on the class of decision functions given by $f(\mathbf{x}) = \text{sign}((\mathbf{w} \cdot \mathbf{x}) + b)$. It can be shown by statistical learning theory that by finding the optimal hyperplane $(\mathbf{w} \cdot \mathbf{x}) + b$, defined as the one with the maximal margin of separation between the two classes, we can maximize the prediction capability of a learning machine. This hyperplane can be obtained by solving a constrained quadratic optimization problem whose solution \mathbf{w} has an expansion $\mathbf{w} = \sum_i c_i \mathbf{x}_i$ in terms of a subset of training data that lie on the margin (support vecotrs).

The property that the final decision functions $f(\mathbf{x}) = \text{sign}(\sum_i c_i(\mathbf{x} \cdot \mathbf{x}_i) + b)$ depend only on the dot product $\mathbf{x}_i \cdot \mathbf{x}_j$ allows us to map the data to some other dot product space (feature space) F via a nonlinear transformation using the *kernels*, $k(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{y})$, such as

$$k(\mathbf{x}, \mathbf{y}) = \begin{cases} (\mathbf{x} \cdot \mathbf{y} + a)^d; & \text{polynomial,} \\ \exp(-\|\mathbf{x} - \mathbf{y}\|^2 / 2\sigma^2); & \text{radial basis function (RBF),} \\ \tanh(\kappa(\mathbf{x} \cdot \mathbf{y}) + \theta); & \text{sigmoid,} \end{cases} \quad (14)$$

and perform the same linear algorithm in F . The decision is then made based on the outcome of the nonlinear decision function,

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^l c_i k(\mathbf{x} \cdot \mathbf{x}_i) + b\right). \quad (15)$$

6 Optimization and Performance

Choosing an optimal set of weighting factors for the energy terms, including the gap opening and gap extension penalties, is crucial to the performance of PROSPECT. The performance of PROSPECT is measured in two aspects: alignment accuracy and fold recognition. It is likely that the optimal parameters for the alignment accuracy are not necessarily optimal for the fold recognition. Our strategy is that we optimize the parameters only for the alignment accuracy. The optimality for the fold recognition is achieved by training SVM. We used the same training set that had been used in our previous work [Xu and Xu, 2000], consisting of 174 structurally-aligned protein pairs that have less than 30% sequence identity between them. The alignment accuracy was measured by comparing the alignments with the structural alignments generated by SARF [Alexandrov, 1996]. Shifts from the exact alignments within 4 residues are counted as correct alignments. To check if the optimized parameters were overtrained for the training set, the alignment accuracy was calculated for the test set consisting of 137 protein pairs, the same set used in our previous work. Second, to further evaluate the alignment accuracy, benchmark set prepared by Sippl’s group (prosup set) [Domingues *et al.*, 2000] was tested. Third, from the same training and test sets, we created the query sequence set for the fold recognition, containing 236 sequences. Then, we performed all-against-all threadings. Among $236 \times 235 = 55460$ query-template pairs excluding self-threadings, all pairs that share the same SCOP superfamily number are classified as a true pair. Otherwise, they are classified as a false pair. Half of total 55460 pairs were randomly selected and used for training the SVM, and the other half were used for testing the trained SVM. To evaluate the performance of new PROSPECT, we test our method on benchmark sets created by Fischer *et al.* (Fischer set) [Fischer *et al.*, 1996], and Lindahl and Elofsson (Lindahl set) [Lindahl and Elofsson, 2000].

6.1 Alignment Accuracy

Before searching for the optimum combination of the weighting factors, we examined the accuracy of each energy term and its relative importance in determining the alignment accuracy. We calculated the alignment accuracies for both the training set and the prosup set after the alignments were calculated only using a single energy term (mutation, singleton, pairwise). The gap penalty and baseline level were optimized for the training set. The results are summarized in Table 1.

method/set	mutation	singleton	pairwise
profile/train	75.0%	67.7%	54.1%
no profile/train	63.7%	57.8%	42.3%
profile/prosup	72.6 %	65.1%	49.2%
no profile/prosup	55.2%	56.7%	38.4%

Table 1: Alignment accuracy on the training set by using only a single energy term. For mutation, singleton, dynamic programming algorithm is used to find the global alignment between a query sequence and the templates. For pairwise energy, divide-and-conquer algorithm is used.

As evidently seen in Table 1, the profile information greatly increases the alignment accuracy in each case. Especially, with profile-profile mutation energy alone, we have the alignment accuracy as high as 75% for the training set, roughly 11% higher than the 63.7% alignment accuracy achieved by the conventional the sequence-sequence alignment with PAM250 matrix. This result is consistent with other works [Jones, 1999, Jaroszewski *et al.*, 2000] that the alignment methods such as sequence-profile or profile-profile alignment algorithm that utilize the profile information, but not the structural information, can generate reasonably good alignments among the remotely related proteins. Similarly high alignment accuracies were obtained for the prosup set. The reason that the alignment accuracy is higher for the training set is not only that the gap penalty and baseline level parameters were optimized for the training set, but rather that the pairs of proteins of the training set were prepared from the proteins with the at least “superfamily” level similarity [Xu and Xu, 2000]. Nonetheless, the alignment accuracy for the prosup set is comparable to that for the training set when the profile information was used. The results also suggest that the most important factor in determining the alignment accuracy is the profile-profile alignment accuracy. It is also interesting to note that the singleton energy term with profile information can generate reasonably good alignment of 67.7% for the training set and 65.1% for the prosup set. The pairwise term without profile information is least effective in alignment accuracy. However, the pairwise term with profile increases the alignment accuracy.

The alignment experiments with single energy term suggest several points. First, we have learned that including the profile information greatly increases the alignment accuracy, roughly 10% increase in all the cases. Second, the profile-profile alignment energy is the most important factor, therefore the quality of the profile is crucial for the alignment accuracy. Third, the pairwise energy term is least helpful. These findings are consistent with the previous work [Xu and Xu, 2000] in that although the pairwise energy increases the correct fold recognition rate, the effect on the alignment accuracy is rather small.

The fact that the pairwise energy term is least helpful in determining alignment accuracy suggests an efficient threading strategy; the pairwise energy can be ignored during we search for the optimal alignment and it is evaluated based on the optimal alignment after the optimal alignment is found. We have searched for the optimal combination of weighting factors with and without the pairwise energy term systemetically using a combination of exhaustive search and the simulated annealing strategy. The best alignment accuracies we found for both cases are nearly identical at 79%. Although divide-and-conquer algorithm is computationally efficient, overall computational speed for the alignment without the pairwise energy is more than one order of magnitude faster than the alignment with the pairwise energy. Based on this observation, we decided to adopt the conventional dynamic programming algorithm without the pairwise

set	Exact	4 residue shifts
training	61.3%	80.6%
test	65.5%	81.1%
prosup	57.7%	75.8%

Table 2: Alignment accuracy on the training, test, and prosup sets with the optimal parameters.

interaction in order to get the optimal alignment. After the optimal alignment is found, the pairwise interaction energy is evaluated using distance-dependent pairwise energy parameters. The results are shown in Table (2). Similar performances on both training set and test set indicate that the optimal parameters are not over-trained. The result on prosup set is slightly worse, which indicates that prosup set is more difficult than our training and test sets. However, the exact match score, 57.7%, is roughly 10% higher than the best score (48.0%) reported in the original paper [Domingues *et al.*, 2000].

6.2 Fold Recognition

A good feature selection is crucial to maximize the SVM performance. We have tested a variety of combinations of the features. The final feature vectors are 10-dimensional. They consist of (1) query sequence size, (2) template sequence size, (3) threading score, (4) mutation energy, (5) singleton energy, (6) gap penalty, (7) pairwise energy, (8) secondary structure score, (9) the number of identical residues among aligned sequences, and (10) alignment length. We have tried several other features. One Example is the energy statistics of each template collected from the threadings against entire templates in the library in order to compensate a certain template’s tendency to produce higher (or lower) energy compared to other templates regardless of a query sequence. The other example is the energy statistics of a query sequence collected in the same way as an attempt to reduce the “composition effect” [Bryant and Altschul, 1995]. Although these features clearly increase the SVM performance, more systematic study need to be done. We used the program SVM^{light} [Joachims, 1999] to train SVM.

One practical issue in training SVM is the unbalance between the numbers of true and false pairs. This problem is handled by using “cost factor” [Joachims, 1999], which is an adjustable parameter that controls the relative weights between training errors on true pairs and false pairs. The kernels we have trained are the linear, the polynomial and the radial basis function (RBF) models. The RBF models turn out be most effective. The results for the RBF models are shown in Table (3). As it can be seen in Table (3), indicated by low training error and poor performance on Fischer benchmark, there is danger for the over-training with RBF model especially when the width scale parameter $g = 1/2\sigma^2$ is large. It is also noticeable that training error and the performance are not sensitively dependent on the choice of parameters, as long as we avoid the over-training situation. Based on these observations, the RBF model with $j = 5$ and $g = 0.005$ is chosen as our SVM model.

The fold recognition performance of PROSPECT was tested on two benchmark sets; Fischer [Fischer *et al.*, 1996] and Lindahl bechmark set[Lindahl and Elofsson, 2000]. The Fischer benchmark set consists of 68 structurally related protein pairs with low sequence similarity and 300 templates. In order to access the test, we used the criteria by Fischer, which is if no incorrect folds have a better score than the expected match, it is considered correct. Then, PROSPECT

Kernel	Parameters (j, g)	Training Error	Test Error	Fischer (Top 1)
RBF	1, 0.1	1.2%	1.1%	43
RBF	5, 0.1	1.0%	1.2%	36
RBF	10, 0.1	1.5%	1.8%	35
RBF	1, 0.05	1.2%	1.2%	47
RBF	5, 0.05	1.1%	1.2%	45
RBF	10, 0.05	1.6%	1.8%	47
RBF	1, 0.005	1.3%	1.3%	54
RBF	5, 0.005	1.2%	1.2%	55
RBF	10, 0.005	1.7%	1.7%	54
RBF	1, 0.0001	1.4%	1.4%	48
RBF	5, 0.0001	1.3%	1.2%	51
RBF	10, 0.0001	1.8%	1.9%	46

Table 3: The performance of various SVM models. The parameters, j, g , are the cost factor and RBF width scale parameter, $g = 1/2\sigma^2$, respectively. “Top 1” refers to the number of query sequences among 68 Fischer benchmark pairs that recognize the correct templates at top position.

can correctly recognize 55 pairs out of 68 pairs. The Lindahl set consists of 976 protein sequences. The performance on all against all comparison of these 976 sequences is measured in three different similarity levels: family, superfamily and fold. The results are summarized in Table (4). It can be seen that the performances of PROSPECT on all similarity levels are significantly better than or comparable to any other method. On family level, all methods except THREADER perform well, and the performance of PROSPECT is slightly worse than that of the best-performing method, FUGUE. On superfamily and fold levels, PROSPECT performs better than any other method. It is noticeable that although the focus of new PROSPECT is on detecting family/superfamily level similarity it also performs relatively well in detecting fold level similarity, better than THREADER. The unique feature of PROSPECT is that unlike other methods it performs well on all similarity levels; for example, FUGUE performs best at family/superfamily levels and THREADER performs best at fold level.

One of the most important requirements for the protein structure prediction method is the ability to tell the reliability of a prediction. It is even more critical for the genome-scale applications. The reliability assessment is based on the SVM output. If we make a binary classification based on the SVM output, all the templates with the SVM output greater than zero should be assigned a correct template. However, we found that this simple scheme produces too many false positives and false negatives. In Figure 4, we plot two characteristics of the SVM predictor on test set. The solid line represents the probability of being a true pair given an SVM score, and the dotted line is the ratio of true pairs being above a given SVM score. At an SVM score of 2 and above, the reliability of our SVM predictor is 100%. At the score cutoff of 1, the confidence level drops to approximately 80%. However, a problematic fact is that at the score cutoff values of 1 or 2 our predictor recognizes only about 30% or 20% of the true pairs. This problem is contrary to the fact that PROSPECT is very good at locating a pair at the top position. The problem stems from the composition effect, a consequence of using energy parameters derived from a particular database with a certain composition [Bryant and Altschul, 1995]. The distri-

%pairs at top 1/top 5 Method	Family only		Superfamily only		Fold only	
	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5
PROSPECT	79.2	87.0	46.3	64.0	20.8	50.0
FUGUE	82.2	85.8	41.9	53.2	12.5	26.8
PSI-BLAST	71.2	72.3	27.4	27.9	4.0	4.7
HMMER-PSIBLAST	67.7	73.5	20.7	31.3	4.4	14.6
SAMT98-PSIBLAST	70.1	75.4	28.3	38.9	3.4	18.7
BLASTLINK	74.6	78.9	29.3	40.6	6.9	16.5
SSEARCH	68.6	75.7	20.7	32.5	5.6	15.6
THREADER	49.2	58.9	10.8	24.7	14.6	37.7

Table 4: The performance of PROSPECT at different similarity levels:family, superfamily and fold. For comparison, the results of FUGUE and other popular methods are shown.

bution of threading energy of a query sequence over templates widely varies depending on its amino acid composition. For example, if a query sequence has more of attracting amino acid types, the pairwise energy tends to be low regardless of templates, therefore producing more of templates with high SVM score.

The same symptom can be seen in Figure (5), where we plot Spec-Sens scan on Lindahl set [Lindahl and Elofsson, 2000]. The specificity is defined by

$$\text{Specificity}(\text{score}) = \frac{\text{TP}(\text{score})}{\text{TP}(\text{score}) + \text{FN}(\text{score})}, \quad (16)$$

where $\text{TP}(\text{score})$ and $\text{FN}(\text{score})$ denote the number of true positives above score and the number of false negatives above score. It measures the probability that a pair with a score greater than a certain cutoff score is a true pair. The sensitivity is defined by

$$\text{Sensitivity}(\text{score}) = \frac{\text{TP}(\text{score})}{\text{TP}(\text{score}) + \text{FP}(\text{score})}, \quad (17)$$

where $\text{FP}(\text{score})$ is the number of false positives with a score less than score. It is the fraction of the number of true positives. In low specificity region, the performance of PROSPECT is comparable to that of FUGUE. However, in high specificity region, the sensitivities are below those of FUGUE and some other methods. Nonetheless, it is interesting to observe that in fold level PROSPECT outperforms all other methods in entire specificity region. Research is undergoing in our lab to improve Spec-Sens performance.

7 Conclusions

In this paper, we describe a new component of our protein structure prediction program PROSPECT that can computationally efficiently and reliably recognize distant homologues by sequence-structure comparison. The key features of our method are (1) an efficient way to utilize the evolutionary information in an optimal way, (2) global sequence-structure alignment by dynamic programming algorithm, (3) the SVM training using the alignment models derived from the alignments to recognize distance homologues, and (4) confidence score for the prediction.

In addition to the new profile-profile alignment method, we introduced the way to utilize the evolutionary information when we calculate threading potentials including singleton and pairwise energies. We found that new energy scheme utilizing the evolutionary information not only greatly increases the alignment accuracy but also improves the fold recognition performance. The alignment accuracy of our method tested on the benchmark set prepared by Sippl [Domingues *et al.*, 2000] is roughly 10% higher than the previously reported best score. The fold recognition test on two benchmark sets by Fischer *et al.* [Fischer *et al.*, 1996], and Lindahl and Elofsson [Lindahl and Elofsson, 2000] indicates that the performance of PROSPECT is superior or comparable to any other fold recognition method at all similarity levels. The unique advantage of PROSPECT is that it performs well at all similarity levels unlike other methods. One drawback is that the confidence score at family/superfamily levels is less sensitive than the leading fold recognition methods, while the confidence score at fold level is better than other methods. We argue that the problem stems from the composition effect, and research is undergoing in our lab to tackle this problem.

An emphasis of new approach is on the computational efficiency. The main focus is to develop a computational method that can be applied to genome-scale applications. The pairwise interaction is ignored when we search for the optimal alignment solely for the computational efficiency reason. Although the success rate of our new method in finding a correct template at fold level approaches 20%, it is still too low. We believe that rigorous treatment of the pairwise interaction may be crucial to improve success rate at fold level similarity. The gap penalty we have adopted is a simple affine gap penalty. It is generally believed that a proper gap penalty should be position-dependent. In this regard, a program developed by Shi *et al.* [Shi *et al.*, 2001] is noteworthy. They determined the gap penalty at each position of a structural template according to its structural characteristics. Although it is nontrivial to find and optimize the position-dependent gap penalty parameters, we believe it will significantly improve the performance of PROSPECT. As mentioned earlier, the alignment model should be refined. Two alignment schemes are implemented in PROSPECT: global and global-local. By default, the global and global-local alignment schemes are used for the chain library and domain library, respectively. It is known that each alignment scheme has its merits and shortcomings. More flexible and elaborated alignment scheme should be developed. Main effort is being devoted to improving the sensitivity of our confidence score, and applying our method to several microbial genomes.

References

- [Abagyan and Batalov, 1997] Abagyan, R. A. and Batalov, S. (1997). Do aligned sequences share the same fold? *J. Mol. Biol.* **273**, 355–368.
- [Alexandrov, 1996] Alexandrov, N. N. (1996). SARFing the PDB. *Protein Eng.* **9**, 727–732.
- [Altschul *et al.*, 1997] Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, **25**, 3389–3402.
- [Bowie *et al.*, 1991] Bowie, J. U., Luthy, R., and Eisenberg, D. (1991). A method to identify protein sequences that fold into a known three-dimensional structure. *Science*, **253**, 164–170.

- [Bryant, 1996] Bryant, S. (1996). Evaluation of threading specificity and accuracy. *Proteins: Struct. Funct. Genet.* **26**, 172–185.
- [Bryant and Altschul, 1995] Bryant, S. H. and Altschul, S. F. (1995). Statistics of sequence-structure threading. *Curr. Opinion Struct. Biol.* **5**, 236–244.
- [Burgess, 1998] Burgess, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, **2**, 121–167.
- [Cui and Wong, 2000] Cui, Y. and Wong, W. H. (2000). Multiple-sequence information provides protection against mis-specified potential energy functions in the lattice model of proteins. *Phys. Rev. Lett.* **85**, 5242–5245.
- [Ding and Dubchak, 2001] Ding, C. H. Q. and Dubchak, I. (2001). Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, **17**, 349–358.
- [Domingues *et al.*, 2000] Domingues, F. S., Lackner, P., Andreeva, A., and Sippl, M. J. (2000). Structure-based evaluation of sequence comparison and fold recognition alignment accuracy. *J. Mol. Biol.* **297**, 1003–1013.
- [Finkelstein, 1998] Finkelstein, A. V. (1998). 3d protein folds: homologs against errors—a simple estimate based on the random energy model. *Phy. Rev. Lett.* **80**, 4823–4825.
- [Fischer *et al.*, 1996] Fischer, D., Elofsson, A., Bowie, J. U., and Eisenberg, D. (1996). Assessing the performance of fold recognition methods by means of a comprehensive benchmark. In: *Biocomputing: Proceedings of the 1996 Pacific Symposium*, (Hunter, L. and Klein, T., eds) pp. 300–318. World Scientific Publishing Co. Singapore.
- [Fitch and Smith, 1983] Fitch, W. M. and Smith, T. F. (1983). Optimal sequence alignments. *Proc. Natl. Acad. Sci. U.S.A.* **80**, 1382–1386.
- [Furey *et al.*, 200] Furey, T. S., Cristianini, N., Duffy, N., Schummer, D. W. B. M., and Haussler, D. (200). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, **16**, 906–914.
- [Godzik *et al.*, 1992] Godzik, A., Skolnick, J., and Kolinski, A. (1992). A topology fingerprint approach to the inverse folding problem. *J. Mol. Biol.* **227**, 227–238.
- [Gonnet *et al.*, 1992] Gonnet, G. H., Cohen, M. A., and Benner, S. A. (1992). Exhaustive matching of the entire protein sequence database. *Science*, **256**, 1443–1445.
- [Gribskov *et al.*, 1987] Gribskov, M., McLachlan, A. D., and Eisenberg, D. (1987). Profile analysis: Detection of distantly related proteins. *Proc. Natl. Acad. Sci. USA*, **84**, 4355–4358.
- [Henikoff and Henikoff, 1997] Henikoff, S. and Henikoff, J. G. (1997). Embedding strategies for effective use of information from multiple sequence alignments. *Protein Sci.* **6**, 698–705.
- [Holm and Sander, 1996] Holm, L. and Sander, C. (1996). Mapping the protein universe. *Science*, **273**, 595–602.

- [Hua and Sun, 2001] Hua, S. and Sun, Z. (2001). A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. *J. Mol. Biol.* **308**, 397–407.
- [Jaroszewski *et al.*, 2000] Jaroszewski, L., Rychlewski, L., and Godzik, A. (2000). Improving the quality of twilight-zone alignments. *Protein Sci.* **9**, 1487–1496.
- [Joachims, 1999] Joachims, T. (1999). Making larger-scale svm learning practical. In: *Advances in Kernel Methods-Support Vector Learning*, (Schölkopf, B., Burges, C., and Smola, A., eds), MIT Press.
- [Jones, 1999] Jones, D. T. (1999). GenTHREADER: an efficient and reliable protein fold recognition method for genomic sequences. *J. Mol. Biol.* **287**, 797–815.
- [Jones *et al.*, 1992] Jones, D. T., Taylor, W. R., and Thornton, J. M. (1992). A new approach to protein fold recognition. *Nature*, **358**, 86–89.
- [Karplus *et al.*, 1999] Karplus, K., Barrett, C., Cline, M., Diekhans, M., Grante, L., and Hughey, R. (1999). Predicting protein structure using only sequence information. *Proteins, Suppl 3*, 121–125.
- [Kelly *et al.*, 2000] Kelly, L. A., MacCallum, R. M., and Sternberg, M. J. E. (2000). Enhanced genome annotation using structural profiles in the program 3d-pssm. *J. Mol. Biol.* **299**, 499–520.
- [Lindahl and Elofsson, 2000] Lindahl, E. and Elofsson, A. (2000). Identification of related proteins on family, superfamily and fold level. *J. Mol. Biol.* **295**, 613–625.
- [Lu and Skolnick, 2001] Lu, H. and Skolnick, J. (2001). A distance-dependent atomic knowledge-based potential for improved protein structure selection. *Proteins*, **44**, 223–232.
- [Melo *et al.*, 2002] Melo, F., Sánchez, R., and Sali, A. (2002). Statistical potentials for fold assessment. *Protein Science*, **11**, 430–448.
- [Panchenko *et al.*, 2000] Panchenko, A. R., Marchler-Bauer, A., and Bryant, S. H. (2000). Profiles based on structure alignments increase the sensitivity of protein threading. In: *Quantitative Challenges in the Post-Genome Sequence Era: a Workshop and Symposium*, (The La Jolla Interfaces in Science, ed) p. 2. La Jolla, CA.
- [Reva *et al.*, 1999] Reva, B. A., Skolnick, J., and Finkelstein, A. V. (1999). Averaging interaction energies over homologs improves protein fold recognition in gapless threading. *Proteins*, **35**, 353–359.
- [Rychlewski *et al.*, 2000] Rychlewski, L., Jaroszewski, L., Li, W., and Godzik, A. (2000). Comparison of sequence profiles. strategies for structural predictions using sequence information. *Protein Sci.* **9**, 232–241.
- [Schäffer *et al.*, 2001] Schäffer, A. A., Aravind, L., Madden, T. L., Shavirin, S., Spouge, J. L., Wolf, Y. I., Koonin, E. V., and Altschul, S. F. (2001). Improving the accuracy of psi-blast protein database searches with composition-based statistics and other refinements. *Nucleic Acids Res.* **29**, 2994–3005.

- [Shi *et al.*, 2001] Shi, J., Blundell, T. L., and Mizuguchi, K. (2001). Fugue: sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties. *J. Mol. Biol.* **310**, 243–257.
- [Sjlander *et al.*, 1996] Sjlander, K., Karplus, K., Brown, M., R, H., Krogh, A., Mian, I., and Haussler, D. (1996). Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology. *Comput. Appl. Biosci.* **12**, 327–345.
- [Tatusov *et al.*, 1994] Tatusov, R. L., Altschul, S. F., and Koonin, E. V. (1994). Detection of conserved segments in proteins: Iterative scanning of sequence databases with alignment blocks. *Proc. Natl. Acad. Sci. USA*, **91**, 12091–12095.
- [Vapnik, 1995] Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. New York: Springer.
- [Vapnik, 1998] Vapnik, V. (1998). *Statistical Learning Theory*. New York: John Wiley and Sons, Inc.
- [Xu and Xu, 2000] Xu, Y. and Xu, D. (2000). Protein threading using PROSPECT: Design and evaluation. *Proteins: Struct. Funct. Genet.* **40**, 343–354.
- [Yona and Levitt, 2002] Yona, G. and Levitt, M. (2002). Within the twilight zone: a sensitive profile-profile comparison tool based on information theory. *J. Mol. Biol.* **315**, 1257–1275.
- [Zien *et al.*, 2000] Zien, A., Rätsch, G., Mika, S., Schölkopf, B., Lengauer, T., and Müller, K. R. (2000). Engineering support vector machine kernels that recognize translation initiation sites. *Bioinformatics*, **16**, 799–807.

Figure Caption

- Figure 1. The expected singleton energy based on the new and old formulation of the singleton energy parameters. The structural variables, H, B, L, B, I, E denote helix, beta sheet, loop, buried, intermediate, exposed, respectively.
- Figure 2. The distance-dependent pairwise energy parameters.
- Figure 3. Comparison between distance-dependent pairwise energy vs. distance-independent pairwise energy with 7 Å cutoff.
- Figure 4. The confidence score of SVM output. The solid line represents the probability of being a true positive at a given SVM score. The dotted line represents the fraction of true positives being above a given SVM score.
- Figure 5. Spec-Sens curves on Lindahl's benchmark set. Specificity represents the probability that a pair with a score greater than a certain cutoff score is a true pair. Sensitivity is the fraction of the number of true positives. (a) Family level only, (b) Superfamily level only, (c) Fold level only.