J Mol Model (2009) 15:1093–1108
DOI 10.1007/s00894-009-0454-9

ORIGINAL PAPER

Solvent accessible surface area approximations for rapid and accurate protein structure prediction

Elizabeth Durham · Brent Dorr · Nils Woetzel · René Staritzbichler · Jens Meiler

Received: 16 November 2008 / Accepted: 2 January 2009 / Published online: 21 February 2009

Abstract The burial of hydrophobic amino acids in the protein core is a driving force in protein folding. The extent to which an amino acid interacts with the solvent and the protein core is naturally proportional to the surface area exposed to these environments. However, an accurate calculation of the solvent-accessible surface area (SASA), a geometric measure of this exposure, is numerically demanding as it is not pair-wise decomposable. Furthermore, it depends on a full-atom representation of the molecule. This manuscript introduces a series of four SASA approximations of increasing computational complexity and accuracy as well as knowledge-based environment free energy potentials based on these SASA approximations. Their ability to distinguish correctly from incorrectly folded protein models is assessed to balance speed and accuracy for protein structure prediction. We find the newly developed "Neighbor Vector" algorithm provides the most optimal balance of accurate yet rapid exposure measures.

Keywords Environment free energy · Protein structure prediction · Solvent accessible surface area

E. Durham · B. Dorr · N. Woetzel · R. Staritzbichler · J. Meiler
Departments of Chemistry, Pharmacology, and Biomedical Informatics and Center for Structural Biology, Vanderbilt University,
465 21st Ave South, BioSci/MRB III, Room 5144B,
Nashville, TN 37232-8725, USA

J. Meiler (✉)
Department of Chemistry, VU Station B # 351822,
Vanderbilt University,
7330 Stevenson Center,
Nashville, TN 37235-1822, USA
e-mail: jens.meiler@vanderbilt.edu
URL: http://www.meilerlab.org/

Abbreviations

ANN	artificial neural network
AUC	Area under the receiver operating characteristic curve
CASP	Critical assessment of structure prediction
KBP	Knowledge-based potential
MSMS	Maximal speed molecular surfaces
NC	Neighbor count
NV	Neighbor vector
OLS	Overlapping spheres
PDB	Protein data bank
ROC	Receiving operating characteristic
RMSD	Root mean square deviation
rSASA	Per-residue solvent-accessible surface area
SASA	Solvent-accessible surface area
VMD	Visual molecular dynamics

Introduction

Computational protein structure prediction gains importance in the post-genomic area

Genome sequencing has provided a wealth of information about the amino acid sequence of proteins. While x-ray crystallography and nuclear magnetic resonance spectroscopy made great progress in elucidating the structure of many of these proteins, these experimental techniques are laborious and are not feasible for use on all proteins [1]. In particular, membrane proteins, which comprise greater than 50% of all drug targets [2], and large protein complexes evade experimental structure elucidation. While up to 35% of all proteins are membrane proteins [3], less than 2% of structures deposited in the PDB belong to this class (as of

02/2008). Therefore, there has been an increased demand for computational methods to predict the structure for such proteins and to assist in structure elucidation from sparse or low-resolution experimental data generated by complementary techniques such as electron paramagnetic resonance spectroscopy [4], x-ray crystallography [5], and cryo-electron microscopy [5].

Protein structure prediction techniques can be categorized into comparative modeling techniques that build a model of the target protein based on the known structure of a related template protein, and *de novo* structure prediction techniques that can be used in the absence of a suitable template structure [6]. Proteins usually fold into the conformation with the lowest free energy, so protein structure prediction is essentially a search amongst all possible conformations of an amino acid sequence for the conformation with the lowest free energy. While both classes of protein structure prediction techniques depend critically on energy functions to evaluate the candidate conformations (also commonly called models), *de novo* structure prediction in particular requires very rapid yet accurate energy evaluation functions in order to search a large conformational space in a short period of time [6]. These energy evaluation functions approximate the energy of a given protein model and thus provide a way to "score" each model. Both comparative modeling and *de novo* structure prediction methods have been evaluated in recent critical assessment of structure prediction (CASP) experiments [7] during which computational methods have repeatedly predicted protein structures *de novo* to within 5 Å $C_{\propto}$ *rmsd* [8].

Knowledge-based energy functions allow accurate and rapid calculation of classical energy terms

Energetic terms, such as hydrogen bonding, electrostatics, and van der Waals forces contribute to the interactions of atoms within a protein as well as between the protein and solvent [9]. While molecular mechanics force-fields seek to individually describe each of these starting from first principles, knowledge-based potentials (KBPs) seek to derive energy functions that describe the net effect of all these contributions in a specific setting, e.g., protein structures [10]. Hence, they approximate the overall free energy more generally, and frequently encompass multiple classical energy terms associated with a physical interaction [11]. KBPs have been shown to be an effective alternative to using atomic solvation parameters to more precisely model the folding process [12].

KBPs relate the probability of a conformation to the energy associated with that conformation using an inverse Boltzmann relation [13]:

$$\Delta G = -RT \ln \mathrm{P}$$

which provides a means for the derivation of a free energy from a propensity. Advantages of knowledge-based potentials include the comprehensive and unbiased inclusion of all experimentally elucidated protein structures. Disadvantages are the requirement of a vast knowledge-base [11], potential biases in the knowledge-base that translate into the potentials [11], and difficulty aligning components of the knowledge-based energy contributions with classical energy terms [11]. Nevertheless, the widespread use of knowledge-based free energy potentials in predicting protein structure [14–18], protein-protein interactions [19, 20], protein-ligand interactions [21–24], and in protein design [25, 26] underlines their success in recent years. Knowledge-based energy terms have been derived for all levels of protein architecture, most notably atoms [15, 27], amino acids [18], secondary structure elements [28], and the overall protein fold [29]. Often, several of these knowledge-based free energy approximations are linearly combined into a single composite energy function without addressing the certain overlap between the individual terms that results from the description of the same classical terms, mostly on different levels of architecture.

Amino acid environment energy depends on an accurate yet rapid estimation of solvent accessible surface area (SASA)

The amino acid "environment free energy" [30, 31] encompasses amino acid interactions with the solvent (solvation) as well as with the protein core and integrates hydrogen bonding, electrostatics, and van der Waals forces among others. It is an important driving force in protein folding as it maps to effects like surface area minimization, burial of hydrophobic side chains, and side-chain packing density [30].

The extent to which an amino acid interacts with its environment, the solvent and the protein core, is naturally proportional to the degree to which it is exposed to these environments [32]. The solvent-accessible surface area (SASA) is a geometric measure of this exposure, and therefore a dependency exists between SASA and environment free energy [33, 34]; some approaches even assume a strictly linear relation between the two values [32, 35]. An explicit calculation of the SASA is computationally intractable as this value is, by nature, not pair-wise decomposable [36]. Hence an accurate but pair-wise decomposable approximation of SASA is often used in conjunction with KBPs to describe environment free energy [18].

A precise calculation of solvent accessible surface area is numerically demanding and not practical for computational protein structure prediction

SASA is typically calculated by methods involving the *in-silico* rolling of a spherical probe, which approximates a

water molecule, around a full-atom protein model. Lee and Richards presented the first algorithm for calculating the solvent-accessible surface area (SASA) of a molecular surface [37]. Their method involved the extension of the van der Waals radius for each atom by 1.4 Å (the radius of a polar solvent probe) and the calculation of the surface area of these expanded-radius atoms. The Shrake and Rupley algorithm [38] involves the testing of points on an atom's van der Waals surface for overlap with points on the van der Waals surface of neighboring atoms. Many SASA approximations have been developed including spline approximations [39] and approximations that take advantage of boolean logic and look-up tables [40]. Wodak and Janin's statistical SASA approximation algorithm is a function of only interatomic distances that approximates each amino acid by one sphere at the center of mass [41]. Many approaches employ a lattice surrounding the protein to approximate its SASA [42–44].

A pairwise-decomposable method of SASA approximation is desirable as it can then be employed in minimization approaches, such as dead end elimination. One SASA approximation that achieves this criteria is the method of Street and Mayo in which a scaled two-body approximation of the buried area is subtracted from the total surface area in order to approximate SASA [36]. The method of Zhang et al. improved upon the Street and Mayo method by accounting for its shortcoming, the overlapping burial of core residues. Areas were calculated in the presence of generic side chains rather than the backbone alone, which reduced the error of the area calculations [45]. One of the more efficient non-pairwise-decomposable algorithms is the maximal speed molecular surfaces (MSMS) algorithm which fits spherical and toroidal patches onto the surfaces of atoms based on which points on the atom are accessible to a spherical probe that approximates a solvent molecule [46].

Several approximations for burial are based upon "neighborhood densities [47]," a weighted sum of neighboring atoms, which take advantage of the idea that neighborhood density is inversely related to SASA. The method used to approximate burial in an early version of Rosetta, a state-of-the-art protein structure prediction algorithm, uses the number of C_β atoms within 10 Å of the C_β of the amino acid of interest [18]. Since that time, this has been modified slightly so that centroids, pseudo-atoms located at the side chain's center of mass, rather than C_βs are used [48]. Other work has examined various burial approximations and found that the number of C_β atoms within 14 Å of the C_β of the amino acid of interest is most conserved in structural alignments, most predictable from amino acid sequence, and provides the greatest utility in fold recognition and sequence alignment [49]. A shortcoming of burial approximations is their inability to take into account the spatial orientation of neighboring atoms (illustrated in Fig. 3). A method that calculates burial by examining neighborhood densities in four different tetrahedral directions attempts to address this shortcoming [50]. The "neighbor vector" algorithm introduced in this manuscript attempts to address this shortcoming as well.

As is evidenced by the wealth of related literature, this area has been researched extensively and many SASA approximations have been developed. While many of the discussed

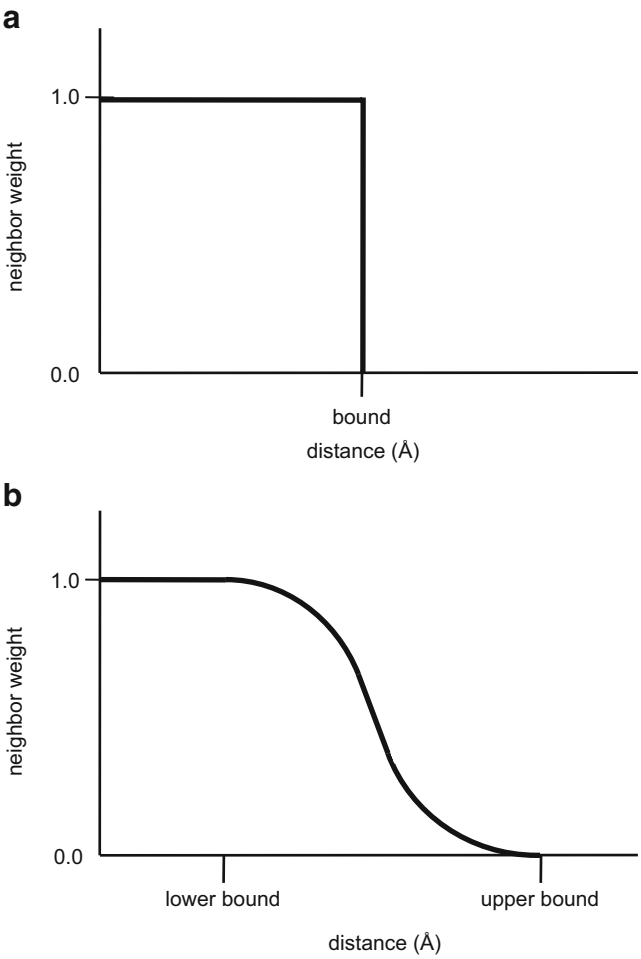


Fig. 1 This figure depicts ways in which a "neighboring" amino acid can be defined. **a**) Previous work uses a step function with a hard boundary to determine which amino acids are neighbors. Any amino acids lying within that boundary are considered neighbors and any amino acids lying outside of that boundary are not considered neighbors. **b**) An expanded definition of neighbor that includes a smooth transition function is used in the neighbor count algorithm. Rather than a single boundary, a lower and upper boundary are designated. Amino acids lying within the lower boundary are considered complete neighbors and are assigned a neighbor weight of 1.0. Amino acids lying outside of the upper boundary are not considered neighbors at all and are assigned a neighbor weight of 0.0. Amino acids lying between the lower and upper bounds are assigned a weight between 0.0 and 1.0 based on their proximity to the amino acid of interest

methods are very accurate, they are also time-consuming and not tractable for use in protein structure prediction, where thousands of protein models need to be evaluated. Additionally, the majority of these methods work on full-atom protein models whereas reduced amino acid representations are often used in early stages of protein structure prediction. Finally, many of these methods return the SASA of the protein model as a whole rather than the SASA of each amino acid (known as rSASA or per-residue SASA), which is necessary in order to take advantage of the knowledge-based potentials.

In this manuscript, the authors seek to build upon several of these approaches and refine them specifically for use in protein structure prediction. While this manuscript focuses on the benefits of a rapid SASA approximation method for protein folding, there are additional areas that would benefit from such a method, such as protein binding and design. Specifically, hydrophobic surface patches, which are important in molecular recognition processes, constitute up to 60% of the SASA of a protein, and methods for their rapid identification based on SASA calculation have been developed [51]. The rSASA calculated by the MSMS algorithm is used as reference standard throughout the present work.

Four SASA approximation algorithms are presented that reflect the trade-off between accuracy and speed

This manuscript systematically introduces and compares a series of rSASA approximations of increasing complexity. KBPs describing the environment free energy of an amino acid in dependence of these SASA approximations have been derived. All approximations are examined in terms of both runtime and the ability to discriminate native-like from nonnative-like protein models obtained in structure prediction applications, in order to fine-tune the balance between algorithm speed and accuracy.

Materials and methods

Exposure algorithms of increasing complexity

Neighbor count (NC) The central idea behind the neighbor count algorithm is that the number of neighboring amino acids is inversely proportional to the exposure of an amino acid. The definition of a "neighbor" is expanded in this work by assigning a weight between 0.0 and 1.0 to all amino acids in the protein model based on their proximity to the amino acid of interest. A lower boundary and an upper boundary are chosen such that all amino acids whose

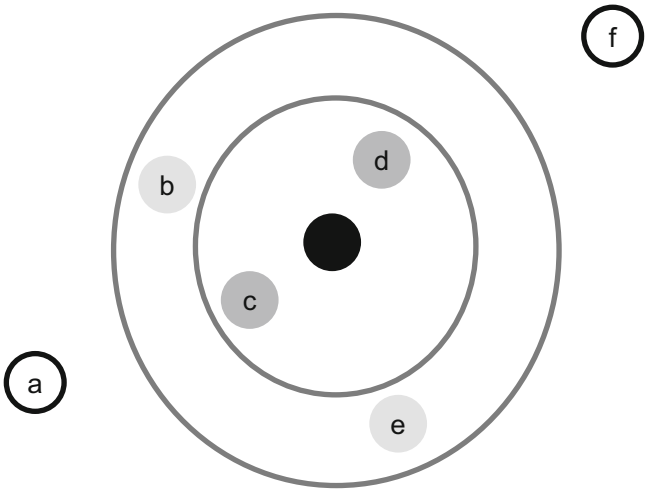


Fig. 2 This figure depicts the neighbor count algorithm. The *inner* and *outer gray rings* represent the lower and upper bounds respectively. The *small circles* represent the C_β atoms of amino acids. The *black circle* represents the amino acid of interest. Amino acids a and f are assigned a neighbor weight of 0.0 because they are outside of the upper bound. Amino acids b and e are assigned a weight between 0.0 and 1.0 because they lie between the upper and lower bounds. Amino acids c and d are counted as one complete neighbor each because they lie within the lower bound

C_β lies at a distance less than or equal to the lower boundary are assigned a neighbor weight of 1.0 (i.e., they are counted as complete neighbors), amino acids whose C_β lies at distance greater than the upper boundary are assigned a neighbor weight of 0.0 (i.e. they are not considered neighbors at all), and amino acids whose C_β lies at a distance between the lower and upper bounds are assigned a weight between 0.0 and 1.0 (see Fig. 1). For glycine, a pseudo- C_β atom is introduced at the geometric position where an actual C_β would sit. This expansion of the definition of "neighbor" allows for amino acids that are spatially close to the amino acid of interest to have a greater weight in determining the neighbor count keeping the potential continuously differentiable at the same time, a characteristic essential for gradient-based minimization.

$$NeighborWeight(distance,\ lower\,bound,\ upper\,bound) = \begin{cases} 1,\ if\ distance \le lower\,bound \\ \frac{1}{2}\left[cos\left(\frac{distance - lower\,bound}{upper\,bound - lower\,bound} \times \pi\right) + 1\right], \\ if\ lower\,bound < distance < upper\,bound \\ 0,\ if\ distance \ge upper\,bound. \end{cases}$$

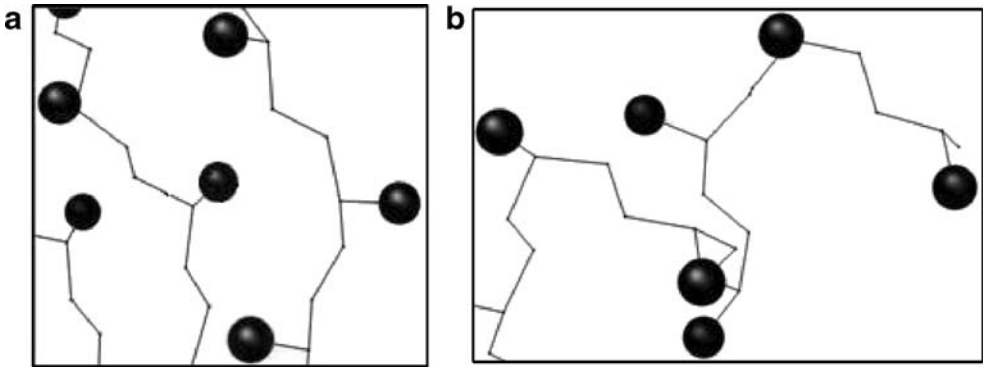


Fig. 3 This figure depicts a shortcoming of the neighbor count algorithm. *Lines* are drawn from the amino acid of interest in this case to all neighboring (as defined by the neighbor count algorithm) amino acids. Two scenarios are shown for which the neighbor count algorithm returns a value of five. However, these two scenarios depict two very different exposure states

The neighbor count value for each amino acid is generated by adding the neighbor weight values of all other amino acids in the protein model as shown in the equation below and Fig. 2.

$$NeighborCount(aa_i) = \sum_{j \neq i} NeighborWeight\left(dist\left(aa_i, aa_j\right),\ lower\ bound,\ upper\ bound\right)$$

A shortcoming of using the number of neighboring amino acids as a measure of burial is that this approach disregards the spatial distribution of its neighbors. Figure 3 shows two examples that represent different exposure scenarios, yet return the same neighbor count value.

Neighbor vector (NV) The neighbor vector algorithm is an extension of the neighbor count algorithm that takes into account the spatial orientation of neighboring amino acids.

$$NeighborVector\left(aa_j\right) = \left\| \frac{\sum_{j \neq i} \left(\overrightarrow{vector_{1j}} \Big/ \left\| \overrightarrow{vector_{1j}} \right\| \right) * NeighborWeight(dist(i,j), lower\ bound,\ upper\ bound)}{NeighborCount(aa_i)} \right\|$$

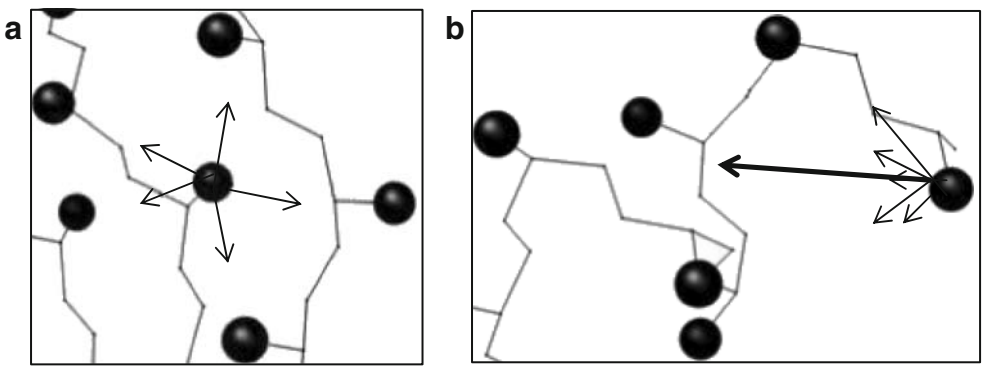


Fig. 4 This figure depicts the neighbor vector algorithm. The vectors drawn to the C_βs of neighboring amino acids are shown in *black* and the vector sum is shown in *heavyweight black*. **a)** When summed, the vectors essentially cancel out yielding a vector of zero length which indicates burial. **b)** When summed, the vectors yield a vector with a large magnitude which indicates exposure

The neighbor vector is a vector associated with each amino acid whose length can range between 0.0 and 1.0. A neighbor vector of length ≅1.0 implies high exposure whereas a neighbor vector of length ≅0.0 implies low exposure (i.e. burial). This is shown graphically in Fig. 4. Note that the neighbor vector is still a pair-wise decomposable measure of exposure.

Artificial neural network (ANN) As input for an ANN that approximates SASA, an additional term not used in previous measures is introduced: the dot product of the $(C_\alpha - C_\beta)$ vector with the neighbor vector (NV $(C_\alpha - C_\beta)$). Recall that the side chain atoms extend from the C_β atom. Therefore, this dot product term provides information about the orientation of the side chain of the amino acid of interest, with respect to neighboring amino acids. If the $(C_\alpha - C_\beta)$ vector points in the same direction as the neighbor vector, the angle between these vectors will be small and the dot product will be ≅+1.0. If the $(C_\alpha - C_\beta)$ vector points in the opposite direction as the neighbor vector, the angle between these vectors will be large and the dot product will be ≅−1.0 (see Fig. 5). Therefore, this dot product provides additional information about the position of the side chain atoms with respect to the neighboring amino acids. The neighbor count, neighbor vector, and $NV \bullet (C_\alpha - C_\beta)$ are input to the ANN.

The ANN contains a single hidden layer with three neurons. The ANN was trained using a feed-forward algorithm with back-propagation over 2670 steps (5000 steps were allowed, but the training terminated early due to convergence). The data was split into a training set (80% of the data), a monitor set (10% of the data), and an independent set (10% of the data). The learning rate η was 0.01 and the momentum α was 0.5.

Overlapping spheres (OLS) The overlapping spheres algorithm is a variant of the Shrake and Rupley [38] algorithm for calculating molecular surfaces with the exception that spheres surround amino acids rather than atoms. In this algorithm, a sphere is placed around each C_β and points are placed on the surface of the sphere surrounding the amino acid of interest. The fraction of points on an amino acid's sphere that do not overlap with any other sphere is used as a measure of exposure (see Fig. 6). The spheres where chosen to have a uniform size regardless of amino acid type. Usage of amino acid specific radii did not lead to a significant improvement in rSASA calculation (data not shown). While the optimal number of points placed on the sphere has been investigated [52], this parameter was not optimized. Points were distributed uniformly every 5° along the surface of the sphere.

Establishment of rSASA reference standard

The maximal speed molecular surfaces (MSMS) [46] algorithm as implemented in the visual molecular dynamics (VMD) [53] molecular visualization package serves as the reference standard method for rSASA. Protein models with the hydrogen atoms removed are used in order to ensure a consistent representation. In order to convert this rSASA measure into a relative exposure, the rSASA for each amino acid in the protein is divided by the rSASA for that amino

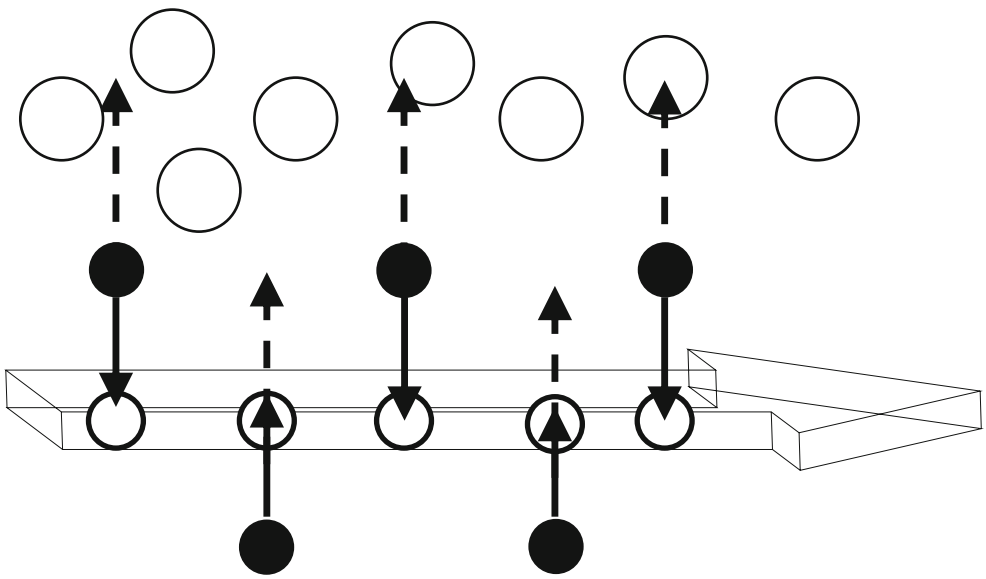


Fig. 5 A β-strand is shown where the C_α atoms and C_β atoms of the strand are represented by *black* and *white* circles respectively. The C_βs of neighboring amino acids are represented by white circles. The neighbor vectors are shown as *dashed lines*. The $(C_\alpha - C_\beta)$ vectors are shown as *solid lines*. The dot product of the neighbor vector and the $(C_\alpha - C_\beta)$ vector gives information about the angle between the two vectors and hence the orientation of the side chain atoms with respect to the neighboring amino acids (large open circles)

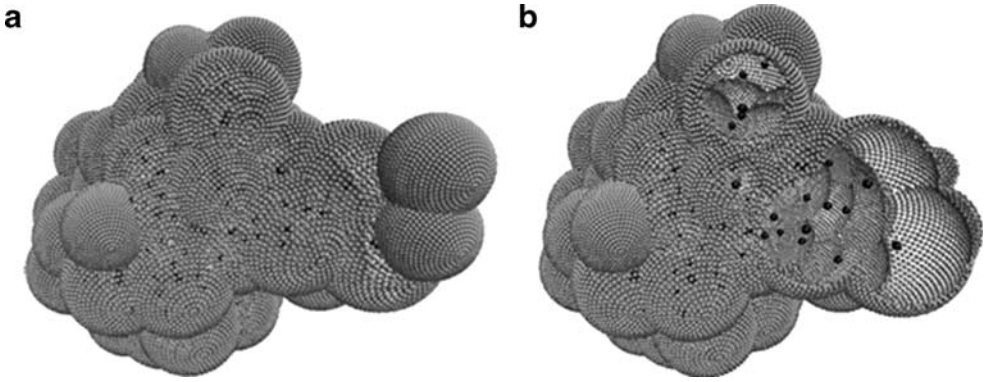


Fig. 6 The overlapping spheres algorithm places a sphere around each C_β and places points on the surface of the spheres. The points that do not overlap with the spheres of any other amino acids are used as a measure of relative exposure. The C_β atoms are colored in black and the points that do not overlap with any other spheres are colored in gray. **a**) the exterior of the protein **b**) a cut away of the protein

acid alone in space (i.e., all other amino acids in the protein were removed). This gives a relative exposure for each amino acid in the protein with a minimum exposure of 0.0 (completely buried) and a maximum exposure of 1.0 (completely exposed).

Optimization of parameters for each approximation algorithm

In order to determine the optimal parameters for each SASA approximation, a Monte Carlo parameter optimization method is used. The parameter set that produces the output that correlates most highly with the rSASA reference standard is selected as optimal. 90% of the proteins in the representative protein database (described below) are used in parameter optimization while 10% was withheld. The correlations reported in Table 1 are based only upon the withheld 10%.

The optimal parameters found for each exposure algorithm are shown. The parameters that maximized the correlation of exposures produced by each algorithm with exposures produced by the rSASA reference standard are selected as optimal.

Table 1 Optimal parameters

Algorithm	Optimal parameters
Neighbor count	lower boundary: 4.0 Å, upper boundary: 11.4 Å
Neighbor vector	lower boundary: 3.3 Å, upper boundary 11.1 Å
Artificial neural network	nine inputs are provided to the ANN: - NC(2.0, 9.4), NV(1.3, 9.1), & NV(1.3,9.1) • $C_\alpha - C_\beta$ - NC(4.0, 11.4), NV(3.3, 11.1), & NV(3.3, 11.1) • $C_\alpha - C_\beta$ - NC(6.0, 13.4), NV(5.3, 13.1), & NV(5.3, 13.1)• $C_\alpha - C_\beta$
Overlapping spheres	sphere radius: 4.75 Å

Establishment of representative protein database for generation of KBPs

Statistics are generated for each amino acid type and each of the exposure algorithms by analysis of the representative protein database described in Table 2. This database contains high resolution (<2.5 Å) structures with <25% homology. The complete list of proteins from the PDB was submitted to the PISCES server [54, 55] to identify proteins with low sequence similarity. The input parameters used for culling are the following: sequence percentage identity <=25%, resolution=0.0 Å–3.0 Å, R-factor=0.3, sequence length 40–10,000 amino acids, non X-ray entries were excluded as were C_α − only entries. The resulting database of unique structures contained 1795 soluble proteins. Information about the proteins used to create the KBPs is summarized in Table 2.

Generation of knowledge-based environment potentials using inverse Boltzmann relation

The following equation describes how histograms are generated for each amino acid type.

$$propensity\_aa_1[j] = \frac{\left[1+\sum_i^n equal\_exposure\left(aa_i, e_j\right)\right]}{\sum_k^m histogram\_aa_1[k]} * m$$

$$equal\_exposure\left(aa_i, e_j\right) = \begin{cases} 1, & e(aa_i) = e_j \\ 0, & e(aa_i) \neq e_j \end{cases}$$

Table 2 Proteins used in KBP generation

Protein category	# Proteins	# Amino Acids	# α-helices	# β-strands
Soluble proteins	1795	884,529	32,075	32,641

Table 3 Relationship between probabilities, propensities, and energies

Probability	Propensity	Energy
Probability > P_{random}	Propensity >1	Energy <0 (favoured)
Probability = P_{random}	Propensity=1	Energy=0 (neutral)
Probability <P_{random}	0 < Propensity <1	Energy >0 (penalized)

where aa_i is amino acid type i, n is the number of amino acids of type i in the database, j is a specific exposure value, e_j is the range of exposure values j associated with that bin, and m is the number of bins (20 bins are used for all algorithms). Prior to multiplication by the number of exposure values, the values in each bin are probabilities ($0 \leq$ probability ≤ 1). Multiplying by the number of bins converts these probabilities to propensities ($0 \leq$ propensity $\leq$ number of bins). Propensities are then converted to energies according to the inverse Boltzmann relation discussed earlier.

The relationship between probabilities, propensities, and energies as used in creation of KBPs is shown in Table 3. P_{random} is defined as 1/# possible exposure values. States found rarely are associated with high energy whereas states found frequently are associated with low energy.

Essentially, exposure values that are seen rarely in native proteins are associated with high energy values whereas exposure values that are seen often in native proteins are associated with low energy values. A spline is used to smooth the bins into a differentiable potential. A pseudocount of 1 is added to each bin so that exposure values that are never seen (i.e., have a count of 0) are not associated with an infinitely large energy.

Benchmark proteins are selected such that 10% of the protein models are "native-like"

Nineteen benchmark proteins are selected for analysis of the exposure algorithms. The decoys were generated by the Rosetta folding algorithm and are a subset of the Rosetta benchmark set. For each of the benchmark proteins, multiple protein models are included in the benchmark (between 70 and 1030 depending on the availability of protein models for each benchmark protein). *Rmsd*100, a normalized form of *rmsd* [56], is used to examine the deviation of each protein model from the native conformation. Protein models having an *rmsd*100 value <5 Å are referred to as "native-like" whereas protein models that have an *rmsd*100 value ≥5 Å are referred to as "nonnative-like." Additional values (between 4 Å and 7 Å) were also tested as a threshold for the definition of "native-like" and yielded similar results. Protein models are selected such that 10% of the decoys are "native-like" and 90% of the protein models are "nonnative-like". This provided a "level playing field" and basis for comparison as the maximum enrichment for all benchmark proteins with this distribution of protein models is 10.0.

The protein models analyzed are a subset of the protein models available for a given benchmark protein and are randomly selected from this larger group. This random selection procedure is repeated ten times to provide standard deviations of the evaluation criteria (read below). Additionally, proteins of various sizes, secondary structure composition, and CATH classifications are chosen to ensure a representative benchmark set (see Table 4).

Table 4 Summary of benchmark proteins used in KBP analysis

PDB ID	CATH classification	# Residues	# Models with *rmsd*100 <5 Å	# Models with *rmsd*100 ≤5 Å	# Models available
1ail	mainly alpha	70	11	99	120
1e6i	mainly alpha	136	7	63	120
1enh	mainly alpha	54	48	432	1120
1r69	mainly alpha	69	11	99	1120
1a19	alpha beta	180	57	513	1120
1iib	alpha beta	212	68	612	1120
1scj	alpha beta	346	11	99	120
1acf	alpha beta	125	103	927	1120
1bm8	alpha beta	99	72	648	1120
1cc8	alpha beta	73	71	639	1120
1ctf	alpha beta	74	90	810	1120
1hz6	alpha beta	216	45	405	1120
1opd	alpha beta	85	95	855	1120
1tig	alpha beta	94	17	153	1120
1b3a	mainly beta	134	64	576	1120
1bq9	mainly beta	54	12	108	120
1c9o	mainly beta	132	49	441	1120
1fna	mainly beta	91	67	603	1120
1shf	mainly beta	118	13	117	1120

Table 5 Average SASA values for amino acids

Amino acid	Average SASA (Å^2)	Standard deviation	n
ALA	209.02	5.22	18,352
ARG	335.73	9.48	40,715
ASN	259.85	7.37	35,232
ASP	257.99	7.31	38,428
CYS	240.50	5.68	10,750
GLN	286.76	8.24	30,958
GLU	285.03	8.28	53,663
GLY	185.15	4.50	48,071
HIS	290.04	7.74	18,812
ILE	273.46	6.50	47,414
LEU	278.44	7.27	75,574
LYS	303.43	8.47	45,807
MET	291.52	8.11	11,698
PHE	311.30	8.59	34,128
PRO	235.41	6.13	38,319
SER	223.04	5.93	46,935
THR	243.55	5.97	46,626
TRP	350.68	10.27	11,853
TYR	328.82	8.94	27,671
VAL	250.09	5.83	59,959

Table 4 provides information about the benchmark proteins used for analysis of the KBPs based upon each exposure algorithm. Proteins with multiple types of secondary structural elements and of various sizes are included.

Average rSASA values are used to convert the actual rSASA into a relative exposure for benchmark proteins

In order to facilitate comparison amongst the exposure algorithms, rSASA values computed with the VMD implementation of the MSMS algorithm are converted from actual areas in Å^2 to relative exposures (on a scale of 0.0 (completely buried) to 1.0 (completely exposed)). To convert areas into relative exposures, the rSASA is divided by the average rSASA for that amino acid type alone in space. The average values for each amino acid type alone in space are shown in Table 5 along with the standard deviations and the number of amino acids (n) used in determining the average.

Evaluation metrics: enrichment, receiver operating characteristic (ROC) curves, and Z-scores are measures of the KBP's discriminatory power

In order to evaluate the KBPs based upon each exposure algorithm, the ability of each KBP to discriminate between native-like and nonnative-like models is examined. The KBP for each algorithm is used to evaluate the energy of all protein models for each benchmark protein. The metric enrichment is used to evaluate the ability of each KBP to distinguish between native-like and nonnative-like protein models.

$$enrichment = \frac{\left(\frac{\#\ of\ native-like\ models\ in\ lowest\ 10\%\ of\ energy\ scores}{\#\ of\ native-like\ models}\right)}{percentage\ of\ native-like\ models}$$

As 10% of the protein models for each benchmark protein are native-like, the maximum enrichment possible for each KBP is 10.0 and a random enrichment is an enrichment of 1.0.

ROC curves display the true positive rate versus the false positive rate for a binary classification system. In this case, the ability of the KBPs based on the approximation algorithms to correctly classify native-like and nonnative-like protein models, is examined. Additionally, the area under the ROC curve (AUC) is determined from these ROC curves. An AUC of 1.0 indicates perfect classification whereas an AUC of 0.5 is representative of a random measure.

Z-scores are calculated for each KBP. A random KBP is expected to achieve a z-score of 0.0. A more negative z-score indicates greater power of the KBP in distinguishing between native-like and nonnative-like protein models.

$$z-score = \frac{(average\ score\ of\ native-like\ models)-(average\ score\ of\ all\ models)}{standard\ deviation\ of\ the\ scores\ of\ all\ models}$$

Results

Increasing algorithm complexity corresponds to a more accurate rSASA approximation yet slower run times

In order to determine how well each exposure algorithm approximates rSASA, the correlation of exposure values produced by each algorithm to the exposure values given by the reference standard rSASA algorithm is examined. Results displaying the correlation with the reference standard rSASA and run times for each algorithm are

Table 6 Exposure algorithm performance

Exposure algorithm	Average runtime per amino acid (seconds)	Correlation with rSASA reference standard
rSASA	3.94e-2	1.0000
Neighbor count	1.13e-5	−0.8526
Neighbor vector	1.87e-5	0.8757
Artificial neural network	5.78e-5	0.8906
Overlapping spheres	3.11e-3	0.8842

shown in Table 6. The rSASA reference standard method takes several orders of magnitude longer (0.39e-2 seconds per amino acid for the rSASA reference standard compared to <6e-5 seconds per amino acid for NC, NV, and ANN and 3e-3 for OLS) than any of the approximation methods indicating its infeasibility for use in rapid protein structure prediction. As expected, as the algorithm complexity increases, the runtime increases as well. Of note, the OLS algorithm is two orders of magnitude slower than the other approximation algorithms but still 12 times faster than the rSASA reference standard algorithm. The correlation of the neighbor count algorithm is negative due to the fact that the number of neighbors is *inversely* proportional to the rSASA. As algorithm complexity increases, the correlation with the rSASA reference standard also increases. The ANN approxima-

a rSASA KBP

	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1.0	
A	-2.27	-0.82	-0.54	-0.31	-0.22	-0.15	0.11	0.30	0.68	1.48	2.18	2.90	3.89	3.77	3.87	4.99	7.07	7.76	7.76	7.76	A
R	-0.68	-0.59	-0.68	-0.79	-0.85	-0.83	-0.66	-0.54	-0.30	0.10	0.36	0.97	1.39	2.18	3.12	4.01	4.87	5.47	7.26	6.57	R
N	-1.37	-0.71	-0.61	-0.65	-0.69	-0.64	-0.55	-0.37	-0.15	0.08	0.55	1.03	1.88	3.28	4.20	4.89	5.99	6.39	7.09	7.09	N
D	-1.03	-0.62	-0.65	-0.69	-0.77	-0.81	-0.66	-0.48	-0.26	0.02	0.42	1.11	1.85	3.44	4.28	5.11	5.81	7.42	7.42	7.42	D
C	-2.54	-1.13	-0.57	0.06	0.51	1.04	1.71	2.25	2.42	2.69	3.24	3.64	4.56	5.94	4.56	4.84	5.94	5.94	5.94	5.94	C
Q	-1.06	-0.55	-0.57	-0.67	-0.72	-0.75	-0.74	-0.60	-0.33	0.01	0.50	1.07	2.08	2.97	3.33	4.03	5.03	5.87	6.97	6.97	Q
E	-0.57	-0.30	-0.36	-0.55	-0.70	-0.79	-0.85	-0.78	-0.61	-0.30	0.10	0.71	1.35	2.40	3.67	5.16	4.72	6.17	7.56	7.56	E
G	-2.04	-0.95	-0.82	-0.61	-0.42	-0.30	-0.12	0.20	0.86	1.61	2.78	3.58	3.76	3.49	4.69	6.25	6.94	7.64	7.64	7.64	G
H	-1.60	-1.04	-0.81	-0.76	-0.62	-0.44	-0.25	0.01	0.32	0.62	1.04	1.28	2.15	2.88	3.52	3.90	4.67	6.46	6.46	6.46	H
I	-2.49	-0.95	-0.50	-0.10	0.26	0.58	0.82	1.15	1.66	2.15	2.76	2.98	3.99	4.90	5.19	6.00	6.00	7.39	7.39	7.39	I
L	-2.41	-1.00	-0.58	-0.24	0.03	0.43	0.71	1.08	1.39	1.87	2.31	2.80	3.48	4.30	5.37	5.45	6.06	6.75	7.85	7.85	L
K	0.08	0.02	-0.28	-0.57	-0.74	-0.85	-0.87	-0.83	-0.72	-0.44	-0.08	0.42	1.14	2.01	3.20	3.66	4.42	6.03	6.73	7.42	K
M	-2.34	-0.90	-0.54	-0.23	0.09	0.31	0.31	0.77	0.88	1.27	1.54	1.81	2.45	2.81	2.92	3.70	3.70	5.78	6.48	5.78	M
F	-2.41	-1.14	-0.64	-0.24	0.26	0.58	0.98	1.23	1.56	1.91	2.23	2.61	3.27	3.81	5.24	4.83	5.93	5.64	7.03	6.33	F
P	-1.48	-0.72	-0.66	-0.68	-0.63	-0.59	-0.50	-0.38	-0.24	0.24	0.88	1.61	2.47	3.45	3.83	3.94	5.55	7.16	7.16	7.16	P
S	-1.81	-0.86	-0.68	-0.69	-0.62	-0.49	-0.26	0.01	0.26	0.76	1.33	2.10	2.97	3.53	3.60	4.18	6.30	7.40	7.40	7.40	S
T	-1.81	-0.89	-0.76	-0.62	-0.58	-0.56	-0.37	0.09	0.46	0.88	1.31	2.11	3.12	3.77	4.03	4.27	5.75	7.36	6.67	7.36	T
W	-2.12	-1.32	-1.01	-0.57	-0.12	0.29	0.64	1.17	1.30	1.68	2.31	2.69	3.31	4.34	3.87	5.95	4.85	4.56	5.95	5.95	W
Y	-2.00	-1.23	-1.01	-0.69	-0.33	0.04	0.36	0.75	0.98	1.56	1.78	2.38	2.99	3.94	4.11	5.27	5.78	5.49	6.19	6.88	Y
V	-2.45	-0.92	-0.51	-0.20	0.09	0.37	0.62	0.99	1.55	1.98	2.57	3.04	4.12	5.31	5.13	5.31	5.82	7.62	7.62	7.62	V
	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1.0	

b NC KBP

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
A	8.12	3.58	2.66	1.26	0.60	-0.09	-0.23	-0.33	-0.29	-0.33	-0.50	-0.68	-0.93	-0.99	-0.76	-0.05	0.90	2.02	3.38	6.04	A
R	7.61	4.00	2.79	1.36	0.53	-0.18	-0.71	-1.00	-1.07	-1.06	-0.94	-0.71	-0.41	0.28	1.13	2.22	5.04	5.66	7.61	7.61	R
N	7.43	4.60	2.49	0.47	-0.13	-0.59	-0.81	-0.87	-0.78	-0.78	-0.62	-0.51	-0.39	-0.18	0.33	1.18	2.53	3.77	6.33	6.74	N
D	7.75	4.35	2.15	0.34	-0.30	-0.76	-0.98	-1.02	-0.87	-0.74	-0.57	-0.38	-0.09	0.28	0.88	1.59	2.96	4.86	6.36	7.75	D
C	6.28	4.49	4.49	3.19	2.47	1.94	1.24	0.52	0.21	-0.33	-0.71	-1.04	-1.29	-1.30	-0.95	-0.32	1.08	2.17	4.67	6.28	C
Q	7.32	4.10	2.57	1.01	0.29	-0.50	-0.86	-1.02	-1.00	-0.88	-0.66	-0.57	-0.30	-0.03	0.69	1.47	2.86	4.75	5.37	7.32	Q
E	7.90	3.72	2.32	0.45	-0.23	-0.88	-1.07	-1.16	-0.97	-0.76	-0.48	-0.19	0.10	0.57	1.33	2.24	3.85	4.76	7.90	7.90	E
G	7.29	3.68	2.25	0.47	-0.16	-0.52	-0.61	-0.51	-0.45	-0.40	-0.48	-0.50	-0.62	-0.66	-0.38	0.19	1.06	2.21	3.71	5.21	G
H	6.81	3.05	2.24	1.03	0.56	0.05	-0.26	-0.55	-0.74	-0.90	-0.93	-1.01	-0.82	-0.41	0.11	1.17	2.35	4.41	5.71	6.81	H
I	7.72	4.89	3.36	2.73	2.15	1.55	0.67	0.26	-0.14	-0.52	-0.88	-1.11	-1.32	-1.26	-0.66	0.47	1.97	3.98	7.72	7.72	I
L	8.18	4.85	3.66	2.59	1.86	1.30	0.57	0.10	-0.35	-0.68	-0.95	-1.19	-1.30	-1.09	-0.41	0.79	2.40	4.07	6.10	6.57	L
K	7.75	3.76	2.18	0.56	-0.07	-0.72	-1.03	-1.17	-1.08	-0.95	-0.66	-0.22	0.24	0.94	1.90	2.81	4.57	5.80	7.75	7.75	K
M	6.12	3.10	2.25	1.72	1.15	0.76	0.22	-0.08	-0.34	-0.60	-0.78	-1.03	-1.17	-1.12	-0.45	0.32	1.56	3.41	4.51	6.12	M
F	7.37	4.54	3.71	2.51	1.90	1.33	0.75	0.21	-0.25	-0.72	-0.98	-1.27	-1.37	-1.06	-0.25	0.95	2.39	4.48	6.28	7.37	F
P	7.51	3.52	1.99	0.21	-0.37	-0.72	-0.79	-0.71	-0.70	-0.64	-0.58	-0.50	-0.49	-0.23	0.39	1.23	2.55	3.77	5.43	7.51	P
S	7.74	3.45	2.24	0.65	0.03	-0.50	-0.71	-0.70	-0.61	-0.60	-0.52	-0.53	-0.60	-0.54	-0.19	0.51	1.66	2.81	4.27	5.66	S
T	7.70	3.71	2.61	1.13	0.53	-0.18	-0.64	-0.72	-0.70	-0.77	-0.76	-0.70	-0.62	-0.54	-0.16	0.62	1.70	3.16	4.75	6.31	T
W	6.30	4.91	3.59	2.63	1.81	1.24	0.47	-0.05	-0.42	-0.79	-1.20	-1.30	-1.33	-0.81	0.28	1.47	3.81	5.20	5.60	6.30	W
Y	7.24	5.44	3.62	2.41	1.76	1.05	0.47	0.02	-0.39	-0.84	-1.12	-1.26	-1.29	-0.90	0.04	1.50	2.68	5.04	6.54	7.24	Y
V	7.96	4.56	3.25	2.31	1.72	1.06	0.45	0.02	-0.20	-0.47	-0.74	-0.96	-1.25	-1.24	-0.76	0.22	1.44	3.07	4.87	7.96	V
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	

c NV KBP

	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1.0	
A	-0.42	-1.42	-1.05	-0.60	-0.38	-0.08	-0.01	0.14	0.10	-0.04	0.06	0.40	0.26	0.85	1.34	2.19	2.93	3.88	4.23	5.35	A
R	1.90	0.14	-0.44	-0.57	-0.63	-0.63	-0.72	-0.70	-0.68	-0.67	-0.38	0.05	0.33	0.86	1.73	2.34	3.26	4.39	4.31	7.61	R
N	0.77	-0.49	-0.51	-0.40	-0.33	-0.39	-0.39	-0.42	-0.45	-0.55	-0.44	-0.20	-0.12	0.20	0.60	1.41	3.11	4.49	5.64	6.04	N
D	1.56	0.05	-0.28	-0.31	-0.29	-0.28	-0.41	-0.46	-0.55	-0.71	-0.59	-0.35	-0.36	0.02	0.44	1.25	2.54	4.20	4.92	7.06	D
C	-0.66	-1.66	-1.44	-0.94	-0.67	-0.16	0.11	0.43	0.74	1.12	1.37	2.22	2.43	3.10	3.45	3.88	5.59	6.28	4.49	6.28	C
Q	1.17	-0.24	-0.43	-0.41	-0.41	-0.43	-0.41	-0.53	-0.67	-0.74	-0.57	-0.13	-0.01	0.53	1.09	1.91	2.93	3.98	4.75	6.62	Q
E	1.99	0.45	0.03	-0.04	-0.12	-0.27	-0.33	-0.45	-0.66	-0.89	-0.77	-0.47	-0.52	-0.03	0.45	1.36	2.66	3.96	4.40	6.51	E
G	0.24	-1.00	-0.89	-0.54	-0.28	-0.17	-0.08	-0.09	-0.10	-0.26	-0.26	-0.25	-0.02	0.09	0.63	2.04	3.45	3.83	4.43	5.35	G
H	0.75	-0.59	-0.99	-0.90	-0.79	-0.58	-0.52	-0.38	-0.26	-0.14	0.08	0.37	0.62	1.02	1.40	2.11	3.07	3.68	4.17	5.43	H
I	-0.49	-1.64	-1.43	-0.91	-0.52	-0.22	0.05	0.29	0.53	0.70	1.03	1.57	2.21	2.65	3.04	3.98	5.08	5.32	5.42	7.03	I
L	-0.22	-1.50	-1.43	-0.98	-0.63	-0.35	-0.12	0.19	0.39	0.60	0.96	1.57	1.77	2.47	2.73	3.75	4.49	5.14	5.62	7.08	L
K	2.73	0.97	0.26	-0.03	-0.24	-0.46	-0.61	-0.68	-0.78	-0.84	-0.78	-0.39	-0.28	0.26	0.77	1.44	2.51	3.99	4.42	5.67	K
M	-0.14	-1.47	-1.30	-0.87	-0.64	-0.26	-0.07	0.00	0.19	0.37	0.65	1.06	1.36	1.62	2.13	2.35	3.10	3.17	3.59	5.20	M
F	-0.03	-1.48	-1.51	-1.09	-0.71	-0.38	-0.03	0.28	0.59	0.81	1.23	1.66	2.11	2.44	3.06	3.79	4.43	4.74	5.77	6.28	F
P	0.67	-0.55	-0.66	-0.51	-0.31	-0.26	-0.29	-0.29	-0.26	-0.36	-0.28	-0.35	-0.24	0.08	0.34	1.22	2.25	3.75	4.18	5.31	P
S	0.15	-0.90	-0.71	-0.46	-0.36	-0.32	-0.21	-0.20	-0.28	-0.38	-0.37	-0.13	0.01	0.42	0.82	1.52	2.55	3.70	4.00	5.54	S
T	0.26	-0.95	-0.78	-0.59	-0.47	-0.38	-0.34	-0.35	-0.36	-0.37	-0.31	0.03	0.33	0.93	1.36	2.29	3.23	3.96	4.33	6.31	T
W	0.42	-1.16	-1.48	-1.20	-0.87	-0.52	-0.29	0.18	0.36	0.52	1.09	1.62	1.71	2.47	3.00	3.52	4.10	4.91	6.30	5.20	W
Y	0.37	-1.17	-1.41	-1.16	-0.86	-0.61	-0.30	0.04	0.37	0.59	0.88	1.46	1.67	2.26	2.92	3.47	4.53	5.29	5.63	7.24	Y
V	-0.60	-1.65	-1.29	-0.75	-0.44	-0.23	0.00	0.21	0.30	0.39	0.69	1.35	1.69	2.13	2.68	3.53	4.59	4.56	5.76	6.35	V
	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1.0	

d ANN KBP

	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1.0	
A	-2.22	-0.88	-0.37	-0.17	-0.14	-0.23	-0.09	-0.05	0.39	1.04	2.18	3.96	8.12	8.12	8.12	8.12	8.12	8.12	8.12	8.12	A
R	-1.07	-1.01	-0.93	-0.96	-0.96	-0.92	-0.57	-0.20	0.44	1.11	2.39	4.17	7.61	7.61	7.61	7.61	7.61	7.61	7.61	7.61	R
N	-1.42	-0.76	-0.64	-0.65	-0.68	-0.77	-0.67	-0.49	-0.18	0.13	1.96	5.23	7.43	7.43	7.43	7.43	7.43	7.43	7.43	7.43	N
D	-1.01	-0.72	-0.57	-0.64	-0.77	-0.93	-0.84	-0.67	-0.33	-0.01	1.70	4.92	7.75	7.75	7.75	7.75	7.75	7.75	7.75	7.75	D
C	-2.51	-1.22	-0.47	0.00	0.39	0.82	1.28	2.09	2.47	3.06	4.08	4.67	6.28	6.28	6.28	6.28	6.28	6.28	6.28	6.28	C
Q	-1.25	-0.80	-0.68	-0.73	-0.89	-0.95	-0.74	-0.46	0.07	0.73	2.10	4.54	7.32	7.32	7.32	7.32	7.32	7.32	7.32	7.32	Q
E	-0.69	-0.49	-0.50	-0.64	-0.86	-1.08	-0.99	-0.83	-0.40	0.16	1.74	4.09	7.90	7.90	7.90	7.90	7.90	7.90	7.90	7.90	E
G	-1.92	-0.71	-0.40	-0.28	-0.31	-0.42	-0.48	-0.44	-0.27	0.26	1.76	4.25	7.99	7.99	7.99	7.99	7.99	7.99	7.99	7.99	G
H	-1.72	-1.25	-0.90	-0.70	-0.57	-0.40	-0.09	0.15	0.49	0.76	2.22	3.31	6.81	6.81	6.81	6.81	6.81	6.81	6.81	6.81	H
I	-2.43	-1.17	-0.58	-0.22	0.21	0.38	0.85	1.49	2.09	2.52	3.58	5.08	7.72	7.72	7.72	7.72	7.72	7.72	7.72	7.72	I
L	-2.33	-1.23	-0.72	-0.35	-0.05	0.32	0.77	1.35	1.81	2.35	3.47	5.47	8.18	8.18	8.18	8.18	8.18	8.18	8.18	8.18	L
K	-0.33	-0.57	-0.68	-0.89	-1.00	-1.08	-0.95	-0.69	-0.19	0.27	1.67	4.09	7.75	7.75	7.75	7.75	7.75	7.75	7.75	7.75	K
M	-2.29	-1.10	-0.62	-0.34	-0.17	0.11	0.46	0.83	1.14	1.52	2.05	3.38	6.81	6.81	6.81	6.81	6.81	6.81	6.81	6.81	M
F	-2.35	-1.32	-0.77	-0.25	0.13	0.51	1.01	1.40	1.84	2.44	3.40	5.43	7.37	7.37	7.37	7.37	7.37	7.37	7.37	7.37	F
P	-1.48	-0.79	-0.57	-0.51	-0.48	-0.58	-0.62	-0.57	-0.45	-0.13	1.51	3.54	7.51	7.51	7.51	7.51	7.51	7.51	7.51	7.51	P
S	-1.77	-0.77	-0.56	-0.43	-0.49	-0.63	-0.58	-0.43	0.01	0.40	1.82	3.55	7.74	7.74	7.74	7.74	7.74	7.74	7.74	7.74	S
T	-1.79	-0.92	-0.68	-0.61	-0.59	-0.61	-0.55	-0.14	0.45	0.89	2.29	4.11	7.70	7.70	7.70	7.70	7.70	7.70	7.70	7.70	T
W	-2.15	-1.51	-0.89	-0.44	0.00	0.23	0.72	1.25	1.77	2.53	3.40	5.20	6.30	6.30	6.30	6.30	6.30	6.30	6.30	6.30	W
Y	-2.15	-1.46	-0.92	-0.48	-0.03	0.27	0.66	1.16	1.67	2.21	3.52	5.63	7.24	7.24	7.24	7.24	7.24	7.24	7.24	7.24	Y
V	-2.41	-1.03	-0.53	-0.22	-0.01	0.14	0.58	1.13	1.57	2.18	3.15	5.07	7.96	7.96	7.96	7.96	7.96	7.96	7.96	7.96	V
	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1.0	

e OLS KBP

	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1.0	
A	-2.41	-0.49	-0.15	-0.05	0.07	-0.15	0.30	0.22	0.59	1.06	1.63	1.97	3.17	3.90	3.79	4.36	7.43	8.12	8.12	8.12	A
R	-1.54	-0.98	-0.95	-0.84	-0.78	-0.71	-0.23	0.04	0.58	1.09	1.69	2.02	3.16	3.84	4.52	4.56	7.61	7.61	7.61	7.61	R
N	-1.73	-0.74	-0.60	-0.53	-0.56	-0.60	-0.34	-0.20	0.05	0.41	0.68	1.19	2.58	4.00	5.03	5.48	7.43	7.43	7.43	7.43	N
D	-1.40	-0.69	-0.59	-0.63	-0.68	-0.79	-0.46	-0.43	-0.08	0.23	0.50	1.09	2.33	3.52	4.45	5.27	7.75	7.75	7.75	7.75	D
C	-2.72	-0.66	-0.04	0.44	0.88	1.10	1.80	2.20	2.62	3.23	3.79	3.71	4.89	6.28	4.49	6.28	6.28	6.28	6.28	6.28	C
Q	-1.61	-0.79	-0.63	-0.71	-0.78	-0.80	-0.34	-0.21	0.21	0.81	1.21	1.76	2.92	3.40	4.32	5.12	6.62	7.32	7.32	7.32	Q
E	-1.12	-0.56	-0.55	-0.65	-0.82	-0.97	-0.59	-0.64	-0.15	0.22	0.70	1.17	2.55	3.82	3.84	4.76	7.90	7.90	7.90	7.90	E
G	-2.13	-0.49	-0.23	-0.17	-0.15	-0.24	-0.20	-0.19	-0.02	0.22	0.70	1.51	2.92	4.12	3.91	4.59	6.89	7.99	7.29	7.99	G
H	-2.09	-1.06	-0.68	-0.56	-0.24	-0.15	0.19	0.46	0.65	1.14	1.22	1.83	3.00	3.48	3.44	4.33	6.12	6.81	6.81	6.81	H
I	-2.62	-0.80	-0.24	0.15	0.48	0.67	1.23	1.71	2.23	2.66	3.13	3.49	4.78	5.08	5.42	5.52	7.72	7.72	7.72	7.72	I
L	-2.56	-0.90	-0.44	0.04	0.32	0.59	1.17	1.52	2.05	2.50	2.79	3.44	4.65	5.04	5.00	6.10	8.18	8.18	8.18	8.18	L
K	-0.89	-0.74	-0.82	-0.86	-0.87	-0.93	-0.58	-0.45	-0.04	0.48	0.79	1.27	2.26	3.57	3.90	4.61	7.06	7.75	7.75	7.75	K
M	-2.48	-0.82	-0.34	-0.17	0.11	0.37	0.74	1.10	1.49	1.81	1.94	2.49	2.84	3.35	3.01	4.25	6.81	6.81	6.81	6.81	M
F	-2.58	-0.98	-0.36	0.18	0.49	0.86	1.29	1.53	1.93	2.78	3.02	3.35	4.48	4.67	5.30	5.77	6.68	7.37	7.37	7.37	F
P	-1.80	-0.63	-0.48	-0.36	-0.37	-0.34	-0.34	-0.31	-0.17	0.01	0.41	0.96	2.03	3.66	3.75	4.29	6.82	7.51	7.51	7.51	P
S	-2.02	-0.64	-0.44	-0.35	-0.38	-0.50	-0.24	-0.13	0.30	0.56	0.89	1.49	2.48	3.83	3.66	4.03	7.74	7.74	7.74	7.74	S
T	-2.05	-0.78	-0.60	-0.48	-0.44	-0.43	-0.21	0.11	0.63	1.06	1.37	2.05	2.92	4.03	3.96	4.75	7.00	7.70	7.70	7.70	T
W	-2.48	-1.10	-0.56	-0.02	0.27	0.49	1.05	1.41	1.98	2.34	2.74	3.73	4.35	4.50	5.60	5.20	6.30	6.30	6.30	6.30	W
Y	-2.47	-1.14	-0.53	-0.01	0.30	0.53	1.07	1.35	1.76	2.42	2.79	3.23	4.02	5.29	5.85	5.44	7.24	7.24	7.24	7.24	Y
V	-2.58	-0.70	-0.24	0.02	0.26	0.38	0.92	1.36	1.85	2.21	2.67	3.28	4.20	4.63	4.87	5.56	7.96	7.96	7.96	7.96	V
	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1.0	

Fig. 7 The knowledge-based potentials based upon each exposure algorithm are shown and colored by value where *white* represents low values and *dark gray* represents high values. A visual inspection of the KBPs confirms that the energies shown in the KBPs agree with expectations. For example, one expects a hydrophobic amino acid, for example valine (V), to prefer a low exposure value, a large number of neighbors, and a low neighbor vector magnitude. This is in fact what is seen as indicated by the minima in the plots. Conversely, one expects a hydrophilic amino acid, such as lysine (K) to prefer a high exposure value, a small number of neighbors and a high neighbor vector magnitude. This is also what is seen in the plots

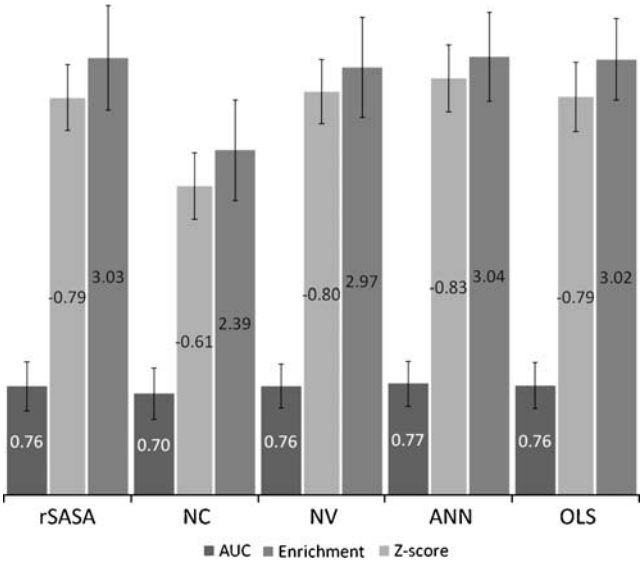


Fig. 8 The average enrichment, z-score, and area under the ROC curve (AUC) is shown for each exposure algorithm over all benchmark proteins. The z-scores are in light gray, the AUC values are in medium gray, and the enrichment values are in dark gray. The neighbor count algorithm performs the least favorably according to all of the evaluation measures whereas the remaining algorithms perform approximately the same with the ANN generally performing slightly better than the others

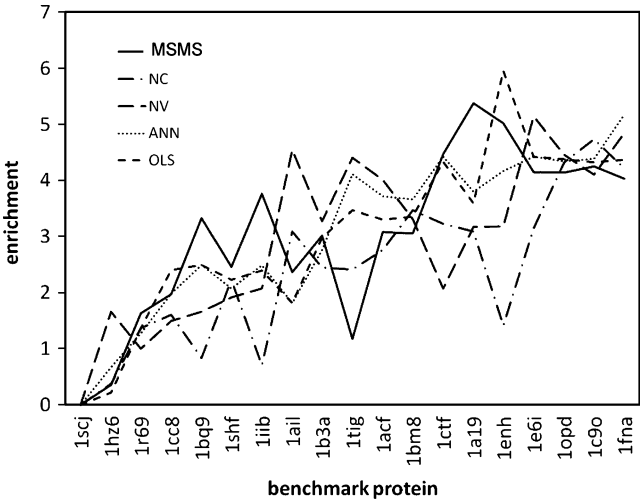


Fig. 9 The enrichment is shown for each algorithm over all benchmark proteins. There are some proteins for which none of the exposure algorithms provided an enrichment (for example 1scj) while there are some benchmark proteins for which many of the exposure algorithms provided good enrichments. There are also proteins for which the enrichment produced by each algorithm increased with algorithm complexity as expected (for example 1enh)

tion algorithm correlates most highly (r=0.89) with the rSASA reference standard.

Visual inspection of KBPs confirm expected trends

A visual inspection of the KBPs ensures that the potentials agree with expectations (see Fig. 7). For example, one expects for hydrophobic amino acids in solution to prefer burial. This is in fact what is seen. Consider the preference of hydrophobic amino acids, such as valine (V), methionine (M), and phenylalanine (F) for a large number of neighbors, a small neighbor vector magnitude, and small relative exposures. Additionally, one expects hydrophilic amino acids to prefer exposure in solution. This is also the case. Consider the preference of the hydrophilic amino acids lysine (K), asparagine (N), and glutamine (Q) for low neighbor counts, a large neighbor vector magnitude, and large relative exposures.

Evaluation metrics indicate that the neighbor count algorithm does not perform as well as other approximation algorithms

As evidenced by the enrichment values in Fig. 8, the rSASA reference standard and the neighbor vector, artificial neural network, and overlapping spheres algorithms perform similarly (enrichment=~3.0) and all outperform the NC method (enrichment <2.5). While no single method clearly dominates the others, some trends can be seen (Fig. 9). In several cases (i.e., 1bq9, 1iib, 1enh), the neighbor count algorithm does not perform as well as the other algorithms. While the rSASA reference standard

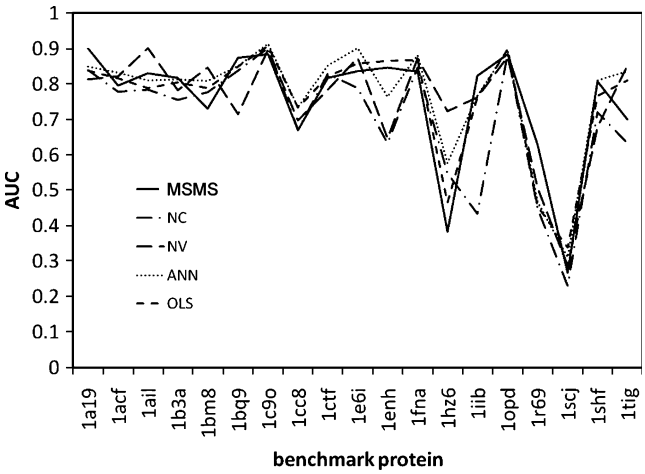
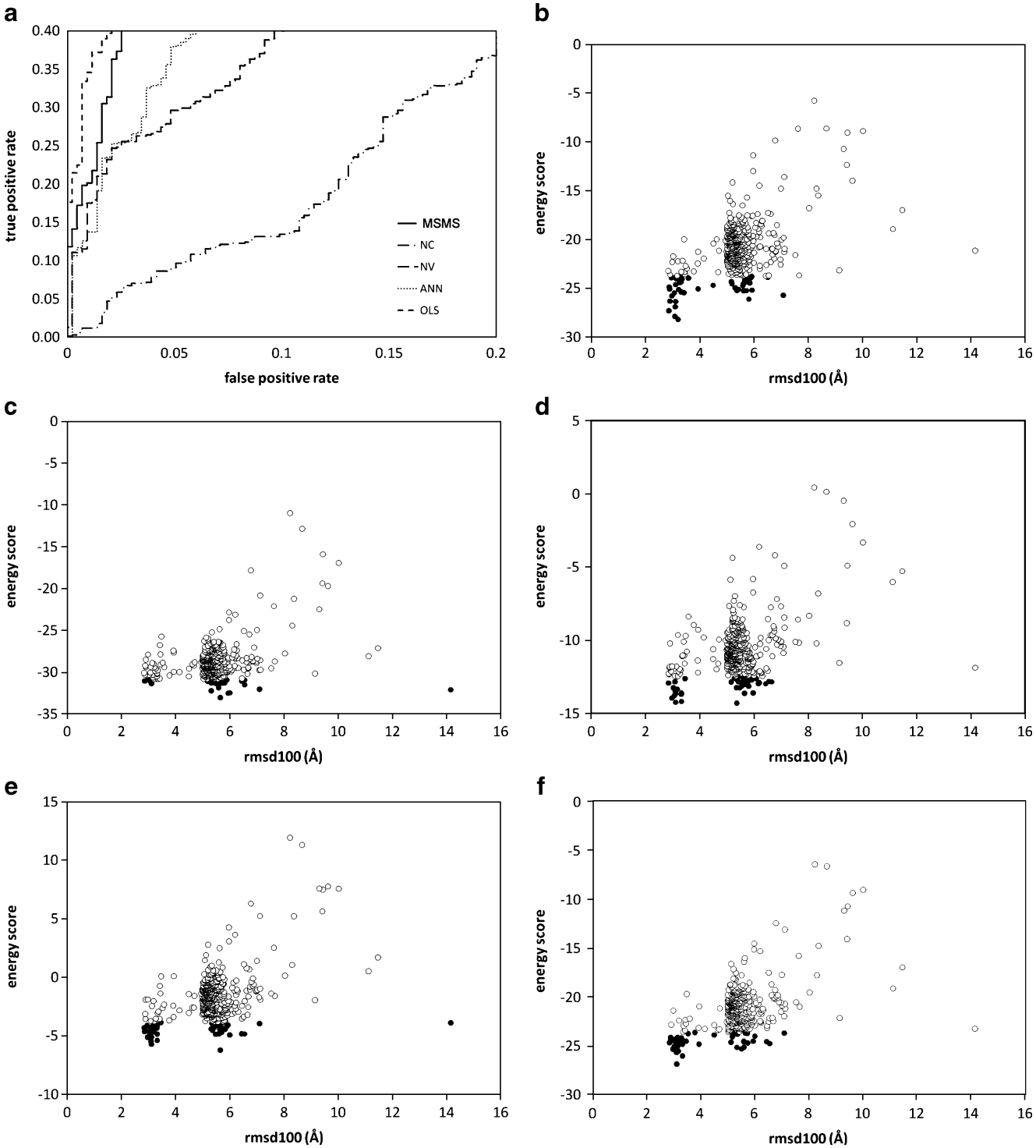


Fig. 10 The area under the ROC curve (AUC) is shown for each exposure algorithm over all benchmark proteins. The AUC varies widely over the benchmark proteins. There are some proteins for which all algorithms perform very well (for example, 1c9o) while there are some proteins for which none of the algorithms perform well (for example, 1scj)



algorithm often provides the greatest enrichment (i.e., 1bq9, 1iib, 1a19), there are several cases in which the neighbor vector algorithm provides the better results (i.e., 1ail, 1b3a, 1e6i).

Additionally, the area under the ROC curve (AUC) is examined for the KBPs over the benchmark proteins (see Fig. 10). Again, the AUC values vary widely across benchmark proteins. However, the neighbor count algorithm (AUC=0.7) lags a bit behind the neighbor vector, artificial neural network, overlapping spheres, and reference standard rSASA algorithms (AUCs ≥0.75).

The z-scores also support the trends shown by the other evaluation metrics. The neighbor count has the least

◀ **Fig. 11** **a**) The ROC curve for 1enh. As the algorithm complexity increases, the area under the ROC curve increases. In this case, the OLS algorithm is able to distinguish between native-like and nonnative-like models more effectively than the reference standard rSASA algorithm. **b**) rSASA, enrichment: 5. **c**) neighbor count, 1.46. **d**) neighbor vector, 3.13. **e**) ann, 4.58. **f**) ols, 6.67. In **b**) – **f**) the energy scores assigned to each protein model (each protein model is represented by one point) is plotted against the *rmsd*100 value of that model. Models assigned an energy score in the lowest 10% (most energetically favorable) are shown as *solid circles* whereas models assigned an energy score in the highest 90% (least energetically favorable) are shown as *open circles*. If the energy potential is able to perfectly distinguish between native-like (<5 Å *rmsd*100) and nonnative-like (≥5 Å *rmsd*100) models, the 10% of models identified as most energetically favorable (shown in *black*) would have an *rmsd*100 value <5 Å. As the algorithm complexity increases, the potential based on the algorithm is able to more effectively distinguish between native-like and nonnative-like models as also indicated by the increasing enrichment values. Interestingly, the OLS algorithm achieves a higher enrichment value than the true rSASA value indicating that additional factors must be taken into account in order to capture all aspects of environment free energy

negative z-score (−0.61) whereas the artificial neural network has the most negative z-score (−0.83) with neighbor vector coming in a close second (−0.80).

A detailed analysis of the benchmark protein 1enh

The benchmark protein 1enh is an example where the potentials are able to distinguish between native-like and nonnative-like models to an extent that corresponds to the complexity of each algorithm (i.e., the NC algorithm is the least effective and the OLS algorithm is the most effective). This is indicated by the increasing area under the ROC curve (see Fig. 11a) moving from NC to NV to ANN to OLS. This can also be seen when the *rmsd*100 is plotted against the energy score assigned to each protein model (see Fig. 11b-f). As the algorithm complexity increases, the KBP is able to more effectively identify native-like protein models. Of note, the OLS KBP yields a higher enrichment than the rSASA reference standard. This indicates that environment free energy KBPs based on rSASA approximation alone may not be a complete picture of the environment free energy and that additional factors should be taken into account in order to more completely capture the environment free energy. Further examination is necessary to explore this question.

For a specific example, consider ALA5 of a 1enh protein model (Fig. 12). The rSASA method determines that the relative exposure of ALA5 is 0.375, ranked the 13th most exposed amino acid of the 54 amino acids in the protein model. The NC algorithm calculated that ALA5 has 6.495 neighbors and ranked ALA5 as the 21st most exposed amino acid in the protein model. However, the NV algorithm was able to discern that the majority of ALA5's neighbors are on one side of the amino acid leaving the other side relatively exposed. The NV algorithm assigned ALA5 a vector of magnitude 0.568 and ranked ALA5 as the 19th most exposed amino acid in the model, closer to its true rank. The ANN predicted a relative exposure of 0.348 for ALA5 and ranked it as the 18th most exposed amino acid in the protein model, again closer to its true rank than the ranks achieved by the NC and NV algorithms. The OLS algorithm returned a relative exposure of 0.372 for ALA5

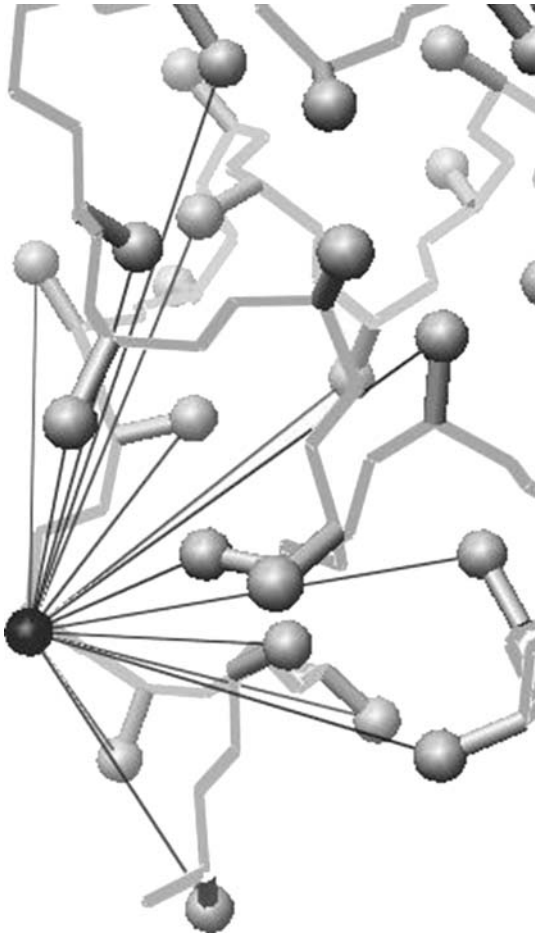


Fig. 12 The backbone and C_βs are shown in *gray*. The ALA5 C_β is shown in *black*. The actual relative rSASA as determined by the reference standard method of ALA5 is 0.375 and it is the 13th most exposed exposed amino acid in the protein model. *Lines* are drawn from the ALA5 C_β to all C_βs assigned a neighbor weight >0 as determined by the neighbor count algorithm. Although ALA5 has many neighbors, all of the neighbors are on one face of the amino acid leaving the other face exposed. Therefore, the neighbor count algorithm ranks ALA5 only as the 21st most exposed amino acid. The neighbor vector algorithm is able to distinguish that most of the neighboring amino acids are on one face of ALA5 and ranks ALA5 as the 19th most exposed amino acid in the protein model. The ANN is able to use the NC, NV, and $NV \bullet (C_\alpha - C_\beta)$ information to more accurately determine the actual exposure and rank ALA5 as the 18th most exposed amino acid in the protein model. The OLS algorithm ranks ALA5 as the 13th most exposed amino acid in the model, its true rank

and ranked it as the 13th most exposed amino acid in the protein model, which is in fact its correct ranking. The exposure value given for the ALA5 of a 1enh protein model as well as the rank of ALA5 amongst the 54 amino acids in the protein model is shown in Table 7.

Discussion

Four algorithms for determining the relative exposure on a reduced protein model are presented. The complexity of these algorithms varies and as expected, the simplest algorithms are the most efficient in terms of runtime but less effective in approximating the reference standard rSASA method and distinguishing between native-like and nonnative-like protein models. Also as expected, the more complex algorithms, such as the artificial neural network and overlapping spheres, achieve more accurate exposure measures and are more effectively able to distinguish between native-like and nonnative-like protein models.

Neighbor count is the simplest measure of exposure and achieves the lowest average enrichment. Also as expected, as the algorithms increase in complexity, they are able to achieve a higher enrichment. The ANN is particularly effective at this task and achieves enrichments on reduced protein models that are nearly as high as the enrichments achieved by the rSASA on full-atom protein models.

As the Rosetta models used for benchmarking were generated using the Rosetta environment score, most of these models bury apolar amino acids and expose polar amino acids and fulfill overall the generally expected environment architecture within proteins. Hence the enrichment test performed in this work is a stringent test that measures improvement over the Rosetta energy function which explains the rather moderate enrichment values. Substantially higher enrichments can be obtained if models are created without the use of the environment score.

Table 7 Exposure algorithm performance for ALA5

Exposure algorithm	Exposure value given for ALA5 by each exposure algorithm	Rank of ALA5 amongst 54 all amino acids in the protein model given by each exposure algorithm
rSASA reference Standard	0.375 relative exposure	13th most exposed amino acid
Neighbor count	6.495 neighbors	21st most exposed amino acid
Neighbor vector	0.568 NV magnitude	19th most exposed amino acid
Artificial neural network	0.348 relative exposure	18th most exposed amino acid
Overlapping spheres	0.372 relative exposure	13th most exposed amino acid

As is seen in Fig. 9 and indicated by the large standard deviations shown in Fig. 8, the degree to which the algorithms are able to recognize native-like protein varies widely. Consider the high enrichments produced for the protein 1e6i. In this case, the algorithms are fairly effective in distinguishing between native-like and nonnative-like protein models. However, there are proteins that are "hard," for example 1scj. All algorithms produce an enrichment of 0.0 (worse than random).

In all cases, the maximum possible enrichment of 10.0 is not achieved by any algorithm, including the rSASA reference standard. This indicates that the environment free energy approximations based on SASA contain a limited amount of information and additional energy terms should be considered in order to achieve additional discriminatory power.

The large standard deviations of the enrichment values (shown in Fig. 8) indicate that further improvements to these algorithms are possible. The fact that the reference standard rSASA method does not always perform best in terms of ability to distinguish between native-like and nonnative-like protein models is unexpected (for example, consider the benchmark protein 1tig). The assumption that environment free energy is directly proportional to SASA should be investigated further to determine if this is strictly the case or if there may be other crucial contributions to environment free energy as well.

Future work includes an in depth analysis of various histogram sizes used in creation of the KBPs and optimizing parameters with the standard for optimality being the parameters that yield the greatest enrichment for protein structure prediction rather than correlation with the reference standard rSASA.

Conclusions

Four exposure algorithms of varying complexities are presented that efficiently produce exposures on reduced protein models that closely correlate with the exposure measures given by the rSASA reference standard on a full-atom model. These exposure measures can be used to derive KBPs that provide discriminatory power in distinguishing between native-like and nonnative-like models. This measure of environment free energy is an important energy term but is best utilized as part of a more comprehensive energy evaluation function. For use in computational protein structure prediction, the neighbor vector algorithm provides the most optimal balance of accurate yet very rapid exposure measures. The assumption that environment free energy is directly proportional to SASA will be investigated further.

Acknowledgments The authors would like to thank all members of the Meiler Lab for helpful discussions and support. This work was supported by grant R01-GM080403 from the National Institute of General Medical Sciences and training grant 2-T15 LM07450-06 from the National Library of Medicine.

References

1. Baker D, Sali A (2001) Protein structure prediction and structural genomics. Science 294:93–96
2. Fang Y, Frutos AG, Lahiri J (2002) Membrane protein microarrays. J Am Chem Soc 124(11):2394–2395
3. Wiener MC (2004) A pedestrian guide to membrane protein crystallization. Methods 34(3):364–372
4. Alexander N et al (2008) *De novo* high-resolution protein structure determination from sparse spin-labeling EPR data. Structure 16(2):181–195
5. Jiang W et al (2001) Bridging the information gap: computational tools for intermediate resolution structure interpretation. J Mol Biol 308(5):1033–1044
6. Baker D, Sali A (2001) Protein structure prediction and structural genomics. Science 294(5540):93–96
7. Bourne PE (2003) CASP and CAFASP experiments and their findings. Methods Biochem Anal 44:501–507
8. Bradley P et al (2005) Free modeling with Rosetta in CASP6. Proteins 61(Suppl 7):128–134
9. Dill KA (1990) Dominant forces in protein folding. Biochemistry 29(31):7133–7155
10. Lazaridis T, Karplus M (2000) Effective energy functions for protein structure prediction. Curr Opin Struct Biol 10(2):139–145
11. Sippl MJ (1995) Knowledge-based potentials for proteins. Curr Opin Struct Biol 5(2):229–235
12. Juffer AH et al (1995) Comparison of atomic solvation parametric sets: applicability and limitations in protein folding and binding. Protein Sci 4(12):2499–2509
13. Boas FE, Harbury PB (2007) Potential energy functions for protein design. Curr Opin Struct Biol 17(2):199–204
14. Chen CT et al (2006) HYPLOSP: a knowledge-based approach to protein local structure prediction. J Bioinform Comput Biol 4 (6):1287–1307
15. Lu H, Skolnick J (2003) Application of statistical potentials to protein structure refinement from low resolution ab initio models. Biopolymers 70(4):575–584
16. Ferrada E, Melo F (2007) Nonbonded terms extrapolated from nonlocal knowledge-based energy functions improve error detection in near-native protein structure models. Protein Sci 16 (7):1410–1421
17. Casadio R et al (2007) Thinking the impossible: how to solve the protein folding problem with and without homologous structures and more. Methods Mol Biol 350:305–320
18. Simons KT et al (1997) Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. J Mol Biol 268(1):209–225
19. Audie J, Scarlata S (2007) A novel empirical free energy function that explains and predicts protein-protein binding affinities. Biophys Chem 129(2–3):198–211
20. Darnell SJ, Page D, Mitchell JC (2007) An automated decision-tree approach to predicting protein interaction hot spots. Proteins 68(4):813–823
21. Evers A, Gohlke H, Klebe G (2003) Ligand-supported homology modelling of protein binding-sites using knowledge-based potentials. J Mol Biol 334(2):327–346
22. Roche O, Kiyama R, Brooks CL 3rd (2001) Ligand-protein database: linking protein-ligand complex structures to binding data. J Med Chem 44(22):3592–3598
23. Gohlke H, Hendlich M, Klebe G (2000) Knowledge-based scoring function to predict protein-ligand interactions. J Mol Biol 295(2):337–356
24. Grzybowski BA et al (2002) From knowledge-based potentials to combinatorial lead design in silico. Acc Chem Res 35(5):261–269
25. Poole AM, Ranganathan R (2006) Knowledge-based potentials in protein design. Curr Opin Struct Biol 16(4):508–513
26. Isogai Y et al (2005) Design of lambda Cro fold: solution structure of a monomeric variant of the *de novo* protein. J Mol Biol 354 (4):801–814
27. DeBolt SE, Skolnick J (1996) Evaluation of atomic level mean force potentials via inverse folding and inverse refinement of protein structures: atomic burial position and pairwise non-bonded interactions. Protein Eng 9(8):637–655
28. Frishman D, Argos P (1995) Knowledge-based protein secondary structure assignment. Proteins 23(4):566–579
29. Domingues FS et al (1999) Sustained performance of knowledge-based potentials in fold recognition. Proteins Suppl 3:112–120
30. Koehl P, Delarue M (1994) Polar and nonpolar atomic environments in the protein core: implications for folding and binding. Proteins 20(3):264–278
31. Koehl P, Levitt M (1999) Structure-based conformational preferences of amino acids. Proc Natl Acad Sci USA 96(22):12524–12529
32. Ooi T et al (1987) Accessible surface areas as a measure of the thermodynamic parameters of hydration of peptides. Proc Natl Acad Sci USA 84(10):3086–3090
33. Viacarra C, Mayo S (2005) Electrostatics in computational protein design. Curr Opin Chem Biol 9(6):622–626
34. Pokala N, Handel TM (2004) Energy functions for protein design I: efficient and accurate continuum electrostatics and solvation. Protein Sci 13(4):925–936
35. Gordon DB, Marshall SA, Mayo SL (1999) Energy functions for protein design. Curr Opin Struct Biol 9(4):509–513
36. Street AG, Mayo SL (1998) Pairwise calculation of protein solvent-accessible surface areas. Fold Des 3(4):253–258
37. Lee B, Richards FM (1971) The interpretation of protein structures: estimation of static accessibility. J Mol Biol 55 (3):379–400
38. Shrake A, Rupley JA (1973) Environment and exposure to solvent of protein atoms. Lysozyme and insulin. J Mol Biol 79(2):351–371
39. Colloc'h N, Mornon JP (1990) A new tool for the qualitative and quantitative analysis of protein surfaces using B-spline and density of surface neighborhood. J Mol Graphics 8:133–140
40. Le Grand SM, Merz KM Jr (1992) Rapid approximation to molecular surface area via the use of boolean logic and look-up tables. J Comp Chem 14(3):349–352
41. Wodak SJ, Janin J (1980) Analytical approximation to the accessible surface area of proteins. Proc Natl Acad Sci USA 77(4):1736–1740
42. Pearl LH, Honegger A (1983) Generation of molecular surfaces for graphic display. J Mol Graphics 1(1):9–12
43. You T, Bashford D (1994) An analytical algorithm for the rapid determination of the solvent-accessibility of points in a three-dimensional lattice around a solute molecule. J Comp Chem 16 (6):743–757
44. Juffer AH, Vogel HJ (1998) A flexible triangulation method to describe the solvent-accessible surface of biopolymers. J Comput Aided Mol Des 12(3):289–299

45. Zhang N, Zeng C, Wingreen NS (2004) Fast accurate evaluation of protein solvent exposure. Proteins 57(3):565–576
46. Sanner MF, Olson AJ, Spehner JC (1996) Reduced surface: an efficient way to compute molecular surfaces. Biopolymers 38 (3):305–320
47. Stouten PFW et al (1993) An effective solvation term based on atomic occupancies for use in protein simulations. Mol Simul 10 (2–6):97–120
48. Rohl CA et al (2004) Protein structure prediction using Rosetta. Methods Enzymol 383:66–93
49. Karchin R, Cline M, Karplus K (2004) Evaluation of local structure alphabets based on residue burial. Proteins 55(3):508–518
50. Weiser J, Shenkin PS, Still WC (1999) Approximate solvent-accessible surface areas from tetrahedrally directed neighbor densities. Biopolymers 50(4):373–380
51. Eisenhaber F, Argos P (1996) Hydrophobic regions on protein surfaces: definition based on hydration shell structure and a quick method for their computation. Protein Eng 9(12):1121–1133
52. Flower DR (1997) SERF: a program for accessible surface area calculations. J Mol Graph Model 15(4):238–244
53. Humphrey W, Dalke A, Schulten K (1996) VMD: visual molecular dynamics. J Mol Graph 14(1):33–38 27–28
54. Wang GL, Dunbrack RL (2005) PISCES: recent improvements to a PDB sequence culling server. Nucleic Acids Res 33:W94–W98
55. Wang GL, Dunbrack RL (2003) PISCES: a protein sequence culling server. Bioinformatics 19(12):1589–1591
56. Carugo O, Pongor S (2001) A normalized root-mean-square distance for comparing protein three-dimensional structures. Protein Sci 10(7):1470–1473