PROTEINS
STRUCTURE ■ FUNCTION ■ BIOINFORMATICS

Coarse- and fine-grained models for proteins: Evaluation by decoy discrimination

Chris Kauffman,[1,2]* and George Karypis[2]

[1] Department of Computer Science, George Mason University, Fairfax, Virginia 22030

[2] Department of Computer Science, University of Minnesota, Minneapolis, Minnesota 55455

ABSTRACT

Coarse-grained models for protein structure are increasingly used in simulations and structural bioinformatics. In this study, we evaluated the effectiveness of three granularities of protein representation based on their ability to discriminate between correctly folded native structures and incorrectly folded decoy structures. The three levels of representation used one bead per amino acid (coarse), two beads per amino acid (medium), and all atoms (fine). Multiple structure features were compared at each representation level including two-body interactions, three-body interactions, solvent exposure, contact numbers, and angle bending. In most cases, the all-atom level was most successful at discriminating decoys, but the two-bead level provided a good compromise between the number of model parameters which must be estimated and the accuracy achieved. The most effective feature type appeared to be two-body interactions. Considering three-body interactions increased accuracy only marginally when all atoms were used and not at all in medium and coarse representations. Though two-body interactions were most effective for the coarse representations, the accuracy loss for using only solvent exposure or contact number was proportionally less at these levels than in the all-atom representation. We propose an optimization method capable of selecting bead types of different granularities to create a mixed representation of the protein. We illustrate its behavior on decoy discrimination and discuss implications for data-driven protein model selection.

Proteins 2013; 00:000–000.

Key words: protein decoy discrimination; coarse-grained models; n-body interactions; machine learning; protein model selection.

INTRODUCTION AND BACKGROUND

Modeling protein structures continues to garner great interest for its applications in drug discovery, disease study, and bio-products. However, it remains an extremely difficult task due to the large size of protein systems. Single protein systems involve thousands of atoms and even short runs of molecular dynamics require large computational resources.

A promising avenue to surmount this hurdle is to use coarse-grained (CG) models. The central idea is very simple: to avoid the cost of modeling all atoms, merge atoms into groups with a single interaction center. The merged object is referred to here as a *bead*. Appropriate merging choices should preserve most aspects of the physical system reasonably in the CG model while reducing the calculations required for simulations. Coarse-grained models are increasingly used in general molecular dynamics,[1] whereas a wide variety of CG models specific to proteins have been proposed to overcome the tremendous number of variables in these systems (see the thorough review by Tozzini[2]). Researchers have merged all atoms in a residue into a single bead or limited number of main and side chain beads since the inception of protein modeling.[3,4] Side chain interactions of proteins of particular importance leading some models such as SICHO use a single interaction center centered on the sidechain.[5] The popular and successful ROSETTA approach to protein structure prediction relies on a model in which all heavy atoms of the backbone are used but sidechain atoms are merged into a bead.[6] Recent years have seen the advent of other models such as the

Additional Supporting Information may be found in the online version of this article.

Grant sponsor: NSF; Grant numbers: IIS-0905220, OCI-1048018, IOS-0820730; Grant sponsor: DOE grant USDOE/DE-SC0005013 and the Digital Technology Center at the University of Minnesota.

*Correspondence to: Chris Kauffman, Department of Computer Science, George Mason University, 4400 University Drive MSN 4A5 Fairfax, VA 22030. E-mail: kauffman@cs.gmu.edu

Received 28 August 2012; Revised 12 November 2012; Accepted 14 November 2012

Published online 27 November 2012 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/prot.24222

Table I
The Combined Dataset of Decoys used for all Experiments

Set	Natives	Decoys	Total	Reference	Source
fisa	4	200	204	6	http://dd.compbio.washington.edu/
fisa3	5	250	255	6	http://dd.compbio.washington.edu/
4state	7	300	307	18	http://dd.compbio.washington.edu/
lattice	8	400	408	48	http://dd.compbio.washington.edu/
lmds	9	450	459	42	http://dd.compbio.washington.edu/
casp5	17	267	284	49	http://www.fiserlab.org/potentials/casp_decoys/
moulder	20	1000	1020	50,51	http://salilab.org/decoys/
casp6	24	447	471	49	http://www.fiserlab.org/potentials/casp_decoys/
tsai	30	1500	1530	52	http://depts.washington.edu/bakerpg/decoys/
casp7	34	755	789	49	http://www.fiserlab.org/potentials/casp_decoys/
rose	42	2100	2142	6	http://depts.washington.edu/bakerpg/decoys/
skol	47	2350	2397	43	http://cssb.biology.gatech.edu/amberff99
ro62	59	2950	3009	45,46	http://depts.washington.edu/bakerpg/decoys/
casp8	68	1159	1227	49	http://www.fiserlab.org/potentials/casp_decoys/
lkf	115	5318	5433	36	http://titan.princeton.edu/2010-10-11/Decoys/
Combined	415	19446	19861		

Proteins were drawn from 15 decoy sets generated by previous researchers. The columns are (Set) the decoy set, (Natives) the number of distinct native proteins in the set, (Decoys) the number of decoys in each set limited to 50 per native, (Total) the total structures in the set, (Reference) a citation describing the production of the decoy set, and (Source) the URL from which the decoy set was downloaded. Some native proteins belong to multiple decoy sets thus the Natives and Total column do not total to the the Combined row.

two-center per residue UNRES force field,[7,8] the three-center CABS approach,[9] and MARTINI force field which groups four heavy atoms together.[10] Kurkcuogu *et al.* showed that coarse-graining preserves the vibrational modes of two proteins even when reducing 5, 10, or 20 heavy atoms into a single interaction center.[11] They also explored using a fine-grained view of "interesting parts" of a protein while coarse-graining the remainder. The notion of a such a mixed representation is explored in the latter part of our work here.

Despite such attention, it is still not clear how much modeling accuracy is lost by switching to coarse representations. Part of the difficulty is that even coarse-grained models require heavy computation to perform molecular dynamics. Though such simulations are beginning to become tractable, it is still difficult to sample protein state space enough to evaluate a variety of models using dynamics. An example from a recent study of alpha-helical proteins using the coarse-grained UNRES force field found that for the 66-residue GCN4 protein, 4 of 10 simulation runs folded to near-native conformations for a total cost of around 99 h of CPU time[12] (Table II). Directly optimizing force field parameters using simulation is still largely out of reach.

An alternative vehicle for assessing protein models is through *decoy discrimination*. In this setting, one or more correctly folded proteins (natives) has associated with it incorrectly folded structures (decoys). The goal is to develop a scoring function that differentiates a native structure from its decoys. Since there are no dynamics involved, decoy discrimination is much cheaper as means to quickly evaluate models. One can also control the number of decoys and their characteristics directly if greater sampling of the state space is desired. Decoy discrimination has a long history in protein structure prediction and analysis. Scoring functions go by a variety of names including empirical force field, knowledge-based potential, and statistical potential. The idea is always to assign an extreme score to natives and the opposite extreme to decoys. If low scores are assigned to natives, the score can be interpreted as a kind of energy function due to the widely held belief that native structures are at the protein's global potential energy minimum.

In this work, we evaluate three levels of protein model granularity using decoy discrimination. At each granularity level, we assessed a variety of *feature types* including *n*-body interactions, solvent exposure, and dihedral angle bending. This gives insight into which features are informative at high versus low granularity and which may be discarded without affecting accuracy. For robustness, we used four different machine learning techniques to determine the model parameters. Comparing their relative performance illustrates aspects of linear versus nonlinear estimation and shows whether binary classification is a suitable means to determine model parameters. We adhered to a strict cross-validation methodology: models are assessed on a large dataset of 15 decoy sets and performance is measured only on structures that were not seen during parameter estimation. Two styles of cross-validation were used: balancing decoys amongst folds and leaving whole decoy sets out. Both make for a strong test of whether the models generalize and allows us to identify difficult decoy sets.

Finally, we propose a new method which can select bead types from a mixture of model granularities while maximizing the discrimination of native from decoys. This is a first step towards a data-driven method for protein model selection. We illustrate its behavior on the full set of decoys and explore how bead types from the different levels of granularity are combined.

Table II
Comparison of Linear and Nonlinear SVM learners

Feature	Level	Method	Rank	Top-1	Z-Score	Params
Two-body	res	svm	7.23 (1.11)	0.499 (0.029)	−2.09 (0.11)	1073
	res	svmrbf	6.66 (0.82)	0.549 (0.036)	−2.25 (0.07)	1073
	mc1	svm	3.50 (0.62)	0.771 (0.040)	−3.24 (0.07)	1189
	mc1	svmrbf	3.46 (0.71)	0.771 (0.032)	−3.29 (0.06)	1189
	t32	svm	2.62 (0.52)	0.889 (0.033)	−4.44 (0.24)	1584
	t32	svmrbf	2.57 (0.44)	0.896 (0.029)	−5.11 (0.34)	1584
(Two − and three-body	res	svm	7.52 (1.87)	0.410 (0.041)	−1.92 (0.08)	2682
	res	svmrbf	7.25 (1.70)	0.456 (0.033)	−2.01 (0.06)	2682
	mc1	svm	4.36 (0.50)	0.694 (0.019)	−2.81 (0.09)	3075
	mc1	svmrbf	5.16 (0.55)	0.634 (0.030)	−2.62 (0.14)	3075
	t32	svm	1.97 (0.36)	0.911 (0.020)	−4.25 (0.28)	7567
	t32	svmrbf	2.58 (0.53)	0.870 (0.025)	−4.15 (0.28)	7567
Contacts	res	svm	8.28 (0.89)	0.417 (0.050)	−1.74 (0.09)	212
	res	svmrbf	7.36 (0.80)	0.492 (0.075)	−1.96 (0.20)	212
	mc1	svm	4.17 (0.54)	0.730 (0.017)	−2.81 (0.09)	222
	mc1	svmrbf	4.92 (0.58)	0.655 (0.057)	−2.73 (0.13)	222
	t32	svm	3.69 (0.76)	0.781 (0.083)	−3.21 (0.21)	204
	t32	svmrbf	4.20 (1.04)	0.749 (0.061)	−3.20 (0.24)	204
Exposure	res	svm	7.62 (0.88)	0.409 (0.052)	−1.75 (0.11)	1266
	res	svmrbf	7.21 (0.64)	0.496 (0.049)	−2.31 (0.19)	1266
	mc1	svm	5.17 (0.82)	0.660 (0.037)	−2.54 (0.09)	1360
	mc1	svmrbf	4.82 (0.57)	0.687 (0.024)	−3.06 (0.17)	1360
	t32	svm	2.92 (0.85)	0.750 (0.045)	−2.98 (0.12)	1386
	t32	svmrbf	2.82 (1.00)	0.769 (0.025)	−4.95 (0.42)	1386
Angles	3 groups	svm	7.65 (0.88)	0.407 (0.030)	−1.62 (0.06)	598
	3 groups	svmrbf	6.12 (0.49)	0.492 (0.050)	−1.86 (0.08)	598
	20 groups	svm	7.76 (0.97)	0.511 (0.047)	−2.01 (0.10)	25,221
	20 groups	svmrbf	7.35 (1.00)	0.518 (0.054)	−2.06 (0.11)	25,221

Results of the 4CV experiment are given divided by the type of feature used (Feature), the level of representation (Level), and the discrimination method used to learn models (Method). The performance metrics Rank, Top-1, and Z-Score are described in Section Performance metrics. The mean across four cross-validation folds is given along with the standard deviation in parenthesis. The final column (Params) is the number of parameters in the row model.

MATERIALS AND METHODS

Dataset details

We combined decoys from 15 different sets of decoys that have been reported in literature, several of which are available from the Decoys R Us project.[13] The dataset is summarized in Table I. The number of decoys associated with each native in different decoy sets varies. To keep the size of data manageable, we limited the number of decoys per native per decoy set to 50 structures. For example, though there are more decoys available in it, we used only 50 decoy structures for each of the four natives in the fisa set giving a total of 200 decoys and 204 total structures. We sampled the 50 decoys from those available to give each native both high-RMSD decoys which were badly misfolded and low-RMSD decoys which resemble the native structure closely. In several decoy sets, such as casp sets, each native had fewer than 50 decoys in which case we used all decoys.

Some native proteins appear in several decoy sets. This is why adding each entry in the Natives column of Table I gives more than the 415 natives in the Combined row. The combined set was pruned to ensure that no identical proteins were present. The 415 proteins share less than 90% sequence identity with one another. Some close relatives were kept to keep the set as large as possible but the majority of entries share little sequence similarity: there are 376 sequence clusters using blastclust at the 30% sequence identity threshold.

Cross-validation

In cross-validation, the available data is divided into multiple folds. We used four-fold cross-validation so that in the experiments of Section Four-fold cross-validation experiment (4CV), we trained models on 3/4 of the proteins and tested the learned model on 1/4 of the proteins. This process was done four times with a different quarter of the data left out each time. Performance statistics were collected for each fold and their mean and standard deviation are given in the experimental results. For the results in Section Four-fold cross-validation experiment, each decoy set was evenly divided amongst the folds so that each fold had examples from every decoy set. We also performed

cross-validation where whole decoy sets were left out for a total of 15 folds (DCV). At each step, one whole decoy set such as fisa or ro62, was left out and all remaining data was used to estimate model parameters.

As noted in Section Four-fold cross-validation experiment, some natives are present in multiple decoy sets. During experiments, this handled in the following way. In the cross-validation experiments of Section Four-fold cross-validation experiment, whenever a particular native was selected for training, all decoys from all sets associated with that native were also used for training. For whole decoy cross-validation experiments in Section Four-fold cross-validation experiment, we divided the data on decoy sets. To test performance on a decoy set, the natives and decoys in it were removed from the training set. In addition, any decoys from different sets which were associated with a left out native were omitted from both training and testing. This prevents models from learning from any direct information on the test proteins.

Fine-, medium-, and coarse-grained representations

The first key design choice of an empirical forcefield is the type of body which will be represented. This choice has the largest impact on how accuracy will be traded for efficiency. We use the generic term *bead* when referring to an object in a protein representations. In the fine-grained model, beads are physical atoms while at coarser levels of representation several atoms are merged into a single bead. We use three granularities of models.

Fine-grained: t32

We adopted the model of Qiu and Elber which assigns all atoms to 32 types[14] and is referred to as the t32 representation. This set was chosen as the original study showed expanding to 46 types of atoms did not improve the discriminative power of the model and the t32 set prove quite robust on an evaluation of atomic and coarse-grained potentials to detect decoys using support vector machines by Zhang and Zhou.[15] An alternative would be the RAPDF/DFIRE set of 167 atom types which have been widely used.[16,17] These proved less effective in Zhang and Zhou's evaluation potentially due to the large number of parameters which must be estimated.

Medium-grained: mc1

The physical atoms of each residue were assigned to either the main-chain or side-chain giving each residue except glycine two beads. The barycenter (mean XYZ coordinate) of physical atoms in main-chain or side-chain groups determined the coordinates of each mc1 bead. This is an intermediate representation, more coarse than the atomic level but still allowing independent interaction centers for each residue. Each side-chain was assigned a type based on the amino acid. Glycine, alanine, and proline were treated specially: each was assigned a single interaction point specific to the residue. All other amino acids were assigned a specific side-chain atom and a generic main-chain atom. There are a total of 21 bead types in mc1. The mc1 model is similar to several prior models which use two beads per residue.[7,18,19]

Coarse-grained: res

All physical atoms in a residue were merged into a single bead at their barycenter. The res representation has 20 types of beads corresponding to each of the amino acids.

Types of features

After choosing a level of representation, a variety of structural features may be calculated for a protein. We explored a range of generic features that represent common energy terms in empirical forcefields.

Two- and three-body interaction features

Interactions between two bodies are the most prevalent features in empirical force fields, particularly for decoy detection. Most empirical force fields take a discretized approach to two-body interactions: distances between each bead pair are assigned to a distance bin and a separate parameter is associated to each bin and pair-type.

For the fine-grained t32 representation, we adopted the same three distance bins as the original study by Qiu and Elber:[14] 2.0–3.5, 3.5–5.0, and 5.0–6.5 Å. Atom pairs not in one of these distance ranges were ignored. There were $3 \times (32 \times (32 + 1))/2 = 1584$ features of this type. This forcefield appeared as t32S3 in the original and subsequent studies.[14,15] For the medium-grained mc1 and coarse-grained res types, we included two additional bins for a total of five bins: 2.0–3.5, 3.5–5.0, 5.0–6.5, 6.5–8.0, and 8.0–10.0Å. No attempt was made to optimize distance bins for predictive performance. The arbitrary nature of how bin cut-offs must be chosen is unsatisfactory and deserves further investigation into a more disciplined approach. These are refereed to as the *two-body features.*

Examples of a potential energy functions which calculate interactions higher than two were first explored in by Munson and Singh.[20] They analyzed two-, three-, and four-body potentials and found that four-body potentials explain patterns of four-body contacts in a statistically superior fashion to lower order interactions. However, two-body potentials recognized native sequence-structure pairs equally as well as three- and four-body potentials in threading, the only difference being better Z-scores in the higher body case. There has been some recent work ana-

lyzing four-body potentials mainly for their use in threading.[21–23] The number of parameters that must be estimated increases exponentially with n for n-body interactions. Estimating a large number of parameters given the limited number of native protein structures available can compromise the generality of such models. To assess whether this happens, we computed three-body interactions for the three representation levels. To avoid an explosion of parameters, three-body features used only a single distance bin: 2.0–6.5 Å; for t32 (fine-grained) and 2.0–10.0 Å; for mc1 and res (medium- and fine-grained). In our experiments, three-body features were always included in addition to the two-body features, the two-body having the distance bins described above. These are referred to as two- and three-body features.

Proteins represented by two-body and two- and three-body features are simply vectors of counts. However, longer proteins tend to have much larger total counts than their shorter counter-parts as they number of two-body interactions increases quadratically with the length of the protein. Data with drastically different scales tend to degrade the performance of machine learners we used. We adopted a simple normalization: count vectors were normalized by the number of atoms (t32) or pseudoatoms (mc1,res) in the protein. A similar normalization procedure was used in previous decoy studies.[15]

Contact (single-body) features

Rather than distinguish interactions by the types of both bodies, a forcefield may instead limit consideration how densely individual bodies are packed. Typically, this is done by counting the number of beads in a volume centered on a bead of one type. The count is attributed to the central bead type. This amounts to a sort of single-body energy as the types of the other beads are ignored. The density can correlate with a bead's placement at the surface of the protein (less crowded) or the interior (more crowded). Single-body potentials are referred to as "contact numbers" in some bioinformatics literature[24,25] and we follow that convention referring to the feature as *contacts*. We calculate contacts using the same bins as are used for two-body interactions above except that interactions count towards the total for both bodies (e.g., an alanine–arginine interaction counts towards both the contacts of alanine and arginine).

Solvent exposure features

Solvent accessibility was calculated for each bead by a sampling algorithm: 100 evenly spaced points were placed on the surface of each bead and were counted as buried if they were inside the radius of another bead and exposed if not. The fraction of exposed points was multiplied by the surface area of the bead to get the area.

The exposed areas were converted into discretized features using binning. Initially, we experimented with fixed bin widths but determining appropriate cutoffs for each type of bead proved difficult. Beads which agglomerated several physical atoms do not have established radii. Their radii must be estimated from the data. Some decoy structure contain unrealistic bond lengths which can make the maximum surface area for a bead-type abnormally large. In turn, this large maximum distorts binning based on the proportion of a beads surface area to the maximum observed. A more robust strategy is required.

We discretized by calculating the empirical distribution of the surface areas of a bead-type across the entire data set and used quantiles to determine bins. Beads are, therefore, evenly divided into the bins: if four bins are desired, each contains 25% of the beads. After examining the distribution for a number of bead types, it was not clear which number of bins was appropriate to use. Some bead types had complex distributions which would cause information loss if too few bins are used, whereas other distributions were flat requiring only a few bins to represent. To avoid information loss, we included multiple overlapping bins and allowed the feature selection to determine which were important for identifying decoys. We included 5, 10, and 20 bins for each bead type. This mixed-quantile strategy gave better performance than a fixed number of bins according to initial tests with glmnet and we report it as the *exposure* feature subsequently.

Angle features

Angles were generated by first examining all phi–psi angle pairs for each residue in the dataset (aside from N- and C-terminal residues). These were then clustered in two different ways. In the first, each residue was assigned to one of eight clusters of phi–psi angle pairs which were determined according to K-means clustering as implemented by the kmeans function of the R package stats.[26] Counts of cluster membership were used as features for each protein giving $8 \times 20 = 160$ features. In addition, we counted transitions between two angle states as giving $160^2 = 25600$ features. Since not every transition occurred in the data, there fewer angle features than the combined total of individual and transition clusters: on 25,221, total features were observed rather than 25,760. These are referred to as the angle features with *20 groups* of amino acids.

We noted that most amino acids adopt similar phi–psi angle distributions and can be grouped together. On looking at the distribution of clusters, only proline and glycine had significantly different cluster arrangements. To reduce the number of angle features, eight clusters of phi–psi angles were computed for proline, eight for glycine, and eight for the combination of all other residue types. Transitions between these 24 clusters were also

counted as for the 20 groups. A total of 204 single and transition features were used for the angle features with three *groups* of amino acids.

Our treatment of angular features was inspired by the clustering of angle states and transitions used by Zhang *et al.*[27] and Bahar et al.[28] Both used reduced alphabets of amino acids in determining angle states but favored the use of reduced model angles rather than phi–psi angles which require all atoms of the backbone to be present.

Discrimination methods

Once the level of representation has been set and the structural features selected, a method must be selected to determine parameters (feature weights) for the final model. For decoy discrimination, the goal is to establish a set of model parameters that differentiate native proteins from decoys. We assessed four methods for discrimination and parameter estimation.

Support vector machines (SVM)

A linear SVM learns a vector of parameters w to representing a separating hyperplane between the positive (native protein) and negative (decoy) classes. A nonlinear SVM also separates the positive and native classes but uses a kernel, in our case the radial basis function (RBF) kernel (svmrbf). The kernel allows nonlinear boundaries to be learned at the cost of not being able to determine the parameter vector w for the structural features. We used a customized R[26] interface to LIBSVM[29] to train SVM models. We used a grid of values for the SVM cost parameter C and RBF kernel parameter gamma during cross-validation and report the best performing models.

The SVMRANK package was used to generate linear ranking SVM models (svmrank).[30,31] We did not explore nonlinear ranking as the linear and nonlinear results on the standard SVM were similar and the computational requirements for nonlinear ranking problems is prohibitive. We tuned the SVM cost parameter for svmrank over a grid of values and report the best result.

Penalized regression models

A penalized logistic regression model is learned by optimizing the following.

$$\max_{w} \sum_{i=1}^{N} \left[y_i w^T x_i - \log(1 + e^{w^T x_i}) \right] - \lambda \left[(1-\alpha)||w||_2 + \alpha||w||_1 \right] \quad (1)$$

The left term represents the loss of the model and is the conditional log-likelihood of observing the entire data set of size N with features x_i and classes y_i. The right term is regularizer. As in the SVM models, the end result of a logistic regression model is a vector w of feature weights. As the penalty parameter λ is increased, elements of w shrink to 0 which allows feature selection to be done. The α parameter controls the relative L1 and L2 penalty on the model. We set $\alpha = 0.9$ which introduces a small amount of L2-regularization on feature selection along with L1-regularization. This was found to improve overall performance. The glmnet R package was used to train L1-penalized logistic regression models.[32] This package efficiently solves for all levels of the penalty parameter λ. We used 10-fold internal cross-validation with evaluation based on the area under the ROC curve to determine the optimal λ value for each model.

Decoy separation with bead selection (BSM)

The glmnet method performs feature selection but it has the following limitation. For two-body features, individual pair-wise parameters such as R.res-A.res (a res level interaction) may be driven to zero. However, in another parameter associated with R.res, such as R.res-C.res is nonzero, C.res still plays a part in the model. Rarely does glmnet drive all parameters associated with a bead type to zero simultaneously. As long as some pairwise interactions are nonzero for a bead type, it cannot be dropped.

We surmounted this limitation by designing a method which discriminates natives from decoys while doing bead selection. As it is a *bead selection method*, we refer to as *BSM*. BSM is designed to simultaneously drive all parameters associated to a bead type to zero together thereby allowing the bead to be eliminated.

As an optimization problem, BSM takes the following form. The vector of parameters or feature weights w and must be chosen so that the decoys have higher energy than natives. Formally, this is

$$w^T x_{\text{decoy}_i} - w^T x_{\text{native}} > 0 \quad (2)$$

where x are the feature vectors for a decoy and its associated native structure. We constructed an $n \times f$ decoy matrix D which is the difference of feature vectors between each decoy and its corresponding native protein. The rows of D are the term on the left-hand side of Eq. (2); columns correspond specific feature differences. The matrix vector product Dw gives a vector of the energy differences between decoys and natives. In this formulation, we only compare natives to their associated decoys as in svmrank. Also as in support vector machine approaches, we used the hinge loss to encourage a large energy gap between decoys and associated natives. The hinge loss is $h(z) = \max(0, 1 - z)$ and when z is a vector, it produces a positive vector. It reaches a minimum of zero when input z is 1 or greater. The loss function is denoted $h(Dx)$: any decoy not exceeding the native in energy by 1 unit has nonzero loss. To balance the loss function, we applied regularization to the parameters w. This took a

special form where the penalty applied to groups of variables associated with a single bead type. In the case of two-body interactions, each feature was associated with two bead types such as the interaction of $w_{\mathrm{CAH}\ -\ R.sc}$ where bead types CAH and R.sc are from the t32 and mc1 representations, respectively. For a particular bead type A, we compute the maximum absolute value, $\max_X |w_{A\ -\ X}|$, where X can be any bead type. This coefficient is penalized during parameter estimation. As in glmnet, the absolute value or L1-penalty induces sparsity in parameters driving some of w to zero. The max or L-∞ norm has also been used in literature for regularization, and their combination has come under some scrutiny recently.[33]

The final form of our optimization problem is then

$$\min_w h(Dw) + \lambda \sum_A \max_X |w_{A-X}| \qquad (3)$$

where the fixed parameter λ governs the trade-off between loss and regularization. The problem is convex so that it has a global minimum but nonsmooth due the hinge, L1, and L-∞ loss. We explored several methods solve the optimization problem for BSM. Equation (3) can easily be cast as a linear program, but standard LP solvers have memory requirements that scale quadratically with the problem size. In our situation, we are using a mixture of all two-body features from the res, mc1, and t32 representations so that D is large and dense, around 20K by 10K with 45% nonzero entries. This proved to much for standard solvers. Coordinate descent is another reasonable choice as it is used to great effect in approaches such as glmnet.[32] However, careful analysis of Eq. (3) reveals that the regularization term is nonseparable. In such cases, coordinate descent is not guaranteed to converge.[34] Instead we used the subgradient descent method which is very general but suffers from limited accuracy and speed.[35] In our case, tractability and solvabilty out-weight speed concerns.

In Eq. (3), we started λ at a very large value which drives the entire parameter vector w to zero and gradually reduced the magnitude of λ. This is identical to the regularization path approach of glmnet in that bead types will enter the model by becoming nonzero at different points along the path. We used 2500 subgradient steps at each value of λ. Step sizes were reduced in the subgradient method using $1/\sqrt{k}$, where k was the subgradient iteration.

Performance metrics

Decoy discrimination is an interesting problem from the machine learning standpoint as it is always unbalanced: for every positive instance which is the native protein structure there are potentially many negative instances which are misfolded. Performance is measured only on the ability to identify from amongst a pool of structure for a single protein the single native structure (or closest to native). For that reason, typical classification metrics such as ROC are unsuitable. We used several metrics commonly used in other decoy discrimination literature.

Mean native rank (rank)

The native and associated decoy proteins are ranked by their prediction score and the rank of the native is taken. In cross-validation, we report the mean of these ranks. A lower rank is better with mean native rank of 1 being the perfect prediction.

Top-1 fraction

In a given set of natives and decoys, we report the fraction of natives that are ranked higher than all their associated decoys (those that have native rank of 1). A higher Top-1 Fraction is better with 1.0 being the perfect.

Z-score

The native protein structure is believed to have a lower free energy than misfolded decoys. Interpreting the prediction scores produced by an SVM or glmnet method as an energy, the Z-score is defined

$$Z = \frac{\mu_{\mathrm{decoy}} - E_{\mathrm{native}}}{\sigma_{\mathrm{decoy}}} \qquad (4)$$

where μ_{decoy} and σ_{decoy} are the mean and standard deviation of the decoy prediction scores and E_{native} is the prediction score for the native protein. A larger more negative Z-score corresponds to better separation of decoys from natives.

RESULTS

Four-fold cross-validation experiment (4CV)

Proteins were represented at the coarse *res* level, medium *mc1* level, and fine-grained *t32* level to determine trade-offs associated with each representation. At each of these levels, features were calculated for each protein including two-body interactions, two- and three-body interactions (called two- and three-body, *contact* counts (or one-body interactions), and solvent *exposure*. Proteins were also represented using only their backbone *angle* data grouping residues into 3 or 20 groups for angle binning. Each representation/feature combination has a number of model parameters associated with it which may be set to discriminate native from decoy proteins. Section Fine-, medium-, and coarse-grained representation describes the representation level and Section Types of features describes structure features.

We considered four methods to fit model parameters: linear support vector machine training (*svm*), nonlinear support vector machine training (*svmrbf*), ranking support vector training (*svmrank*), and penalized logistic regression (*glmnet*). These are primarily classification methods which learn parameters to discriminate between two classes, in our case native and decoy structures. Section Discrimination methods gives details of these methods.

In our first experiment, 415 proteins with associated decoys (total 19,861 structures) were divided into four folds, each fold having a balanced number of proteins from each decoy set. We refer to this experiment as *four-fold cross-validation* (4CV). At each step, three folds were used for training and the remaining fold was used for evaluation. Performance is averaged over the four folds. The results are used to compare aspects of the parameter learning models and also evaluate the viability of each type of feature in each representation. The comparison is done based on the mean rank of the native structure (*Rank*), the fraction of all natives ranked in the top position (*Top-1*), and the *Z-score* which gives a normalized score (or energy) separation between natives and decoys. These performance measures are detailed in Section Performance metrics.

Linear versus nonlinear classification

We first focused on linear and nonlinear SVMs (*svm and svmrbf*). Table II compares svm and svmrbf in the four-fold cross-validation experiment. The two classifiers have very similar performance. Of particular note are the two-body results in the top section of Table II as they are most directly comparable to the results from Dong and Zhou.[15] With two-body interaction features, we see a small benefit at the residue-level representation for using a nonlinear kernel, but at finer-grained representations there is little to no benefit over the linear version of SVM. This trend is also present in the two-and three- body interactions and the contact/one-body interactions: some benefit is given at the coarsest representation level by using svmrbf but no such benefit is present at the finer mc1 and t32 levels. Solvent exposure features follow this trend but to a weaker extent with svmrbf only slightly out-performing svm at each level of granularity. Finally, angle data definitely benefits from the nonlinear SVM though it is comparatively a weak feature for identifying decoys.

The near equivalence of linear and nonlinear SVMs (svm and svmrbf) conflicts with earlier work which indicates linear SVMs or inferior to their nonlinear counterparts[15] Our best explanation for this difference is that experiments in the previous work were restricted to single decoy sets for training and testing. For example, the two cross-validation experiments were done within the LKF and CASP7 datasets separately. Since decoy sets vary greatly in how the structures are generated, it is possible that characteristics of those datasets lent themselves to nonlinear separation. However, the model learned does not transfer to a decoy set with different characteristics. The experiment in which potentials were transferred to new decoy sets in Ref. [15] (Table IV) indicated that the linear and nonlinear potentials behave similarly on truly new data. The issue of how well any potential can be applied to a truly new set of decoys is taken up in Section whole decay set cross-validation experiment (DCV). T4

Despite their slightly superior performance on a few of the protein representations, there is a major disadvantage of nonlinear SVM models. Both linear and nonlinear SVMs tend to learn classification models based on support vectors which are simply specific training examples of some importance. In the linear case, through simple algebraic operations, the parameters for each feature can be recovered so we may know how each interaction affects the likelihood of being a decoy. This is not so for nonlinear SVMs: they learn a model that is implicitly embedded in a higher dimensional space (infinite dimensional in the case of the svmrbf) which makes it very difficult to relate features in the original space to the likelihood of a protein being native or decoy. Due to this difficulty in interpretation and the fact that only marginal performance gains come from using a nonlinear kernel, we omit svmrbf from further discussion.

Regularized logistic regression vs. SVM classification

With the number of features in representations ranging from 204 to 25,221, there is potential to over-fit parameters to training data which decreases the generalization of a model. A regularized method such as glmnet is designed to avoid this by charging a cost for the inclusion of any feature while learning model. Such method tend to generate sparse models with zero parameters associated to many features. SVMs do not encourage sparse models explicitly.

Table III and Figure I show a comparison of the performance of the linear SVM against the regularized logistic regression classifier glmnet. The svmrank classifier in this table is discussed later. Included in Table III are the number of parameters in the model (*Params*) which is also the number of structure features, how many parameters were nonzero, (*Selected*), and the fraction of nonzero parameters (*Frac.*). Also present are measures of model stability amongst the four cross-validation folds: the correlation of parameters learned and the overlap of nonzero parameters. T3

In all representations, the effectiveness of regularization is apparent. The glmnet method performed equal to or better than svm while simultaneously selecting a relatively small number of important features. While svm tended to provide a slightly better Z-score than glmnet, glmnet dominated svm in providing a better mean native rank and top-1 fraction for natives.

Table III
Comparison of Methods, Representations, and Features on four-fold cross validation (4CV)

Feature	Level		Method	Rank	Top-1	Z-score	Params	Nonzero	Frac.	Correlation	Overlap
Two-body	res	1	glmnet	6.45 (1.18)	0.532 (0.040)	−2.19 (0.10)	1073	485 (95)	0.452	0.790 (0.022)	0.820 (0.046)
		2	svm	7.23 (1.11)	0.499 (0.029)	−2.09 (0.11)	1073	1054 (12)	0.982	0.720 (0.010)	0.994 (0.004)
		3	svmrank	6.65 (0.99)	0.549 (0.039)	−2.24 (0.04)	1073	1064 (12)	0.992	0.777 (0.016)	1.000 (0.000)
	mc1	4	glmnet	2.96 (0.53)	0.771 (0.027)	−3.15 (0.08)	1189	650 (26)	0.547	0.811 (0.006)	0.822 (0.014)
		5	svm	3.50 (0.62)	0.771 (0.040)	−3.24 (0.07)	1189	1154 (14)	0.971	0.804 (0.009)	0.989 (0.002)
		6	svmrank	3.08 (0.24)	0.785 (0.035)	−3.26 (0.23)	1189	1176 (16)	0.989	0.786 (0.018)	0.997 (0.002)
	t32	7	glmnet	1.69 (0.50)	0.920 (0.017)	−4.34 (0.16)	1584	845 (113)	0.533	0.805 (0.021)	0.842 (0.029)
		8	svm	2.62 (0.52)	0.889 (0.033)	−4.44 (0.24)	1584	1577 (2)	0.996	0.937 (0.009)	0.999 (0.001)
		9	svmrank	1.93 (0.53)	0.920 (0.015)	−4.44 (0.13)	1584	1578 (2)	0.996	0.830 (0.036)	0.999 (0.001)
Two - and three-body	res	10	glmnet	6.92 (1.48)	0.470 (0.035)	−2.12 (0.16)	2682	774 (387)	0.289	0.680 (0.075)	0.738 (0.091)
		11	svm	7.52 (1.87)	0.410 (0.041)	−1.92 (0.24)	2682	2652 (48)	0.989	0.688 (0.011)	0.999 (0.000)
		12	svmrank	6.97 (0.86)	0.482 (0.021)	−2.11 (0.13)	2682	2657 (46)	0.991	0.742 (0.014)	0.999 (0.000)
	mc1	13	glmnet	3.59 (0.17)	0.713 (0.032)	−2.80 (0.11)	3075	1530 (328)	0.498	0.706 (0.024)	0.817 (0.036)
		14	svm	4.36 (0.50)	0.694 (0.019)	−2.81 (0.09)	3075	3031 (75)	0.986	0.747 (0.007)	0.999 (0.001)
		15	svmrank	4.02 (0.30)	0.682 (0.009)	−2.76 (0.18)	3075	3037 (74)	0.988	0.738 (0.019)	1.000 (0.000)
	t32	16	glmnet	1.56 (0.45)	0.911 (0.021)	−4.08 (0.19)	7567	2220 (125)	0.293	0.761 (0.025)	0.733 (0.022)
		17	svm	1.97 (0.36)	0.911 (0.020)	−4.25 (0.28)	7567	7538 (5)	0.996	0.827 (0.010)	0.998 (0.000)
		18	svmrank	2.17 (0.72)	0.908 (0.025)	−4.24 (0.24)	7567	7562 (4)	0.999	0.822 (0.033)	1.000 (0.000)
Contacts	res	19	glmnet	7.13 (1.63)	0.472 (0.077)	−1.97 (0.23)	212	189 (17)	0.891	0.490 (0.328)	0.976 (0.028)
		20	svm	8.28 (0.89)	0.417 (0.050)	−1.74 (0.09)	212	212 (0)	1.000	0.581 (0.075)	1.000 (0.000)
		21	svmrank	6.64 (0.87)	0.499 (0.054)	−2.06 (0.19)	212	212 (0)	1.000	0.806 (0.032)	1.000 (0.000)
	mc1	22	glmnet	3.82 (0.37)	0.730 (0.049)	−2.83 (0.12)	222	192 (18)	0.865	0.737 (0.119)	0.973 (0.025)
		23	svm	4.17 (0.54)	0.730 (0.017)	−2.81 (0.09)	222	222 (0)	1.000	0.850 (0.018)	1.000 (0.000)
		24	svmrank	3.33 (0.61)	0.742 (0.054)	−2.80 (0.28)	222	222 (0)	1.000	0.752 (0.034)	1.000 (0.000)
	t32	25	glmnet	3.35 (0.56)	0.771 (0.068)	−3.15 (0.12)	204	160 (34)	0.784	0.456 (0.272)	0.970 (0.018)
		26	svm	3.70 (0.76)	0.781 (0.083)	−3.21 (0.21)	204	204 (0)	1.000	0.829 (0.012)	1.000 (0.000)
		27	svmrank	2.95 (0.45)	0.769 (0.040)	−3.14 (0.19)	204	204 (0)	1.000	0.796 (0.045)	1.000 (0.000)
Exposure	res	28	glmnet	6.66 (1.03)	0.491 (0.063)	−2.03 (0.16)	1266	148 (16)	0.117	0.742 (0.029)	0.691 (0.040)
		29	svm	7.62 (0.88)	0.409 (0.052)	−1.75 (0.11)	1266	1266 (0)	1.000	0.606 (0.025)	1.000 (0.000)
		30	svmrank	6.71 (0.58)	0.499 (0.053)	−2.18 (0.17)	1266	1259 (1)	0.995	0.877 (0.009)	1.000 (0.001)
	mc1	31	glmnet	4.55 (1.11)	0.696 (0.043)	−2.74 (0.13)	1360	156 (18)	0.115	0.788 (0.024)	0.716 (0.038)
		32	svm	5.18 (0.82)	0.660 (0.037)	−2.54 (0.09)	1360	1360 (0)	1.000	0.754 (0.015)	1.000 (0.000)
		33	svmrank	4.12 (0.67)	0.641 (0.067)	−2.63 (0.18)	1360	1352 (1)	0.994	0.809 (0.016)	1.000 (0.001)
	t32	34	glmnet	2.56 (0.91)	0.778 (0.027)	−3.10 (0.12)	1386	183 (20)	0.132	0.740 (0.049)	0.712 (0.037)
		35	svm	2.92 (0.85)	0.750 (0.045)	−2.98 (0.12)	1386	1386 (0)	1.000	0.707 (0.020)	1.000 (0.000)
		36	svmrank	2.32 (0.70)	0.807 (0.028)	−3.26 (0.08)	1386	1370 (2)	0.989	0.755 (0.051)	1.000 (0.001)
Angles	3 groups	40	glmnet	7.33 (0.49)	0.487 (0.033)	−1.86 (0.10)	598	461 (6)	0.771	0.795 (0.026)	0.923 (0.007)
		41	svm	7.65 (0.88)	0.407 (0.030)	−1.62 (0.06)	598	503 (3)	0.841	0.677 (0.020)	0.978 (0.006)
		42	svmrank	7.35 (0.41)	0.525 (0.027)	−2.08 (0.10)	598	542 (5)	0.906	0.766 (0.014)	0.988 (0.006)
	20 groups	37	glmnet	8.13 (0.82)	0.523 (0.029)	−1.97 (0.08)	25,221	6916 (2833)	0.274	0.647 (0.027)	0.835 (0.049)
		38	svm	7.76 (0.97)	0.511 (0.047)	−2.01 (0.10)	25,221	15530 (136)	0.616	0.671 (0.012)	0.873 (0.004)
		39	svmrank	7.85 (0.99)	0.561 (0.017)	−2.04 (0.12)	25,221	20511 (188)	0.813	0.731 (0.019)	0.936 (0.005)

The first series of columns are identical to Table II. The rightmost columns give statistics on the models learned. They are (Nonzero) the mean number of nonzero parameters in the row model, (Frac.) the fraction of nonzero parameters, (Correlation) the mean Pearson correlation coefficient between the parameter vectors of the four models, and (Overlap) the mean fraction of parameters which are nonzero in pairs of models. Standard deviations are given in parentheses.

The Params, Nonzero, and Frac. columns of Table III give information on the size of the models learned in each case. The Nonzero column gives the average number of nonzero parameters in the model and Frac. relates this to the total possible number of nonzeros which is Params. The tendency of SVMs to produce dense models is apparent as in nearly all representations a large fraction of parameters are nonzero. Conversely, glmnet produced relatively sparse models everywhere except when the number of features was small (contacts and angles with three groups).

Table IV
Overall Best Method for Each Representation and Feature

	Feature	Level	Method	Rank	Top-1	Z-score	Params	Selected
1	Two - and three-body	t32	glmnet	1.56 (0.45)	0.911 (0.021)	−4.08 (0.19)	7567	2220 (125)
2	Two-body	t32	glmnet	1.69 (0.50)	0.920 (0.017)	−4.34 (0.16)	1584	845 (113)
3	Exposure	t32	svmrank	2.32 (0.70)	0.807 (0.28)	−3.26 (0.08)	1386	1370 (2)
4	Contacts	t32	svmrank	2.95 (0.45)	0.769 (0.040)	−3.14 (0.19)	204	204 (0)
5	Two-body	mc1	glmnet	2.96 (0.53)	0.771 (0.027)	−3.15 (0.08)	1189	650 (26)
6	Contacts	mc1	svmrank	3.33 (0.61)	0.742 (0.054)	−2.80 (0.28)	222	222 (0)
7	Two - and three-body	mc1	glmnet	3.59 (0.17)	0.713 (0.032)	−2.80 (0.11)	3075	1530 (328)
8	Exposure	mc1	svmrank	4.12 (0.67)	0.641 (0.067)	−2.63 (0.18)	1360	1352 (1)
9	Three groups	angles	svmrbf	6.12 (0.50)	0.492 (0.050)	−1.86 (0.08)	598	—
10	Two-body	res	glmnet	6.45 (1.18)	0.532 (0.040)	−2.19 (0.10)	1073	485 (95)
11	Contacts	res	svmrank	6.64 (0.87)	0.499 (0.054)	−2.06 (0.19)	212	212 (0)
12	Exposure	res	glmnet	6.66 (1.03)	0.491 (0.063)	−2.03 (0.16)	1266	148 (16)
13	Two - and three-body	res	glmnet	6.92 (1.48)	0.470 (0.035)	−2.12 (0.11)	2682	774 (387)
14	Twenty groups	angles	svmrbf	7.35 (1.00)	0.518 (0.054)	−2.06 (0.11)	25,221	—

Columns are identical to those given in Table III. The rows are ordered by the Rank performance statistic from best to worst.

The two rightmost columns of Table III give information on the stability of the learned models by giving the mean correlation of parameters and the fraction of overlap of selected features amongst the four models learned during cross-validation. Both glmnet and SVM tend to produce fairly stable models and despite glmnet selecting a small fraction of features, there is a high degree of overlap of those selected different data is left out. This will be discussed further in the results of the cross-training experiment.

Binary classification and grouped separation

The learning paradigm exercised by binary classifiers like SVMs and logistic regression is to distinguish all native proteins from all decoys. This is done by assigning model parameters that give a lower score to all native proteins than any decoy protein. Technically, this formulation is more restrictive than needed as in reality, we should only require a native structure to be lower in energy than its associated decoys, not the decoys of a different protein. For example, it may be difficult for a binary classifier to assign model parameters such that a very large native structure has a lower energy than a much smaller decoy that is close to its native structure. We have used normalization on the sizes of proteins which may mitigate this to some extent. However, it is still interesting to examine what happens when we relax the requirement that all natives are lower in energy than all decoys. We will refer to these two formulations as the *binary classification* formulation and the *grouped separation* formulation. The grouped separation approach has a longer history with many recent examples [14,36–38] whereas the advent of machine learning in structural biology has led to the classification approach receiving some attention.[15,27,39] Grouped separation is typically solved using algorithms for linear programming, whereas the binary classification problem is usually addressed with one of a plethora of machine learning tools.

To investigate the merits of the grouped separation model, we used a ranking SVM (svmrank) in the same four-fold cross-validation framework as the svm and glmnet (both binary classifiers). The ranking SVM learns a model in which data are grouped and parameters are sought to create a desired ranking within each group. In our case, the groups were the 415 proteins and the members of each group were a native along with all decoys associated with that native. In each group, the native was to be ranked lower in energy than the decoys, but there was no penalty for ranking a native higher in energy than a decoy in a different group. This was a relaxation over the svm and glmnet methods which did penalize ranking a native above a decoy in a different group.

The performance of the ranking SVM is reported in Table III as *svmrank* along with the svm and glmnet. In most cases, svmrank improves slightly over the performance of svm and approaches the accuracy of glmnet. On the contact, exposure, and angle features svmrank produces a better mean native rank and top-1 fraction than svm and glmnet. The comparison illustrates an important point: standard binary classification restricts parameter estimation unnecessarily for decoy discrimination. This is important in situations where the protein is represented using a limited number of structure features used as in the case of contacts (204–222 features) where the additional flexibility of svmrank led to improvement in the mean native rank statistic.

The svm method was out-performed by both glmnet, a binary classification with regularization, and by svmrank, a group separation method with no regularization. Estimating parameters using both regularization to induce sparsity and the grouped separation formulation for flexibility could result in robust estimates. To our knowledge, there are no machine learning methods that specifically address this formulation. We incorporated grouped discrimination

Table V
Average Number of Nonzero Parameters for two - and three-body features Determined by glmnet During 4CV

Level	Two-body	Two - and three-body	Total
res	239/1073	534/1609	773/2682
mc1	509/1189	1021/1886	1530/3075
t32	480/1584	1740/5983	2220/7567

Parameters are divided by type (two-body or three-body) and the possible nonzero parameters is given.

and regularization into our optimization method for mixed representation selection which is discussed in Section Mixed model model selection experiment.

Comparison of representation levels

A primary concern of this study is to examine how the granularity of a protein representations affects the accuracy it can achieve on some task, in our case decoy identification. Table IV shows the best mean native rank achieved by any method in four-fold cross-validation. The table is sorted by the mean native rank (the top-1 fraction and z-score follow nearly the same ordering).

As expected, accuracy strongly correlates with the granularity. The fine-grained t32 atomic features occupy the highest accuracy slots, whereas the coarse-grained res and angle features are at the bottom of the table. There appears to be great promise in using the mc1 representation or coarse-grained models akin to it. The t32 representation uses all atoms and at best identifies 92% of natives using two-body interactions (line 2 of Table IV). Alternatively, mc1 uses a maximum of two beads per residue and gets 77% of natives correct using two-body interactions (line 5). Between levels, this is a $\frac{1584-1189}{1584} = 25\%$ reduction in parameters for a $\frac{0.920-0.771}{0.920} = 16\%$ decrease in performance. Using a single interaction point per residue in res representation gives a larger drop, down to a best top-1 fraction of 53% using two-body interactions. This is a smaller step in parameter reduction ($\frac{1189-1073}{1189} = 10\%$) for a larger drop in accuracy ($\frac{0.771-0.532}{0.771} = 31\%$). The best mean native rank approximately doubles between representations: 1.56 at t32, 2.96 at mc1, and 6.45 at res. These together indicate that the models coarser than two beads per residue will be greatly handicapped in approximating the protein structure.

There appears to be little to no benefit from using two- and three-body interactions over two-body interactions. Only at the t32 level is a slight benefit observed, while at coarser granularity no such benefit occurs. This casts a dim picture on the utility of considering higher body interactions despite their use in recent studies.[21–23] However, there are many ways to construct higher body features and our method, grouping all three-body interactions into a single distance bin, may not be optimal for the task of decoy discrimination. Our choice was based on a desire to prevent the feature space from becoming intractably large while retaining informative interactions but we may have lost some key three-body information with our binning procedure. It is essential that three-body interactions show significant generalization in a test set. Table V shows the nonzero parameters in the (two-and three-)-body model selected by glmnet. At all three levels, three-body features are selected with nonzero weights indicating that in the training set they appeared discriminative. The (two-and three-)-body models do badly on the test sets at the res and mc1 in cross-validation, badly at least compared to their two-body counterparts. This indicates that many three-body features do not generalize well and the training set sizes are not large enough to properly identify this fact. Further development of higher body features will require careful validation to ensure that they do not suffer from over-fitting.

Coarse-grained, two-body potentials have long been used in protein structure analysis but recent work by Pokarowski *et al.* has shown that many published two-body interaction potentials are essentially the sum of one-body energies.[10,11] Our use of contacts, which are one-body potentials, serves as a performance validation of that work. Note that when creating the feature vector for a protein, observing beads for alanine and arginine between 2.0 and 3.5Å; apart has the following effect: for a two-body potential the count on feature A_R_2–3.5 is increased by 1; for one-body potentials, the count on feature A_2–3.5 is increase by 1 as is the count on feature R_2–3.5. When the machine learner determines parameters for the two-body potential, it assigns a single weight to A_R_2–3.5 count which is multiplied by the count and added to the total score. This weight is distinct from other two-body features such as A_G_2–3.5. In the one-body case, parameter weights are set for both the count of A_2–3.5 and the count of R_2–3.5 separately and their sum contributes to overall score. This additivity is like a constraint that any A-R interactions are the sum of two one-body paramters associated with A and R. For that reason, the "contacts" feature is equivalent to the Pokarowski's reduction of two-body terms to sums of one-body terms. Their results indicate that one-body terms should do equally well to two-body terms in prediction tasks. Pokarowski et al. examined coarse-grained potentials (our res level) and used only a single distance bin for the potentials. Our results in Table III show at the res level that contacts (one-body interactions) have nearly the same performance (mean rank 6.64) as two-body features (mean rank 6.45). This is in good agreement with the notion that the coarse interaction of two residues is essentially the sum of two one-body terms. Results at the mc1 level are similar: two-body features achieved mean rank 2.96, whereas one-body features were close at mean rank 3.33. These findings expand on Pokarowski and coworkers studies in that they are a true illustration of the predictive power of one- versus two-body potentials and

Table VI
Results of leaving whole decoy sets out (DCV)

	Rank			Top-1			Z-score					Cor_{all}		
Data	res	mc1	t32	res	mc1	t32	res	mc1	t32	N	Mam	res	mc1	t32
fisa	4.000	9.750	1.500	0.750	0.750	0.750	−3.322	−3.438	−4.011	204	8.39	0.992	0.990	0.989
fisa3	3.000	2.600	1.000	0.600	0.800	1.000	−2.375	−3.633	−7.021	255	7.86	0.996	0.997	0.994
4state	2.857	1.714	2.000	0.571	0.714	0.429	−2.241	−2.931	−2.684	307	9.03	0.989	0.988	0.976
lattice	8.375	1.625	1.125	0.375	0.875	0.875	−2.369	−4.416	−4.442	408	7.13	0.985	0.988	0.972
lmds	11.778	7.000	7.667	0.222	0.667	0.556	−1.788	−2.807	−2.414	459	7.36	0.977	0.981	0.962
casp5	1.882	1.118	1.000	0.765	0.882	1.000	−1.950	−3.106	−3.520	284	11.09	0.994	0.994	0.989
moulder	2.800	2.900	1.000	0.550	0.850	1.000	−2.379	−3.497	−5.049	1020	10.96	0.978	0.989	0.996
casp6	3.083	1.250	1.042	0.458	0.833	0.958	−1.577	−2.753	−3.560	471	9.19	0.990	0.991	0.986
tsai	19.900	8.233	4.867	0.067	0.333	0.433	−0.353	−1.678	−2.421	1530	7.30	0.924	0.943	0.922
casp7	2.029	1.353	1.000	0.529	0.765	1.000	−1.672	−2.388	−3.101	789	13.74	0.986	0.983	0.983
rose	8.262	2.905	2.119	0.357	0.619	0.929	−1.743	−3.371	−5.654	2142	9.03	0.930	0.934	0.936
skol	13.809	7.787	5.128	0.213	0.426	0.447	−1.103	−1.825	−2.358	2397	8.24	0.904	0.913	0.916
ro62	12.017	8.458	2.593	0.254	0.441	0.814	−1.359	−1.823	−3.467	3009	8.77	0.914	0.898	0.929
casp8	1.559	1.088	1.000	0.647	0.941	1.000	−2.097	−2.965	−3.896	1227	10.31	0.968	0.976	0.987
lkf	3.070	2.130	2.139	0.739	0.817	0.826	−3.598	−4.149	−4.894	5433	8.50	0.884	0.843	0.804
Cor_N	0.205	0.127	0.148	−0.076	−0.300	−0.086	−0.133	0.027	−0.081	1.000	−0.118	−0.919	−0.979	−0.933
Cor_{Mam}	−0.581	−0.474	−0.498	0.434	0.376	0.535	−0.020	0.111	0.039	−0.118	1.000	0.277	0.226	0.312
Mean DCV	6.561	3.994	2.345	0.473	0.714	0.801	−1.995	−2.985	−3.899			0.961	0.961	0.956
Mean 4CV	6.450	2.960	1.690	0.532	0.771	0.920	−2.190	−3.150	−4.340					
SD DCV	5.558	3.210	1.994	0.218	0.184	0.225	0.807	0.815	1.330			0.038	0.045	0.050
SD 4CV	1.180	0.530	0.500	0.040	0.027	0.017	0.097	0.079	0.164					

Only two-body interactions were used as structure features and only the glmnet method was used for parameter estimation. The decoy set *left out* during training and used as the test is listed in the first column. Performance statistics by representation level are listed in subsequent columns. The N column gives the number of structures in each decoy set. The Mam column gives the average Mammoth structure alignment score between natives in the row decoy set the best structure in a different decoy set. The Cor_{all} columns give the Pearson correlation coefficient of the row model with the model trained on all decoy sets. The middle row, Cor_N, gives the correlation coefficient of each column with the N (decoy set size) column. The lower part of the table compares the overall mean and standard deviation of statistics when leaving one decoy set out at a time (DCV, this table) versus leaving one balanced fold out as was done in the previous experiment (4CV, Table III).

they are not restricted to a single distance bin (res and mc1 models used five distance bins). However, at the atomic level (t32), the reduction from two- to one-body terms gave a larger drop in prediction performance (mean rank 1.56 for two-body versus 2.95 for one-body). For a fine-grained interactions and energy, the one-body approximation apparently breaks down.

Both the contact features and exposure features did surprisingly well at each level of representation. At the coarse res level, they provided nearly the same amount of information as two-body interactions. Contacts were quite effective at the mc1 level, exposure was less so. Both contacts and exposure were more distant from two-body interactions at the t32 level, though they still provide a higher degree of discriminatory power than the two coarse-grained representations. Along with the failure of (two and three)-body features at the res and mc1 levels, this suggests coarse-grained models may benefit from pursuing simpler features such as solvent exposure and the density of bead packing.

The angle representation was and oddity in that svmrbf method proved most effective at fitting its parameters, though other methods came close in terms of the top-1 fraction. The svmrbf model for angles clustered into three groups surprising achieves a better mean native rank than any res level features. For all methods, clustering the angles into three groups provided better performance than dividing into 20 groups based on the amino acid type. This may be in part explained due to model additivity. Interactions between beads can be considered somewhat independently in that two good contacts are more energetically favorable than two bad contacts with one good and one bad somewhere in between. This property lends itself reasonable well to linear models (svm, svmrank, and glmnet). Angle bending is not quite so independent: a two locally favorable bends may be globally unfavorable if they create clashes or near clashes in the protein chain. The additivity property is no longer a good approximation and linear estimation methods will miss such relations. Nonlinear learning methods, such as svmrbf, are better at deriving models which incorporate nonadditivity.

Whole decoy set cross-validation experiment (DCV)

In this experiment, a whole decoy set was left out during training and then used to evaluate the learned model. We refer to this methodology as *decoy set cross-validation* (*DCV*). DCV is more challenging than (4CV) as decoys from different sets are generated using different methodologies. Decoys used for training may have different characteristics than those that appear in testing. DCV identi-

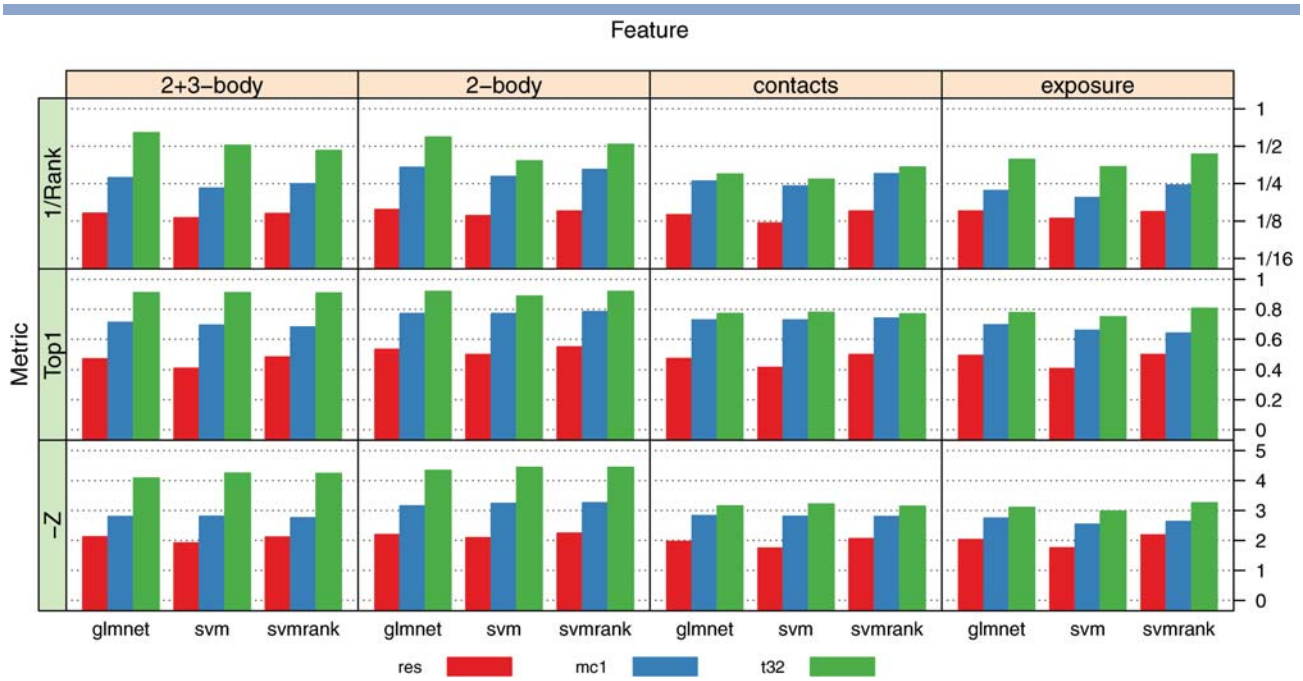


Figure 1

Results of four-fold cross-validation (4CV). The model features vary horizontally and the performance metric vertically. Within each cell, bars are grouped by the discrimination method used. Color indicates the representation level. The mean native rank statistic has been inverted to 1/Rank and Z-score to −Z so that larger bars indicate better performance.

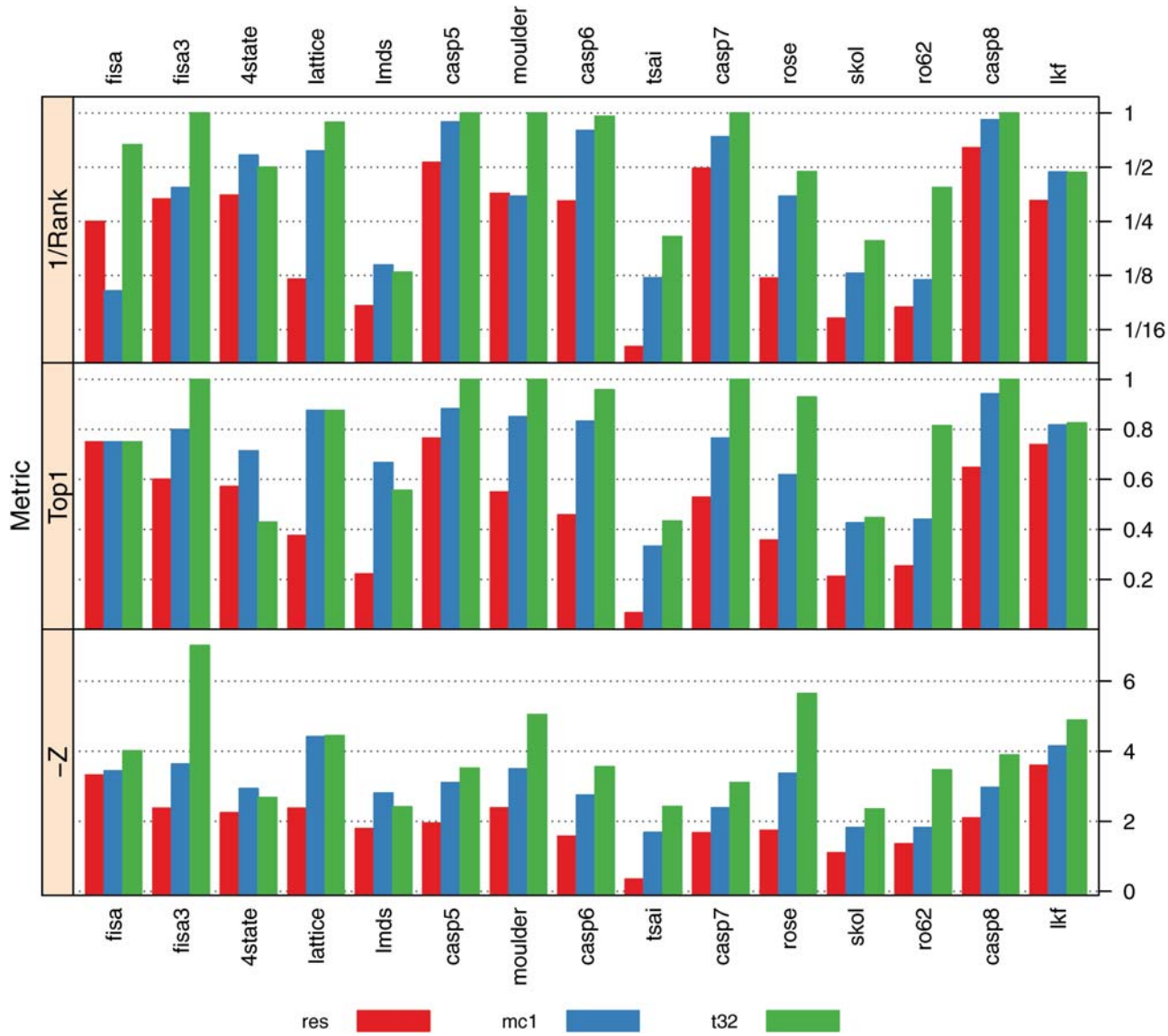
fies the difficult decoy sets and tests whether patterns learned on decoy sets generalize to truly new data.

In the DCV experiment, we limited ourselves to two-body interactions at the res, mc1, and t32 representation levels. We used only the glmnet method for parameter estimation. This combination (glmnet with two-body features) was representative of the best performance according to Section Four-for cross validation experiment and should prove representative of varying structural features and parameterization method.

T6 F2 Table VI presents numerical results for the cross-training evaluation whereas Figure 2 gives a visual summary of the results. As the representation varies from coarse-grained res to fine-grained t32, performance generally improves on all decoy sets. A few exceptions are the 4state and lmds sets in which the atomic detail of t32 performs worse than the two-interaction point model of mc1. The 4state decoy set was originally created using a reduced representation[18] which may explain why mc1 and res perform favorably on it compared to t32. Though lmds decoys were created using an all-atom model,[42] global functional forms were used to explicitly smooth out local energy minima in the decoys. Without unfavorable atomic interactions, the t32 features are not as informative explaining why the mc1 representation, which does not rely on atomic clashes, transfers from other decoy sets to lmds more readily.

Performance across decoy sets varied drastically. The sets lmds, tsai, and skol proved very challenging for all levels. When left out, the best representation for each data set achieved 66% (lmds/mc1), 43% (tsai/t32), and 45% (skol/t32) Top-1 recognition of native proteins over decoys. This is compared to rates in the 80–95% range for most other large sets. Decoys in these sets were all subjected to some energy minimization or structural relaxation to remove many obvious atomic clashes, procedure that is known to substantially increase difficulty.[43,44] Future work on decoy should focus on making improvements on this kind of decoy set. The ro62 set also provided a challenge for the coarse-grained representations but was handled readily by the t32 level. This set was produced using the ROSETTA software but incorporated a feedback loop to increase the number of decoys near the native structure ([45,46] Rhiju Das personal communication). Studies of coarse-grained models would benefit from analyzing this set.

The columns for Cor_{all} of Table VI indicates how much a decoy set affects learned parameters. It gives the Pearson correlation coefficient between the parameter vector of the model learned when the row's decoy set is left and the model learned when all decoy sets are used. An important point is that *training on all decoy sets together leads to a perfect model with Rank and Top-1 of 1.0 on all sets at all levels of granularity.* This is clearly an over-fit of the data that will not generalize to new types of decoys. However, analyzing the influence each decoy set has on the all-decoy-set parameters paints an interesting picture.



COLOR

Figure 2

Summary of performance statistics when leaving whole decoy sets out during cross-validation (DCV). Only two-body interactions trained by glmnet were used. The decoy set left out varies horizontally across the cells, performance metric varies vertically. Color indicates the representation level. The mean native rank (Rank) and Z-score (Z) have been inverted to 1/Rank and −Z so that larger bars indicate better performance.

When correlation is low, it indicates the decoy set is exerting influence on the all-set model as the left-out model parameters differ from the all-set parameters. Confounding this reasoning is the variance in size of decoy sets. The center row of Table VI labeled Cor_N indicates that the influence decoy sets exert on the overall model correlates very well with their size. The performance statistics (Rank, Top-1, Z-score) do not correlate well with the decoy set size (row Cor_N) but the model stability measure Cor_{all} exhibits high-negative correlation to decoy set size (rightmost columns of row Cor_N). When larger decoy sets are left out, parameter estimates drift farther from the estimates based on all decoy sets. However, for a small but difficult decoy set like lmds, parameters are similar whether the decoy set is used or left out (Cor_{all} = 0.962 for t32). Clearl some small changes in the parameters have a big impact on performance: training with all sets including lmds gives a mean native rank of 1.0 on lmds, whereas training with all sets except lmds gives a mean native rank of 11.8 on lmds. A simple correlation coefficient between model parameters does not seem an adequate measure of the stability of those models nor how they will generalize to new data.

Table VII
Difficulty of discriminating decoys generated with and without templates

			Rank			Top-1			Z-score		
Templ?	#Sets	Stat	res	mc1	t32	res	mc1	t32	res	mc1	t32
No	6	Mean	4.763	2.927	2.442	0.522	0.766	0.802	−2.045	−2.936	−3.497
Yes	9	Mean	9.259	5.595	2.201	0.401	0.636	0.800	−1.920	−3.060	−4.503
No	6	SD	6.155	3.589	1.439	0.244	0.213	0.200	1.017	1.081	1.632
Yes	9	SD	4.613	2.604	2.374	0.198	0.153	0.252	0.697	0.652	0.988
Yes v. No		p-*value*	0.163	0.154	0.811	0.337	0.232	0.989	0.800	0.808	0.216

Columns are (Templ?) whether templates influenced the decoy generation, (#Sets) the number of data sets in the group, (Stat) mean or standard deviation, and (remaining columns) the aggregate statistic for each performance measure. The upper portion of the table shows the mean and standard deviation of performance measures on DCV (rows of V). Data sets which used templates in decoy generation are 4state, lmds, moulder, skol, casp5–8, lkf. Those that did not use tempaltes are fisa, fisa3, lattice, rose, tsai, ro62. The bottom row shows *p*-values for a two-tailed T-test on whether the means of each stastistic are different from one another. High *p*-values indicate the means are not likely to be different.

As an alternative to simple correlations, we examined structural relations between proteins in different decoy sets. We aligned all native structures in a decoy set against all other natives using the Mammoth structure alignment program.[47] The best structure alignment score for each native in a decoy set was recorded and the average over all natives in a decoy set is given in the Mam column of Table VI. A low Mam value indicates the natives in a decoy set share few structural characteristics with representatives in other decoy sets. The row Cor_{Mam} gives the correlation Mam with performance statistics and model stability. It has moderate correlation to Rank and Top-1 and weak correlation to model stability (Cor_{all}). The correlation adds to the explanation of why decoy sets like lmds, skol, and tsai are difficult: they contain so distinct structures with few similar structures in other sets from which to learn. Counter examples are fisa3 and lattice which have low Mam scores but good performance in terms of Rank and Top-1. However, these sets are small. The combination of structural distinctness and aforementioned energy minimization is likely the full reason why lmds, tsai, and skol are so difficult.

The difficulty of leaving whole decoy sets out is further illustrated by comparing performance on this experiment (Mean DCV) and the results obtained from 4-4CV Mean in which decoy sets were balanced across the four folds. Mean performance statistics are shown near the bottom of Table VI. The 4CV experiment has generally better performance statistics than DCV and the standard deviation of DCV folds is much wider than in 4CV. This underscores the fact that testing a model on a new decoy set is a true out-of-sample estimate, where the decoys may be drawn from an entirely different distribution than the training data.

Predicting a completely new protein structure is much more difficult than and predicting the structure of a protein with an identified structural template. Templates influence decoy data sets in that the decoy generation mechanism uses some knowledge of the native structure or a related template. To assess how much this affects our own study, we looked at the results on DCV aggregated over decoy sets which used knowledge of the native or a template to generate decoys versus those that did not. Ostensibly, the use of a native or good template should produce decoys with more native-like characteristics which should conversely make decoy discrimination harder: there are fewer differences between a native and template-influenced decoys. Aggregated results are shown in Table VII along with a listing of which decoy sets were influenced in some way by a template. The reasons for a decoy set qualifying as template-influenced (Templ = yes) are described in detail in Supporting Information; use of a close structural relative or the native protein itself or fixing native secondary structure confers template influence. The means for template-free generation ("No" rows in Table VII) are indeed better indicating that these decoys are easier to identify than those based on templates. However, none of the performance measures exhibits a statistical difference between template-based and template-free decoy generation sets. This is due to the large variance of performance between the datasets in each group.

Mixed model selection experiment

To date, efforts to derive reduced protein representations have primarily focused on choosing the model according to physical intuition. After choosing a representation and functional form, force field parameters a determined to reproduce experimental results or discriminate structural decoys. Representations such as t32, mc1, and res, are derived from a priori knowledge of what seems sensible for modeling purposes and other features are discarded to avoid computational costs.

A pure data-mining viewpoint takes the opposite approach of including all potentially useful features in an unbiased way. The useful features are selected during parameterization to maximize accuracy. It is not clear whether this philosophy can be directly incorporated in molecular dynamics. However, for the decoy discrimination setting, it is readily usable to select beads from different representations to create a mixed model.

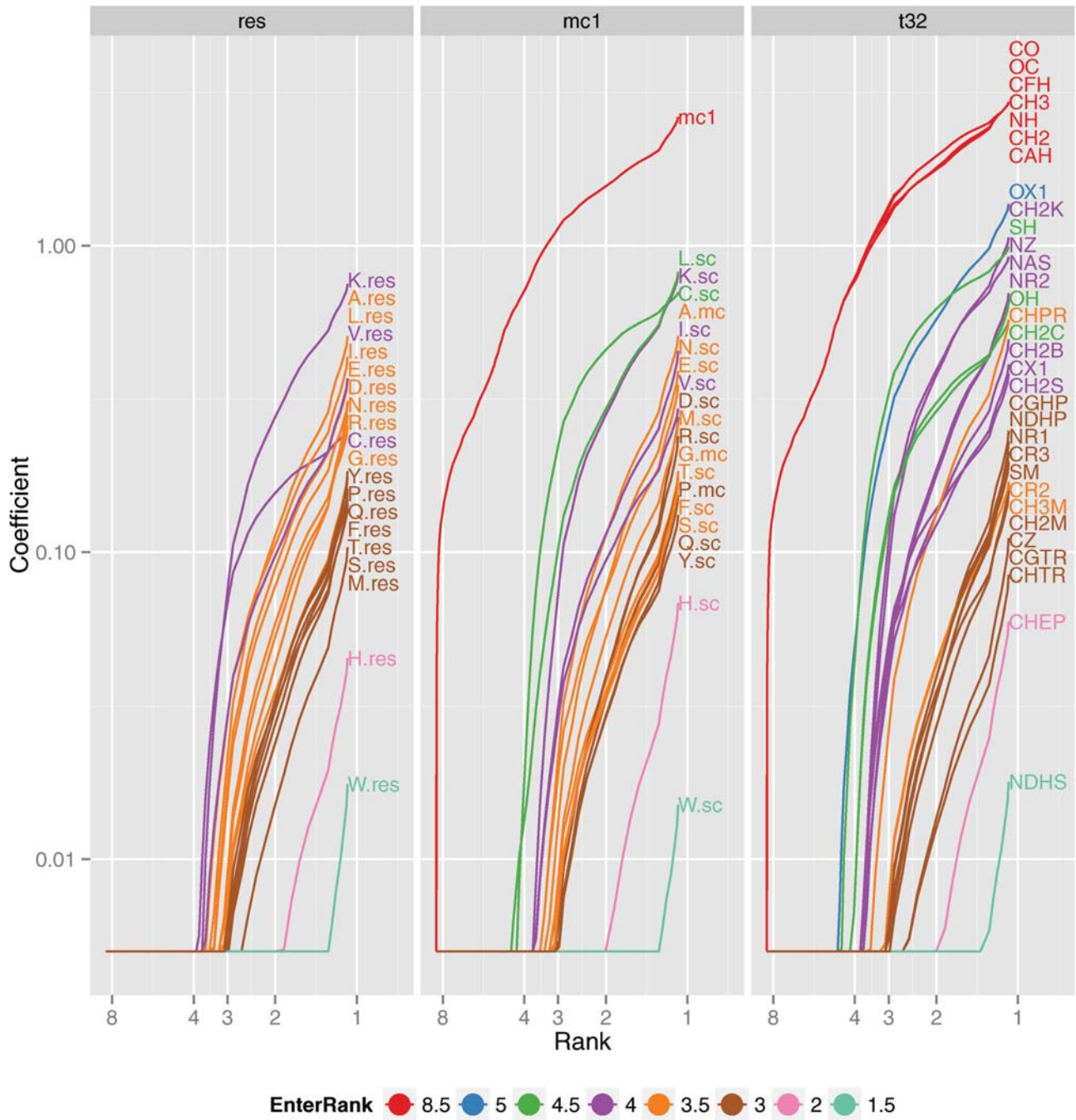


Figure 3

Results of mixed bead selection by BSM. The *x*-axis shows the mean native rank of the model generated. Rank improves from left to right. The *y*-axis shows the coefficient associated with each bead type. A zero coefficient means a bead type may be eliminated from the model. Color indicates the approximate rank at which beads enter the model as the regularization penalty varies.

In this experiment, each protein was simultaneously represented using res, mc1, and t32, and all two-body interactions between beads were counted. This included cross-level interactions such as those between the mc1 bead R.sc (arginine sidechain) and the t32 atom type OX1. We developed an optimization procedure to do decoy discrimination in this mixed representation. It is a *bead selection method* thus we abbreviate it as *BSM*. Details of BSM are given in Section Decoy separation with bead selection. Briefly, it uses a regularization path approach similar to glmnet while doing grouped decoy separation as svmrank does. At high-regularization levels,

few beads are in the model while progressively lowering regularization allows more beads to enter the model and lets us observe the trade-off between model complexity and performance. Beads are selected from any of the three granularity levels giving us insight into which parts of the protein may be approximated using coarse-grained beads. Parameters were fit using all proteins in the 15 decoy sets so there is no cross-validation in this experiment.

Analysis of selected beads

The main results of this experiment are shown in Figure 3. Our BSM procedure varies a regularization penalty which gradually allows more bead types to enter the model. Allowing more beads into the model tends to give better performance so that each regularization point corresponds to a performance statistic. The plot in Figure 3 contrasts the mean native rank statistic against the coefficients associated with each type of bead in the three representation levels. Color indicates when the bead types enter the model by going from a zero to a nonzero coefficient. The bead types are listed on the right side in positions roughly corresponding to their weight when the mean native rank reaches 1 (e.g., all native structures have rank 1 for perfect performance).

The first beads to enter the model were related to the protein backbone. From t32 the types NH, CAH, CO, and OC are backbone atoms as is the mc1 bead mc1 (the bead type and representation level are both named mc1). These types overlap in that they represent the same parts of the protein, but according to the model's behavior, including both is beneficial. Additional t32 atom types that entered immediately are CH2, CH3, and CFH, the beta and gamma carbons of most residues. Using only these eight bead types, a mean native rank of 4.6 was achieved. This is interesting in that it indicates a large number of decoys must contain obvious backbone defects.

Next to enter was the t32 atom OX1: it represents the charged oxygens in the side chain of aspartic and glutamic acid. It was followed shortly afterward by t32 atoms SH and CH2C, the sulfur and beta carbon in cysteine, and OH, the oxygen in serine, threonine, and tyrosine. Additionally, the mc1 beads L.sc (leucine sidechain) and C.sc (cysteine sidechain) entered around this rank. It is interesting to again see overlapping elements, cysteine sidechain beads from both t32 and mc1, enter at approximately the same time. At this regularization level, 14 bead types were used which gave 3.9 mean native rank.

The bead types then entered in larger groups with a variety of t32 and mc1 beads activating. Beads from the res representation also entered. Several coherent groups representing charged side chains entered at approximately the same rank including lysine (K.res, K.sc, CH2K, NZ), aspartic/glutamic acid (CH2B, CX1, D.sc), and arginine (NR2). The cysteine res bead, C.res, also entered at this regularization level, well after the t32 and mc1 representations.

The remaining beads entered in large groups except for six outliers which entered very late. These were related to histitdine (H.res, H.sc, CHEP) and tryptophan (W.res, W.sc, NDHS). Their late inclusion indicates they do not factor into decoy discrimination heavily.

It seems a great deal of discriminatory power resides in only a few bead types. Modeling the backbone properly gives the initial and largest performance boost to achieve a mean native rank of 4.6. Adding a few select bead types that model charged groups and cysteine brings the rank down to 3.9. After that, a wider variety of bead types is required to get better rank.

Behavior of bead selection

The behavior shown in Figure 3 illustrates several deficiencies of BSM. Bead types representing the same part of the protein at different granularities seem to enter at the same time. Ideally, we would like BSM to prevent such redundancies. Figure 3 does not illustrate the maximum performance achievable by models using a subset of bead types. It may be that using only the first 14 bead types selected, a lower rank can be achieved, but this would require parameterizing only on these types. The BSM allows other bead types to enter as the regularization level changes giving only a rough idea of how effective each group of beads will be in isolation.

While it is tempting to compare the ranks achieved by the mixed selection of BSM to those presented in Sections Four-fold cross-validation experiment and whole decoy set cross-validation experiment, those experiments used a cross-validation framework that give more robust estimates of performance on future data. BSM was evaluated on all data and may over-estimate the achievable rank by the selected models. Our purpose was to explore the potential of automated model selection. Testing the mixed models produced by BSM will be the subject of future work.

DISCUSSION AND CONCLUSIONS

Two over-arching observations emerged from our comparison of three granularities of protein representation and variety of energy terms in them. First, the atomic-level detail (t32 model) gives the best performance definitively, but great improvement over single bead per residue (res model) can be gained by differentiating side-and main-chain interactions (mc1 model). The best mean native rank over 415 native proteins in 4CV are 6.45 for res and 2.96 for mc1. This improvement comes at a very low cost in terms of the number of parameters associated with the mc1 model: for two-body interactions there are

1073 parameters to learn in res versus 1189 in mc1, an increase of only 116 parameters. Going to atomic detail in t32 requires 1584 parameters for two-body interactions.

The second broad message is that low-resolution features (contact counts and solvent accessible surface area), provide a surprisingly large amount of discriminatory power regardless of their representation level. This is compared to two-body and (two- and three-)-body interactions. The decrease in performance for using contact counts or solvent exposure rather than two-body interactions at the t32 level gives a drop in mean native rank about 1.1; the average drop is only 0.77 at mc1 level and 0.20 at the res level. This is a sign that using lower resolution energy terms when low-resolution models are used does not compromise accuracy. At least, the exclusive use of contacts or exposure does not compromise accuracy much more than the initial choice of a coarse-grained representation.

In terms of parameter estimation methodology, three additional technical results come from our analysis. There is little benefit from using nonlinear parameter estimation techniques for the protein representations and features examined. The nonlinear svm models performed little better than linear versions and have several drawbacks (two hyper-parameters vs. one for linear, longer training times, and a lack of explicit parameter representations, Section Linear versus nonlinear classified). It also seems that training models through grouped separation rather than binary classification, as was done with svmrank, deserves additional exploration. Combining this training approach with the sparsity-inducing regularization of glmnet could produce more robust parameters. We are currently testing methods to do this. Finally, performance on a single decoy set, even within cross-validation, is not indicative how well a model will generalize to new decoy sets. It is difficult to assess how stable any of these models might be as small changes in parameter can drastically alter performance on difficult decoy sets.

Our intention with this study is not to suggest a particular scheme by which to do structure prediction, but instead to get at whether coarse-grained models limit the accuracy of prediction methods. An oft-employed protein prediction strategy is the following. We want to formulate the best structure prediction within time T. The first step is to search for a structural template using one of many good methods. Should a template or templates be found, the amount of conformational sampling required is reduced by many orders of magnitude by searching near the template. The remaining time up to T is probably best spent using a fine-grained model like t32 as the representation does not limit the accuracy of predictions much. If no template is found, then we must do a large-scale sampling of the protein's conformational space to get a sense of low-energy shapes. Certainly using a coarse-grained model will limit the accuracy of predictions, but much more conformational ground can be covered using a coarse-grained model due to the smaller number of beads in the model. Our results indicate coarse-grained models provide enough fidelity to guide sampling to reasonably close approximations of native structures. So lacking a good template, most of the prediction time up to T should be spent on coarse-grained sampling, perhaps with some subsequent fine-grained refinement. To our knowledge, most successful prediction schemes work roughly in this way with possible iteration between coarse and fine modes. From the stand-point of template utilization, our work confirms that this strategy is quite reasonable. Further inquiry is required to determine whether analysis of template-based or template-free decoys can yield insight into specific prediction tasks. For example, restricting parameter estimation to template-based decoys only may increase our understanding model refinement while the template-free setting may be more useful to derive parameters for new fold predictions.

Our data-driven approach to selecting mixed representations (BSM) led to modest insight into mixing beads from different granularities (see Section Behavior of bead selection). Rather than select one coarse representations for low performance and gradually shift to a finer-resolution as regularization is eased, BSM seemed to select similar beads from multiple representation levels at the same regularization points. Backbone beads are selected initially, then a few important side chain beads, particularly cysteine, then equivalent res/mc1/t32 beads at similar levels of regularization. While this gives some indication of the relative importance of different bead types, in most modeling situations we would not use redundant representations such as the mc1 backbone bead along with t32 backbone atoms like CAH, NH, and CO. In general, enforcing mutual exclusion in training would destroy the convexity of the optimization problem making it much less tractable to solve numerically. While difficult, it is worth additional work to determine if alternative formulations exist which produce less redundant models as this would have much greater impact in the modeling community.

ACKNOWLEDGMENTS

The authors would like to thank Rhiju Das for providing insight into the nature of the ro62 decoy set and Christodoulos Floudas for making the lkf decoy set available. The anonymous reviewers have our thanks for providing insightful feedback which greatly improved the above work. Access to research and computing facilities was provided by the Digital Technology Center and the Minnesota Supercomputing Institute.

REFERENCES

1. Volker Baschnagel J, Binder K, Doruker P, Gusev AA, Hahn O, Kremer K, Mattice WL, Muller-Plathe F, Murat M, Paul W, Santos S, Suter UW. Tries. Bridging the gap between atomistic and coarse-

grained models of polymers: status and perspectives. Inviscoelasticity, Atomistic models, Statistical Chemistry; Advances in Polymer Sciences, Vol.152, Advances in Polymer Science. Springer Berlin Heidelberg: Berlin, Heidelberg, July, 2000.
2. Tozzini V. Coarse-grained models for proteins. Curr Opin Struct Biol 2005;15:144–150.
3. Levitt M. A simplified representation of protein conformations for rapid simulation of protein folding. J Mol Biol 1976;104:59–107.
4. Ueda Y, Taketomi H, Gō N. Studies on protein folding, unfolding, and fluctuations by computer simulation. II. A. Three-dimensional lattice model of lysozyme. Biopolymers 1978;17:1531–1548.
5. Kolinski A, Skolnick J. Assembly of protein structure from sparse experimental data: an efficient Monte Carlo model. Proteins 1998;32:475–494.
6. Simons KT, Kooperberg C, Huang E, Baker D. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. J Mol Biol 1997;268:209–225.
7. Liwo A, Odziej S, Pincus MR, Wawak RJ, Rackovsky S, Scheraga HA. A united-residue force field for off-lattice protein-structure simulations. I. Functional forms and parameters of long-range side-chain interaction potentials from protein crystal data. J Comput Chem 1997;18:849–873.
8. Liwo A, Arłukowicz P, Czaplewski C, Ołdziej S, Pillardy J, Scheraga HA. A method for optimizing potential-energy functions by a hierarchical design of the potential-energy landscape: application to the UNRES force field. Proc Natl Acad Sci 2002;99:1937–1942.
9. Zhang Y, Kolinski A, Skolnick J. TOUCHSTONE II: a new approach to ab initio protein structure prediction. Biol J 2003;85:1145–1164.
10. Marrink SJ, Risselada H, Yefimov S, Tieleman DP, de Vries AH. The MARTINI force field: coarse grained model for biomolecular simulations. J Phys Chem B 2007;111:7812–7824.
11. Kurkcuoglu O, Jernigan RL, Doruker P. Mixed levels of coarse-graining of large proteins using elastic network model succeeds in extracting the slowest motions. Polymer 2004;45:649–657.
12. Rojas AV, Liwo A, Scheraga HA. Molecular dynamics with the United-residue force field: ab initio folding simulations of multi-chain proteins. J Phys Chem B 2007;111:293–309.
13. Samudrala R, Levitt M. Decoys 'R' Us: a database of incorrect conformations to improve protein structure prediction. Protein Sci 2000;9:1399–1401.
14. Qiu J, Elber R. Atomically detailed potentials to recognize native and approximate protein structures. Proteins 2005;61:44–55.
15. Dong Q, Zhou S. Novel nonlinear knowledge-based mean force potentials based on machine learning. IEEE/ACM Trans Comput Biol Bioinform 2011:476–486.
16. Samudrala R, Moult J. An all-atom distance-dependent conditional probability discriminatory function for protein structure prediction. J Mol Biol 1998;275:895–916.
17. Zhang C, Liu S, Zhou H, Zhou Y. An accurate, residue-level, pair potential of mean force for folding and binding based on the distance-scaled, ideal-gas reference state. Protein Sci 2004;13:400–411.
18. Park B, Levitt M. Energy functions that discriminate X-ray and near native folds from well-constructed decoys. J Mol Biol 1996;258:367–392.
19. Liwo A, Arłukowicz P, Ołdziej S, Czaplewski C, Makowski M, Scheraga HA. Optimization of the UNRES force field by hierarchical design of the potential-energy landscape. 1. Tests of the approach using simple lattice protein models. J Phys Chem B 2004;108:16918–16933.
20. Munson PJ, Singh RK. Statistical significance of hierarchical multi-body potentials based on Delaunay tessellation and their application in sequence-structure alignment. Protein Sci 1997;6:1467–1481.
21. Feng Y, Kloczkowski A, Jernigan RL. Four-body contact potentials derived from two protein datasets to discriminate native structures from decoys. Proteins 2007;68:57–66.
22. Krishnamoorthy B, Tropsha A. Development of a four-body statistical pseudo-potential to discriminate native from non-native protein conformations. Bioinformatics 2003;19:1540–1548.
23. Gniewek P, Leelananda SP, Kolinski A, Jernigan RL, Kloczkowski A. Multibody coarse-grained potentials for native structure recognition and quality assessment of protein models. Proteins: Struct Funct Genet 2011;79:1923–1929.
24. Kinjo AR, Horimoto K, Nishikawa K. Predicting absolute contact numbers of native protein structure from amino acid sequence. Proteins 2005;58:158–165.
25. Yuan Z. Better prediction of protein contact number using a support vector regression analysis of amino acid sequence. BMC Bioinform 2005;6:248.
26. R Development Core Team, R: A Language and Environment for Statistical Computing, vol. 1, no. 2.11.1. Vienna, Austria: R Foundation for Statistical Computing, 2011, p. 409.
27. Zhang J, Chen R, Liang J. Empirical potential function for simplified protein models: combining contact and local sequence-structure descriptors. Proteins 2006;63:949–960.
28. Bahar I, Kaplan M, Jernigan RL. Short-range conformational energies, secondary structure propensities, and recognition of correct sequence-structure matches. Proteins 1997;29:292–308.
29. Chang C-c, Lin C-j. LIBSVM: a library for support vector machines. Science 2011;2:1–39.
30. Joachims T. Training linear SVMs in linear time. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD 06, Philadelphia, PA, USA: ACM Press, New York, NY, USA. pp. 217–226. 2006.
31. Joachims T, Finley T, Yu C-NJ. Cutting-plane training of structural SVMs. Mach Learn 2009;77:27–59.
32. Friedman J, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. J Stat Softw 2010;33:1–22.
33. Quattoni A, Carreras X, Collins M, Darrell T. An efficient projection for L1, infinity regularization. In Proceedings of the 26th Annual International Conference on Machine Learning ICML 09, Montreal, QC, Canada, June 14-18, 2009: ACM Press, New York, NY, USA. pp.1–8, 2009.
34. Tseng P, Yun S. A coordinate gradient descent method for nonsmooth separable minimization. Math Program 2007;117:387–423.
35. Bertsekas DP. Nonlinear programming. 2nd ed. Athena Scientific, Belmont, MA, USA, 1999.
36. Loose C, Klepeis JL, Floudas CA. A new pairwise folding potential based on improved decoy generation and side-chain packing. Proteins 2004;54:303–314.
37. Rajgaria R, McAllister SR, Floudas CA. A novel high resolution C-alpha C-alpha distance dependent force field based on a high quality decoy set. Proteins: Struct Funct Bioinform 2006;65:726–741.
38. Rajgaria R, McAllister SR, Floudas CA. Distance dependent centroid to centroid force fields using high resolution decoys. Proteins 2008;70:950–970.
39. Tan C-W, Jones DT. Using neural networks and evolutionary information in decoy discrimination for protein tertiary structure prediction. BMC Bioinform 2008;9:94.
40. Pokarowski P, Kloczkowski A, Jernigan RL, Kothari NS, Pokarowska M, Kolinski A. Inferring ideal amino acid interaction forms from statistical protein contact potentials. Proteins: Struct Funct Bioinform 2005;59:49–57.
41. Pokarowski P, Kloczkowski A, Nowakowski S, Pokarowska M, Jernigan RL, Kolinski A. Ideal amino acid exchange forms for approximating substitution matrices. Proteins 2007;69:379–393.
42. Keasar C, Levitt M. A novel approach to decoy set generation: designing a physical energy function having local minima with native structure characteristics. J Mol Biol 2003;329:159–174.
43. Wroblewska L, Skolnick J. Can a physics-based, all-atom potential find a protein's native structure among misfolded structures? I.

Large scale AMBER benchmarking. J Comput Chem 2007;28: 2059–2066.
44. Handl J, Knowles J, Lovell SC. Artefacts and biases affecting the evaluation of scoring functions on decoy sets for protein structure prediction. Bioinformatics 2009;25:1271–1279.
45. Das R, Qian B, Raman S, Vernon R, Thompson J, Bradley P, Khare S, Tyka MD, Bhat D, Chivian D, Kim DE, Sheffler WH, Malmström L, Wollacott AM, Wang C, Andre I, Baker D. Structure prediction for CASP7 targets using extensive all-atom refinement with Rosetta@home. Proteins 2007;69:118–128.
46. Bradley P, Misura KMS, Baker D. Toward high-resolution de novo structure prediction for small proteins. Science 2005;309:1868–1871.
47. Ortiz AR, Strauss CEM, Olmea O. MAMMOTH (Matching molecular models obtained from theory): an automated method for model comparison. Protein Sci 2002;11:2606–2621.
48. Samudrala R, Xia Y, Levitt M, Huang ES. A combined approach for ab initio construction of low resolution protein tertiary structures from sequence. Pac Symp Biocomput 1999;516:505–516.
49. Rykunov Eac D, Fiser A. New statistical potential for quality assessment of protein models and a survey of energy functions. BMC Bioinformatics 2010;11:128.
50. John B, Sali A. Comparative protein structure modeling by iterative alignment, model building and model assessment. Nucleic Acids Res 2003;31:3982–3992.
51. Eramian D, Shen M-y, Devos D, Melo F, Sali A, Marti-Renom MA. A composite score for predicting errors in protein structure models. Protein Sci 2006;15:1653–1666.
52. Tsai J, Bonneau R, Morozov AV, Kuhlman B, Rohl CA, Baker D. An improved protein decoy set for testing energy functions for protein structure prediction. Proteins 2003;53:76–87.