

Polarizable Protein Packing

ALBERT H. NG, CHRISTOPHER D. SNOW

*Division of Chemistry and Chemical Engineering, California Institute of Technology,
1200 E. California Blvd., MC 210-4, Pasadena, California 91125*

Received 9 December 2009; Revised 12 October 2010; Accepted 17 October 2010

DOI 10.1002/jcc.21714

Published online 24 January 2011 in Wiley Online Library (wileyonlinelibrary.com).

Abstract: To incorporate protein polarization effects within a protein combinatorial optimization framework, we decompose the polarizable force field AMOEBA into low order terms. Including terms up to the third-order provides a fair approximation to the full energy while maintaining tractability. We represent the polarizable packing problem for protein G as a hypergraph and solve for optimal rotamers with the FASTER combinatorial optimization algorithm. These approximate energy models can be improved to high accuracy [root mean square deviation (rmsd) < 1 kJ mol⁻¹] via ridge regression. The resulting trained approximations are used to efficiently identify new, low-energy solutions. The approach is general and should allow combinatorial optimization of other many-body problems.

© 2011 Wiley Periodicals, Inc. J Comput Chem 32: 1334–1344, 2011

Key words: polarizable force field; protein structure prediction; AMOEBA; rotamer optimization; ridge regression

Introduction

The need for increased accuracy in biophysical protein models has led researchers to develop energy functions that include many-body effects. However, these energy functions are computationally expensive and incompatible with typical combinatorial algorithms for optimizing protein structures. To bridge this disconnect, we decompose multibody energies into low order terms suitable for traditional combinatorial optimization algorithms. Previously, Mayo and coworkers have developed decomposable approximations for solvent-accessible surface area terms¹ and for Poisson–Boltzmann solvent polarization.^{2,3} Here, we develop approximate models of multibody protein polarization. Several recent energy functions such as AMOEBA include many-body protein polarization.^{4–9} We decompose the many-body polarization energy of the AMOEBA force field into terms involving 1, 2, 3, or 4 residues. We demonstrate that the resulting truncated approximations are both accurate and amenable to combinatorial protein optimization. Furthermore, training the approximations resulted in remarkably accurate energy predictions ($R^2 = 0.99$) for low-energy rotamer combinations.

More detailed biophysical models that include polarization may improve accuracy in protein structure prediction and design. Here, we concentrate entirely on the fixed-backbone sidechain optimization subproblem. Many existing software packages optimize sidechain packing.^{10–12} Given a quantitative scoring function, the typical search algorithm attempts to identify the sidechain configuration of lowest energy. To reduce the problem to a combinatorial search-space, these programs force sidechains to adopt preferred conformations usually represented as discrete rotational isomers, or rotamers.^{13,14} Ideally, with a sufficiently

accurate energy function and rotamer library, the lowest energy rotamer combination will approximate the experimental native state of the protein. Diverse combinatorial optimization algorithms have been devised to obtain the lowest energy rotamer combination.^{15–19} Here, we use FASTER,^{20,21} a robust algorithm well suited to large combinatorial search spaces (Methods).

One benefit of the fixed-backbone rotamer optimization framework is that improvements made in the context of identifying low-energy rotamer combinations may apply to other problems. For example, the same methods constitute protein design calculations if the population of rotamers contains distinct amino acids. Another extension to the rotamer optimization framework is the consideration of entropic effects.²² Methods such as generalized belief propagation can be used to calculate the marginal probabilities of all rotamers and identify the rotamer distribution of the lowest free energy.^{23–25}

Combinatorial rotamer optimization has usually been applied to simple pairwise energy functions that allow rotamer–rotamer energies to be precalculated. However, traditional optimization algorithms are not suited for energy functions that do not consist of purely rotamer one-body and pairwise energies. For example, AMOEBA, a recently developed polarizable force field, models electrostatics using self-consistent-induced dipoles in addition to permanent multipoles.⁵ As a result, the AMOEBA energy cannot

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

Correspondence to: C. D. Snow; e-mail: csnow@alum.mit.edu

Contract/grant sponsor: King Abdullah University of Science and Technology (KAUST); contract/grant numbers: KUS-F1-028-03

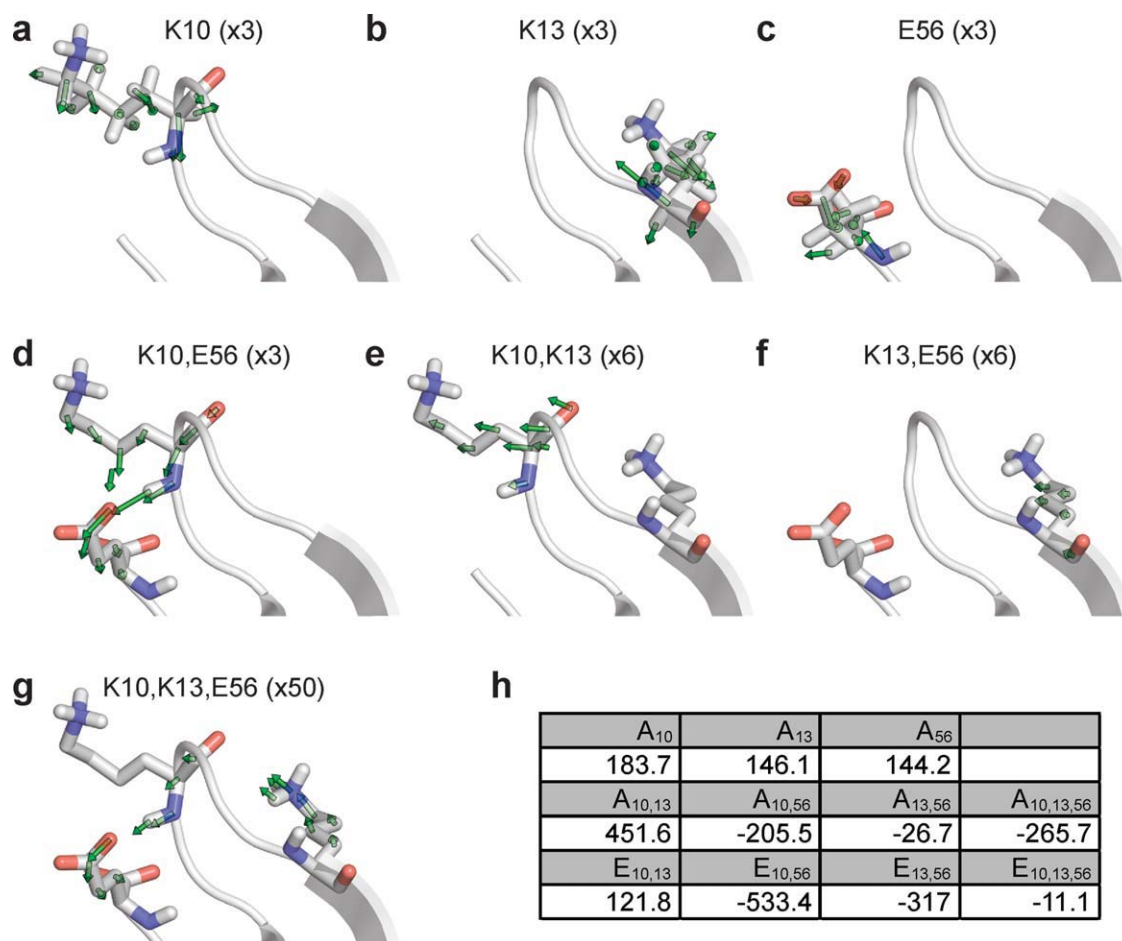


Figure 1. Illustrated derivation of a favorable triplet term from protein G for K10, K13, and E56 (the C-terminal residue). See text. The local backbone is cartooned for context; only atoms from the listed residues are present for the calculations. We plot the induced dipoles d_i from AMOEBA calculations for each isolated residue (a–c), the induced dipole correction vectors for each second-order term (d–f) $\delta_{ij} = d_{ij} - d_i - d_j$, and the 3-body term (g) $\delta_{ijk} = d_{ijk} - d_{ij} - d_{ik} - d_{jk} + d_i + d_j + d_k$. For clarity, we scale the induced dipole correction vectors by the multiplier shown (x3, x6, or x50) and omit induced dipole correction vectors below a size threshold (respectively, 1/6, 1/12, or 1/100 of a Debye). Panel (h) lists raw AMOEBA energy values (A_{10} , A_{13} , A_{56} , $A_{10,13}$, $A_{10,56}$, $A_{13,56}$, $A_{10,13,56}$), the derived second-order terms ($E_{10,13}$, $E_{10,56}$, $E_{13,56}$), and the derived third-order term ($E_{10,13,56}$) in kJ mol^{-1} . This figure was created using PyMOL (www.pymol.org).

be exactly decomposed into rotamer–rotamer energies. However, the full energy of an N -residue protein may be decomposed into a sum of energy terms, from 1-body to n -body (Methods). An illustrative example of the derivation of a 3-body term is provided in Figure 1. Calculating higher order terms (e.g., 4-residue and 5-residue) rapidly becomes intractable due to combinatorial explosion. However, if lower order effects dominate as expected, a truncated summation of energy terms provides a useful approximation to the full energy. Truncation allows us to avoid an exponential increase in the number of required higher order terms.

It remains to be seen to what extent protein polarization contributes to protein sidechain positioning. There are a number of scenarios where including this level of detail could be particu-

larly advantageous. Polarization is most likely to play a role in accurate accounting of the energetics of buried polar interaction networks. For example, the much debated extent to which salt bridges contribute to protein stability depends upon polarization.^{26–30} Although the typical two-body energy function was parameterized to implicitly account for a fixed degree of polarization, the true charge will vary depending on solvent exposure and the presence or absence of complementary charges. Explicitly modeling polarization effects should therefore enable more accurate models of context-dependent salt bridge energetics. Another potential application is to model the titration of enzyme active sites. When combined with a sophisticated solvent model, the methods described here may be particularly well suited for predicting pK_a shifts, because they account for protein polarization in addition to

robust sampling of the potentially astronomical number of relevant sidechain positions and protonation states.³¹

First, we show that this decomposition scheme allows us to approximate the AMOEBA energy of static protein models. Next, we demonstrate that the approximate energy models effectively score protein G rotamer combinations. We then demonstrate methods to tune the approximate energy models to high accuracy. Finally, we show that these approximations are compatible with combinatorial rotamer optimization, completing the demonstration of polarizable protein packing.

Methods

AMOEBA Decomposition

We set each 1-body energy term, E_i , equal to the AMOEBA energy of the corresponding isolated residue, A_i as calculated by the TINKER program `analyze.x`. The isolated residue includes both the sidechain and backbone atoms. Fortunately, incomplete valence of backbone atoms does not prevent `analyze.x` from completing the calculation. Higher order terms were calculated by the inclusion–exclusion principle; each term was a correction to cognate lower order terms. By cognate, we denote a term that applies to a subset of the corresponding residues. For example, a 2-body energy term E_{ij} was calculated by subtracting the cognate 1-body terms from the calculated AMOEBA energy A_{ij} of the residue pair: $E_{ij} = A_{ij} - A_i - A_j$. Likewise, $E_{ijk} = A_{ijk} - A_{ij} - A_{ik} - A_{jk} + A_i + A_j + A_k$. We defined approximate energy landscapes by summing all applicable terms: first-order (ΣE_i), second-order ($\Sigma E_i + \Sigma E_{ij}$), and third-order ($\Sigma E_i + \Sigma E_{ij} + \Sigma E_{ijk}$). AMOEBA calculations for severely clashing sidechains occasionally failed to converge. When `analyze.x` failed to converge, the term in question was assigned a stiff arbitrary penalty of 10^5 kJ mol⁻¹. Any decomposition scheme involves breaking the protein into fragments. Any given term involves partial valence where adjacent residues are absent, but the effects cancel with the appropriate inclusion–exclusion of lower order terms. We contemplated and rejected an alternate decomposition scheme where sidechains and peptide groups would be distinct groups because the latter scheme would greatly increase the number of terms to calculate.

An illustrated example of a favorable third-order term is given in Figure 1. To understand the energy correction terms, it is helpful to inspect the induced dipoles (green vectors). First, we calculate AMOEBA energies for each individual residue, A_i , and plot the final induced dipoles d_i from the calculations (Figs. 1a–1c). Second, we calculate AMOEBA energies for each pair of residues, A_{ij} , and compute the second-order term $E_{ij} = A_{ij} - A_i - A_j$. Similarly, we compute the induced dipole correction vectors $\delta_{ij} = d_{ij} - d_i - d_j$ and plot these (Figs. 1d–1f). Third, we calculate the AMOEBA energy for the residue triplet, A_{ijk} , and compute the third-order term $E_{ijk} = A_{ijk} - A_{ij} - A_{ik} - A_{jk} + A_i + A_j + A_k$. We plot the induced dipole corrections $\delta_{ijk} = d_{ijk} - d_{ij} - d_{ik} - d_{jk} + d_i + d_j + d_k$ associated with the triplet (Fig. 1g). The magnitudes of the induced dipole corrections are much smaller for distant interactions and higher order corrections (note the scale factors). For visual clarity, we omit nonpolar hydrogens for the higher order calculations. Incomplete valences

in the backbone result in large induced dipoles for backbone atoms. This intraresidue effect cancels when the higher order terms are calculated. In this example, the hydrogen bond between the sidechain of E56 and the backbone of K10 (Fig. 1d) is the dominant pairwise effect and is associated with large favorable induced dipoles. The addition of K13 (Fig. 1g) augments the favorable polarization between K10 and E56. The pertinent energy terms (in kJ mol⁻¹) are shown (Fig. 1h).

Calculations

Calculations were performed on a 16-thread, 2.8-GHz, dual Xeon 5560 computer. Calculation times are tabulated in Supporting Information Table I. We used the SHARPEN protein modeling library in conjunction with the TINKER 5.0 analysis tools.^{10,32} AMOEBA calculations were performed with `analyze.x` and `amoebapro.prm`. Python scripts to perform the calculations described in this work have been added to the SHARPEN documentation (<http://www.sharp-n.org/>).

The default SHARPEN rotamer library is the Dunbrack backbone-dependent rotamer library.¹¹ We select rotamers from this library using a SHARPEN rotamer generator object with default behavior that matches a Rosetta³³ protocol that provides additional rotamers for buried residues. Briefly, we first compute a neighbor map that indicates which pairs of residues may potentially interact. Two residues are neighbors if their C_β (C_α for Gly) atoms are within a cutoff distance. The distance cutoffs range between 6–14 Å and depend on the amino acid types. Next, we use this neighbor map to assign residues to either the surface class or buried class. Buried residues are residues with at least 18 neighboring residues. For each $10^\circ \phi, \psi$ bin, the backbone-dependent rotamer library lists rotamers by decreasing frequency (along with the standard deviation for each chi value). If a residue is buried, we take up to 30 of the most common rotamers, or up to 98%, whichever limit occurs first. If a residue is in the surface class, we take up to 24 of the most common rotamers, or up to 95%, whichever limit occurs first. Finally, if a residue is buried, additional rotamers are generated by varying $\chi_1 \pm 1$ standard deviation. For large buried residues (Phe, Tyr, or Trp), we also vary $\chi_2 \pm 1$ standard deviation.

Rather than filling elements of a matrix, we store energy terms for individual rotamers and pairs of rotamers inside the vertices and edges of a SHARPEN EnergyGraph data structure, the underlying data structure of which is a Boost graph (<http://www.boost.org/>). This way we avoid allocating space for absent interactions. Also, we can more easily generalize to higher order interaction terms. Specifically, when we wish to store an energy term for a collection of three or more rotamers, we store the term at a hyperedge within a SHARPEN EnergyHyperGraph data structure.¹⁰ There is overhead associated with using a hypergraph; optimization algorithms are less efficient when passed an EnergyHyperGraph. To conveniently store large collections of energy terms on disk, we used the `pytables` module (<http://www.pytables.org/>).

Non-Clashing Rotamer Combinations

Random rotamer combinations rarely avoid high-energy van der Waal (vdw) clashes. To identify nonclashing but otherwise ran-

dom rotamer combinations for protein G, we first constructed an energy model that only penalized clashing rotamer pairs. Specifically, the rotamer–rotamer energy $E_{ij} = 0$ unless the vdw $E_{ij} > 100 \text{ kJ mol}^{-1}$, in which case $E_{ij} = \text{vdw } E_{ij}$. To sample nonclashing rotamer combinations, we started with 3200 random combinations and proceeded to optimize each rotamer combination with a steepest descent Monte Carlo trajectory that only accepts new rotamers if energy is reduced (i.e., $T = 0$, the number of iterations was 3390, five times the total number of rotamers). We scored the 3200 resulting combinations with AMOEBA and selected the 200 rotamer combinations with the lowest AMOEBA energy. These combinations successfully avoided penalized clashes and were diverse. To quantify the diversity of the rotamer combinations, we define the site entropy as the sum over observed rotamers of $P_{\text{rot}} \cdot \log(P_{\text{rot}})$ where P_{rot} is the observed probability of a particular rotamer. The net entropy is the sum of the site entropy for each residue. Here, net entropy = 94.5. For comparison, the entropy of 200 completely random protein G rotamer combinations was 110.3 ± 0.1 .

A Low-Energy Rotamer Combination

Having computed all 1-body and 2-body energy terms for protein G, we constructed a SHARPEN EnergyGraph (defined above). FASTER repeatedly identified the same putative minimum-energy rotamer combination (approximation $E = -6686 \text{ kJ mol}^{-1}$, actual AMOEBA $E = -5626 \text{ kJ mol}^{-1}$). We used this favorable rotamer combination as a starting point to calculate ensembles of low-energy combinations using the two following methods.

Low-Energy Rotamer Combinations (Method 1)

We performed Monte Carlo on the second-order approximate energy landscape in lieu of the true many-body energy landscape. Starting from the putative minimum-energy rotamer combination (above), we collected 10,546 distinct rotamer combinations from a Monte Carlo trajectory of 6.5×10^5 steps. Empirically, we found that a low-temperature ($K_B T = 8 \text{ kJ mol}^{-1}$) Metropolis criterion sampled the low-energy regime of interest. We scored these rotamer combinations with AMOEBA (median $E = -5535 \text{ kJ mol}^{-1}$). We removed 23 high-energy combinations (at least 200 kJ mol^{-1} worse than $-5626 \text{ kJ mol}^{-1}$ the best AMOEBA energy encountered). From the remaining pool, we randomly selected (without replacement) a 5000-member training set and a 5000-member test set. The entropy (defined above) of the Method 1 training set was 38.1. There were no identical combinations in the training and test sets. However, some “overlap” is essential for fitting. As desired, most 1-body, 2-body, and 3-body terms that were found in the test set were also present in the training set (Supporting Information Fig. 1). The average (standard deviation) hamming distance was 19.2 (2.8) rotamer substitutions between all pairs of training and test combinations.

Low-Energy Rotamer Combinations (Method 2)

Starting from the putative minimum-energy rotamer combination (above), we collected 16 Monte Carlo trajectories ($K_B T = 8 \text{ kJ}$

mol^{-1}), each consisting of 1000 AMOEBA calculations. We pooled the 2586 distinct rotamer combinations accepted by the Monte Carlo trajectories and removed 11 high-energy combinations (at least 200 kJ mol^{-1} worse than $-5633 \text{ kJ mol}^{-1}$ the best AMOEBA energy found during the Monte Carlo trials). From the remaining pool of 2575 low-energy rotamer combinations, we randomly selected (without replacement) a 500-member training set and 500-member test set. The entropy (defined above) of the Method 2 training set was 34.7.

Ridge Regression

We use ridge regression to learn a set of correction parameters to augment an initial energy model.²⁰ Each correction parameter is associated with one rotamer (or a collection of rotamers). Given a training set of rotamer combinations (instances), we create an array Y where each element is the unexplained difference between the actual AMOEBA energy for that instance and the energy predicted by the initial model. To eliminate the need for a constant correction term, we center the Y array by subtracting the mean, $\langle Y \rangle$, from each element. The matrix X describes which correction terms (columns) apply to each instance (rows). This matrix is large and sparse. Accordingly, we used the sparse matrix capabilities of Python module `cvxopt`³⁴ (<http://abel.ee.ucla.edu/cvxopt>) a free (GPL) convex optimization software package. We also use the `cvxopt` package to solve the linear system efficiently with Cholesky factorization. Specifically, to obtain the regression coefficients for each term, we used `cvxopt.cholmol.linsolv` ($X^T \times X + k \times I, X^T \times Y$) where k is the free ridge parameter and I is the identity matrix. Unless otherwise noted, reported calculations used a regularization parameter of 0.01 and a 500-member rotamer combination training set.

Results

Static Structures

First, we tested our decomposition scheme by approximating the many-body AMOEBA energy for static protein models. Our test panel was comprised of four small protein structures: protein G (1PGB), Rubredoxin (1B13), HPr protein (1CM2), and PAS factor (2B8I). Heteroatoms were removed and each model was pre-treated with REDUCE to obtain reasonable hydrogen positions.³⁵ For n from 1 to 4 we compared the cumulative sum of all n -body energy terms to the AMOEBA energy of each complete protein (Fig. 2). As expected, successive approximations approached the many-body energy (gray line). Including all 3-body terms greatly reduced the error of the approximation, compensating for an overly favorable second-order approximation. We found no compelling reason to include 4-body or higher order terms. Including all 4-body energy terms greatly increased the calculation time (Supporting Information Table I) without consistently improving the approximation (Table 1).

To understand the AMOEBA decomposition, we computed histograms of the energy terms for the static protein G model (Fig. 3). Many higher order terms were negligible. In particular, discontinuous y -axes were necessary to accommodate the histogram peaks at 0. For the 27,720 protein G 3-body terms, the

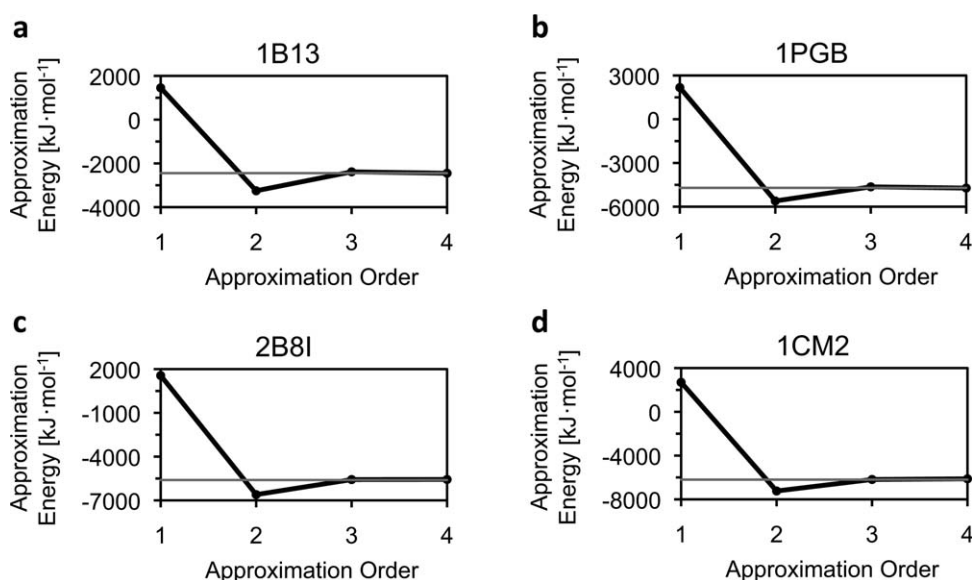


Figure 2. AMOEBA energy decomposition for four small proteins.

standard deviation (mean) was 0.82 (0.035) kJ mol^{-1} . Although most 3-body terms were small, large outliers spanned observed extremes of -29 and 28 kJ mol^{-1} . Together, 163 of the larger terms ($|E_{ijk}| > 4 \text{ kJ mol}^{-1}$) summed to 739 kJ mol^{-1} . However, the 27,557 smaller 3-body terms also had a significant aggregate contribution of 229 kJ mol^{-1} . The skew of the distribution to positive values is visible (Fig. 3c). Compared with the 3-body distribution, the 4-body distribution is narrow, with a standard deviation (mean) of 4.2×10^{-2} (-3.8×10^{-6}) kJ mol^{-1} . The largest 4-body term magnitude was 4.3 kJ mol^{-1} and only 13 of the 4-body terms had a magnitude greater than 2 kJ mol^{-1} .

Avoiding Negligible 3-Body Terms

For computational tractability, it is desirable to avoid explicitly calculating 3-body terms that turn out to be negligible. Ideally, we could avoid most terms of low magnitude (a low false positive rate) while retaining almost all terms of significant magnitude (a high true positive rate).

To develop a filtering method, we started by looking at the native structure of protein G (1PGB). Initially, we tried to predict the magnitude of 3-body terms as a function of the distance between the residues involved. We found that a large-magnitude 3-body term ($|E_{ijk}| > 4 \text{ kJ mol}^{-1}$) is unlikely if one residue is distant from the other residues. However, we found that inspecting the distances between residues was an inferior method to inspecting the 2-body terms that directly quantify the strength of the interaction between two residues.

Specifically, we compared the magnitude of each 3-body energy term $|E_{ijk}|$ from the native state of protein G to the magnitudes of the three cognate 2-body energy terms: $|E_{ij}|$, $|E_{ik}|$, and $|E_{jk}|$. Given these three values, we considered several metrics: the sum, the maximum, the minimum, the sum of the two larger, the sum of the two lesser, and the median. The last two metrics had superior performance as assessed with receiver-operator curve

(ROC) analysis (Supporting Information Fig. 2). Accordingly, when we “filter” potential 3-body terms below, we are discarding potential 3-body terms when the sum of the absolute value of the two smaller cognate 2-body terms fails to exceed a cutoff. This metric makes sense; 3-body coupling is unlikely to arise if one of the residues interacts weakly with the other two residues.

To identify a generally useful filter cutoff value we repeated the ROC analysis for the other members of the test panel: 1B13, 1CM2, and 2B8I (Supporting Information Fig. 2c). In each case, it was possible to find a cutoff value where 99% of the larger 3-body terms ($|E_{ijk}| > 4 \text{ kJ mol}^{-1}$) were retained and 93–98% of the smaller 3-body terms ($|E_{ijk}| \leq 4 \text{ kJ mol}^{-1}$) were discarded. Depending on the protein, the lowest filter cutoff that retained at least 99% of the larger terms was between 10 – 18 kJ mol^{-1} . A 10 kJ mol^{-1} filter cutoff reduced the number of 3-body energy terms by 90–96%. Removing these terms did not significantly degrade the approximation of the full AMOEBA energy (E_{filter} in Table 1). In the next section, we discuss the performance of the same filtering method in the context of scoring random protein G rotamer combinations.

Notably, it was also possible to identify significant 4-body terms by inspecting 3-body terms (Supporting Information Fig. 3). For example, the 13 largest 4-body terms ($|E_{ijkl}| > 2.0 \text{ kJ}$

Table 1. Cumulative Approximation Energy in kJ mol^{-1} for the Test Proteins.

Protein	Length	ΣE_i	$+\Sigma E_{ij}$	$+\Sigma E_{ijk}$	$+\Sigma E_{ijkl}$	E_{filter}^a	AMOEBA E
1B13	54	1454	-3252	-2386	-2450	-2393	-2446
1PGB	56	2172	-5606	-4638	-4722	-4692	-4707
2B8I	77	1560	-6605	-5574	-5566	-5627	-5604
1CM2	85	2697	-7239	-6185	-6125	-6207	-6199

$$^a E_{\text{filter}} = \Sigma E_i + \Sigma E_{ij} + \Sigma \text{filtered } E_{ijk}.$$

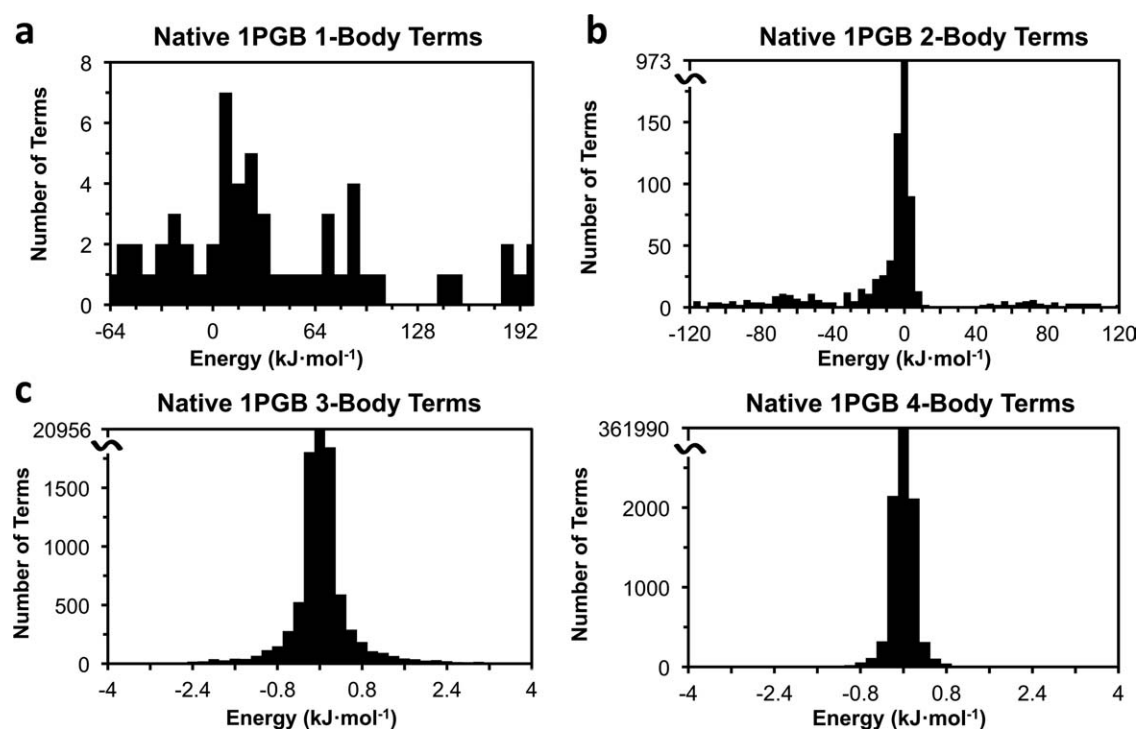


Figure 3. Distributions of first- to fourth-order terms for the native protein G model. (a) The energy for isolated residues ranged between -62 and 200 kJ mol^{-1} . Discontinuous y-axes are required for the higher order (second to fourth) distributions due to sharp peaks at 0. The second-order term distribution (b) is skewed left, causing an overly favorable second-order approximation. The third-order term distribution (c) is narrow and skewed right, compensating for the second-order approximation. Continuing the pattern, the fourth-order term distribution (d) narrows and is slightly skewed toward negative values.

mol^{-1}) for protein G could be identified by restricting attention to the 461 qualifying 4-body terms with large cognate 3-body terms ($|E_{ijk}| + |E_{ijl}| + |E_{ilk}| + |E_{jlk}| \geq 20$ kJ mol^{-1}) (Supporting Information Fig. 3b). This represents a useful reduction from the total number of 367,290 4-body terms.

Random Rotamer Combinations

To further test the AMOEBA decomposition scheme, we generated a test set of 200 protein G rotamer combinations. Other than avoiding large vdw clashes, the rotamer combinations were random (Methods). These diverse combinations ranged in energy between -4321 and -3786 kJ mol^{-1} .

For each rotamer combination, we summed terms corresponding to each rotamer (1-body) and rotamer-pair (2-body) to calculate the second-order approximation. We additionally included all rotamer triplet (3-body) terms to calculate the third-order approximation. Both the second-order and third-order approximations were highly correlated with the true many-body landscape (Fig. 4a). However, the third-order approximation had less noise and greater absolute accuracy.

Reduced noise is more important than absolute accuracy for correctly ranking rotamer combinations. Therefore, to compare predicted energies from different approximations, we treat the

slope and intercept of the approximation as free parameters. In other words, the figure of merit we use to assess an approximation is the post-linear-regression rmsd of predicted energies from the many-body AMOEBA energies. For example, to compare the second-order and third-order approximations for random protein G rotamer combinations, we plot the residual deviations after linear regression (Fig. 4b). Adding all 3-body terms reduced the rmsd of the predicted energies from 35.0 to 9.4 kJ mol^{-1} .

Computing all 1-body and 2-body terms for protein G (numbering 678 and 221,530, respectively) required only 45 min (Supporting Information Table I). However, computing the 3×10^6 distinct 3-body terms present within the random test set (out of 4.7×10^7 possible 3-body terms) required over 10 h. The long computation time motivates the use of a filtering method (see previous section). Having already computed all of the 3-body terms, we retrospectively tested the performance of filtered third-order approximations. Testing a number of potential cutoff values, we found a good cutoff value of 5 kJ mol^{-1} (Fig. 5). For this particular protein G rotamer problem, applying this cutoff would have eliminated 79% of the 3-body terms and computation time. Despite the computational savings, the filtered approximation was nearly as accurate (rmsd = 12.3 kJ mol^{-1}) as the model that included all 3-body effects (rmsd = 9.4 kJ mol^{-1}). For simplicity, the results below do not use a filter.

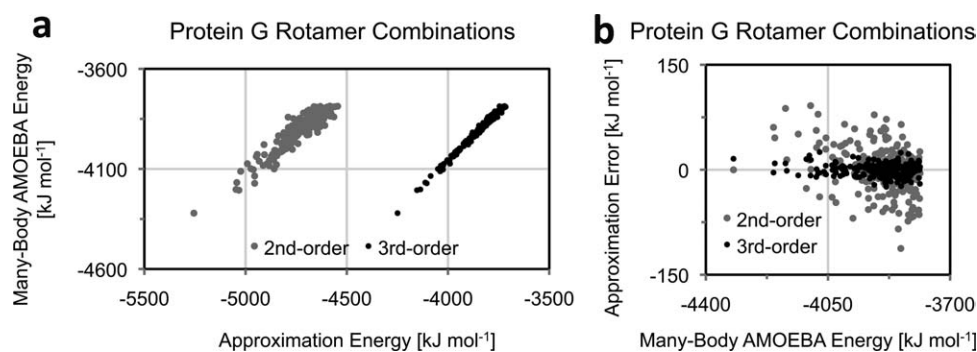


Figure 4. (a) Second-order (gray) and third-order (black) approximations for 200 rotamer combinations of protein G selected to avoid van der Waal clashes. (b) Residual errors after fitting the slope and intercept for each approximation.

Sampling Low-Energy Combinations

Although the results with random rotamer combinations were informative, high-energy combinations are irrelevant for most applications. Ideally, we could benchmark our approximate energy models against a test set composed of rotamer combinations that are low energy but otherwise random. Obtaining such a test set is not trivial because brute force sampling is intractable.

Therefore, we used Monte Carlo to sample low-energy rotamer combinations (Methods). Our first strategy (Method 1) sampled from the second-order approximation energy landscape. We used this computationally efficient approach to generate training and test sets, each with 5000 distinct rotamer combinations. However, we were concerned that sampling rotamer combinations using the approximation might lead to bias when assessing the approximation quality. Therefore, for comparison we generated alternate training and test sets (500 members each) by directly Monte Carlo sampling rotamer combinations from the many-body AMOEBA energy landscape (Method 2). Fortunately, no bias was apparent; both methods produced similar results. Therefore, calculations below refer to the training and test sets generated by the more efficient strategy (Method 1).

Low-Energy Approximation Performance

Low-energy rotamer combinations proved to be more easily and accurately modeled than random combinations. The prediction rmsd of the second-order approximation for protein G test set rotamer combinations was 11.6 kJ mol⁻¹ rather than 35.0 kJ mol⁻¹ (Fig. 6a). This is unsurprising; low-energy rotamer combinations were less diverse and occupied a smaller energy range: 195 kJ mol⁻¹ rather than 535 kJ mol⁻¹. Despite the much larger size (10,000) of the combined training and test sets, lower entropy resulted in fewer distinct 3-body terms (989,018). After computing the 3-body terms, we found that the rmsd of the third-order approximation for test set rotamer combinations was 2.3 kJ mol⁻¹ rather than 9.4 kJ mol⁻¹ (Fig. 6b). This latter result was intriguing; potentially we need only explicitly calculate a limited pool of 3-body terms to achieve chemical accuracy for low-energy rotamer states.

Regression

We sought to further increase precision and decrease computational expense. The relatively small magnitude of the majority of the higher order terms suggested that regression-based approaches might allow us to avoid explicitly calculating a large number of higher-order terms. Instead, we would calculate the AMOEBA energy for a training set of rotamer combinations and calculate effective correction terms using regression. The cluster expansion methods described by Keating and coworkers are one such approach.^{36,37} Briefly, Grigoryan et al. (i) build and score a large number of combinations for a training set and (ii) build an approximate energy model by iterative feature selection with fast cross validation. In contrast, our method involved (i) directly computing 1-body and 2-body energy terms to form a second-order approximate energy landscape, (ii) building and scoring a moderate number of low-energy rotamer combinations for a training set, and (iii) a single ridge regression calculation to fit selected correction terms. Originally, we had intended to fit 3-body correction terms. Unfortunately, accurately fitting a

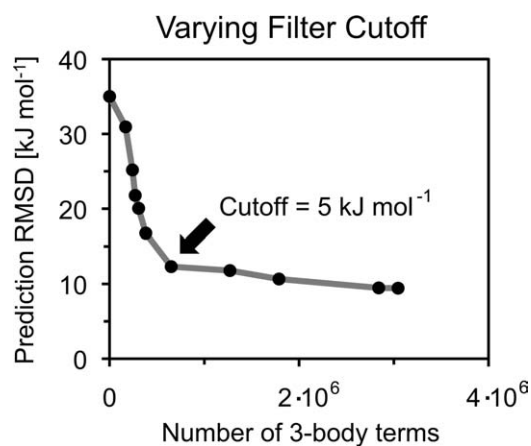


Figure 5. Filtering potential 3-body terms leads to performance intermediate between the second-order approximation (rmsd = 35.0 kJ mol⁻¹) and the third-order approximation (rmsd = 9.4 kJ mol⁻¹).

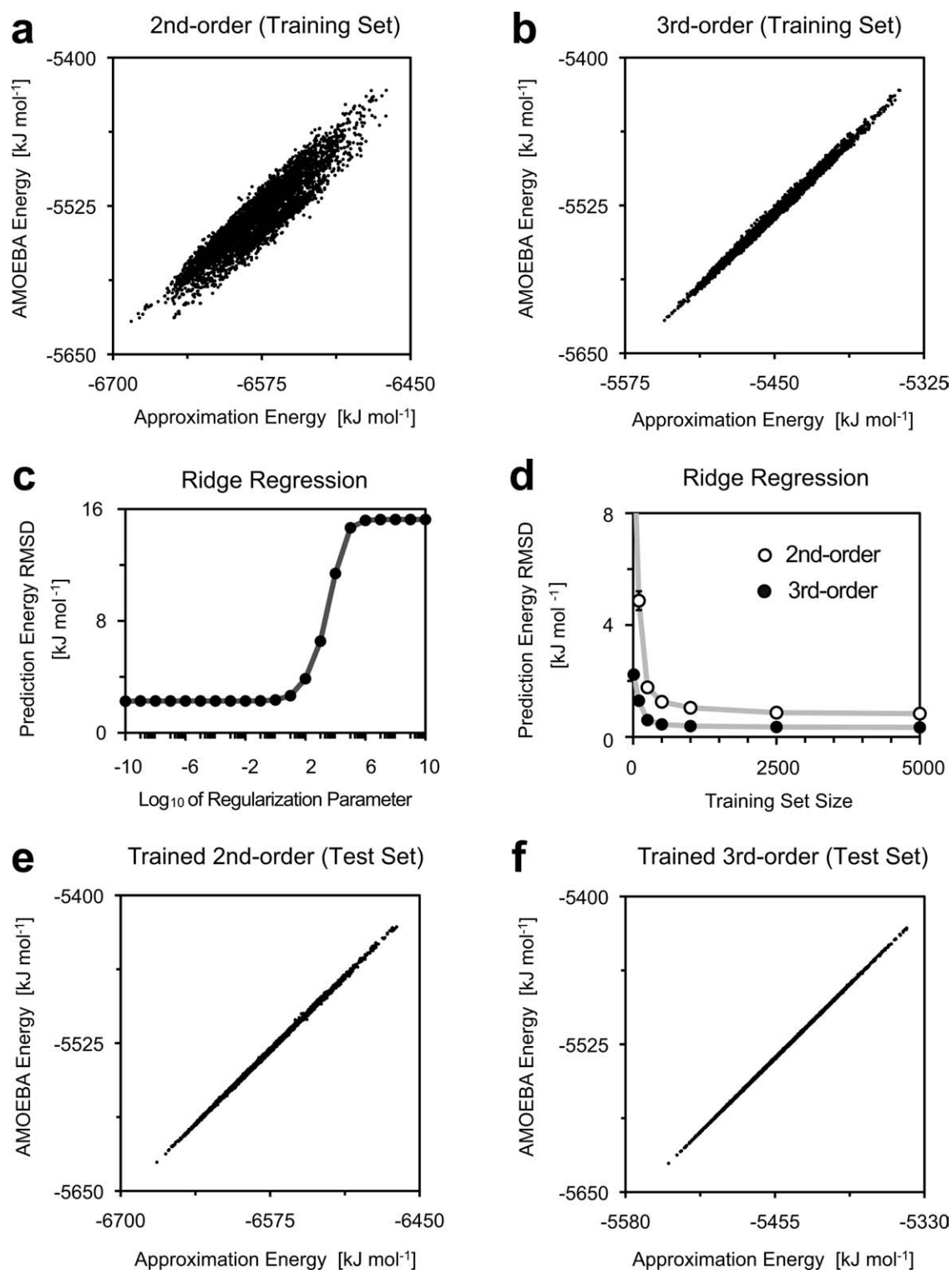


Figure 6. Predicted energies for 5000 test set rotamer combination from (a) second-order and (b) third-order approximation were highly correlated with the true many-body AMOEBA energies. Ridge regression performance varied with (c) the choice of regularization parameter and (d) training set size. The (e) trained second-order and (f) trained third-order models were particularly accurate with respective rmsd of 0.8 and 0.3 kJ mol^{-1} .

large number of correction terms requires a large training set. Instead, we were pleased to discover that fitting only 1-body correction terms to a relatively modest training set was usually sufficient for high accuracy (see the superoxide dismutase example below for an exception).

We began by calculating explicit 1-body and 2-body energy terms for each rotamer and rotamer pair, forming a second-order approximate energy landscape. This approximate energy landscape was useful for the next step: building a moderate number of low-energy rotamer combinations (e.g., 500). For each of these training set rotamer combinations, we calculated both the full AMOEBA energy and the predicted energy of the second-order approximation. We then fit to the prediction error (the full AMOEBA energy minus the approximation prediction) using ridge regression (Methods). The fitting parameters were 1-body correction terms corresponding to each distinct rotamer present in the training set rotamer combinations. In other words, starting from an approximation that contains both 1-body and 2-body terms, we tuned only the 1-body terms to better fit the full AMOEBA energies found for the training set.

Ridge regression is a flexible strategy (Methods). For low values of the regularization parameter, the method becomes a convenient form of unconstrained regression. On the other hand, for high values of the regularization parameter, ridge regression relies heavily on the *a priori* estimate for each parameter and provides only small correction terms. Thus, modulating the regularization parameter can address overfitting. Figure 6 illustrates results for protein G. After testing values for the regularization parameter between 10^{-10} and 10^{10} , we selected a regularization parameter of 0.01 to fully benefit from the regression accuracy improvements without unnecessary risk of overfitting (Fig. 6c). We found that trained models had impressive accuracy and that performance depended strongly on training set size (Fig. 6d). Similarity to the test set, particularly coverage of the 1-body terms found within the test set, was a major factor. Term corrections, the output of the regression, were small with respect to the initial 1-body energy terms (Supporting Information Fig. 4). The highest accuracy was found using the largest training set size (5000 combinations): 0.8 kJ mol^{-1} (0.3 kJ mol^{-1}) rmsd for the second-order (third-order) approximation (Figs. 6e and 6f). However, our default training set size (500 members) also performed well: 1.3 kJ mol^{-1} (0.4 kJ mol^{-1}) rmsd for the second-order (third-order) approximation.

To further validate this approach, we proceeded to test the other proteins in the test panel. In each case, tuning 1-body energies by training the second-order approximation with 500 rotamer combinations (from relatively short Monte Carlo trajectories) yielded a high accuracy model with prediction rmsd less than 1 kJ mol^{-1} (Supporting Information Fig. 5). The trained second-order approximation was relatively easy to compute and offered impressive precision. As a control, we tested performance after discarding the most similar training and test combinations (ensuring that no training-test pair was within five rotamer substitutions) and found that accuracy was still quite high; training with 500 combinations led to rmsd = 1.7 (0.56) kJ mol^{-1} for the tuned second-order (third-order) approximation. As a further control, we investigated the performance of first-order models. The explicitly calculated 2-body terms were essential for the

high accuracy. Before fitting, an energy model consisting solely of the explicitly calculated 1-body terms models the training set poorly (29.5 kJ mol^{-1} rmsd). In contrast, when trained with 500 (5000) examples, a tuned first-order model predicted the test set energies with 7.8 (5.3) kJ mol^{-1} rmsd.

Interpretation of the perturbations to the value of the single-body terms is challenging. These correction terms implicitly model thousands of higher order effects observed in the training set. Interpretation depends critically on the nature of the higher order effects that are omitted from the initial energy model. For example, the single-body correction terms change significantly when they implicitly represent effects higher than second-order (Supporting Information Fig. 4a) or effects higher than third-order (Supporting Information Fig. 4b). If the ultimate aim is to calculate physically interpretable correction terms via regression, additional care will be required to construct the training set (e.g., to ensure that the factors of interest vary independently).

Packing

As described above, we developed second-order and third-order approximations that were quite accurate. We proceeded to verify that these energy models were compatible with combinatorial rotamer optimization methods. Applying FASTER to either the second-order or the trained second-order model was no more time consuming than a standard rotamer optimization calculation (Methods). Optimizing with the untrained second-order model produced a rotamer combination with low AMOEBA energy = $-5625.7 \text{ kJ mol}^{-1}$. Optimizing with the trained second-order approximation was superior, leading to a rotamer combination with AMOEBA energy = $-5646.3 \text{ kJ mol}^{-1}$.

It was significantly more challenging to perform combinatorial optimization using the explicit third-order approximation. Computation time was the first hurdle. Despite applying the 5 kJ mol^{-1} filter discussed above, 1.4×10^7 of 4.7×10^7 potentially significant 3-body terms remained (calculations required 2 days). The second hurdle was holding the resulting data structure in memory. To avoid inflating the required memory, we excluded the smallest 3-body terms ($|E_{ijk}| < 0.5 \text{ kJ mol}^{-1}$) when building a SHARPEN EnergyHyperGraph.¹⁰ Removing the smaller terms reduced the necessary memory from 4.2 to 1.7 Gb. The final hurdle was the increased cost of combinatorial optimization. Running a single FASTER round required 16 min (the current implementation of FASTER was written for sparse EnergyGraphs rather than EnergyHyperGraphs with high edge density). Despite these hurdles, explicit inclusion of higher order terms is a viable option. The resulting rotamer combination had AMOEBA energy of $-5641.9 \text{ kJ mol}^{-1}$. Considering computational expense and accuracy, we recommend the trained second-order approximation over the third-order approximation for the polarizable packing problem.

Repacking Accuracy

Proper quantification of repacking accuracy statistics will require much larger test sets and the inclusion of a solvent model. That said, repacking protein G with the trained AMOEBA approxima-

tion was comparable with repacking using the OPLSaa energy function (also lacking a solvent model) and with the all-atom Rosetta energy function (which includes the Lazaridis–Karplus solvation terms). Specifically, 31, 32, or 30 out of 46 χ_1 values were recapitulated within 30° for the AMOEBA, OPLSaa, and Rosetta cases. One of our future goals is to apply the methods developed here to accurately approximate the multibody Poisson–Boltzmann continuum electrostatics solvation energy. Presumably, solvent screening will somewhat reduce the magnitude of protein–protein polarization effects.

Local Optimization of a Polar Sidechain Network

Global repacking calculations are not the only potential application for the decomposition and approximation scheme delineated in this work. Detailed energy models for interacting networks of polar sidechains, including sidechain conformational variability, might benefit analyses of salt bridge and hydrogen bond energetics or improve pK_a prediction algorithms. To demonstrate, we consider a toy problem: modeling an apo state for superoxide dismutase from thermophile *Alvinella pompejana* (3F7L). After removing the metals, we generated rotamers for the local amino acid network (Supporting Information Fig. 6a) of 11 rotatable sidechains (H44, H46, H61, H69, H78, H118 in addition to D81, V116, D122, T135, and R141). We included three protonation states for six histidine sidechains, for a total of 729 protonation state combinations and 6.1×10^{17} discrete conformations.

To train an energy approximation applicable to low-energy combinations, we followed the general procedure described above. First, we calculated explicit 1- and 2-body terms as above. Second, we obtained the putative minimum-energy rotamer combination for the approximate energy landscape (actual AMOEBA $E = -15343$ kJ mol⁻¹). Third, we collected a 2×10^5 step Monte Carlo trajectory using the approximate energy landscape. To ensure diverse training and test sets, we reduced the 6449 unique combinations encountered during Monte Carlo to 1184 combinations such that no two combinations were within one rotamer substitution. The first 500 combinations were selected for a training set. The second 500 combinations were selected for a test set. We calculated the actual AMOEBA energy for both sets of rotamer combinations. We next tried to use ridge regression to calculate 1-body correction terms, but the trained second-order approximation had a relatively high prediction rmsd of 16 kJ mol⁻¹. Instead, we proceeded by using ridge regression to calculate correction terms for all 1-body terms and all 2-body energy terms where $|E_{ij}| > 1$ kJ mol⁻¹. This trained second-order approximation recapitulated the training set with rmsd = 0.6 kJ mol⁻¹ and the test set with rmsd = 1.6 kJ mol⁻¹ (Supporting Information Fig. 6b). Most important, the trained approximation was useful; optimization using the trained approximation resulted in discovery of the rotamer solution with the most favorable AMOEBA energy = -15459 kJ mol⁻¹ (Supporting Information Fig. 6c). AMOEBA favors positively charged histidine, with the exception of H46, which accepts a hydrogen bond from R141. Combined, the calculations to reach this point required 76 min (Supporting Information Table I).

Given our particular interest in the relative energy of the histidine protonation state combinations, we also tried an alternate

training set comprised of the putative minimum-energy rotamer combination (according to the second-order approximate energy landscape) for all 729 protonation state combinations. Calculating the actual AMOEBA energy for each of these combinations, we found that the untrained second-order approximation correlated reasonably well ($R^2 = 0.99$) with the large differences in the AMOEBA energy for the protonation state combinations (which ranged between -15,450 and -12,606 kJ mol⁻¹) albeit with significant noise (rmsd = 63 kJ mol⁻¹). The untrained second-order model systematically disfavored solutions with fewer charged histidines. Training an energy model on these data (Supporting Information Fig. 6d) reduces the training set rmsd to 12 kJ mol⁻¹, but the resulting “big-picture” approximate energy model was insufficiently precise for discovery of improved low-energy combinations. Unlike the energy model trained on low-energy rotamer combinations, combinatorial optimization using the energy model trained on protonation state combinations did not identify a solution with a more favorable AMOEBA energy.

Discussion

The prototype calculations presented here are more computationally demanding than typical sidechain repacking calculations in current use. However, it is important to note that our computational burden consists of many small calculations. Current trends in CPU design toward multicore processors are very favorable for reducing the cost of these calculations. Also, the Folding@Home project provides an example of distributing TINKER calculations to tens of thousands of volunteer processors.³⁸ We are interested in methods for breaking large calculations (e.g., rotamer optimization with a multibody energy function) into small problems that may be calculated in parallel.

By demonstrating polarizable protein packing, we open the door for studies that compare the utility of polarizable and non-polarizable potentials for structure prediction. Additional trials on a large set of proteins will be required for statistically significant energy function comparison. Before undertaking this comparison, it will be crucial to model solvation and the crystallographic packing environment as accurately as possible. Otherwise, small differences in the accuracy of the protein model will be obscured. The natural solvent model to adopt will be the Poisson–Boltzmann continuum solvent model. First, an AMOEBA Poisson–Boltzmann treatment has previously been developed.⁶ Second, Mayo and coworkers have shown that Poisson–Boltzmann solvation energies can be roughly decomposed into 2-body terms (rmsd = 2.7 kJ mol⁻¹ for sidechain desolvation energy).^{2,3} The method described in this work should be suitable for developing high-accuracy approximations to the many-body Poisson–Boltzmann energy.

The strategies developed here are also applicable to classical fixed-backbone protein design. In this case, the population of rotamers at designed positions will expand to include new amino acids. For computational tractability, it will be important to select a modest number of positions to design. This is not necessarily a new limitation; designing a large number of positions can consume a prohibitive amount of memory.

In conclusion, we found that quantifying 3-body polarization effects resulted in an accurate approximation to the many-body AMOEBA energy. Most 3-body terms were small. In aggregate, 3-body effects were unfavorable, more often reflecting energetic frustration than cooperativity. Approximating the many-body energy allowed us to formulate a polarizable packing problem and to solve with the FASTER algorithm. Furthermore, our approximations could be significantly improved with ridge regression. Within the limited domain of low-energy rotamer combinations, the resulting energy models predicted the many-body energy with remarkable accuracy ($\text{rmsd} < 1 \text{ kJ mol}^{-1}$).

Acknowledgments

The authors thank Frances H. Arnold for support. The authors thank Phillip A. Romero, Gevorg Grigoryan, and an anonymous reviewer for useful suggestions. A.H.N. was supported by the Caltech Summer Undergraduate Research Fellowship program (SURF).

References

1. Street, A. G.; Mayo, S. L. *Fold Des* 1998, 3, 253.
2. Marshall, S. A.; Vizcarra, C. L.; Mayo, S. L. *Protein Sci* 2005, 14, 1293.
3. Vizcarra, C. L.; Zhang, N.; Marshall, S. A.; Wingreen, N. S.; Zeng, C.; Mayo, S. L. *J Comput Chem* 2008, 29, 1153.
4. Cieplak, P.; Caldwell, J.; Kollman, P. *J Comput Chem* 2001, 22, 1048.
5. Ren, P.; Ponder, J. W. *J Comput Chem* 2002, 23, 1497.
6. Schnieders, M. J.; Baker, N. A.; Ren, P.; Ponder, J. W. *J Chem Phys* 2007, 126, 124114.
7. Patel, S.; Brooks, C. L., III. *J Comput Chem* 2004, 25, 1.
8. Patel, S.; Mackerell, A. D., Jr.; Brooks, C. L., III. *J Comput Chem* 2004, 25, 1504.
9. Anisimov, V. M.; Lamoureux, G.; Vorobyov, I. V.; Huang, N.; Roux, B.; MacKerell, A. D. *J Chem Theory Comput* 2004, 1, 153.
10. Loksha, I. V.; Maiolo, J. R., III; Hong, C. W.; Ng, A.; Snow, C. D. *J Comput Chem* 2009, 30, 999.
11. Rohl, C. A.; Strauss, C. E.; Misura, K. M.; Baker, D. *Methods Enzymol* 2004, 383, 66.
12. Krivov, G. G.; Shapovalov, M. V.; Dunbrack, R. L., Jr. *Proteins: Struct Funct Bioinf* 2009, 77, 778.
13. Janin, J.; Wodak, S.; Levitt, M.; Maigret, B. *J Mol Biol* 1978, 125, 357.
14. Bower, M. J.; Cohen, F. E.; Dunbrack, R. L., Jr. *J Mol Biol* 1997, 267, 1268.
15. Yanover, C.; Schueler-Furman, O.; Weiss, Y. *J Comput Biol* 2008, 15, 899.
16. Desmet, J.; De Maeyer, M.; Lasters, I. *Pac Symp Biocomput* 1997, 122.
17. Xu, J.; Jiao, F.; Berger, B. *Proc IEEE Comput Syst Bioinform Conf* 2005, 247.
18. Gordon, D. B.; Hom, G. K.; Mayo, S. L.; Pierce, N. A. *J Comput Chem* 2003, 24, 232.
19. Pierce, N. A.; Spriet, J. A.; Desmet, J.; Mayo, S. L. *J Comput Chem* 2000, 21, 999.
20. Desmet, J.; Spriet, J.; Lasters, I. *Proteins* 2002, 48, 31.
21. Allen, B. D.; Mayo, S. L. *J Comput Chem* 2006, 27, 1071.
22. Zhang, J.; Liu, J. S. *PLoS Comput Biol* 2006, 2, e168.
23. Fromer, M.; Yanover, C. *Proteins: Struct Funct Bioinf* 2009, 75, 682.
24. Sciretti, D.; Bruscolini, P.; Pelizzola, A.; Pretti, M.; Jaramillo, A. *Proteins: Struct Funct Bioinf* 2009, 74, 176.
25. Kamisetty, H.; Xing, E. P.; Langmead, C. J. *J Comput Biol* 2008, 15, 755.
26. Hendsch, Z. S.; Tidor, B. *Protein Sci* 1994, 3, 211.
27. Kumar, S.; Nussinov, R. *J Mol Biol* 1999, 293, 1241.
28. Bosshard, H. R.; Marti, D. N.; Jelesarov, I. *J Mol Recognit* 2004, 17, 1.
29. Kumar, S.; Ma, B.; Tsai, C.-J.; Nussinov, R. *Proteins: Struct Funct Genet* 2000, 38, 368.
30. Luisi, D. L.; Snow, C. D.; Lin, J. J.; Hendsch, Z. S.; Tidor, B.; Raleigh, D. P. *Biochemistry* 2003, 42, 7050.
31. Stanton, C. L.; Houk, K. N. *J Chem Theory Comput* 2008, 4, 951.
32. Ponder, J. W.; Richards, F. M. *J Comput Chem* 1987, 8, 1016.
33. Das, R.; Baker, D. *Annu Rev Biochem* 2008, 77, 363.
34. Dahl, J.; Vandenberghe, L. CVXOPT. <http://abel.ee.ucla.edu/cvxopt>
35. Word, J. M.; Lovell, S. C.; Richardson, J. S.; Richardson, D. C. *J Mol Biol* 1999, 285, 1735.
36. Grigoryan, G.; Zhou, F.; Lustig, S. R.; Ceder, G.; Morgan, D.; Keating, A. E. *PLoS Comput Biol* 2006, 2, e63.
37. Grigoryan, G.; Reinke, A. W.; Keating, A. E. *Nature* 2009, 458, 859.
38. Snow, C. D.; Nguyen, H.; Pande, V. S.; Gruebele, M. *Nature* 2002, 420, 102.