Feature Space Resampling for Protein Conformational Search

Ben Blum
UC Berkeley
benblum@gmail.com

Michael I. Jordan
UC Berkeley
jordan@cs.berkeley.edu

David Baker
UW
dabaker@u.washington.edu

November 30, 2009

5 Supplementary Material

5.1 Native feature value predictor

Let $X_1, X_2, \ldots, X_n$ be all features from a single class (for example, all torsion features). Let $x_i^1, x_i^2, \ldots, x_i^{m_i}$ represent the possible values of feature X_i, and let x_i^* be the native value. Each feature value x_i^j—for instance, bin "B" of torsion angle 34 for protein 1dcj, or the beta barrel topology for protein 1acf—has a corresponding numeric property vector $[P_{samp}(x_i^j), lowE(x_i^j), minE(x_i^j), \ldots]$ consisting of a mix of energy and distribution statistics from the initial round of Rosetta models. Brief descriptions of the properties we use are given in Table I of the main article, along with the predictive power of each (as measured by the percentage of native feature values from all proteins in our benchmark that can be identified using this property alone). A justification for our choice of properties is given in the next section. Note that even though P_{samp} is in some sense "special," as it gives the prior belief distribution to be updated, it is treated just like the other properties for the purpose of the predictor.

Some properties are transformed, either to make their ranges comparable to one another or for reasons of mathematical convenience; for instance, the $P_{samp}(x_i^j)$ term is transformed to $\log(P_{samp}(x_i^j))$. Let $f_k(x_i^j)$ be the k^{th} transformed property of x_i^j and let $\Phi(x_i^j)$ be the vector of all single transformed property terms and their pairwise combinations. We compute the dot product between a weight vector β and $\Phi(x_i^j)$ via $\beta^t \Phi(x_i^j) = \sum_{k \neq k'} \beta_{k,k'} f_k(x_i^j) f_{k'}(x_i^j) + \sum_k \beta_k f_k(x_i^j)$. The presence of the pairwise combinations of features allows our model to take joint effects into

account. Given a weight vector β, the predicted probability that x_i^j is native in our model is

$$P_{pred}(x_i^j) = \frac{e^{\beta^t \Phi(x_i^j)}}{\sum_{j'=1}^{m_i} e^{\beta^t \Phi(x_i^{j'})}}.$$

The form of the predictor allows it to output P_{samp} unmodified, given the proper setting of the weights (one for $log(P_{samp})$ and zero for all others), so it is theoretically possible for the trained predictor to make no changes to the Rosetta sampling distribution. This is why we transform P_{samp} to $\log(P_{samp})$ in the property vector.

The free parameters in the predictor are the components of the weight vector β, which must be fitted by maximizing an objective function. Rather than fit some standard measure of belief accuracy, we aim to directly maximize the predictor's effectiveness as input to our resampling method. As outlined in Section 2.3 of the main article, the resampling step of our algorithm attempts to modify Rosetta search to sample features according to the distribution P_{pred} instead of according to P_{samp}. Accordingly, we use as an objective function the *sampling efficiency* of P_{pred}, which we define as $\prod_{i=1}^{n} P_{pred}(x_i^*)$, the estimated probability of encountering a fully native structure in a single Rosetta run with feature distribution P_{pred}. The inverse of this quantity can be regarded as an approximation, ignoring correlations between features, of the expected number of Rosetta samples required to produce a native-like structure. In order to incorporate training data from multiple proteins into the objective function, the sampling efficiencies of each of the proteins in the training set are multiplied (in fact, since we work on a log scale for numerical stability, their logarithms are summed).

The fitted weights for the various predictors (trained on a benchmark of 28 proteins) are shown in Table 5.1.

5.2 Resampling strategies

The probability of seeing the fully native feature string $\mathbf{x}^*$ in any single Rosetta trajectory is, under our independence model, $P_{resamp}(\mathbf{x}^*) = \prod_{i=1}^{k} P_{resamp}(X_i = x_i^*)$. But we are not sure, a priori, of which values are in fact the native ones. Under our belief distribution P_{pred}, the chance that $\mathbf{x}$ is the native feature string is (making the same independence assumption for beliefs as for sampling distributions) $P_{pred}(\mathbf{x}) = \prod_{i=1}^{n} P_{pred}(x_i)$. The *expected* chance of seeing the native in any given sample, with respect to our beliefs, is then $\sum_{\mathbf{x}} P_{pred}(\mathbf{x}) P_{resamp}(\mathbf{x})$. Maximization of this expectation with respect to P_{resamp}, subject to normalization constraints, can be solved in closed form using a standard Lagrangian multiplier argument. The maximum is given by $P_{resamp}(\hat{\mathbf{x}}) = 1$ for $\hat{\mathbf{x}} = \operatorname{argmax} P_{pred}(\mathbf{x})$ and 0 elsewhere. Given just a single sample, the optimal strategy is to try our single best guess for the native feature string.

At the other extreme, in the limit of infinite samples, the chance of sampling feature string $\mathbf{x}$ at least once is the step function $\mathcal{I}(P_{resamp}(\mathbf{x}) > 0)$, which takes the value 1 if $P_{resamp}(\mathbf{x}) > 0$ and 0 otherwise. The expected chance of seeing a native structure is then $\sum_{\mathbf{x}} P_{pred}(\mathbf{x}) \mathcal{I}(P_{resamp}(\mathbf{x}) > 0)$. This expectation reaches its optimum value of 1 whenever $P_{resamp}(\mathbf{x}) > 0$ for all $\mathbf{x}$ such that $P_{pred}(\mathbf{x}) > 0$. For very large numbers of samples, the optimum strategy is

Native feature value predictors

Torsion predictor

	P_{samp}	*frag*	*lowE*	*minE*	*loop*
P_{samp}	1.11	-0.16	-0.17	-0.43	-0.083
frag		0.51	0.35	-0.022	-0.16
lowE			-0.11	0.094	0.21
minE				-0.90	-0.24
loop					0.29

Secondary structure predictor

	P_{samp}	*minE*	*lowE*	*psipred*	*jufo*
P_{samp}	1.36	-0.35	-0.29	2.35	-0.24
minE		-0.27	-0.23	1.39	0.081
lowE			0.43	0.28	-0.38
psipred				1.10	-0.54
jufo					0.06

Topology predictor

	P_{samp}	*co*	*lowE*	$minE^{all}$
P_{samp}	1.49	-1.57	0.82	-1.37
co		-0.22	-0.043	-1.45
lowE			4.19	-2.11
$minE^{all}$				2.27

Register predictor

	P^{all}_{samp}	*minE*	$lowE^{all}$	*bulge*
P^{all}_{samp}	0.53	-0.042	0.13	1.06
minE		-0.86	0.29	0.50
$lowE^{all}$			0.55	0.074
bulge				0.50

Contact predictor

	P^{all}_{samp}	$lowE^{all}$	*edgedist*	*oddpleat*
P^{all}_{samp}	1.05	2.57	1.79	0.00038
$lowE^{all}$		-0.29	1.28	-0.48
edgedist			-0.14	-0.12
oddpleat				-0.45

Table III: Predictor weights for the five feature classes. Weights for individual feature value properties are on the diagonal, weights for pairwise terms are elsewhere.

to spread sampling as evenly as possible. If, for instance, we are permitted as many samples as there are joint feature strings, the optimal strategy is to try each string exactly once.

The strategy of setting $P_{resamp} = P_{pred}$ maximizes $\sum_{\mathbf{x}} P_{pred}(\mathbf{x}) \log(P_{resamp}(\mathbf{x}))$, in which the term $\log(P_{resamp}(\mathbf{x}))$ is intermediate between the objective functions $P_{resamp}(\mathbf{x})$, which grows linearly in the value of P_{resamp}, and $\mathcal{I}(P_{resamp}(\mathbf{x}) \neq 0)$, which jumps immediately to 1. Maximizing $\sum_{\mathbf{x}} P_{pred}(\mathbf{x}) \log P_{resamp}(\mathbf{x})$ is equivalent to minimizing $\sum_{\mathbf{x}} P_{pred}(\mathbf{x}) \log(1/P_{resamp}(\mathbf{x}))$, the expected number of samples required to find a fully native feature string.

5.3 Impact of energy function inaccuracy

Because the weights in the predictor are learned from examples, they take into account the unreliability of the energy function as an indicator of nativeness, particularly outside the energy funnel near the native conformation. Our resampling algorithm therefore has the potential to be less vulnerable to false valleys in the energy function than other resampling methods. We can roughly identify proteins in the benchmark for which conformations in false valleys received the lowest energies by looking at the difference between the RMSD of the lowest-energy models and of the lowest-RMSD models. There were eight proteins from our benchmark in which the median RMSD of the 25 lowest-energy models from the initial round was at least 3Å worse than the 1% RMSD of the population at large. The average improvement under resampling of the RMSD of the first prediction for these targets was 3.59Å, compared to an average 1.77Å for all targets, and the average improvement of the best-of-five prediction was 1.97Å, compared to 0.42Å

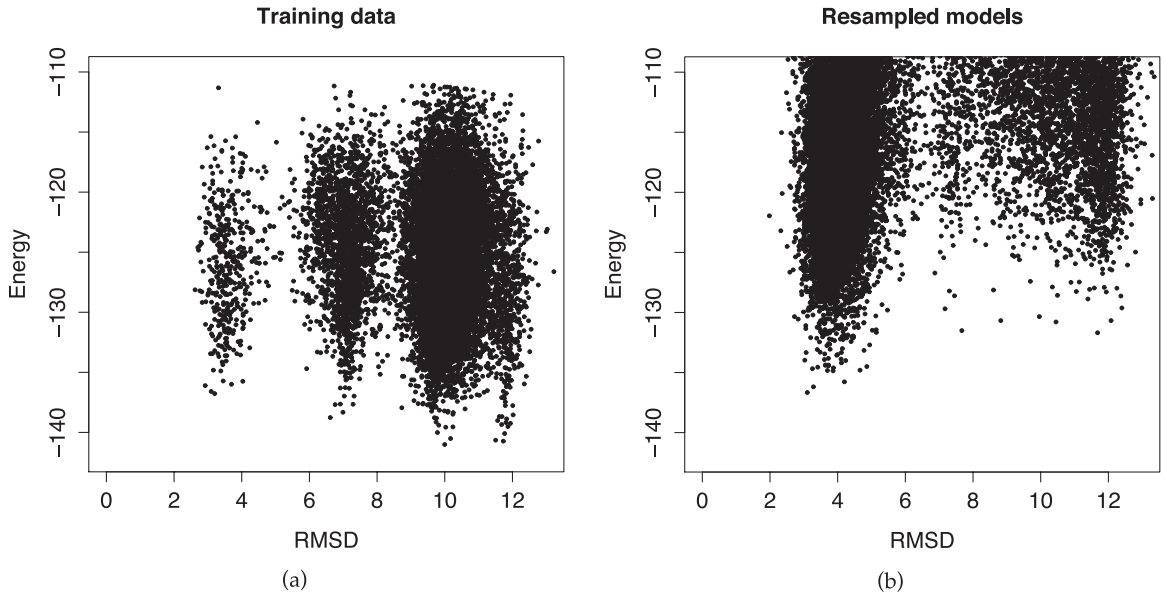

Figure 5: Energy versus RMSD for input and resampling rounds for 1n0u. Each point represents a model produced by Rosetta search. (a) Plot of energy versus RMSD from native for models used as input data for the native feature value predictor. Models fall into several distinct clusters. The largest and lowest-energy clusters are far from the native conformation. (b) Plot of energy versus RMSD for the models resampled using the output of the predictor. The cluster nearest to the native conformation is now the largest, and includes the lowest-energy models.

for all targets.

The resistance of our methods to energy pathologies is readily apparent in the case of 1n0u, a target for which most Rosetta sampling concentrated on an incorrect beta topology. The lowest energy first-round models are concentrated at about 10Å (Figure 5(a)). Only 2% of the first-round models have the native topology. However, the predictor still identifies the native topology with 48% confidence, so in the resampling round, 48% of models have the native topology (Figure 5(b)), including the lowest-energy models.