OPTIMIZING BAYES ERROR FOR PROTEIN STRUCTURE MODEL SELECTION BY STABILITY MUTAGENESIS

Xiaoduan Ye[1], Alan M. Friedman[2*], and Chris Bailey-Kellogg[1*]
[1]*Department of Computer Science, Dartmouth College*
[2]*Department of Biological Sciences, Markey Center for Structural Biology, Purdue Cancer Center, and Bindley Bioscience Center, Purdue University*

Site-directed mutagenesis affects protein stability in a manner dependent on the local structural environment of the mutated residue; e.g., a hydrophobic to polar substitution would behave differently in the core vs. on the surface of the protein. Thus site-directed mutagenesis followed by stability measurement enables evaluation of and selection among predicted structure models, based on consistency between predicted and experimental stability changes ($\Delta\Delta G^\circ$ values). This paper develops a method for planning a set of individual site-directed mutations for protein structure model selection, so as to minimize the Bayes error, i.e., the probability of choosing the wrong model. While in general it is hard to calculate exactly the multi-dimensional Bayes error defined by a set of mutations, we leverage the structure of "$\Delta\Delta G^\circ$ space" to develop tight upper and lower bounds. We further develop a lower bound on the Bayes error of any plan that uses a fixed number of mutations from a set of candidates. We use this bound in a branch-and-bound planning algorithm to find optimal and near-optimal plans. We demonstrate the significance and effectiveness of this approach in planning mutations for elucidating the structure of the pTfa chaperone protein from bacteriophage lambda.

1. INTRODUCTION

With the extensive development of genome projects, more and more protein sequences are available. Unfortunately, while structural genomics efforts have greatly expanded the set of experimentally determined protein structures, the Protein Data Bank (PDB) still has entries for only about 1% of the protein sequences in UniProt. A significant part of the gap between the sequence and structure determination lies with difficulties in crystallization; among the 75104 targets (45391 cloned) in phase one of the Protein Structure Initiative, only 3311 crystallized and only 1307 of these crystals provided sufficient diffraction[1]. At the same time, it has been suggested that only a small number (perhaps a few thousand[2]) of distinct structural organizations, or "folds," exist among naturally-occurring proteins, and many of them can already be found in the current PDB[3]. Therefore, structure *elucidation* (as opposed to experimental structure *determination*) may soon devolve to selecting the correct model among those generated from existing templates.

Since many more proteins are available for structural studies than can be handled by crystallography, we have been developing integrated computational-experimental methods that use relatively rapid, targeted biochemical/biophysical experiments to select among predicted structure models, based on consistency between predicted and observed experimental measurements[4]. Purely computational protein structure prediction methods[5–8] can often produce models close to the correct structure. However, as the series of Critical Assessment of Structure Prediction (CASP) contests illustrates[9], it remains difficult for any method to always select the correct model, particularly in cases where low sequence identity to templates precludes homology modeling. The best model is often among a pool of highly ranked models, but not necessarily the highest-ranked one. Furthermore, different methods often employ different scoring functions and yield different rankings for the same models. Thus using rapid, planned experiments to select the correct one(s) from a given set of predicted models combines the strengths of both computation and experimentation.

This paper focuses on an approach we call "stability mutagenesis," which exploits the relationship between protein structure and thermodynamic stability to perform model selection. A number of methods[10–15] are available for predicting changes

*Contact authors. CBK: 6211 Sudikoff Laboratory, Hanover, NH 03755, USA; cbk@cs.dartmouth.edu. AMF: Lilly Hall, Purdue University, West Lafayette, IN 47907, USA; afried@purdue.edu.

in unfolding free energy ($\Delta\Delta G^\circ$) upon site-directed mutagenesis (i.e., substitution of one amino acid for another at a specific position). These prediction methods provide good accuracy in the aggregate or for defined subsets of mutations, e.g., the FOLD-X method achieved a global correlation of 0.83 between the predicted and experimental $\Delta\Delta G^\circ$ values for 95% of more than 1000 point mutations, with a standard deviation of 0.81 kcal/mol[13]. Since different structure models place some of their equivalent residues in different environments, they yield different predicted $\Delta\Delta G^\circ$ values for those residues. The consistency between predicted and experimentally determined $\Delta\Delta G^\circ$ values thus allows selecting the correct model(s) from a given set.

This paper develops a method for planning the most informative stability mutagenesis experiments for selecting among a given set of protein structure models. In particular, we seek to minimize the expected probability of choosing a wrong model, i.e., the Bayes error. It is difficult to compute exactly the Bayes error in multiple dimensions (here, for sets of mutations), and the general problem of estimating and bounding it has received considerable attention[16–19]. We take advantage of the particular structure of our mutagenesis planning problem in order to derive tight upper and lower bounds on the Bayes error for "$\Delta\Delta G^\circ$ space." In order to efficiently find an optimal set of mutations, we develop a lower bound on the Bayes error of any plan that uses a fixed number of mutations from a set of candidates, along with a branch-and-bound algorithm to identify optimal and near-optimal plans.

2. METHODS

2.1. Bayes Error Bounds

Let $S = \{s_1, s_2, ...s_n\}$ be a given set of predicted protein structure models, and X be a vector of random variables representing the experimental $\Delta\Delta G^\circ$ values with Normal errors (as is standardly assumed). Then each model can be represented as a conditional distribution in the "$\Delta\Delta G^\circ$ space" (Fig. 1), in which each dimension has the $\Delta\Delta G^\circ$ value for one mutation. That is,

$$p(X|s_i) = \mathcal{N}(\boldsymbol{\mu}_i, \sigma^2 I) \tag{1}$$

where $\boldsymbol{\mu}_i$ is the vector of expected $\Delta\Delta G^\circ$ values for model i, and the variance $\sigma^2 I$ (where I is the identity matrix) is mutation independent and model independent.

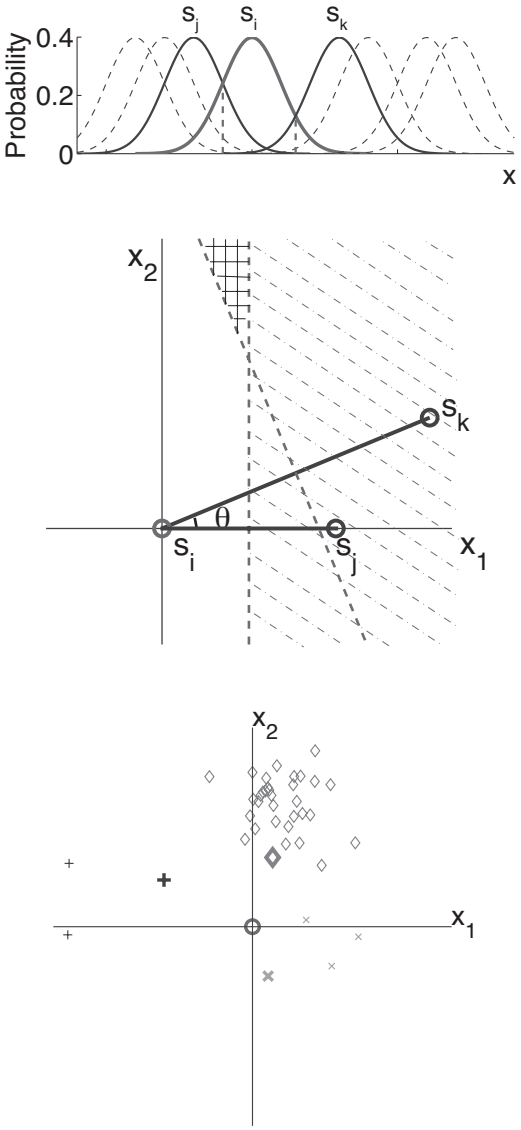

Fig. 1. Intuition for upper bound on ε_i, the Bayes error conditioned on model s_i. (top) In the 1D case, ε_i is determined by s_j and s_k, the closest neighbors on each side of s_i, with no effect from other models (dashed curves). (middle) In higher-dimensional cases, multiple models are unlikely to be collinear. However, if the angle between $\overrightarrow{s_i s_j}$ and $\overrightarrow{s_i s_k}$ is small and s_k is not closer to s_i than s_j is, adding s_k will only increase ε_i by a small amount, the integral of $p(X|s_i)$ over the "#" shaded area. (bottom) To find representative models that are "closest" to s_i, other models are represented as vectors from s_i and hierarchically clustered w.r.t. their angles. Here there are three clusters (different markers), each represented by the model closest to s_i (bold markers) for the purposes of error bounds.

Once the experimental $\Delta\Delta G^\circ$ values have been measured, we will choose the model with the maximum posterior probability. In considering a possible set of mutations during experiment planning, we don't know what the resulting experimental $\Delta\Delta G^\circ$ values will be. Thus we integrate over the possible values, computing the Bayes error ε, formulated as:

$$\varepsilon = \sum_{i=1}^{n} P(s_i)\varepsilon_i \tag{2}$$

where $P(s_i)$ is the prior probability of model s_i and ε_i is the conditional error given that model s_i is correct. By "correct" we mean that the distribution of the measurements X w.r.t. this model is very similar to that for the "true" protein structure. For simplicity, we assume a uniform prior for models, but all discussion applies to the case of non-uniform priors. We define ε_i as:

$$\varepsilon_i = P_i\{p(X|s_i) < \max_{j\neq i} p(X|s_j)\} \tag{3}$$

Here and in the following we use notation $P_i\{e\}$ for the probability of predicate e w.r.t. model s_i:

$$P_i\{e\} = \int p(X|s_i)\cdot I\{e\}\mathrm{d}X \tag{4}$$

where the integral is taken over all possible $\Delta\Delta G^\circ$ values and the indicator function $I\{e\}$ returns 1 if predicate e is true, or 0 if false. The predicate in Eq. 3 evaluates whether a wrong model is selected because the experimental data X is more consistent with it than with the correct model. Weighted integration over all possible datasets then calculates the total probability of error.

Straightforward initial bounds on ε_i can be derived from the Bonferroni inequalities[20]:

$$\varepsilon_i \leq \sum_{j\neq i} P_i\{p(X|s_i) < p(X|s_j)\} \tag{5}$$

$$\varepsilon_i \geq \sum_{j\neq i} P_i\{p(X|s_i) < p(X|s_j)\} - \sum_{j<k\neq i} P_i\{p(X|s_i) < \min(p(X|s_j),\, p(X|s_k))\} \tag{6}$$

The union bound (Eq. 5) evaluates the probability that at least one of the wrong models beats the correct one, which is at most the sum of the probabilities of each individual wrong model beating the correct one. Eq. 6 subtracts out the potential "double-counting" in the union bound, when multiple wrong models beat the correct one but some are better than others. Both bounds are easy to calculate, but are too loose for our purposes here.

Since we assume a common variance for all mutations in all models, the error probability is completely determined by the relative distances among the distribution means. The top and middle panels of Fig. 1 illustrate that in cases where the means are nearly collinear, ε_i is much less than the sum of the individual error probabilities (i.e., the union bound). Conditioning on model s_i, we shift the coordinate system so that $\boldsymbol{\mu}_i$ is at the origin and the rest of the models are represented as vectors from the origin. We cluster these vectors (Fig. 1, bottom) into disjoint sets C_t for $t = 1, 2, \ldots$. We discuss our clustering method below, but for any set of clusters, the following inequality holds:

$$\varepsilon_i \leq \sum_t P_i\{p(X|s_i) < \max_{j\in C_t} p(X|s_j)\} \tag{7}$$

The difference between Eq. 7 and Eq. 5 is that in Eq. 7 the Bonferroni inequality is applied on clusters instead of individual models. Choosing a representative model s_{j_t} from cluster C_t, we have

$$\begin{aligned} &P_i\{p(X|s_i) < \max_{j\in C_t} p(X|s_j)\} \\ &= P_i\{p(X|s_i) < p(X|s_{j_t})\} \\ &\quad + P_i\{p(X|s_{j_t}) < p(X|s_i) < \max_{j\in C_t, j\neq j_t} p(X|s_j)\} \quad (8) \\ &\leq P_i\{p(X|s_i) < p(X|s_{j_t})\} \\ &\quad + \sum_{j\in C_t, j\neq j_t} P_i\{p(X|s_{j_t}) < p(X|s_i) < p(X|s_j)\} \quad (9) \end{aligned}$$

Eq. 8 is just a rewriting of the probability; either model s_{j_t} beats s_i or some other models in cluster C_t beat it. Eq. 9 is obtained by applying the union bound on the second term of Eq. 8, where the first and second terms correspond to the integral of $p(X|s_i)$ over the stripe-shaded area and the "#" shaded area in the middle panel of Fig. 1, respectively.

Turning to the lower bound, we note that Eq. 6 could be very loose (even negative) if models are highly dependent, because the number of pairwise joint probabilities subtracted out could be much larger than the number of individual probabilities added in. For example, consider a variation of the top panel in Fig. 1 where s_j and the models to the left of it have nearly identical distributions and similarly for s_k and the models to the right of it, such that $P_i\{p(X|s_i) < p(X|s_j)\} \approx \epsilon$ for all wrong models. Then Eq. 6 gives a lower bound of -2ϵ (one added and two pairs subtracted on each side). However, we can obtain a tighter lower bound by using a subset of the models that are highly independent;

in the example, one from the left and one from the right. More generally, still conditioning on s_i, let $S' \subset S - \{s_i\}$ be a subset of the remaining models. Then

$$\varepsilon_i \geq P_i\{p(X|s_i) < \max_{j \in S'} p(X|s_j)\} \quad (10)$$

$$\geq \sum_{j \in S'} P_i\{p(X|s_i) < p(X|s_j)\}$$

$$- \sum_{j<k \in S'} P_i\{p(X|s_i) < \min(p(X|s_j), p(X|s_k))\} \quad (11)$$

Eq. 10 holds because the probability that a model in a superset beats s_i is always at least the probability that a model in a subset does. Eq. 11 is just the Bonferroni inequality applied to S'.

We now have lower and upper bounds that are tighter than simply applying the Bonferroni inequalities. The tightness depends on the choices of clustering method (upper bound) and model subset (lower bound). In fact, we can readily trade off tightness and computation, using more, finer clusters and more trial subsets in order to obtain tighter bounds. For the results presented below, we employ an agglomerative approach to cluster models, with distance between two clusters defined as the maximum angle between any two vectors in them. A cutoff θ determines the number of clusters, and then the model with the smallest distance to s_i in each cluster is selected as the representative model for the cluster (bottom panel of Fig. 1). We also use the representative models as the model subset for the lower bound, because these models are likely to be relatively independent and thus the pairwise joint probabilities are smaller and the lower bound tighter. Since the quality of the bounds depends on the choice of θ and the best choice could be model specific and different for the upper bound and the lower bound, we simply try three different values: $\pi/4$, $\pi/3$, and $\pi/2$. The running time is only three times that of using a fixed cutoff, and we found that the result is significantly improved in practice.

2.2. Planning Algorithm

If there are only a few candidate mutations, or a few are to be selected for a plan, we can enumerate all possible plans, calculate their upper and lower bounds, and choose a good one. In terms of Bayes error, plan A is definitely better than plan B if the upper bound for A is less than the lower bound for B. In practice, the computational complexity of such a brute force method becomes prohibitive for even a modest number of mutations.

In cases where the exhaustive method is infeasible, we can use a greedy approach to minimize the upper bound on the Bayes error—select mutations one by one, minimizing the upper bound on Bayes error at each step. A tight upper bound will allow us to identify a set of selected mutations guaranteed to be of high quality. However, we still do not know how close a plan is to the optimal one, and the greedy plan may be far from optimal.

To evaluate the optimality of a given plan M, we compute a lower bound on how its Bayes error compares to that of the best possible (though unknown) plan that uses the same number of mutations from a set of candidates C:

$$\text{Optimality}(M, C) \geq \frac{lb(C, |M|)}{ub(M)} \quad (12)$$

where $ub(M)$ is the upper bound we previously discussed (Eq. 7, Eq. 9) and we develop below $lb(C, |M|)$, a lower bound on the Bayes error of the optimal plan. An Optimality score close to 1 indicates that the plan is guaranteed to be near optimal. A plan with a lower score may still be good, but we just cannot prove it with our bounds. The Optimality score also supports the branch-and-bound algorithm we develop below: we can ignore all plans chosen from mutations in C if the score in Eq. 12 is greater than 1 for some plan M.

To derive $lb(C, |M|)$, the lower bound on the optimal plan, we start from a lower bound on the Bayes error based on pairwise risk functions developed for multi-hypothesis testing[19]:

$$\varepsilon \geq \left(\frac{2}{n}\right)^2 \cdot \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \varepsilon_{ij} \quad (13)$$

We can also prove the following lemma.

Lemma 1. *Let $d^2 = \sum_{i=1}^{n} d_i^2$ be the sum of squares of n positive real numbers d_i, $i = 1, 2, ..., n$, and let ε_i be the cumulative density of Normal distribution $\mathcal{N}(0, \sigma)$ at point $-d_i/2$. Then for a fixed value of d^2, $\sum_{i=1}^{n} \varepsilon_i$ is minimized when $d_i = d_j$ for $1 \leq i, j \leq n$.*

Proof. Suppose we can find $d_i = b$ and $d_j = a$ that are not equal, say $0 < b < a$, and let $c = \sqrt{(a^2 + b^2)/2}$ be new equal values for d_i and d_j, so that the sum of squared values d^2 is not changed. ε_i

decreases more than ε_j increases:

$$\frac{1}{\sqrt{2\pi}} \int_{b/2}^{c/2} e^{-\frac{x^2}{2}} \mathrm{d}x > \frac{1}{\sqrt{2\pi}} \int_{c/2}^{a/2} e^{-\frac{x^2}{2}} \mathrm{d}x \qquad (14)$$

This follows from the fact that $c - b > a - c$, so that the first integral region is larger than the second; along with the fact that the density is higher in the first region because it is closer to the mean. Thus equalizing a pair reduces the total error, and if we could equalize all pairs, we would minimize the sum. □

Combining Lemma 1 and Eq. 13, we have

$$\varepsilon \geq \frac{2(n-1)}{n} \int_{-\infty}^{-r} \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2} \mathrm{d}x \qquad (15)$$

where $r = \frac{1}{2}\sqrt{d^2/\binom{n}{2}}$ and d^2 is the sum of squared distances among all model distribution means:

$$d^2 = \sum_{i<j} \sum_{k} (\mu_{ki} - \mu_{kj})^2 = \sum_{k} \sum_{i<j} (\mu_{ki} - \mu_{kj})^2 \quad (16)$$

where μ_{ki} is the predicted $\Delta\Delta G^\circ$ value of the k^{th} mutation according to model s_i. Since the inner sum on the right-hand side of Eq. 16 is for only one mutation, we can easily maximize d^2 by independently choosing mutations according to their sums of squared distances over all models. With d^2 thus maximized, the lower bound in Eq. 15 is minimized and becomes a lower bound for any plan of the same size, including the optimal plan.

With the lower bound on the optimal plan in place, let us now turn to the branch-and-bound search for an experiment plan (set of mutations). First let us define the structure of the search tree we will use. In this tree, a node corresponds to an index into the list of possible mutations. All mutations on the path to the root have been eliminated, all those with smaller indices but not on the path to the root have been selected, and all those with larger indices are still candidates. In the example in Fig. 2, at the starred node (index 4), mutations A2G, F3A and R4A (indices 1, 2 and 4) have been eliminated, mutation F3L (index 3) has been selected, and mutations R4G and M5A (indices 5 and 6) are still candidates.

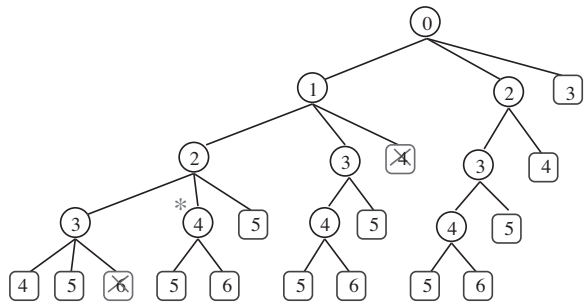

Fig. 2. Example branch and bound search tree structure for choosing two from six mutations at four positions, {A2G, F3A, F3L, R4A, R4G, M5A}. Circles are internal nodes and squares leaf nodes (i.e., plans of the desired size). 'X's indicate violation of a constraint allowing at most one mutation per position.

We now develop our branch-and-bound algorithm, MutPlanBB (Fig. 3), to explore this search tree, using the lower bound on the Bayes error of the optimal plan, along with our earlier upper bound on the Bayes error of a specific plan. The algorithm prunes a node (and its subtree) if the Optimality with respect to the best plan found so far (and thus also to the best one) is more than a user-specified threshold λ. If $\lambda > 1.0$, sub-optimal plans will also be listed. If $\lambda < 1.0$, some good plans might be missed in order to speed up the algorithm, but plans that are "really" good are guaranteed to be kept. For example, if $\lambda = 0.5$, any plan with Bayes error at most half of ub^*, the best upper bound found in the search, cannot be missed. Since ub^* bounds the Bayes error of the optimal plan, this means that we would find all plans with Bayes error no more than half the upper bound on the optimal error. The default value of λ is 1.0, so that all good plans will be listed.

An additional function in the search, *constraintsSatisfied*, checks user-specified constraints, e.g., indicating that at most one mutation per position can be selected. In our branch-and-bound algorithm, we assume that a constraint is monotonic—it is violated by a superset of mutations if it is violated by any subset—as is the case with the one-mutation-per-position constraint. In fact, we can avoid visiting right siblings if any monotonic constraint is violated (a simple modification to the pseudocode in Fig. 3). We can also modify the algorithm in Fig. 3 to handle non-monotonic constraints: only check the satisfaction of constraints on complete plans.

```
MutPlanBB(m, λ, ub*, Ψ, S, C)
  if |S| + |C| = m
      # only one possible plan
      S ← S ⋃ C
      C ← ∅
  if constraintsSatisfied(S) and lb(S ⋃ C, m)/ub* ≤ λ
      if |S| = m
          Ψ ← Ψ ⋃ {S}
          ub* ← min{ub(S), ub*}
      else
          for i from 1 to m − |S| + 1
              # discard C[i] for the i^th child
              S' ← S ⋃ C[1..i − 1]
              C' ← C[i + 1..|C|]
              [ub*, Ψ] ← MutPlanBB(m, λ, ub*, Ψ, S', C')
  return [ub*, Ψ]
```

Fig. 3. Branch and bound algorithm for mutagenesis planning. The inputs include the desired size of plan (m), pruning cutoff (λ), the best upper bound (ub^*) and good plans (Ψ) so far (initially from the greedy approach), and sets of selected and candidate mutations (S and C) at the current node.

There is clearly an exponential number of nodes in the search tree; practical efficiency is attained by effective pruning high up in the tree, so that many nodes need be explicitly visited. As discussed, if desired, the bounds are "tunable"—at more computational cost, we can obtain tighter bounds and thus better pruning. In addition, in order to increase the pruning rate, we initially sort all mutations in ascending order of upper bound on Bayes error, which is easy to calculate in the 1D (single mutation) case. This heuristic[21] structures the search so as to try to exclude good mutations first, so that the error of the remaining mutations is larger, as is the chance of pruning left subtrees, which are larger (see again the tree in Fig. 2). In practice, we found that this reordering improves the pruning rate significantly. Although we can reorder mutations at each level of the search tree, the cost of the sorting may not be worth the benefit, which is not likely to be as significant as the initial sorting.

The cost of visiting an internal node is a table lookup of the Normal cumulative density function (Eq. 15), to compute the lower bound on the optimal plan ($lb(S \bigcup C, m)$). Visiting a leaf node is more expensive, as it requires computing the upper bound ($ub(S)$), by numerical integration in 2D space (Eq. 9).

2.3. Accounting for Bias

While $\Delta\Delta G^\circ$ predictors are based on general determinants of protein stability, some proteins are naturally easier or harder to destabilize than others are. This could lead to bias in the experimental data, which, without care, could result in selection of the incorrect model. For example, if the plan included mutations in which one model was predominantly predicted to be more destabilized than the others, that model would tend to be favored if the protein were relatively easy to destabilize independent of mutation choice. If we knew the bias for a protein, as a single number or a distribution, we could incorporate it into the prediction distribution $p(X|s_i)$. We assume, however, that we only know the range of a constant bias (i.e., a constant offset to $\Delta\Delta G^\circ$, from anywhere in a specified range), because that is a fairly realistic situation in practice.

Conditioning on model s_i (so that its predictions should be biased, as it reflects the native state), its distribution is moved from $\boldsymbol{\mu}_i$ to $\boldsymbol{\mu}_i' = \boldsymbol{\mu}_i + \delta \cdot \mathbf{1}$, for δ in the specified range. Significantly, the error bound expressions (Eq. 9, Eq. 11) are all in terms of only two or three models. Thus, the vector $\overrightarrow{\boldsymbol{\mu}_i\boldsymbol{\mu}_i'}$ can be decomposed into two perpendicular vectors, one parallel and the other orthogonal to line $\boldsymbol{\mu}_i\boldsymbol{\mu}_j$ or plane $\boldsymbol{\mu}_i\boldsymbol{\mu}_j\boldsymbol{\mu}_k$. Since the orthogonal vector does not provide any information for model discrimination (distributions $p(X|s_i)$, $p(X|s_j)$ and $p(X|s_k)$ have identical projections in that direction), such projections lose no information for discrimination. In our implementation, we try bias values within the range $[-2, 2]$ kcal/mol at a resolution of 0.1 and use the worst case (maximum upper bound of Bayes error) as the robustness measurement of a plan. A robust plan will have a biased error bound close to the unbiased one.

3. RESULTS

We employ one representative $\Delta\Delta G^\circ$ prediction method, FOLD-X[13], which predicts stability by developing an empirical effective energy potential including van der Waals, solvation, hydrogen bonding, and electrostatics, and training its parameters and weights using stability data from wild-type and site-directed mutants. We use WHAT-IF[22] to model the mutant structures and version 2.5 of FOLD-X[13] to calculate the stability of mutant and wild-type pro-

teins, and thereby $\Delta\Delta G^\circ$.

In a planning-based framework, we have the luxury of considering only those experiments which we believe to be most reliable. Thus we exclude substitutions involving Cys and Pro, at the first residue position, and in poorly modeled regions. We also adopt FOLD-X's restriction allowing only "non-augmenting" mutations, those whose mutant structures are easy to predict because they involve either a substitution to a smaller sidechain (e.g., Ile → Val) or direct replacement of atoms (e.g., Asp → Asn).

We evaluated the tightness of our bounds using models deposited for a number of different CASP targets. Fig. 4 shows the bounds for four representative test cases from CASP 5, each consisting of the top 10 models by GDT_TS z-score. (Other test cases displayed similar behavior.) Our upper bound is much less than the union bound, and quite close to our lower bound. We have tightly bracketed the Bayes error.

We put our planning mechanism into practice on the pTfa protein of bacteriophage lambda. Lambda pTfa is a small 194 amino acid protein, and, except for our cross-linking work[4, 23], no structural information is available for it or any homolog. The pTfa fragment from residues 1 to 108 forms a stable well-expressed protein that unfolds cooperatively in urea by a two-step mechanism[24]. We previously constructed three high-quality threading models of pTfa 1–108[4], with templates from chaperone DnaK substrate binding domain (PDB id 1dkz), heme chaperone Ccme (PDB id 1liz), and mRNA capping enzyme (PDB id 1ckm). There are altogether 2052 possible substitutions, 19 at each of 108 positions. After applying the restrictions described above, we were left with 192 possible mutations at 77 positions.

Our algorithm first finds a plan by greedily selecting mutations. As Fig. 5 illustrates, the Bayes error has converged fairly well by about 6 mutations (intuitively, 3 mutations distinguishing each pair of models). The Optimality score (Eq. 12) of the six-mutation greedy plan is about 0.6, which means that the Bayes error is within a factor of two of the optimal value. Therefore, we expect a high pruning rate in the branch-and-bound algorithm using this greedy plan as the initial solution. The greedy plan is good in the unbiased case, with a Bayes error of 1.4%. However, we found that with a bias range of $[-2, 2]$ kcal/mol, the Bayes error goes up to 17%.

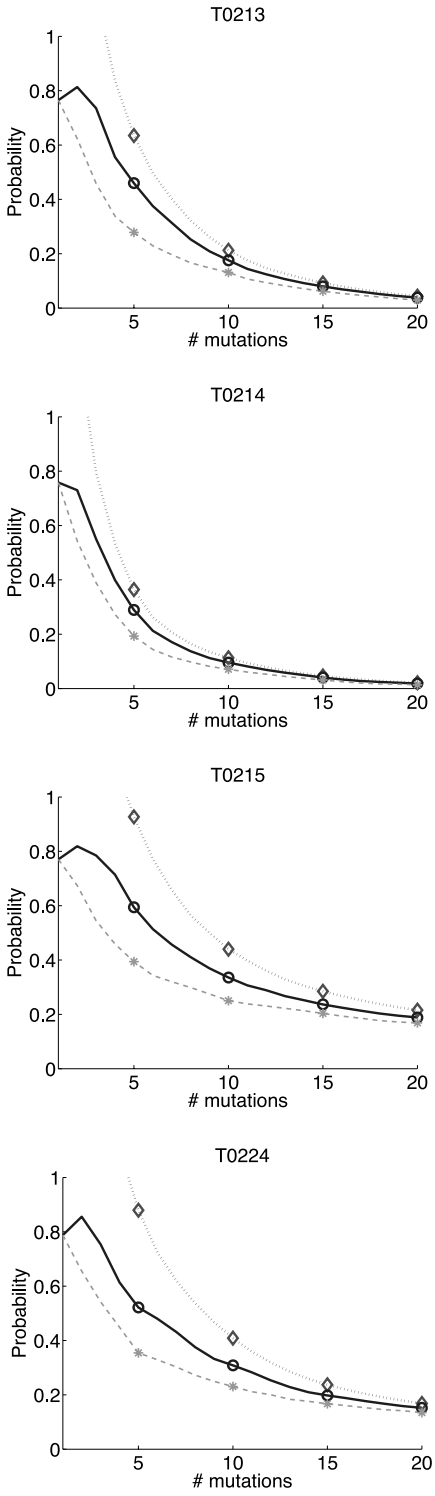

Fig. 4. Error probabilities for greedy plans for four CASP targets: union bound (dotted), tight upper bound (solid), and tight lower bound (dashed).

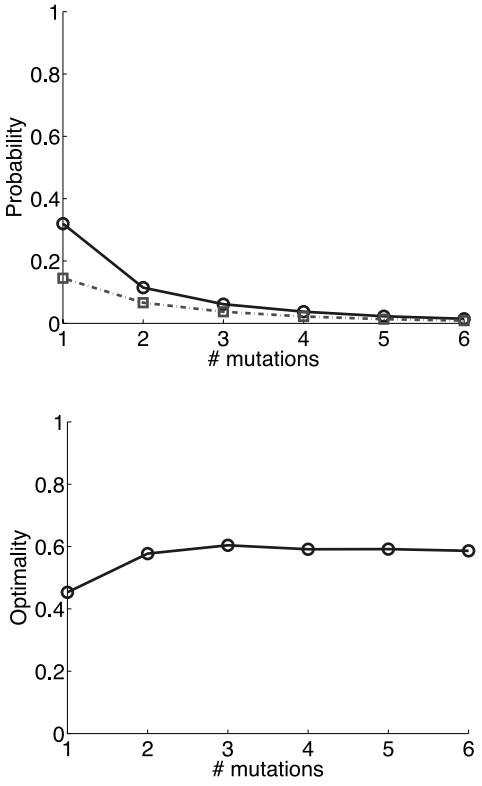

Fig. 5. Greedy plans for three pTfa models. (top) Bayes error of greedy plans (solid line, circles) and lower bound of the optimal plan of the same size (dash-dotted line, squares). (bottom) Optimality, as defined in Eq. 12, of greedy plans.

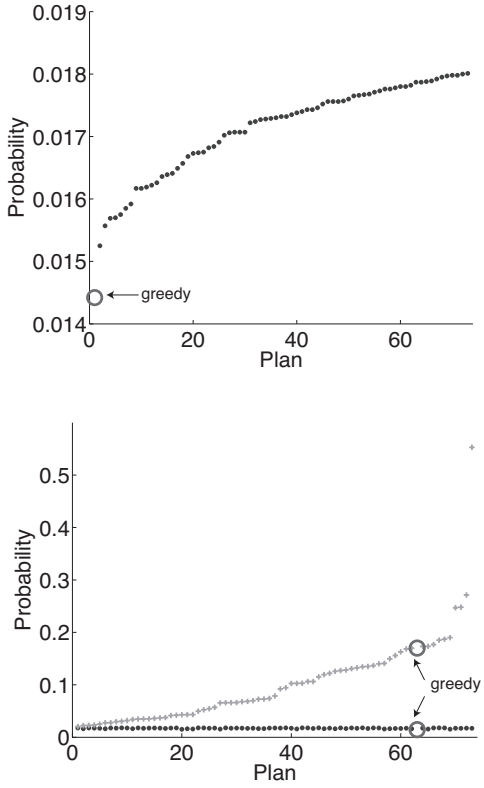

Fig. 6. Bayes error of the six-mutation plans selected by MutPlanBB for three pTfa models. (top) unbiased; (bottom) biased by −2 to 2 kcal/mol. The big circles indicate the Bayes error of the greedy plan in each case.

In order to be robust to a constant bias in how easily pTfa is unfolded, we use our branch-and-bound search to generate a number of good plans, and apply our robustness analysis. With a total of 192 candidate mutations at 77 positions, there are about 5.7×10^{10} possible combinations of six mutations. Our search was much more efficient, visiting a total of 15942 nodes in about 2 hours and identifying 73 good plans at $\lambda = 1$ (the value that ensures finding the optimal plan). Fig. 6 summarizes the Bayes errors for the identified plans, assuming either no bias or bias between −2 and 2 kcal/mol. While the greedy plan happens to be the best if there were no systematic experimental offset, it is much worse in the presence of such possible bias.

Table 1 details three particular plans: the selected plan, the initial greedy plan, and the worst among all plans identified by our branch-and-bound search ("worst-of-bb"). These three plans are comparable without bias, with Bayes errors of 0.018 (selected), 0.014 (greedy), and 0.017 (worst-of-bb). However, the selected plan stands out in the presence of bias, with a significantly smaller Bayes error of 0.020, compared to 0.170 for the greedy and 0.553 for the worst-of-bb. The difference in the presence of bias comes down to a certain "balance" among the selected mutations, in terms of how they discriminate among models. Both D83G (selected) and R10G (greedy) have quite different predictions for the first model and the third model, with a difference of 2.08 kcal/mol for D83G and −2.85 kcal/mol for R10G. Mutation Y77G, common to both plans, also differs for these two models, with a difference of −2.92 kcal/mol. Significantly, this difference has the same sign as that of R10G, but is opposite from

Table 1. Three six-mutation pTfa plans, with the predicted $\Delta\Delta G^\circ$ values for the three models

mut	$\Delta\Delta G^\circ_p$		
N22D	0.68	-3.26	0.02
Y77G	-3.26	0.23	-0.34
T75G	-2.98	-0.75	0.27
F3A	0.15	-0.09	-2.71
D83G	1.73	-1.29	-0.35
T11G	-0.56	0.19	-2.34

selected

mut	$\Delta\Delta G^\circ_p$		
N22D	0.68	-3.26	0.02
Y77G	-3.26	0.23	-0.34
T75G	-2.98	-0.75	0.27
R10G	-1.35	-0.84	1.50
N16D	-0.71	-2.77	-0.17

greedy

mut	$\Delta\Delta G^\circ_p$		
N22D	0.68	-3.26	0.02
Y77G	-3.26	0.23	-0.34
T75G	-2.98	-0.75	0.27
R10G	-1.35	-0.84	1.50
N16D	-0.71	-2.77	-0.17
V79S	-2.45	-2.93	-0.50

worst-of-bb

D83G. Thus a systematic bias would be balanced out in the selected plan but not in the greedy plan.

Two mutations differ between the greedy and the selected plans. In fact, the plans selected by the branch-and-bound search do tend to overlap, as shown in Fig. 7. There are a relatively small number of informative mutations, and the search eliminates the rest while identifying ways to combine the good ones so as to optimize the overall Bayes error.

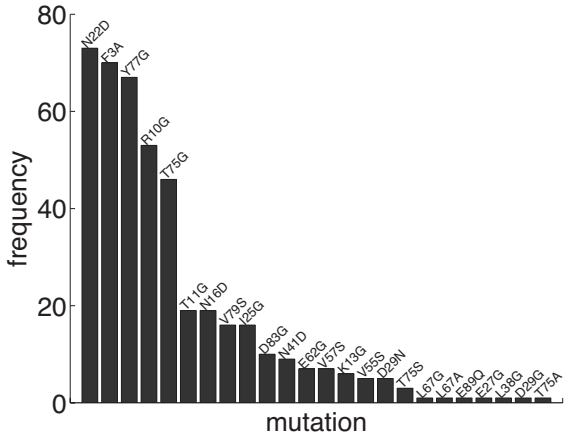

Fig. 7. Frequencies of the 24 unique mutations involved in the six-mutation plans identified by MutPlanBB for three pTfa models.

4. CONCLUSION

Bayes error provides a powerful criterion for evaluating the quality of an experiment plan, assessing how likely we are to make the wrong decision once we have collected the data. Since it is hard to compute Bayes error exactly, we develop here tight error bounds to estimate it for the case of selecting among predicted protein structure models by mutagenesis followed by stability evaluation. We use these error bounds in a branch-and-bound algorithm to optimize experiment plans for model selection. To allow for systematic bias in the experimental data (since proteins vary in how easy or hard they are to destabilize, overall), we consider the predicted performance of possible plans under a range of possible offsets in stability measurement. We demonstrated the tightness of our bounds on several test sets of models, and the effectiveness of our planning mechanism on a system of particular interest to us. Our experimental results for stability mutagenesis will be published separately[24], but we believe the present computational contribution stands on its own as a new solution to the important challenge of planning experiments optimizing Bayes error.

Our approach readily supports several extensions. (1) A mutation may have reliable $\Delta\Delta G^\circ$ predictions in some models but not in all of them. In the calculation of error bounds, what matters is the differences between predictions in different models; thus we set to zero the differences involving missing values, so that they convey no information. (2) In selecting plans, the constraint check can incorporate additional criteria such as the dispersion of selected mutations in the sequence or 3D structure. (3) In a sequential experiment plan, we can seek in each round of experiments to select a "top group" of models rather than a single best; then a subsequent round can focus on selecting among the top models. We can modify our error bounds (Eq. 9 and Eq. 11) so that the correct model will be included in the top group with high probability. To choose a top group of size t, we should ignore the closest $t-1$ neighbors in calculating the error bounds; we will then bound the probability that more than $t-1$ models beat the correct one.

Acknowledgments

This work was supported in part by a grant from NSF SEIII (IIS-0502801) to CBK, AMF, and Bruce Craig.

References

1. Natl. Inst. Gen. Med. Sci. The Protein Structure Initiative. `http://www.structuralgenomics.org`.
2. S. Govindarajan, R. Recabarren, and R. A. Goldstein. Estimating the total number of protein folds. *Proteins*, 35:408–414, 1999.
3. Y. Zhang and J. Skolnick. The protein structure prediction problem could be solved using the current PDB library. *PNAS*, 102:1029–1034, 2005.
4. X. Ye, P. K. O'Neil, A. N. Foster, M. J. Gajda, J. Kosinski, M. A. Kurowski, J. M. Bujnicki, A. M. Friedman, and C. Bailey-Kellogg. Probabilistic cross-link analysis and experiment planning for high-throughput elucidation of protein structure. *Protein Sci.*, 13:3298–3313, 2004.
5. K. T. Simons, C. Kooperberg, E. Huang, and D. Baker. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. *J. Mol. Biol.*, 268:209–225, 1997.
6. D. Kihara, H. Lu, A. Kolinski, and J. Skolnick. TOUCHSTONE: an *ab initio* protein structure prediction method that uses threading-based tertiary restraints. *PNAS*, 98:10125–10130, 2001.
7. A. Godzik. Fold recognition methods. *Methods Biochem. Anal.*, 44:525–546, 2003.
8. M. A. Kurowski and J. M. Bujnicki. Genesilico protein structure prediction meta-server. *Nucleic Acids Res.*, 31(13):3305–3307, 2003. `http://genesilico.pl/meta`.
9. A. Kryshtafovych, C. Venclovas, K. Fidelis, and J. Moult. Progress over the first decade of CASP experiments. *Proteins*, 61:225–236, 2005.
10. C. M. Topham, N. Srinivasan, and T. L. Blundell. Prediction of the stability of protein mutants based on structural environment-dependent amino acid substitution and propensity tables. *Protein Eng.*, 10(1):7–21, 1997.
11. D. Gilis and M. Rooman. PoPMuSiC, an algorithm for predicting protein mutant stability changes. application to prion proteins. *Protein Eng.*, 12:849–856, 2000.
12. C. W. Carter Jr., B. C. LeFebvre, S. A. Cammer, A. Tropsha, and M. H. Edgell. Four-body potentials reveal protein-specific correlations to stability changes caused by hydrophobic core mutations. *J. Mol. Biol.*, 311:621–638, 2001.
13. R. Guerois, J. E. Nielsen, and L. Serrano. Predicting changes in the stability of proteins and protein complexes: a study of more than 1000 mutations. *J. Mol. Biol.*, 320:369–387, 2002.
14. V. Parthiban, M. M. Gromiha, and D. Schomburg. CUPSAT: prediction of protein stability upon point mutations. *Nucleic Acids Res.*, 34:W239–W242, 2006.
15. H. Kamisetty, E.P. Xing, and C.J. Langmead. Free energy estimates of all-atom protein structures using generalized belief propagation. In *Proc. RECOMB*, pages 366–380, 2007.
16. D. G. Lainiotis. A class of upper bounds on the probability of error for multihypothesis pattern recognition. *IEEE Trans. Info. Theory*, 15:730–731, 1969.
17. G. T. Toussaint. Bibliograph on estimation of misclassification. *IEEE Trans. Info. Theory*, 20:472–479, 1974.
18. K. Fukunaga and T. E. Flick. Classification error for a very large number of classes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-6:779–788, 1984.
19. F. D. Garber and A. Djouadi. Bounds on the bayes classification error based on pairwise risk functions. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 10:281–288, 1988.
20. L. Comtet. *Advanced Combinatorics: The Art of Finite and Infinite Expansions*. Springer, 2001.
21. K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Morgan Kaufmann, second edition, 1990.
22. G. Vriend. WHAT IF: a molecular modeling and drug design program. *J. Mol. Graph.*, 8:52–56, 1990.
23. P. O'Neil et al., in preparation.
24. A.N. Foster et al., in preparation.