Supporting Information

Allen et al. 10.1073/pnas.1012985107

SI Text

Combinatorial Library Design. Structure-based computational protein design (CPD) methods can be harnessed to expedite the engineering of proteins by directed evolution. Several methods have been developed to allow the design of combinatorial mutation libraries to be informed by the results of CPD calculations. These approaches allow many specific variants chosen by CPD to be tested experimentally and can facilitate assessment and improvement of the design procedure. Hayes et al. described a method in which a list of low-energy sequences found by CPD is used to generate a table of frequencies for each amino acid type at each position, and then a frequency cutoff is applied to limit the library to only those amino acids found more frequently than the cutoff value at each position (1). Mena and Daugherty developed a similar procedure that produces libraries that include as many of the sequences in the CPD list as possible, while using only those sets of amino acids that can be encoded using degenerate codons (2). This feature helps to ensure that the resulting combinatorial gene libraries can be synthesized quickly and inexpensively. Treynor et al. developed a computational library design method analogous to CPD in which interactions between sets of amino acids at various positions are scored, and this system of interactions is sampled using standard CPD optimization algorithms to find the most favorable degenerate codon sequence (3).

In our view, a procedure that couples CPD to the design of combinatorial protein libraries should provide at least the following:

1. *Explicit consideration of CPD energies.* Methods that ignore CPD energies lead to a weaker correspondence between the final libraries and the original design calculations, limiting the predictive capability of the library design procedure and making improvement of CPD through library screening and analysis more difficult.
2. *Direct specification of the range of library sizes that should be produced.* In general, the desired library size will be a direct function of experimental screening capacity. A method that does not allow the user to specify the library size will either require repeated manual rerunning in an attempt to generate the desired library size, or will waste potentially prohibitive amounts of compute time analyzing libraries with irrelevant sizes.
3. *Control over which sets of amino acids are allowed*. Users with limited resources will usually prefer sets of amino acids that can be encoded using degenerate codons, because the resulting gene libraries can be synthesized in a single reaction with a relatively small number of inexpensive oligonucleotides. Those who can afford larger numbers of oligonucleotides and liquid-handling robots will be able to test libraries made with arbitrary sets of amino acids, which in general should more accurately reflect the sequence preferences of CPD calculations. A robust library design method must therefore handle whatever sets of amino acids the user deems appropriate.
4. *Consideration of all user-allowed sets of amino acids at each position.* Some design methods use heuristics to remove from consideration particular sets of amino acids at each position. Although this process can reduce the computational cost of the library design procedure, it can also result in the elimination of desirable libraries.

Because no previously reported algorithm that we know of satisfies all these criteria, we developed one that does. The new algorithm takes several inputs: (*I*) a list of scored sequences; (*ii*) a list of allowed sets of amino acids (e.g., those that can be encoded using degenerate codons); (*iii*) a range of preferred library sizes; (*iv*) a simulation temperature that controls the degree of preference for sequences with better scores; and, optionally, (*v*) sets of amino acids that are to be required or prohibited at particular positions. Based on these inputs, the algorithm produces a list of combinatorial libraries that are ranked according to the degree to which they satisfy the input list of scored sequences.

The process used by the algorithm to produce a list of combinatorial libraries from a list of scored sequences can be conceptually separated into three steps (Fig. S5).

Step **A**. Scan through the input list of scored sequences and generate a "total diversity" library that includes, at each position, every amino acid seen in the list at that position. This library represents the list optimally but ignores the user's preferred library size and allowed sets of amino acids. If later steps indicate that the size of the problem with this total diversity is insurmountably large, the user can request that the total diversity library be constructed from a subset of the input sequence list. For example, given a list of length 10,000, the user might decide to consider only the best 1,000 sequences in the list during this step.

Step **B**. Enumerate all possible amino acid size configurations that lead to combinatorial libraries within the range of sizes specified by the user. A size configuration is simply a specific number of amino acids at each position in the protein (e.g., 3 amino acids at position 1, 4 amino acids at position 2, etc.). An amino acid set size need not be considered at a particular position if it is larger than the smallest set that includes all amino acids found at that position in the total diversity library. This greatly reduces the total number of size configurations that need to be generated in this step and scored in the next step.

Step **C**. For each size configuration, determine the best set of amino acids of the required size at each position. This is done for each position independently by computing a partition function for each amino acid set with the given size. Amino acid sets that lack user-required amino acids or contain user-prohibited amino acids can be skipped here. Given a position and an allowed set of amino acids, iterate through the list of scored sequences, and for each sequence add to a cumulative partition function the Boltzmann-weight, $\exp(-E/kT)$, where E is the score of the sequence, k is the Boltzmann constant, and T is the simulation temperature. If the amino acid at that position in the current sequence is not found in the amino acid set of interest, nothing is added to the partition function. If the simulation temperature is low, the best-scored sequences will contribute most strongly to the partition function; if the temperature is high, all sequences in the list will contribute similarly. At each position, the set of amino acids with the most favorable partition function (position library score) is chosen. This procedure produces an optimal combinatorial library for each size configuration. The optimal libraries of each possible size configuration can then be ranked based on the sums of their position library scores across all positions.

Computational Performance of Library Design. The CLEARSS library design procedure is most readily applicable to designs with a limited number of variable positions, like the ones we described in this report. For such designs, libraries of virtually any target size can be produced. Table S3 shows that, for a smaller design problem of 10 buried positions, libraries with sizes up to 16 million can be generated without difficulty based on the top 1,000 amino acid sequences of the design. Larger libraries are precluded by the level of diversity found in the original design calculation, rather than for reasons of computational tractability. Generally, very small and very large sized libraries take less processor time than do intermediate sizes, because there are many more size configurations that lead to libraries of intermediate size.

As the number of designed positions and the diversity allowed at each position increase, the combinatorial explosion of size configurations that must be tested may become unwieldy. However, CLEARSS can be applied to larger design problems and target library sizes if the total diversity is limited by using fewer input sequences. Table S3 shows the results of applying CLEARSS to a design problem with 20 variable positions, in a variety of exposed and buried environments. To make these library designs tractable across the entire range of target sizes, the diversity was limited to that found in the top 50 sequences from the original design calculation. These library designs took significantly more time than those in the smaller design problem, especially those targeting intermediate targets sizes (those with millions of members). In general, CLEARSS calculations become intractable as the number of variable positions grows beyond 20, especially if intermediate library sizes (millions of members) are required. We expect that the development of additional performance improvements and heuristics will be able to further expand the set of design problems to which CLEARSS is applicable.

Microtiter Plate-Based Stability Assay Controls. The fluorescence profiles of the GdmCl gradient and the elution buffer show no effect on the shape of the unfolding transition of wild-type Gβ1 (Fig. S6*A*). Sample signal below the elution buffer was interpreted as expression failure; any sample whose data could not be fit yet whose signal was above the elution buffer was deemed expressed but unstable, unfolded, or misfolded. In order to test the accuracy of the microtiter plate-based denaturation assay, Gβ1 unfolding was monitored by circular dichroism (Aviv Biomedical) and tryptophan fluorescence in a fluorimeter (Photon Technology International). The denaturation profiles from these low-throughput experiments were compared to results from the fluorescence plate reader (Fig. S6*B*). The overlapping data points support the use of a two-state unfolding fit during our stability calculations and verify the accuracy of the assay. Next, the unfolding curves from several protein preparations from different concentrations confirmed the assay's precision (Fig. S6*C*). These results support some assumptions that the stability determination method described here makes in order to maintain a high level of throughput. First, we never assay for protein concentration before setting up the GdmCl gradient, relying on the fraction-unfolded plot to remove any concentration bias/effects. Second, the high concentration (250 mM) of imidazole in elution buffer is never dialyzed out of the eluted protein solution. Fig. S6 *B* and *C* show that these discrepancies in protein preparation have no significant effect on fraction-unfolded plots for the wild-type protein.

1. Hayes RJ et al. (2002) Combining computational and experimental screening for rapid optimization of protein properties. *Proc Natl Acad Sci USA* 99(25):15926–15931.
2. Mena MA, Daugherty PS (2005) Automated design of degenerate codon libraries. *Protein Eng Des Sel* 18(12):559–561.
3. Treynor TP, Vizcarra CL, Nedelcu D, Mayo SL (2007) Computationally designed libraries of fluorescent proteins evaluated by preservation and diversity of function. *Proc Natl Acad Sci USA* 104(1):48–53.

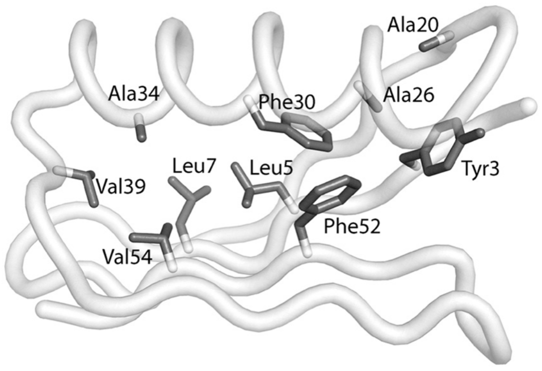

Fig. S1. The core residues of Gβ1 designed in this study. Each of these positions was allowed to assume various rotamers of the hydrophobic amino acids Ala, Val, Ile, Leu, Phe, Tyr, and Trp. Position Trp43 (not shown) was additionally allowed to change rotamer but not amino acid type. All other side chains and the main chain were fixed in the input conformation for the state being modeled in each case.

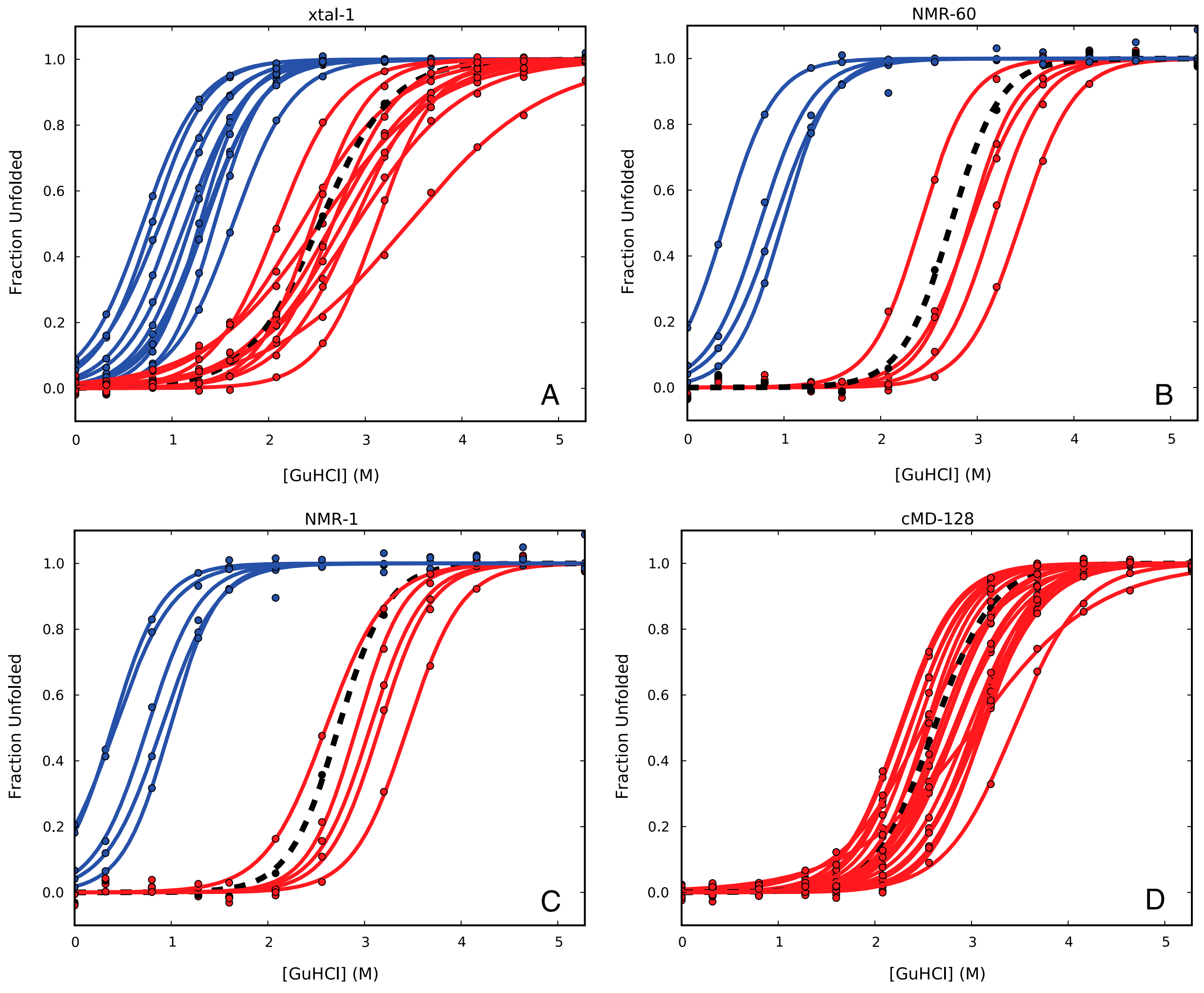

Fig. S2. Fraction-unfolded curves derived from the stability determination of experimental libraries. The dashed black curve denotes variant Y3F, which is the closest library member to the wild type in terms of sequence, and which is known to have a stability very similar to the wild type. The blue curves denote variants with $C_m < 2.0$ M ("destabilized") and the red curves denote variants with $C_m > 2.0$ M ("stabilized"). (*A*) xtal-1 library: Destabilized variants feature Leu at position 5 while stabilized variants feature Ile at position 5. Not pictured: variant Y3F + L5I + L7I, which did not give a signal that could be fit to a two-state unfolding model. (*B*) NMR-60 library: Stabilized variants feature Phe at position 52 while destabilized variants lack Phe52 but have Val at position 39. Not pictured: 14 variants that lack Phe at position 52 and which did not give a signal that could be fit to a two-state unfolding model. (*C*) NMR-1 library: Stabilized variants feature Phe at position 52 while destabilized variants lack Phe52 but have Val at position 39. Not pictured: 13 variants that lack Phe at position 52 and which did not give a signal that could be fit to a two-state unfolding model. (*D*) cMD-128 library: Only stabilized variants are present in this library.

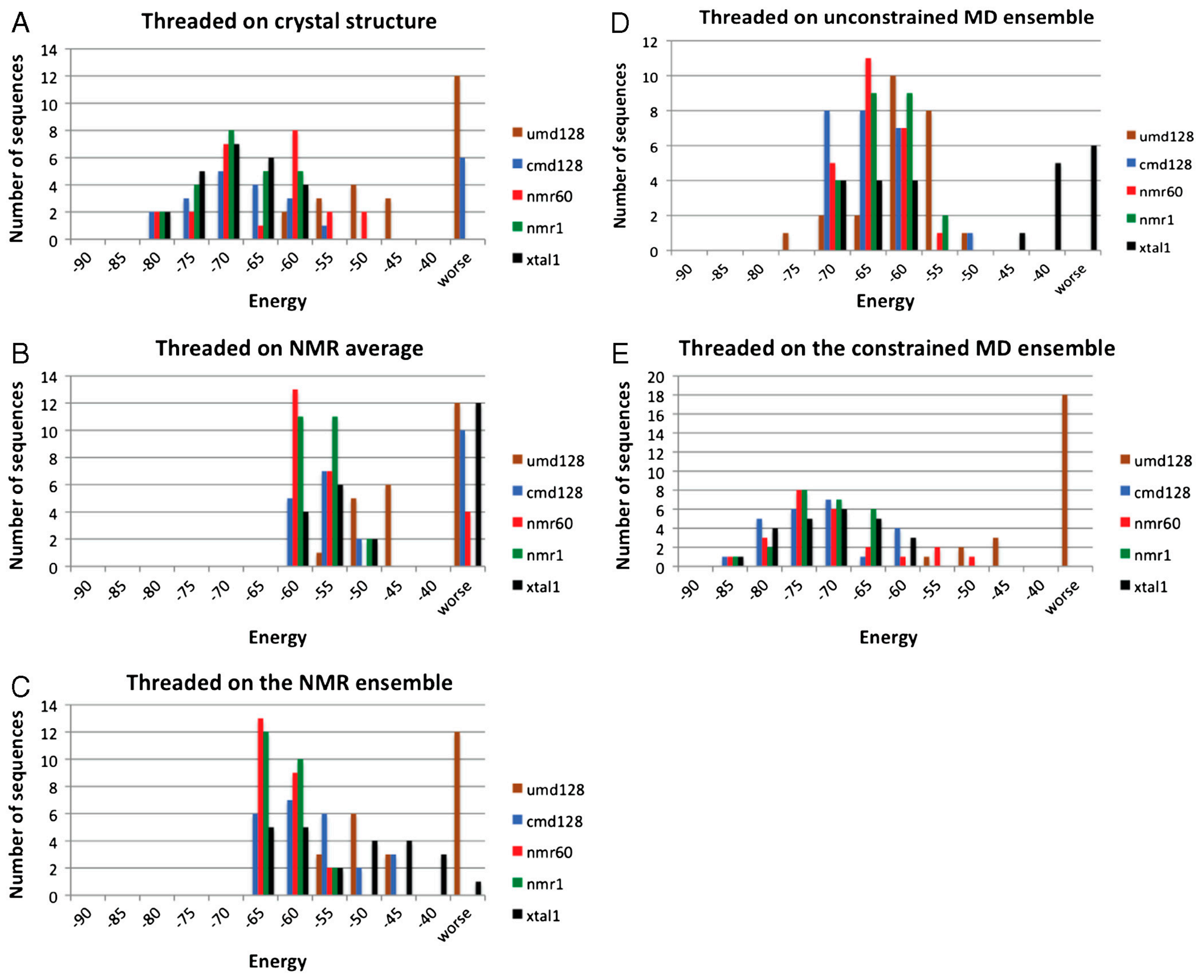

Fig. S3. Energies of the members of each library when threaded on the structural basis for (*A*) the xtal-1 library, (*B*) the NMR-1 library, (*C*) the NMR-60 library, (*D*) the cMD-128 library, and (*E*) the uMD-128 library.

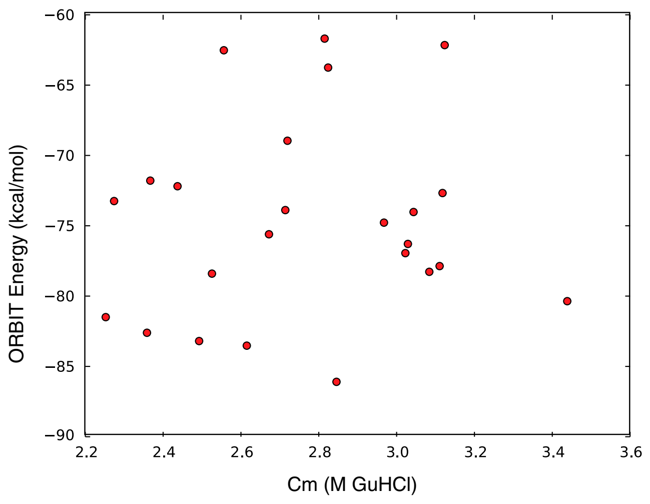

Fig. S4. Correlation between simulation energy and experimental stability for the cMD-128 library. No correlation was observed between the experimentally measured fitness of the sequences and simulation energies that were used to select them for experimental screening.

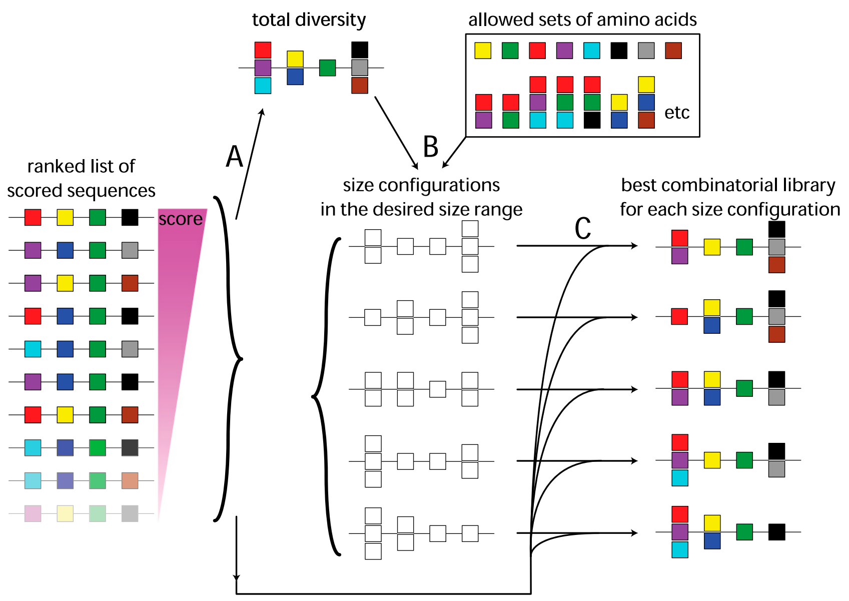
total diversity
allowed sets of amino acids
etc
A
B
ranked list of scored sequences
score
size configurations in the desired size range
C
best combinatorial library for each size configuration

Fig. S5. Detail of the library design method. (*A*) The list of scored sequences defines an initial "total diversity" library that is typically much larger (10^3–10^{15}, or even more) than the desired library size (10^2–10^6). (*B*) This total diversity library and the allowed sets of amino acids are used to construct a set of size configurations that lead to libraries in the desired range of sizes. The boxes in the list of size configurations are unfilled, indicating that the particular amino acids at each position have not yet been determined at this step. (*C*) For each size configuration generated in the previous step, the original list of scored sequences is used to find the optimal set of amino acids of the required size at each position.

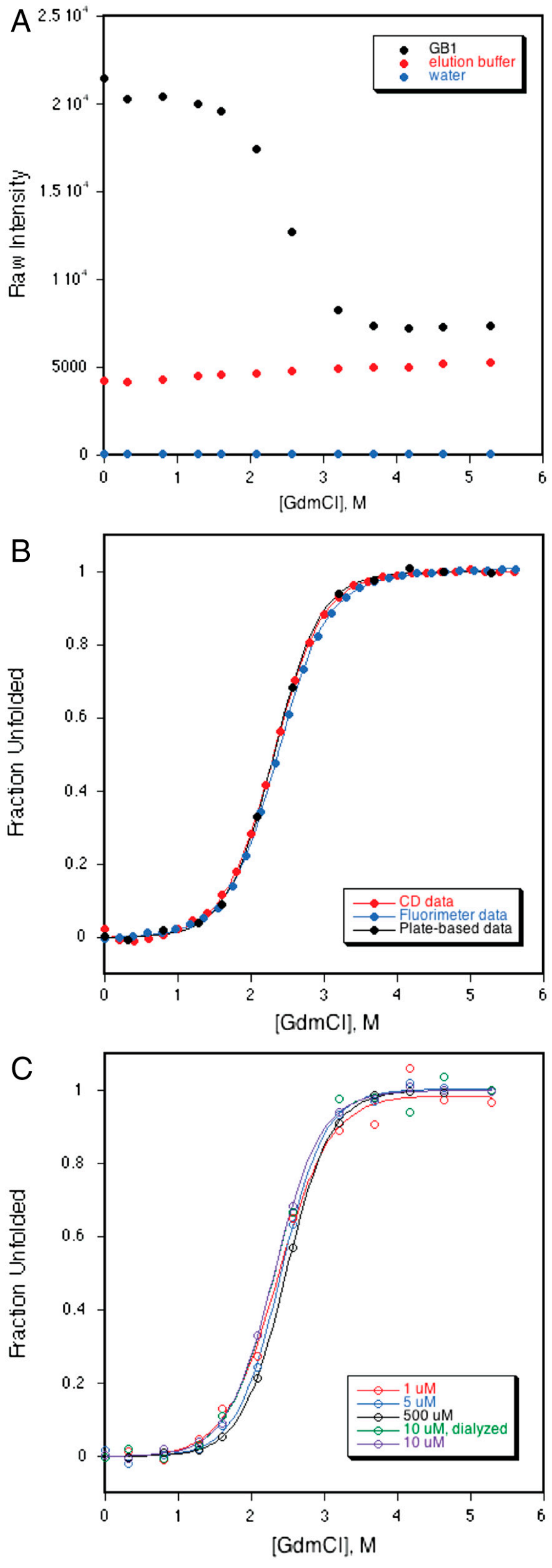

Fig. S6. Microtiter plate-based stability assay controls. (*A*) Denaturation gradient and elution buffer fluorescence profiles. Gβ1 (black) was expressed in a 5 mL culture, purified, and eluted with 500 μL of elution buffer (50 μM $NaPO_4$, 300 mM NaCl, 250 mM imidazole, pH 8). Because each point of the Gβ1 denaturation profile contains 35 μL of eluted protein, the elution buffer profile (red) substitutes protein with 35 μL of elution buffer. Similarly, the water profile (blue) adds 35 μL of water to make up the final volume. Each denaturation profile contains an increasing gradient of GdmCl, 50 μM $NaPO_4$ buffer at pH 6.5, and water. (*B*) Fraction-unfolded profiles between different modes of detection. CD data (red) measured 5 μM Gβ1 titrated with a 5 μM Gβ1/8 M GdmCl solution in 0.2 M steps at 218 nm. Fluorimeter data (blue) measured 5 μM Gβ1 titrated as in the CD experiment with excitation performed at 295 nm and emission recorded at 341 nm with 4 nm bandwidths. Plate-based data (black) measured 12 separate solutions of 10 μM Gβ1 in response to increasing amounts of 8 M GdmCl with fluorescence parameters identical to the fluorimeter data except for 10 nm bandwidths. All samples were measured at 25 °C in 50 μM $NaPO_4$ buffer at pH 6.5. (*C*)

Fraction-unfolded profiles between different protein preparations. Gβ1 was expressed in 100 mL cultures, purified and diluted to 1, 5, 10, and 500 μM in 50 μM $NaPO_4$ buffer at pH 6.5. Another expression culture was dialyzed overnight (Pierce Biotechnology) after purification and diluted to 10 μM in the same buffer. All measurements were taken on a fluorescence plate reader as described in the text.

Table S1. Library coverage

	xtal-1	NMR-1	NMR-60	cMD-128	uMD-128
No. of top-20 list sequences found in library	8	12	10	8	1
No. of top-100 list sequences found in library	15	20	16	16	3

For each design problem, we report the number of top 20 and top 100 designed sequences from each original list that were represented in each corresponding combinatorial library. The maximum possible number of top-20 sequences that could be represented is 20, whereas the maximum number of top-100 sequences is 24 because each library contains only 24 members.

Table S2. Combinatorial library design

Residue	WT	NMR-16D	NMR-16R	cMD-16D	cMD-16R	uMD-16D	uMD-16R
3	Y	F	F	FY	F	W	F
5	L	L	L	L	L	FLV	A
7	L	ILV	IL	IL	IL	I	FL
20	A	A	A	A	A	F	A
26	A	A	A	A	A	A	A
30	F	FILV	F	F	F	FILV	FIL
34	A	A	A	A	A	A	F
39	V	I	IL	ILV	ILV	IV	IL
52	F	FL	FL	F	FL	F	F
54	V	V	ILV	IV	IV	V	AV

Combinatorial libraries designed from the top-16 energy-ranked structures based on two different energy functions. NMR-16D: library based on the top 16 NMR structures ranked by DREIDING energy. NMR-16R: library based on the top 16 NMR structures ranked by Rosetta energy. cMD-16D: library based on the top 16 constrained MD ensemble structures ranked by DREIDING energy. cMD-16R: library based on the top 16 constrained MD ensemble structures ranked by Rosetta energy. uMD-16D: library based on the top 16 unconstrained MD ensemble structures ranked by DREIDING energy. uMD-16R: library based on the top 16 unconstrained MD ensemble structures ranked by Rosetta energy.

Table S3. Performance of the library design procedure when applied to make libraries of various sizes

Library Size	# Size Configurations	Time (s)
Smaller design problem		
$2^4 = 16$	310	1.3
$2^8 = 256$	3, 401	12.8
$2^{12} = 4{,}096$	6, 976	30.0
$2^{16} = 65{,}536$	3, 401	14.7
$2^{20} = 1{,}048{,}576$	310	1.6
$2^{24} = 16{,}777{,}216$	1	0.3
Larger design problem		
$2^{10} = 1024$	836, 158	149.3
$2^{20} = 1{,}048{,}576$	5, 522, 361	1326.9
$2^{30} = 1{,}073{,}741{,}824$	11, 564	60.4

For each target library size, we report the total number of size configuration that had to be scored, and the total time necessary for library design. Smaller design problem: The protein design problem described in the main text for xtal-1 was used; 10 buried positions designed in this problem. Larger design problem: A less constrained protein design problem was used; it had 20 total positions in a variety of surface-exposed and buried environments.