Supplementary Material

The following is optional **Supplementary Material (SM)** that provides additional information to substantiate the claims of the paper. **SM 1** provides details on the peptide-plane geometry assumptions and how they are encoded as equality constraints on the backbone atom coordinates. **SM 2** provides a summary of the CATS algorithm. **SM 3** provides a proof of Theorem 1. **SM 4** discusses the computational complexity of CATS. **SM 5** provides details of the Taylor series computation used in CATS. **SM 6** provides details of the protein design test cases described in Section 3.1. References cited below are given in the References section of the main text.

1 Details of peptide plane geometry assumptions

As mentioned in Section 2.2, we assume (1) that peptide planes are rigid bodies, and (2) that the N-C_α-C' bond angle in each residue is fixed. Crystal structures show that these assumptions are nearly correct (Lovell *et al.*, 2003). CATS could be extended to model slight perturbations to these assumptions. But we will not do so here, since doing so would produce only marginal changes in other atomic interactions. Moreover, the extent of these changes depends on the balance between two completely different types of energies—bond length and angle distortion penalties versus small changes in other interaction energies like van der Waals and electrostatic energies. It is not clear that current energy functions have sufficient quantitative accuracy to make estimation of this balance worthwhile. Our approach here is analogous to the usual approach taken in continuously flexible protein design for sidechains (Desmet *et al.*, 1992; Leach and Lemon, 1998; Desmet *et al.*, 2002; Georgiev *et al.*, 2008b; Gainza *et al.*, 2012), which moves only the sidechain dihedrals while fixing the bond lengths and angles.

Given that the peptide plane is rigid, the conformation of the entire plane is determined by the coordinates of any three of its atoms. Thus, if we determine the coordinates of the two alpha carbons and the nitrogen atom in a peptide plane, the coordinates of any other backbone atoms will be fixed in a basis defined by these three atoms. For each other backbone atom n (the carbonyl carbon, oxygen, or amide hydrogen), we calculate the coordinates of n in this basis from the starting structure, and then use those coordinates whenever we move the alpha carbons and nitrogens using CATS. We use coordinates from the starting structure instead of idealized coordinates because we wish to include the original backbone conformation in the CATS search space.

Even the alpha-carbon and nitrogen atoms cannot move entirely freely: they have only two rather than six degrees of freedom per residue. We thus add four additional constraints on atom-atom distances (Fig. S1). The first three constraints reflect the need for the peptide plane to move only as a rigid body. To enforce this, we fix the side lengths of the triangle formed by the nitrogen and two alpha carbons in each peptide plane. Because triangles whose side lengths are equal are congruent, the peptide plane is restricted to rigid-body motions if and only if these three distance constraints are enforced. Finally, to constrain the N-C_α-C' bond angle, we fix the dot product of the N-C_α and C_α-C' bond vectors. Given peptide plane geometry, the lengths of these two bonds are fixed, so our dot product constraint fixes the angle. The imposed values of these distances and dot product are taken from the original crystal structure. Again, this is done to ensure the original backbone conformation is part of the search space. Thus, these constraints, together with the linear formula for the other backbone coordinates, ensures the N-C_α-C' bond angle will not change relative to the starting structure, while the peptide planes will only move as rigid bodies relative to that structure.

There are four constrained quantities per residue: the squares of the side lengths of the triangle of atoms defining each peptide plane, and the dot product of the N-C_α and C_α-C' bond vectors (see Fig. S1). Thus $\mathbf{c}$ (see Eq. (3)) has four components per peptide plane, which are as follows:

$$||\mathbf{a}(\mathrm{C}_\alpha, 1) - \mathbf{a}(\mathrm{C}_\alpha, 2)||^2 \tag{5}$$

$$||\mathbf{a}(\mathrm{C}_\alpha, 1) - \mathbf{a}(\mathrm{N}, 2)||^2 \tag{6}$$

$$||\mathbf{a}(\mathrm{N}, 2) - \mathbf{a}(\mathrm{C}_\alpha, 2)||^2 \tag{7}$$

$$(\mathbf{a}(\mathrm{C}_\alpha, 1) - \mathbf{a}(\mathrm{N}, 1)) \cdot (\mathbf{a}(\mathrm{C}_\alpha, 1) - \mathbf{a}(\mathrm{C}', 1)) \tag{8}$$

where $\mathbf{a}(n, i)$ denotes the coordinates of atom n of residue i; without loss of generality we set $i = 1$ for the first residue of the peptide plane. For purposes of the last constraint (Eq. 8), the coordinates of the carbonyl carbon C' are expressed as a linear combination of the nitrogen and alpha-carbon coordinates. Thus in the case that the peptide plane is not exactly planar, we are actually constraining the projection of C' into the plane of the nitrogens and alpha carbons, which has essentially the same physical effect but is much simpler mathematically.

2 Summary of algorithm

As discussed in Section 2.1, protein design algorithms like iMinDEE (Gainza *et al.*, 2012) and EPIC (Hallen *et al.*, 2015) convert the continuous degrees of freedom over which they search to all-atom coordinates in order to evaluate energy as a function of the degrees of freedom, and thus design for energetically optimal sequences. We present here a summary of this conversion. The conversion function is called $\mathbf{a}(\mathbf{x})$ in the text and is described primarily in Section 2.3.

Before beginning design, we perform initialization calculations, which define the degrees of freedom. Let $\mathbf{a_{b0}}$ be the initial backbone coordinates of the molecule (e.g., from the crystal structure), and for a vector $\mathbf{z}$ of coordinates let $\mathbf{z}(p, n)$ denote the coordinates in $\mathbf{z}$ of the atom named n in peptide plane p (each peptide plane has two alpha carbons, which we call $C_{\alpha 1}$ and $C_{\alpha 2}$). We initialize as follows:

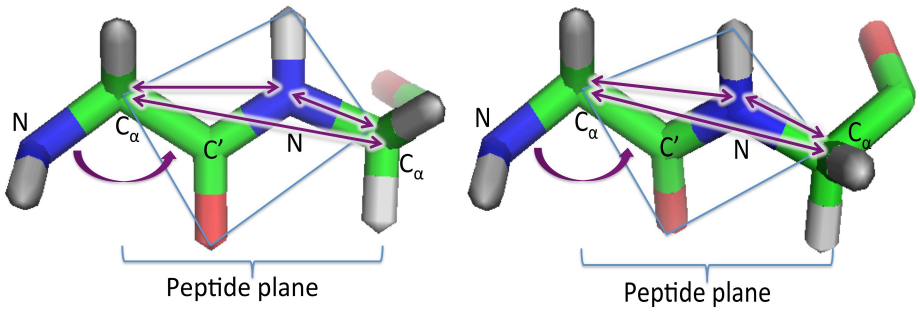

Fig. S1. Geometry assumptions in a section of protein backbone. Each peptide plane moves as a rigid body. This condition is enforced by holding three distances constant (straight purple arrows) and fixing the coordinates of C′ and backbone oxygens and hydrogens in the reference frame of the N and C_α coordinates. We also hold the N-C_α-C′ angle constant (curved arrow); both the distance and angle constraints are expressed as quadratic equalities in the N and C_α coordinates (see SM 1 for details). These constrained quantities are invariant when the backbone dihedrals are rotated (thus, as can be see in the left vs. right panels, they keep the same values). For example, the triangles formed by the straight purple arrows in the two conformations are congruent.

$\mathbf{y_0} \leftarrow$ N-and-C_α-Coordinates($\mathbf{a_{b0}}$)

// Now we express $\mathbf{a_{b0}}$ as a function of $\mathbf{y_0}$, represented as coefficients $\mathbf{v}$
// Loop over all peptide planes containing flexible backbone atoms
for all p in FlexiblePeptidePlanes **do**
  $\mathbf{z_1} \leftarrow \mathbf{a_{b0}}(p, \mathrm{N}) - \mathbf{a_{b0}}(p, \mathrm{C}_{\alpha 1})$
  $\mathbf{z_2} \leftarrow \mathbf{a_{b0}}(p, \mathrm{C}_{\alpha 2}) - \mathbf{a_{b0}}(p, \mathrm{C}_{\alpha 1})$
  $Q_p \leftarrow [\mathbf{z_1}, \mathbf{z_2}, \mathbf{z_1} \times \mathbf{z_2}]$ // make a basis for this peptide plane; $\times$ denotes cross product
  for all n in C′, O, H **do**
    $\mathbf{v}(p, n) \leftarrow Q_p^{-1}\mathbf{a_{b0}}(p, n)$ // These coefficients will stay constant
    // as N and C_α coords, and thus Q_p, change
  end for
end for

// Compute the linear coefficients to define our backbone degrees of freedom
$M_y \leftarrow$ nullspace($\nabla \mathbf{c}(\mathbf{y_0})$) //$M_y$ has orthonormal rows,
// which are orthogonal to the rows of $\nabla \mathbf{c}(\mathbf{y_0})$

// Compute the constraints
$\mathbf{c_0} \leftarrow \mathbf{c}(\mathbf{y_0})$ // see $\mathbf{c}$ definition in Eqs. (5)-(8)

// We now have everything we need to define $\mathbf{f}$
// we now compute derivatives to find the series expansion of $\mathbf{f^{-1}}$
$\mathbf{p} \leftarrow$ ComputeTaylorSeriesCoefficients($\mathbf{a_{b0}}, \mathbf{c_0}, \mathbf{v}, M_y$) // See Eq. (11)

Then $\mathbf{a}(\mathbf{x})$ is computed as follows, where $\mathbf{x}$ is the vector of values of the continuous conformational degrees of freedom. The function $\mathbf{a}(\mathbf{x})$ uses the coefficients $\mathbf{v}$ and $\mathbf{p}$ as parameters:

$(\mathbf{x_b}, \chi) \leftarrow \mathbf{x}$// unpack $\mathbf{x}$ into backbone degrees of freedom and sidechain dihedrals
$\mathbf{y} \leftarrow \mathbf{p}(\mathbf{x_b})$ // compute nitrogen and alpha-carbon coordinates
// Now fill in the rest of $\mathbf{y}$ (other atoms' coordinates)
for all p in FlexiblePeptidePlanes **do**
  // compute other backbone coordinates
  $\mathbf{z_1} \leftarrow \mathbf{y}(p, \mathrm{N}) - \mathbf{y}(p, \mathrm{C}_{\alpha 1})$ // $\mathbf{y}(p, n)$ denotes coordinates in y for atom n in p
  $\mathbf{z_2} \leftarrow \mathbf{y}(p, \mathrm{C}_{\alpha 2}) - \mathbf{y}(p, \mathrm{C}_{\alpha 1})$
  $Q_p \leftarrow [\mathbf{z_1}, \mathbf{z_2}, \mathbf{z_1} \times \mathbf{z_2}]$) // make a basis for this peptide plane
  for all n in C′, O, H **do**
    $\mathbf{y}(p, n) \leftarrow Q_p \mathbf{v}(p, n)$ // compute coordinates of atom n in plane p
  end for
end for

// Place idealized sidechains (see Hallen *et al.*, 2013; Lovell *et al.*, 2003) in coordinates $\mathbf{y}$
$\mathbf{y} \leftarrow$ PlaceIdealizedSidechains($\mathbf{y}$)
$\mathbf{y} \leftarrow$ RotateSidechains($\mathbf{y}, \chi$) // Apply the sidechain dihedral degrees of freedom
Return $\mathbf{y}$

3 Proof of Theorem 1

Theorem 1. *Let $D_\mathbf{b}$ denote the directional derivative in direction $\mathbf{b}$. If $\mathbf{z}(\mathbf{y}) = M_z\mathbf{y} + \mathbf{v_z}$ is an affine function and c satisfies $|\nabla(\mathbf{c}(\mathbf{y_0})^T)\ M_z^T| \neq 0$, then there exists an affine bijection between $Z = \{\mathbf{x_b} \in \mathbb{R}^{2k-6} \,|\, \mathbf{x_b} \neq 0\}$ and $B = \{\mathbf{b} \in \mathbb{R}^{6k} \,|\, \mathbf{b} \neq \mathbf{0},\ D_\mathbf{b}\mathbf{c}(\mathbf{y_0}) = \mathbf{0}\}$.*

Proof. Letting $\mathbf{f} = \{\mathbf{c}, \mathbf{x_b}\}$ and $J = \nabla\mathbf{f}(\mathbf{y_0})$, consider the mapping $\mathbf{m}$ where $\mathbf{m}(\mathbf{b})$ consists of the last $2k-6$ components of $J \cdot \mathbf{b}$. From the assumptions in the theorem, we have $|J| \neq 0$, i.e., the Jacobian of $\mathbf{f}(\mathbf{y_0})$ is nonsingular. Also J^{-1} is well defined, which allows us to construct the inverse mapping $\mathbf{m}^{-1}$: for any $\mathbf{x_b} \in Z$, $\mathbf{m}^{-1}(\mathbf{x_b}) = J^{-1} \cdot \{\mathbf{0}, \mathbf{x_b}\}$. Thus both $\mathbf{m}$ and its inverse are affine.

For any $\mathbf{x_b} \in Z$,

$$D_{\mathbf{m^{-1}}(\mathbf{x_b})}\mathbf{c}(\mathbf{y_0}) = \nabla\mathbf{c}(\mathbf{y_0}) \cdot \mathbf{m}^{-1}(\mathbf{x_b}), \tag{9}$$

which consists of the first $4k+6$ components of

$$J \cdot \mathbf{m}^{-1}(\mathbf{x_b}) = J \cdot J^{-1} \cdot \{\mathbf{0}, \mathbf{x_b}\} = \{\mathbf{0}, \mathbf{x_b}\}. \tag{10}$$

Thus the first $4k+6$ components of Eq. (10) are 0, making Eq. (9) evaluate to $\mathbf{0}$. Also, J^{-1} is nonsingular, so $\mathbf{m}^{-1}(\mathbf{x_b})$ cannot be $\mathbf{0}$. Thus for any $\mathbf{x_b} \in Z$, $\mathbf{m}^{-1}(\mathbf{x_b}) \in B$.

For any $\mathbf{b} \in B$, $\mathbf{m}(\mathbf{b}) = \mathbf{0}$ would imply $J \cdot \mathbf{b} = \mathbf{0}$, which contradicts J being nonsingular. Thus for any $\mathbf{b} \in B$, $\mathbf{m}(\mathbf{b}) \neq \mathbf{0}$ and so $\mathbf{m}(\mathbf{b}) \in Z$.

Therefore, $\mathbf{m}$ is an affine bijection between B and Z. □

4 Computational complexity of CATS

In the worst case, the asymptotic complexity of CATS is the same as for rigid-backbone design with the same sequence space. CATS and rigid-backbone design (as performed with iMinDEE (Gainza *et al.*, 2012), for example) create an RC for each sidechain rotamer; the only difference is that the CATS RC includes backbone degrees of freedom while the iMinDEE RC does not. Suppose there are n mutable positions and q rotamers per mutable position. The conformational space of the entire system then consists of q^n voxels, representing all possible combinations of RCs at different residues. In the worst case (Pierce and Winfree, 2002; Chazelle *et al.*, 2004), the minimized energy (Eq. 2) must be computed separately for each voxel, so both iMinDEE and CATS have worst-case complexity $\mathcal{O}(q^n)$.

In practice, pruning of RCs using dead-end elimination (Desmet *et al.*, 1992; Georgiev *et al.*, 2008b; Gainza *et al.*, 2012) and bounding of the minimum energies of portions of conformational space as part of A* search (Leach and Lemon, 1998; Georgiev *et al.*, 2008b; Gainza *et al.*, 2012; Hallen *et al.*, 2015) ensure that the vast majority of voxels will not require their own energy minimization—neither in CATS nor in iMinDEE. CATS has been implemented to use the EPIC algorithm as part of its conformational search, which tightens energy bounds compared to iMinDEE alone (Hallen *et al.*, 2015). Nevertheless, energy bounds in A*-based protein design (Leach and Lemon, 1998; Gainza *et al.*, 2012; Hallen *et al.*, 2015) tend to become looser as more flexibility is added, causing a greater number of energy minimizations to be performed. This fact, together with the fact that each minimization takes longer when there are more degrees of freedom, causes CATS designs to almost always take longer than comparable rigid-backbone designs.

5 Constructing and validating the Taylor series

For each voxel, we construct a Taylor series to evaluate the inverse mapping from Section 2.3, which we write as $\mathbf{f}^{-1}(\{\mathbf{c}, \mathbf{x_b}\})$ without loss of generality. We need only evaluate this mapping for $\mathbf{c} = \mathbf{c_0}$, that is, with the constrained quantities kept at the values measured using the starting structure (see Section 2.2). Hence, the Taylor series will be centered at $\mathbf{x_b}$=0 and $\mathbf{c}$ at $\mathbf{c_0}$, and all terms in the series that contain a component of $\mathbf{c}$ will be 0:

$$\mathbf{y} = \mathbf{f}^{-1}(\{\mathbf{c_0}, \mathbf{x_b}\}) = \mathbf{y_0} + \frac{\partial \mathbf{f}^{-1}}{\partial \mathbf{x_b}} \cdot \mathbf{x_b} + \frac{1}{2}\mathbf{x_b} \cdot \frac{\partial^2 \mathbf{f}^{-1}}{\partial {\mathbf{x_b}}^2} \cdot \mathbf{x_b} + \ldots \tag{11}$$

where derivatives are evaluated at the central conformation ($\mathbf{x_b} = 0$). To evaluate these derivatives, we observe that the derivatives of $\mathbf{f} = \{\mathbf{c}, \mathbf{z}\}$ with respect to the atomic coordinates $\mathbf{y}$ can be evaluated analytically (since $\mathbf{z}$ and $\mathbf{c}$ are linear and quadratic functions of $\mathbf{y}$ respectively), and the derivatives for the inverse mapping can be obtained from these. For example, the derivatives $\frac{\partial \mathbf{f}^{-1}}{\partial \mathbf{x_b}}$ are components of $(\nabla \mathbf{f})^{-1}$.

The error in the truncated Taylor series converges to 0 as we approach the central conformation, as is always the case for Taylor series. But we must determine how large we can make the voxel without significant error. To do this, we start with a voxel whose range is [-1 Å,1 Å] for each degree of freedom, sample $\mathbf{x_b}$ randomly in the voxel (50 samples per voxel), and evaluate the average error in the constrained dot products and squared distances $\mathbf{c}(\mathbf{a_n}(\mathbf{x_b}))$ calculated at these samples using the truncated series (Eq. 11). If the error is too large, we reduce the allowed range by a factor of 1.3 and repeat validation. This sampling process is analogous to validation in machine learning, although the model we are validating is the analytically-calculated Taylor series rather than a model inferred from training data.

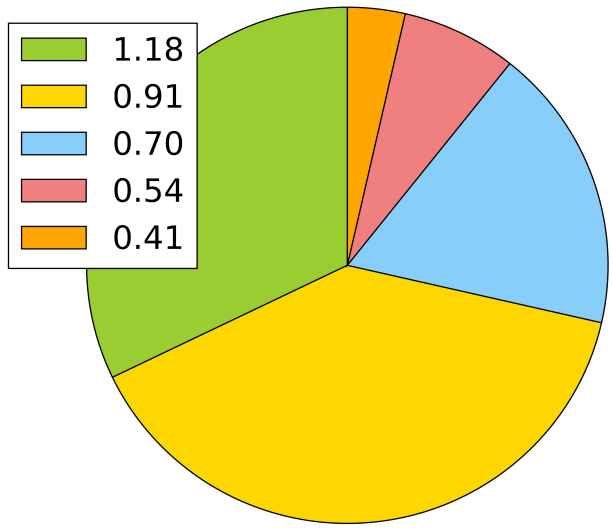

Fig. S2. Distribution of voxel size (range of allowed values for each degree of freedom, in Å), selected as described in SM 5.

We use a fairly stringent criterion for the constraints (RMS error $\leq$ 0.01 Å), but still typically obtain a voxel $\sim$1 Å across (see **SM 6** and Fig. S2). Since we wish to choose voxels small enough to admit local minimization, we believe that searches of larger portions of backbone conformational space should split that space into multiple backbone voxels (Roberts and Donald, 2015). Thus, in this study we consider it sufficient to truncate the Taylor series at second order, though our code supports up to a fourth-order series. Based on visualization of the voxels, we believe that local minimization is a relatively good approximation for global minimization for our voxels (see Fig. S3).

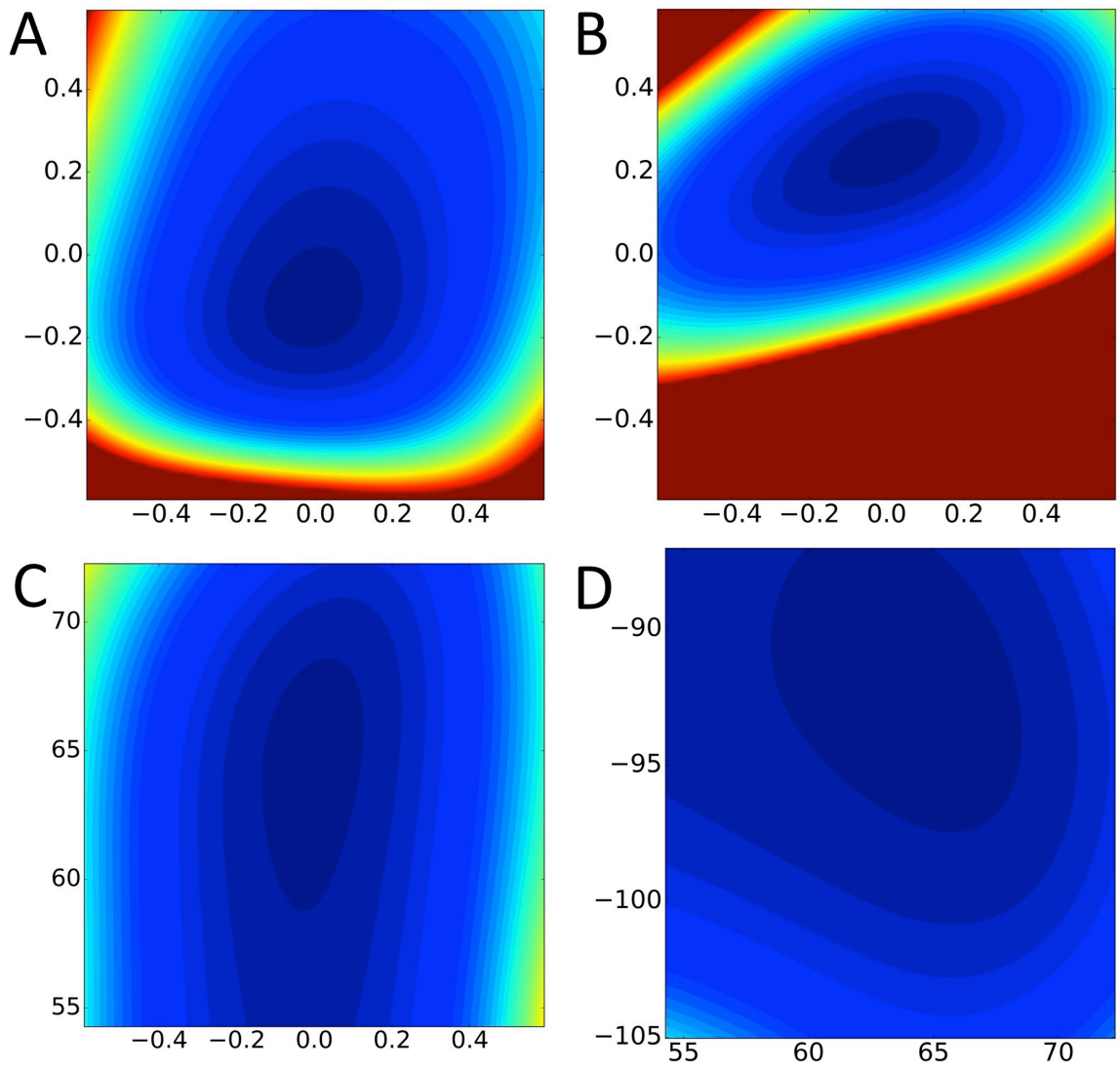

Fig. S3. Contour plots of energies with respect to continuous internal coordinates in the optimal voxel for the CATS design of VRC07 described in Section 3.2. Contours range from -220 kcal/mol (bluest) to -180 kcal/mol and greater (reddest); the deepest red regions thus denote steric clashes. Plots of energy with respect to pairs of degrees of freedom show that the voxel is dominated by a single pronounced minimum, which local minimization will easily find. Degrees of freedom: (A) First CATS backbone coordinate ($x_{b,1}$) versus second ($x_{b,2}$); (B) $x_{b,1}$ versus last CATS backbone coordinate ($x_{b,4}$); (C) $x_{b,1}$ versus χ_1 of Trp 427, chosen because it clashes with the mutated Trp 54 in the rigid-backbone model; (D) χ_1 versus χ_2 of Trp 427. CATS backbone coordinates are in Å; sidechain dihedrals are in degrees.

6 Details of test cases

Individual details of the test cases described in Section 3.1 are provided in Tables S1-S3 and Fig. S4.

Energy improvements were significantly greater for designs (*a*, see Section 3.1) than for minimizations (*c*): see Fig. 2A. This is expected, since the crystal structure backbone is expected to be favorable for the wild-type sequence and rotamers, but may need adjustment to accommodate mutations and sidechain rotations. Similarly, the backbone RMSD between the original and CATS-optimized backbones was greatest for designs (Fig. 2B). CATS could handle and predict larger backbone shifts than previous backbone-flexible protein design algorithms. RMSDs between the original and DEEPer-optimized backbones were smaller than for CATS in 94% of cases.

Additionally, the backbone motion identified by CATS was significantly greater in loops than in α-helices or β-sheets: for designs, 0.21±0.02 Å RMSD in loops compared to 0.14±0.02 for helix and sheet segments (and 0.19±0.03 for mixed segments). This is also expected, since loops are more flexible in most proteins.

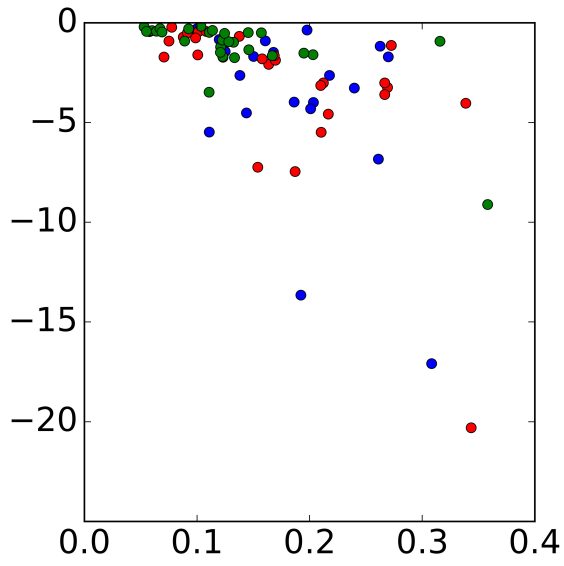

Fig. S4. Energy improvement for CATS compared to rigid-backbone design (kcal/mol), versus RMSD between CATS and original backbones (Å), in design (blue), sidechain placement (red), and single-voxel minimization (green) test cases. As in Section 3.1, design cases search a large sequence space, sidechain placements search the conformation space of the the wild-type sequence, and single-voxel minimizations search the voxel around the wild-type backbone and sidechain conformations.

Table S1. Design test cases. n denotes the number of residues in the flexible backbone segment, which can be in loop (Y), non-loop (N), or mixed (M) secondary structure. E_c and E_d denote the improvements in minimum energy for CATS and DEEPer runs respectively compared to rigid-backbone design. DNF denotes that a run did not finish in the allotted time (six weeks; we do not count these toward the 80 test cases), and F denotes failure of continuous minimization as described in Hallen et al., 2016. RMSD$_c$ and RMSD$_d$ denote the backbone RMSD (over the flexible backbone segment) between the optimal conformations calculated by CATS and DEEPer respectively and the starting backbone. w denotes the width of the voxel (the range over which each backbone degree of freedom was allowed to vary). SD denotes sequence differences between corresponding rigid-backbone (left) and CATS (right) designs.

Protein redesigned	n	PDB id	Loop?	E_c	E_d	RMSD$_c$	RMSD$_d$	w	SD
Atx1 metallochaperone	7	1CC8	Y	-1.71	-1.35	0.27	0.07	1.18	
PA-I lectin	6	1L7L	Y	-3.97	-2.93	0.19	0.07	0.91	A→T, E→D
alpha-D-glucuronidase	5	1L8N	Y	-4.52	-1.88	0.14	0.05	0.91	
Dachshund	8	1L8R	Y	-17.08	F	0.31	F	0.91	
Cytochrome c	8	1M1Q	Y	-1.47	-0.53	0.17	0.06	0.91	S→T
Histidine triad protein	8	2CS7	Y	-2.64	-1.57	0.22	0.05	1.18	
Ponsin	7	2O9S	Y	-3.99	-2.68	0.20	0.05	0.70	
Transcriptional regulator AhrC	6	2P5K	Y	-1.13	-0.45	0.27	0.04	0.70	
Scytovirin	5	2QSK	Y	-0.30	-0.32	0.10	0.06	0.54	
Hemolysin	9	2R2Z	Y	DNF	-1.70	DNF	0.05	0.91	
Bucandin	7	1F94	M	-1.17	-0.13	0.26	0.02	1.18	
Ferredoxin	9	1IQZ	M	F	-2.78	F	0.06	1.18	
gamma-glutamyl hydrolase	5	1L9X	M	-0.69	-0.59	0.09	0.05	0.70	
Sulfite oxidase	7	1MJ4	M	DNF	-5.22	DNF	0.06	0.54	
Dihydrofolate reductase	6	2RH2	M	-3.27	-2.48	0.24	0.04	0.91	V→L, Y→W
Putative monooxygenase	7	2RIL	M	-2.63	-1.17	0.14	0.06	0.41	
alpha-crystallin	6	2WJ5	M	-1.68	-0.84	0.15	0.04	0.91	
Cytochrome c555	7	2ZXY	M	-0.35	-0.34	0.20	0.03	0.91	D→E
High-potential iron-sulfur protein	6	3A38	M	-6.83	-5.66	0.26	0.07	0.91	M→W
Scorpion toxin	7	1AHO	N	-1.43	-0.45	0.12	0.04	1.18	
Cytochrome c553	6	1C75	N	-0.91	0.03	0.16	0.10	0.70	
Nonspecific lipid-transfer protein	6	1FK5	N	-4.31	-1.80	0.20	0.11	0.91	
Transcription factor IIF	7	1I27	N	DNF	-0.62	DNF	0.12	1.18	
Fructose-6-phosphate aldolase	6	1L6W	N	-0.84	-0.83	0.12	0.09	1.18	
Cephalosporin C deacetylase	8	1L7A	N	-5.47	-3.45	0.11	0.14	0.91	
Phosphoserine phosphatase	7	1L7M	N	DNF	-3.49	DNF	0.06	0.70	
Granulysin	7	1L9L	N	-13.65	-9.75	0.19	0.12	1.18	
Ferritin	5	1LB3	N	-0.44	-0.14	0.06	0.05	1.18	

Table S2. Conformational search test cases for wild-type sequences. Columns as in Table S1. The 28 backbone segments used in designs were also used for wild-type conformational search and minimization test cases.

Protein redesigned	n	PDB id	Loop?	E_c	E_d	RMSD$_c$	RMSD$_d$	w
Atx1 metallochaperone	7	1CC8	Y	-2.07	-1.51	0.16	0.06	1.18
PA-I lectin	6	1L7L	Y	-0.67	-0.45	0.14	0.06	0.91
alpha-D-glucuronidase	5	1L8N	Y	-1.71	-1.00	0.07	0.04	0.91
Dachshund	8	1L8R	Y	-20.30	F	0.34	F	0.91
Cytochrome c	8	1M1Q	Y	-1.72	0.43	0.12	0.05	0.91
Histidine triad protein	8	2CS7	Y	-3.01	-1.86	0.21	0.06	1.18
Ponsin	7	2O9S	Y	-4.02	-2.02	0.34	0.06	0.70
Transcriptional regulator AhrC	6	2P5K	Y	-1.13	-0.45	0.27	0.04	0.70
Scytovirin	5	2QSK	Y	-0.22	-0.27	0.08	0.07	0.54
Hemolysin	9	2R2Z	Y	-3.23	-1.73	0.27	0.05	0.91
Bucandin	7	1F94	M	-0.60	-0.22	0.10	0.04	1.18
Ferredoxin	9	1IQZ	M	-5.48	-3.23	0.21	0.06	1.18
gamma-glutamyl hydrolase	5	1L9X	M	-1.60	-0.53	0.10	0.05	0.70
Sulfite oxidase	7	1MJ4	M	-3.59	-3.20	0.27	0.06	0.54
Dihydrofolate reductase	6	2RH2	M	-0.72	-0.88	0.09	0.04	0.91
Putative monooxygenase	7	2RIL	M	-1.87	-0.71	0.17	0.05	0.41
alpha-crystallin	6	2WJ5	M	-1.64	-0.29	0.17	0.02	0.91
Cytochrome c555	7	2ZXY	M	-0.44	-0.63	0.10	0.03	0.91
High-potential iron-sulfur protein	6	3A38	M	-1.80	-0.85	0.16	0.06	0.91
Scorpion toxin	7	1AHO	N	-7.45	-3.10	0.19	0.04	1.18
Cytochrome c553	6	1C75	N	-0.42	0.05	0.11	0.02	0.70
Nonspecific lipid-transfer protein	6	1FK5	N	-4.57	-1.71	0.22	0.11	0.91
Transcription factor IIF	7	1I27	N	-3.14	-2.10	0.21	0.10	1.18
Fructose-6-phosphate aldolase	6	1L6W	N	-0.49	-0.60	0.09	0.09	1.18
Cephalosporin C deacetylase	8	1L7A	N	-7.23	-4.43	0.15	0.14	0.91
Phosphoserine phosphatase	7	1L7M	N	-3.01	-0.47	0.27	0.03	0.70
Granulysin	7	1L9L	N	-0.76	-0.32	0.10	0.07	1.18
Ferritin	5	1LB3	N	-0.91	-1.24	0.07	0.11	1.18

Table S3. Single-voxel minimization test cases. Columns as in Table S1.

Protein redesigned	n	PDB id	Loop?	E_c	E_d	RMSD$_c$	RMSD$_d$	w
Atx1 metallochaperone	7	1CC8	Y	-0.48	-0.15	0.11	0.03	1.18
PA-I lectin	6	1L7L	Y	-0.48	-0.12	0.15	0.03	0.91
alpha-D-glucuronidase	5	1L8N	Y	-0.20	-0.80	0.05	0.02	0.91
Dachshund	8	1L8R	Y	-9.10	-1.02	0.36	0.06	0.91
Cytochrome c	8	1M1Q	Y	-0.97	-0.70	0.13	0.04	0.91
Histidine triad protein	8	2CS7	Y	-1.60	-1.08	0.20	0.05	1.18
Ponsin	7	2O9S	Y	-1.64	-0.87	0.17	0.05	0.70
Transcriptional regulator AhrC	6	2P5K	Y	-0.92	-0.46	0.32	0.04	0.70
Scytovirin	5	2QSK	Y	-0.30	-0.34	0.09	0.06	0.54
Hemolysin	9	2R2Z	Y	-1.21	-0.77	0.12	0.05	0.91
Bucandin	7	1F94	M	-0.39	-0.15	0.06	0.02	1.18
Ferredoxin	9	1IQZ	M	-1.74	-0.40	0.13	0.05	1.18
gamma-glutamyl hydrolase	5	1L9X	M	-0.42	-0.92	0.06	0.05	0.70
Sulfite oxidase	7	1MJ4	M	-0.41	-0.40	0.06	0.05	0.54
Dihydrofolate reductase	6	2RH2	M	-0.49	0.32	0.16	0.04	0.91
Putative monooxygenase	7	2RIL	M	-1.69	-1.11	0.12	0.04	0.41
alpha-crystallin	6	2WJ5	M	-0.86	-0.16	0.12	0.03	0.91
Cytochrome c555	7	2ZXY	M	-0.38	-0.10	0.11	0.02	0.91
High-potential iron-sulfur protein	6	3A38	M	-0.94	-0.25	0.13	0.05	0.91
Scorpion toxin	7	1AHO	N	-1.48	0.32	0.12	0.04	1.18
Cytochrome c553	6	1C75	N	-0.17	0.00	0.10	0.01	0.70
Nonspecific lipid-transfer protein	6	1FK5	N	-0.53	-0.45	0.12	0.10	0.91
Transcription factor IIF	7	1I27	N	-1.52	-0.69	0.19	0.12	1.18
Fructose-6-phosphate aldolase	6	1L6W	N	-0.29	-0.41	0.07	0.07	1.18
Cephalosporin C deacetylase	8	1L7A	N	-3.48	-1.93	0.11	0.10	0.91
Phosphoserine phosphatase	7	1L7M	N	-1.34	-0.27	0.15	0.03	0.70
Granulysin	7	1L9L	N	-0.46	-0.38	0.07	0.08	1.18
Ferritin	5	1LB3	N	-0.91	-1.48	0.09	0.12	1.18