Supplementary Material: COMETS (Constrained Optimization of Multistate Energies by Tree Search): A provable and efficient protein design algorithm to optimize binding affinity and specificity with respect to sequence

Mark A. Hallen[1,3] and Bruce R. Donald[1,2,3,*]

Departments of [1] Computer Science [2] Chemistry, Duke University, Durham, NC 27708 [3] Department of Biochemistry, Duke University Medical Center, Durham, NC 27710 *Corresponding author, brd+jcb15@cs.duke.edu

The following is supplementary material for the following paper:

M.A. Hallen and B.R. Donald. COMETS (Constrained Optimization of Multistate Energies by Tree Search): A provable and efficient protein design algorithm to optimize binding affinity and specificity with respect to sequence.

Section A provides a highly simplified, "toy" example of a multistate protein design calculation to demonstrate the operation of COMETS. Section B provides details of the methods used to compute lower bounds for LMEs over sequence spaces. Section C provides details on the protein design runs described in Section 3 of the main text.

A Toy example

Let us introduce a highly simplified, "toy" example to explain the algorithm. Say we are designing a peptide inhibitor for a protein, "targetin," involved in some disease. We want our peptide to not bind a related protein, "offtargetin," because this binding would cause toxicity. However, we do have a peptide (say, a natural product) that binds both targetin and offtargetin, and we have crystal structures of this peptide with both targetin (structure T) and offtargetin (structure O). Thus, we set up a multistate design as follows. There will be four states: peptide bound to targetin (structure T), unbound peptide in targetin-binding conformation (peptide from structure T), peptide bound to offtargetin (structure O), and unbound peptide in offtargetin-binding conformation (peptide from structure O). These states will be called T-bound, T-unbound, O-bound, and O-unbound respectively. We will optimize the binding energy for the peptide-targetin interaction, which is the difference between the T-bound and T-unbound state energies. We will constrain binding energy for the peptide-offtargetin interaction to be worse than a threshold E_u: say, 10 kcal/mol worse than the wild-type binding energy. We will also constrain the unbound state to be stable, by demanding that the average of the T-unbound and O-unbound state energies be at most 10 kcal/mol worse than wild type. All of these constraints, and the objective function, are simple LMEs.

For simplicity, our toy peptide will only have one mutable residue position (residue 1), to which we may assign either the amino acid type alanine (1 rotamer) or valine (3 rotamers). We will also model only one residue as flexible in each of our proteins: Phe 75 of targetin (4 rotamers) and Leu 75 of offtargetin (5 rotamers). Fig. S1 illustrates the conformational search in COMETS on this problem subject to these modeling assumptions. We also present empirical results for much larger, realistic designs with similar objective function and constraint setups in Section 3 of the main text.

Types of nodes. In this toy example, we create only one node with a partially defined sequence. The sequence space for it is {A,V}, since these are the two sequences available to the peptide. Two nodes with fully defined sequence are created: one for each of the two sequences A and V. For the V node, the conformational tree for each state will have the same structure as a single-state A* tree[5,6,10] for a single-sequence conformational search in which residue 1 is a valine (Fig. S1).

Expansion of nodes. The expansion step for the starting node (sequence space {A,V}) splits the sequence space to create a node for each amino-acid type at residue 1: thus the two nodes will have sequences spaces {A} and {V}. For the node {V}, because the sequence space is fully defined, the expansion step expands nodes in the conformational trees, in the same way that nodes are expanded in single-state A* calculations[5,6,10] (Fig. S1).

B Computing lower bounds for LMEs

Previous A*-based protein design algorithms include methods to compute a lower bound on the energy of a single protein state over a sequence space.[5,6,8] These methods can be modified to provide a lower bound on an LME over a sequence space, with complexity as follows:

Theorem 1. *For any sequence space S defined by specifying the allowed set of amino acid types $S(i)$ at each mutable residue i, the lower bound on the LME Eq.* (1) *can be computed in time $O(n^2r^2s)$, where n is the number of flexible or mutable residues in the system, s is the number of states, and r is the maximum number of RCs available at a given residue.*

We use a different procedure to compute lower bounds for LMEs (linear multistate energies) depending on whether the sequence is fully defined. If it is, a simpler algorithm that often yields a tighter bound is used. If not, we have a generic algorithm, whose running time is bounded as described in Theorem 1 (proven in Section B.3 below).

B.1 Lower-bound algorithm for fully defined nodes

If the sequence is fully defined, then we can bound the LME

$$c_0 + \sum_{a \in A} c_a E_a(\mathbf{s}) \tag{1}$$

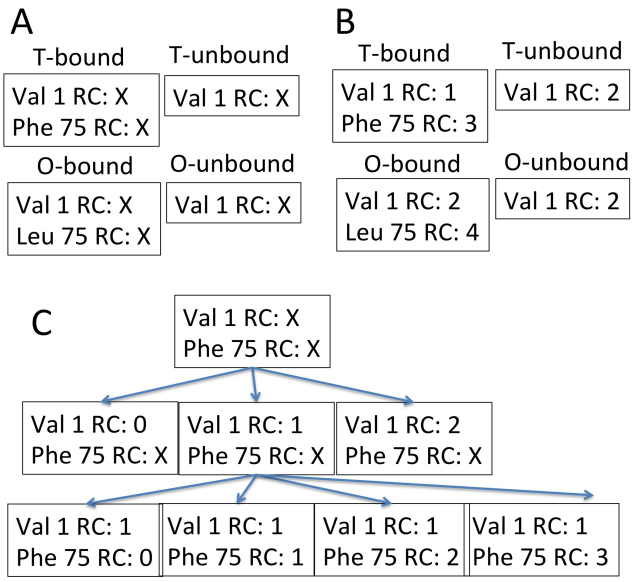

Fig. S1. Expansion of the state conformational trees for the node with sequence V in the toy example. (A) Each state conformational tree starts with a single node, in which all conformational degrees of freedom are undefined. (B) When the sequence tree node is fully processed, the lowest-scoring node in each conformational tree fully defines a conformation. (Only the lowest-scoring node in each tree is shown here: each tree will also have higher-scoring nodes, with varying numbers of unrestricted conformational degrees of freedom). (C) The expansion of the conformational tree for the T-bound state. As in single-state A*, nodes are chosen for expansion based on their scores, which are lower bounds on the energies of conformations in their conformational spaces.

using upper bounds $u_a(\mathbf{s})$ and lower bounds $l_a(\mathbf{s})$ on the optimal conformational energy for our sequence $\mathbf{s}$ in each state $a \in A$. Because we have $u_a(\mathbf{s}) \geq E_a(\mathbf{s})$ and $l_a(\mathbf{s}) \leq E_a(\mathbf{s})$, we have

$$c_0 + \sum_{a \in A,\ c_a < 0} c_a u_a(\mathbf{s}) + \sum_{a \in A,\ c_a > 0} c_a l_a(\mathbf{s}) \tag{2}$$

as a lower bound on Eq. (1). The single-state lower bounds $l_a(\mathbf{s})$ are computed as described in previous work on single-state design.[5,6,8] These methods yield a lower bound that converges to $E_a(\mathbf{s})$ as the conformation trees are fully expanded. Upper bounds $u_a(\mathbf{s})$ are straightforward because any conformation of a given state is an upper bound on the optimal conformational energy for that state. Each time we expand a node in the conformation tree for state a and sequence $\mathbf{s}$, we compute an upper bound u_n on the minimum energy of that node's conformation space by a quick heuristic procedure based on the FASTER algorithm.[2] If u_n is lower than our current estimate of $u_a(\mathbf{s})$, then we set $u_a(\mathbf{s}) = u_n$. As the conformation trees are fully expanded, we must eventually encounter a node in each state's tree whose conformational space is only the optimal conformation (or in the continuous case, only the RC assignment that yields the optimal conformation upon local minimization). At this point, the upper bound becomes tight: $u_a(\mathbf{s}) = E_a(\mathbf{s})$. So we have a method to provide lower bounds for LMEs that converges to the exact value of the LME when a node is fully processed. This is what we need both for constraint enforcement and for minimization of the objective function.

B.2 Lower-bound algorithm for other nodes

If the sequence is not fully defined, then we use an alternate algorithm, which is a direct adaptation of the lower-bounding method used in single-state A* calculations.[10] Nodes without the sequence fully defined cannot be fully processed, so we do not need this algorithm to converge to the exact value of the LME. However, we still prefer a reasonably tight bound in this case for efficiency purposes (to avoid having to generate too many nodes with fully defined sequences).

Let us first consider the rigid case. For the purposes of this algorithm, we will assume a *pairwise* energy function: an energy function that is a sum of terms dependent on at most two residues' conformations, and can thus be expressed in the form

$$\sum_i E(i_{\mathbf{r}}) + \sum_{j<i} E(i_{\mathbf{r}}, j_{\mathbf{r}}) \tag{3}$$

where $i_{\mathbf{r}}$ denotes the RC for our current conformation $\mathbf{r}$ at residue i, $E(\cdot,\cdot)$ is a pairwise interaction energy between RCs, and the one-body terms $E(\cdot)$ measure the internal energy of each RC plus its interactions with non-flexible parts of the system. Let us use E_a to denote a pairwise or 1-body energy for residues in state a. Now, to derive a lower bound, we start with the true minimum value of the LME over our sequence space S, which is (using Eq. 1)

$$c_0 + \min_{\mathbf{s} \in S} \sum_{a \in A} c_a E_a(s). \tag{4}$$

Letting $R_a(\mathbf{s})$ be the conformation space available to a sequence $\mathbf{s}$ in state a, and plugging in Eq. (3), Eq. (4) can be expanded to

$$c_0 + \min_{\mathbf{s}\in S} \sum_{a\in A} c_a \min_{\mathbf{r}\in R_a(\mathbf{s})} \left(\sum_i E_a(i_\mathbf{r}) + \sum_{j<i} E_a(i_\mathbf{r}, j_\mathbf{r}) \right). \tag{5}$$

To obtain a tractable bound, we relax Eq. (5) by moving the minima inside the sums. This operation can only decrease the value of the expression (this is a general property of interchanging summation and minimization), resulting in a valid lower bound. Let $S(i)$ be the set of amino acid types available at residue i, let $R_a(b,i)$ be the set of unpruned RCs available to amino acid type b at residue i in state a, and let us use the notation i_r for an RC at residue i, following previous work.[1,4–6] The relaxation yields

$$c_0 + \sum_i \min_{b\in S(i)} \sum_{a\in A} \min_{i_r\in R_a(b,i)} \left(c_a E_a(i_r) + \sum_{j<i} \min_{b'\in S(j)} \min_{j_s\in R_a(b',j)} c_a E_a(i_r, j_s) \right). \tag{6}$$

So Eq. (6) is a lower bound on Eq. (4), i.e., for the LME we are bounding.

Each state may have flexible residues besides the mutable residues; for the purposes of Eq. (6), these are all considered separately. Hence, the sum over i runs over all mutable residues, which are shared between all states, and also over other flexible residues, which may differ between states and are treated as different residues in the sum (with c_a being nonzero only for the state to which the residue belongs).

This bound, Eq. (6), can also be used with continuous flexibility, with only a change in definitions for the pairwise and 1-body energies E_a. Each of these energies is associated with an RC or a pair of RCs, so when we introduce continuous flexibility, E_a is no longer associated with a single conformation but with a set of conformations defined by bounds on the continuous degrees of freedom of the residues whose energy is being measured. Within these bounds there is a set of "ideal values" for all the degrees of freedom. So, for the terms $c_a E_a$ in Eq. (6), we let E_a denote the pairwise or 1-body energy at the ideal values of all continuous degrees of freedom if $c_a < 0$, but we let E_a denote the minimum value of the 1- or 2-body energy over the allowed ranges for all continuous degrees of freedom if $c_a > 0$. Using these definitions, Eq. (5) is a valid lower bound for the LME Eq. (4) over our node's sequence space, and so Eq. (6) is a valid lower bound, using the same reasoning as in the rigid case.

Regardless of which definition is used, the pairwise and 1-body energies E_a can be precomputed for an RC or a pair of RCs, before COMETS begins. This precomputation is described in previous work on single-state design.[4–6] A tighter bound can be achieved by using a larger rotamer library for the terms with negative coefficients.

B.3 Complexity analysis and proof of Theorem 1

Because the algorithm described in Section B.2 can be used for any sequence tree node, it provides a constructive proof of Theorem 1.

Proof. The bound can be computed using Eq. (6), as described in Section B.2. For each state a and each RC i_r at each residue i, the term $c_a E_a(i_r)$ can be computed in constant time, because $E_a(i_r)$ is simply looked up from the precomputed single-residue energies. Then for each residue j, the term $\min_{b' \in S(j)} \min_{j_s \in R_a(b',j)} c_a E_a(i_r, j_s)$ can be computed in $O(r)$ time, because the minimum is over $O(r)$ precomputed pairwise interaction energies that can be looked up and multiplied by c_a in constant time. So the $\sum_{j<i}$ sum has $O(n)$ terms that can each be computed in $O(r)$ time, meaning the sum can be computed in $O(nr)$ time. Thus, the quantity in large parentheses can be computed in $O(nr)$ time. For each residue i, this quantity needs to be computed $O(rs)$ times. So the cost for each term of the $\sum_i$ is $O(nr^2 s)$. Therefore, the cost to evaluate the entire $\sum_i$, and thus to compute our lower bound, is $O(n^2 r^2 s)$.

C Computational Experiments

Protein design calculations were performed in order to measure the efficiency of COMETS and its ability to design proteins with properties undesignable by single-state methods (Section 3 of the main text). In this section, we provide details of these test cases: the types of systems used (Section C.1) and the results of each test case in table format (Section C.2).

C.1 Types of systems

Systems of four types were used: designs for specificity on a protein that can form two or more different complexes; optimization of the binding energy for a single complex; stabilization of a single protein robust to choice of force field; and stabilization of the reduced form of angiotensinogen relative to the oxidized form or vice versa.

Systems of the first type involve a protein that can form two different complexes whose structures are known. This type can be viewed as a realistic version of the toy example above (Section A). Proteins were redesigned in four different ways: either to prefer one complex over the other or to stabilize both, and with either rigid or continuous flexibility modeling. For designs to favor one complex, the objective function was the binding energy for that complex (the energy of the bound state minus that of the unbound state), and for designs to favor both complexes, the objective function was the sum of the complexes' binding energies. In each case, constraints were placed on all binding energies and on unbound state energies. Desired binding energies were constrained to be better than the wild-type binding energy for the same complex, while undesired binding energies were constrained to be worse than wild-type. To ensure protein stability, the average of the unbound state energies was constrained to be no more than 10 kcal/mol worse than the wild-type average. The pairs of complex structures for these designs were obtained from random entries in the INstruct database.[11] The pairs' PDB ids were 2a40/2a41, 2a5y/3lqr (the angiotensinogen run in Fig. 1), 2gzd/2gzh, 2ngr/1grn, 3egd/3egx, 3efo/3eg9, 3k75/3lqc, 3ktp/3ktr, 3n1f/3n1q, and 3n1g/3n1m. These complexes were drawn from both protein-protein and protein-peptide interactions. In addition to these two-complex systems, a set of

larger designs was performed on a set of ten complexes of bovine trypsin crystallized with ten different variants of bovine pancreatic trypsin inhibitor (BPTI).[9] Designs were performed either to maintain binding by the wild-type BPTI while blocking binding to the nine mutant variants, or to enhance binding to all variants. Protein design to identify enzyme variants resistant to certain inhibitors has been used in the study of bacterial resistance to antibiotics, and has applications to the selection of inhibitors more robust to resistance.[3]

The second type of system was optimization of the binding energy for a single complex, with constraints handled as in the specificity calculations. Treating affinity calculation as a multistate design allows accounting for variations in the unbound state energy, both to ensure the stability of the unbound state and to compute its effect on binding affinity. Structures for these designs were also drawn from the INstruct database:[11] 1b6c, 1nez, 1stf, 1vyh, 2b4s, 2h0d, and 2nqa.

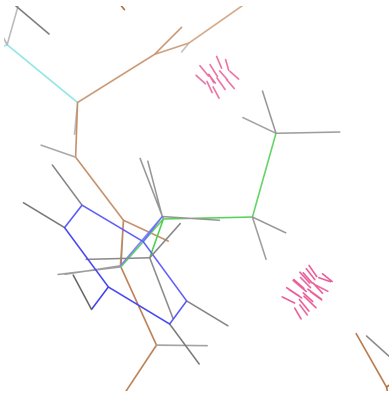

Fig. S2. For the design run optimizing the difference in energy between the reduced and oxidized states of angiotensinogen (PDB ids 2wxy and 2wxx, respectively), the single mutation Y12I (blue to green) was found to fit well into the reduced state, but to cause steric clashes (pink) in the oxidized state. The selection of this single mutant as optimal required explicit multistate design to destabilize the undesired oxidized state, while maintaining the stability of the reduced state: optimization of either the reduced or oxidized state alone yielded aromatic residues at position 12.

The third type of system was stabilization of a single protein, but in a way meant to produce stabilizing mutations that are robust to the choice of force field. This strategy may be useful for obtaining mutations that are more likely to work in practice. For these calculations, the protein's energy was optimized using the AMBER force field, but the energies using both the AMBER and CHARMM force fields were constrained to improve relative to the wild type. The proteins used here were drawn from the test set of protein energy optimizations in Gainza, Roberts, and Donald.[4] The structures' PDB ids were 2o9s, 2qsk, 2ril, 3a38, and 3g36.

Finally, the fourth type of system was the stabilization of the reduced form of angiotensinogen relative to the oxidized form (Fig. S2) or vice versa. This was treated as an unconstrained optimization of the difference between the states, which have been crystallized (PDB ids 2wxy and 2wxx respectively[13]). This type of setup arises in experimental studies of protein function, when one wishes to design a mutant to "freeze" a protein in one of two or more conformational states in order to study the functional role of the chosen state. For example, the N600K mutation in the motor protein Ncd[12] induces Ncd to adopt a conformational state typically only induced by ATP binding; studies of this mutant helped to establish the temporal and mechanistic relationship between the nucleotide binding and force generation of this motor.[7]

C.2 Details of protein design runs

Table 1. Protein design test cases with continuous flexibility, as described in Section 3 of the main text. Type is "aff" for designs for affinity, "stab" for designs for stability robust to force field choice, "spec" for designs to be specific to one complex, "multi" for designs to be multispecific to more than one, and "red" or "ox" for designs to favor either a reduced or oxidized state (each type is described further in Section C.1). As described in Section 3 of the main text, s is the number of states, N is the number of sequences in the designed states, m is the number of sequence tree nodes created, and g is the number of state GMECs computed. Subscripts 1 and 5 denote calculation of the best sequence or enumeration of the best 5 sequences respectively. N_e denotes the number of sequences enumerated (all runs were set to enumerate the five best sequences, but $N_e < 5$ if less than five sequences in the design space satisfy the constraints). Table continues on next page.

Protein redesigned	Mutable residues	PDB id(s)	Type	s	N	N_e	m_1	m_5	g_1	g_5
Beta-2-microglobulin	52, 54, 56, 57, 63	1nez	aff	2	5488	5	2386	2386	84	110
Papain	18, 19, 21, 159, 177, 181	1stf	aff	2	9408	5	5784	5784	7	15
Trypsin	189, 190, 192, 195, 213	BPTI set*	multi	20	896	0				
Trypsin	189, 190, 192, 195, 214	BPTI set*	spec	20	896	0				
Scytovirin	1, 6, 10, 13, 28, 43, 48, 58, 61, 76	2qsk	stab	2	896	1	35		2	
Putative monooxygenase	5, 13, 21, 55, 57, 59, 61, 70	2ril	stab	2	9604	0				
High-potential iron-sulfur protein	6, 7, 14, 19, 23, 26, 41, 60, 64, 69, 73, 74, 78	3a38	stab	2	1882384	5	25708	40141	2	10
dpy-30-like protein	64, 68, 87, 91	3g36	stab	2	196	0				
Actin	24, 25, 26, 28, 345, 349	2a40/2a41	multi	4	392	1	248		97	
CED-4	1, 5, 227, 229, 259, 265, 279, 282	2a5y/3lqr	multi	4	288	5	221	221	85	96
CED-5	1, 5, 227, 229, 259, 265, 279, 283	2a5y/3lqr	spec	4	288	1	217		78	
Rab-11A	44, 46, 47, 48, 50	2gzd/2gzh	multi	4	2744	0				

*BPTI set: 3btd, 3bte, 3btf, 3btg, 3bth, 3btk, 3btm, 3btq, 3btt, 3btw

Protein redesigned	Mutable residues	PDB id(s)	Type	s	N	N_e	m_1	m_5	g_1	g_5
Rab-11A	44, 46, 47, 48, 51	2gzd/2gzh	spec	4	2744	5	601	601	28	40
DNA polymerase beta	291, 309, 311, 322, 324	3k75/3lqc	multi	4	2744	0				
DNA polymerase beta	291, 309, 311, 322, 325	3k75/3lqc	spec	4	2744	3	1388		556	
Polyadenylate-binding protein 1	564, 571, 580, 582, 584	3ktp/3ktr	multi	4	448	1	352		16	
Polyadenylate-binding protein 1	564, 571, 580, 582, 585	3ktp/3ktr	spec	4	448	0				
CDO	867, 872, 874, 901, 918	3n1f/3n1q	multi	4	112	0				
CDO	867, 872, 874, 901, 919	3n1f/3n1q	spec	4	112	1	112		7	
Brother of CDO	753, 756, 758, 760, 789, 804	3n1g/3n1m	multi	4	224	5	205	224	4	25
Brother of CDO	753, 756, 758, 760, 789, 805	3n1g/3n1m	spec	4	224	1	224		6	
Type I TGFbeta receptor	199, 203, 267, 268, 269	1b6c	aff	2	64	5	64	64	2	13
Leupeptin inhibitor	356, 357	2nqa	aff	2	361	5	91	145	5	13
Ponsin	824, 826, 828, 834, 840, 842, 848, 849, 850, 859, 861, 863, 872, 877	2o9s	stab	2	4374	1	665		2	
Angiotensinogen	12, 15, 136, 138, 140	2wxy/2wxx	red	2	224	5	153	180	38	54
Angiotensinogen	12, 15, 136, 138, 141	2wxx/2wxy	ox	2	224	5	203	207	10	22
Sec24d	833, 834, 835, 836, 1025	3efo/3eg9	spec	4	4116	2	1830		565	

Table 2. Protein design test cases without continuous flexibility. Columns as in Table 1. Table continues on next page.

Protein redesigned	Mutable residues	PDB id(s)	Type	s	N	N_e	m_1	m_5	g_1	g_5
Leupeptin inhibitor	356, 357	2nqa	aff	2	361	5	37	73	2	10
Trypsin	189, 190, 192, 195, 213	BPTI set*	spec	20	2476099	5	14707	14707	20	100
Scytovirin	1, 6, 10, 13, 28, 43, 48, 58, 61, 76	2qsk	stab	2	$4.7\text{x}10^{7}$	1	109		2	
Scytovirin	1, 6, 10, 13, 28, 43, 48, 58, 61, 76	2ril	stab	2	$1.7\text{x}10^{10}$	0				
dpy-30-like protein	64, 68, 87, 91	3g36	stab	2	6859	2	73		2	
CED-4	1, 5, 227, 229, 259, 265, 279, 282	2a5y/3lqr	multi	4	87808	5	39	151	4	20
CED-4	1, 5, 227, 229, 259, 265, 279, 282	2a5y/3lqr	spec	4	87808	5	30	87	4	20
Rab-11A	44, 46, 47, 48, 50	2gzd/2gzh	multi	4	2476099	5	42193	45883	4	20
Rab-11A	44, 46, 47, 48, 50	2gzd/2gzh	spec	4	2476099	5	59077	61525	4	20
Cdc42	62, 63, 64	2ngr/1grn	multi	4	6859	0				
Cdc43	62, 63, 65	2ngr/1grn	spec	4	6859	0				
Sec24a	430, 435, 496, 748, 750, 752, 808	3egd/3egx	multi	4	5488	5	1480	1480	18	46
Sec24a	430, 435, 496, 748, 750, 752, 809	3egd/3egx	spec	4	5488	5	1480	1480	4	20
DNA polymerase beta	291, 309, 311, 322, 324	3k75/3lqc	multi	4	2744	0				
DNA polymerase beta	291, 309, 311, 322, 324	3k75/3lqc	spec	4	2744	0				

Protein redesigned	Mutable residues	PDB id(s)	Type	s	N	N_e	m_1	m_5	g_1	g_5
Poly-adenylate-binding protein 1	564, 571, 580, 582, 584	3ktp/3ktr	multi	4	2476099	5	4249	5059	4	20
Poly-adenylate-binding protein 1	564, 571, 580, 582, 584	3ktp/3ktr	spec	4	2476099	5	1315	2593	4	20
CDO	867, 872, 874, 901, 918	3n1f/3n1q	multi	4	2476099	5	50851	514274		20
CDO	867, 872, 874, 901, 918	3n1f/3n1q	spec	4	2476099	5	60661	638474		20
PAF-acetyl-hydrolase	194, 212, 235, 236, 238, 254, 316	1vyh	aff	2	1792	5	1275	1309	2	10
Protein tyrosine phos-phatase 1B	7, 11, 12, 268, 269, 272	2b4s	aff	2	2744	5	647	647	2	10
RING2	22, 26, 29, 32, 36	2h0d	aff	2	2744	5	206	236	2	10
Angio-tensinogen	12, 15, 136, 138, 141	2wxx/2wxy	ox	2	224	5	27	31	2	10
Sec24d	833, 834, 835, 836, 1025	3efo/3eg9	multi	4	4116	5	634	640	12	36
Sec24d	833, 834, 835, 836, 1025	3efo/3eg9	spec	4	4116	4	704		6	

References

1. Johan Desmet, Marc de Maeyer, Bart Hazes, and Ignace Lasters. The dead-end elimination theorem and its use in protein side-chain positioning. *Nature*, 356:539–542, 1992.
2. Johan Desmet, Jan Spriet, and Ignace Lasters. Fast and accurate side-chain topology and energy refinement (FASTER) as a new method for protein structure optimization. *Proteins: Structure, Function, and Bioinformatics*, 48(1):31–43, 2002.
3. Kathleen M. Frey, Ivelin Georgiev, Bruce R. Donald, and Amy C. Anderson. Predicting resistance mutations using protein design algorithms. *Proceedings of the National Academy of Sciences of the USA*, 107(31):13707–13712, 2010.
4. Pablo Gainza, Kyle Roberts, and Bruce R. Donald. Protein design using continuous rotamers. *PLoS Computational Biology*, 8(1):e1002335, 2012.
5. Ivelin Georgiev, Ryan H. Lilien, and Bruce R. Donald. The minimized dead-end elimination criterion and its application to protein redesign in a hybrid scoring

and search algorithm for computing partition functions over molecular ensembles. *Journal of Computational Chemistry*, 29(10):1527–1542, 2008.

6. Mark A. Hallen, Daniel A. Keedy, and Bruce R. Donald. Dead-end elimination with perturbations (DEEPer): A provable protein design algorithm with continuous sidechain and backbone flexibility. *Proteins: Structure, Function and Bioinformatics*, 81(1):18–39, 2013.
7. Mark A. Hallen, Zhang-Yi Liang, and Sharyn A. Endow. Two-state displacement by the kinesin-14 Ncd stalk. *Biophysical Chemistry*, 154(2-3):56–65, 2011.
8. Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.
9. Ronny Helland, Jacek Otlewski, Ottar Sundheim, Michal Dadlez, and Arne O. Smalås. The crystal structures of the complexes between bovine β-trypsin and ten P_1 variants of BPTI. *Journal of Molecular Biology*, 287(5):923–942, 1999.
10. Andrew R. Leach and Andrew P. Lemon. Exploring the conformational space of protein side chains using dead-end elimination and the A* algorithm. *Proteins: Structure, Function, and Bioinformatics*, 33(2):227–239, 1998.
11. Michael J. Meyer, Jishnu Das, Xiujuan Wang, and Haiyuan Yu. INstruct: a database of high-quality 3D structurally resolved protein interactome networks. *Bioinformatics*, 29(12):1577–1579, 2013.
12. Hebok Song and Sharyn A. Endow. Decoupling of nucleotide- and microtubule-binding sites in a kinesin mutant. *Nature*, 396:587–590, 1998.
13. Aiwu Zhou, Robin W. Carrell, Michael P. Murphy, Zhenquan Wei, Yahui Yan, Peter L. D. Stanley, Penelope E. Stein, Fiona Broughton Pipkin, and Randy J. Read. A redox switch in angiotensinogen modulates angiotensin release. *Nature*, 468:108–111, 2010.