Supplementary Information

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

[1] Department of Biochemistry, Duke University Medical Center and [2] Department of Computer Science, Duke University, Durham, NC

This Supplementary Information further describes the methods used in DEEPer by providing additional details and proofs. Section A proves the validity of the indirect pruning algorithms, Algorithms 1 (proved in A.1) and 2 (in A.2), and derives their time complexity (A.3). Section B describes some details of implementation, relating to the application of perturbations (B.1) and the choice of pruning zones for indirect pruning (B.2).

A Proofs of indirect pruning conditions and time complexity analysis

A.1 Proof of single-RC pruning condition

Let us show the indirect pruning algorithm for pruning single RCs, algorithm 1, is correct by proving theorem 1. First of all, we must consider the DEE pruning condition for a conformation $\mathbf{u}$ of Z, where $\mathbf{u}$ is a z-tuple of RCs. We use the iMinDEE minimization-aware version [28] of Goldstein [40]'s DEE pruning condition for tuples.

Lemma 1. *Let* $\mathbf{u}$ *be a* z*-tuple of RCs with* $Z = M_{\mathbf{u}}$*. If there exists a* z*-tuple* $\mathbf{v}$ *of RCs specifying a conformation of* Z *such that*

$$\sum_{h \in Z} \left(E_{\ominus}(h_{\mathbf{u}}) - E_{\ominus}(h_{\mathbf{v}}) + \sum_{h' \in Z, h' < h} \left(E_{\ominus}(h_{\mathbf{u}}, h'_{\mathbf{u}}) - E_{\ominus}(h_{\mathbf{v}}, h'_{\mathbf{v}}) \right) \right) + \sum_{j \in N} \min_{j_s \in R_j} \sum_{h \in Z} \left(E_{\ominus}(h_{\mathbf{u}}, j_s) - E_{\ominus}(h_{\mathbf{v}}, j_s) \right) > E_w + I, \tag{4}$$

then $\mathbf{u}$ *can be pruned, meaning that* Z *is not found in the conformation* $\mathbf{u}$ *in any overall protein conformation whose energy is within* E_w *of the GMEC.*

Proof. We begin with the iMinDEE version of the Goldstein pruning condition, Eq. (1). As noted by [40], this condition need not be applied only to pruning rotamers of single residues; rather, it may be applied to prune conformations of sets of residues ([40] refers to these as *renormalized residues*). In our case, it will be used to prune conformations of the pruning zone Z. Therefore, if there exists a z-tuple $\mathbf{v}$ of RCs specifying a conformation of Z such that

$$E_{\ominus}(\mathbf{u}) - E_{\ominus}(\mathbf{v}) + \sum_{j \in N} \min_{j_s} \big(E_{\ominus}(\mathbf{u}, j_s) - E_{\ominus}(\mathbf{v}, j_s) \big) > E_w + I, \tag{5}$$

then $\mathbf{u}$ can be pruned, with $E_{\ominus}(\cdot)$ denoting the intra-energy lower bound of a conformation and $E_{\ominus}(\cdot,\cdot)$ denoting the pairwise-interaction-energy lower bound of a pair of conformations as usual. The intra-energy of $\mathbf{u}$ is the sum of the intra-energies of its constituent RCs plus the sum of the pairwise energies among these constituent RCs:

$$E_{\ominus}(\mathbf{u}) = \sum_{h \in Z} \left(E_{\ominus}(h_{\mathbf{u}}) + \sum_{h' \in Z, h' < h} E_{\ominus}(h_{\mathbf{u}}, h'_{\mathbf{u}}) \right). \tag{6}$$

The pairwise energy between Z in conformation $\mathbf{u}$ and a given residue j in RC j_s is simply the sum of pairwise energies of the constituent RCs of $\mathbf{u}$ with j_s:

$$E_{\ominus}(\mathbf{u}, j_s) = \sum_{h \in Z} E_{\ominus}(h_{\mathbf{u}}, j_s). \tag{7}$$

Similarly, if Z is in conformation $\mathbf{v}$, then its intra-energy is

$$E_{\ominus}(\mathbf{v}) = \sum_{h \in Z} \left(E_{\ominus}(h_{\mathbf{v}}) + \sum_{h' \in Z, h' < h} E_{\ominus}(h_{\mathbf{v}}, h'_{\mathbf{v}}) \right) \tag{8}$$

and its pairwise energy with j_s is

$$E_{\ominus}(\mathbf{v}, j_s) = \sum_{h \in Z} E_{\ominus}(h_{\mathbf{v}}, j_s). \tag{9}$$

Substituting the energies Eq. (6) through Eq. (9) into the pruning condition Eq. (5) yields Eq. (4), and the lemma follows. □

Now we are ready to prove the validity of indirect pruning.

Theorem 1. *If Algorithm 1 prunes an RC i_r, then no protein conformation whose energy is within E_w of the GMEC will contain i_r.*

Proof. Suppose the algorithm prunes i_r using i_t. Let $\mathbf{v}$ be as in step 2 of the algorithm for this pruning. This means that there exists a z-tuple $\mathbf{v}$ of RCs such that $i_{\mathbf{v}} = i_t$, $h'_{\mathbf{v}} \in R_{h'}(h_{\mathbf{v}})$ for all $h, h' \in Z$ (i.e. $\mathbf{v}$ specifies a valid conformation of the pruning zone), and

$$\begin{aligned} q_f &= \sum_{h \in Z} \min_{h_a \in R_h(i_r)} K(h_a, h_{\mathbf{v}}) \\ &= \sum_{h \in Z} \min_{h_a \in R_h(i_r)} \Bigg(E_{\ominus}(h_a) - E_{\ominus}(h_{\mathbf{v}}) + \sum_{h' \in Z, h' < h} \Big(\min_{h'_{a'} \in R_{h'}(h_a)} E_{\ominus}(h_a, h'_{a'}) \\ &\quad - \max_{h'_{b'} \in R_{h'}(h_{\mathbf{v}})} E_{\ominus}(h_{\mathbf{v}}, h'_{b'}) \Big) + \sum_{j \in N} \min_{j_s \in R_j} \big(E_{\ominus}(h_a, j_s) - E_{\ominus}(h_{\mathbf{v}}, j_s) \big) \Bigg) > E_w + I. \end{aligned} \tag{10}$$

where q_f denotes the value of q when the algorithm is finished. Because $h'_{\mathbf{v}} \in R_{h'}(h_{\mathbf{v}})$ for any residues $h, h' \in Z$, $\max_{h'_{b'} \in R_{h'}(h_{\mathbf{v}})} E_{\ominus}(h_{\mathbf{v}}, h'_{b'}) \geq E_{\ominus}(h_{\mathbf{v}}, h'_{\mathbf{v}})$, which may be substituted into Eq. (10) to yield

$$\begin{aligned} &\sum_{h \in Z} \min_{h_a \in R_h(i_r)} \Bigg(E_{\ominus}(h_a) - E_{\ominus}(h_{\mathbf{v}}) + \sum_{h' \in Z, h' < h} \Big(\min_{h'_{a'} \in R_{h'}(h_a)} E_{\ominus}(h_a, h'_{a'}) \\ &\quad - E_{\ominus}(h_{\mathbf{v}}, h'_{\mathbf{v}}) \Big) + \sum_{j \in N} \min_{j_s \in R_j} \big(E_{\ominus}(h_a, j_s) - E_{\ominus}(h_{\mathbf{v}}, j_s) \big) \Bigg) > E_w + I. \end{aligned} \tag{11}$$

The substitution preserves the inequality because the left-hand-side of Eq. (11), which contains the substitution, is greater than or equal to the left-hand-side of Eq. (10). Now, let U be the

set of conformations of the pruning zone that contain i_r : $U = \{\mathbf{u} \mid M_\mathbf{u} = Z, i_\mathbf{u} = i_r, h'_\mathbf{u} \in R_{h'}(h_\mathbf{u}) \; \forall \; h, h' \in Z\}$. Applying the definition of a minimum to the $\min_{h_a \in R_h(i_r)}$ in Eq. (11), we know that for any z-tuple of RCs $\mathbf{u} \in U$,

$$\sum_{h \in Z} \Bigg(E_\ominus(h_\mathbf{u}) - E_\ominus(h_\mathbf{v}) + \sum_{h' \in Z, h' < h} \big(\min_{h'_{a'} \in R_{h'}(h_\mathbf{u})} E_\ominus(h_\mathbf{u}, h'_{a'}) \\ - E_\ominus(h_\mathbf{v}, h'_\mathbf{v}) \big) + \sum_{j \in N} \min_{j_s \in R_j} \big(E_\ominus(h_\mathbf{u}, j_s) - E_\ominus(h_\mathbf{v}, j_s) \big) \Bigg) > E_w + I. \tag{12}$$

Since $h'_\mathbf{u} \in R_{h'}(h_\mathbf{u})$ for each $h, h' \in Z$, $\min_{h'_{a'} \in R_{h'}(h_\mathbf{u})} E_\ominus(h_\mathbf{u}, h'_{a'}) \leq E_\ominus(h_\mathbf{u}, h'_\mathbf{u})$, which we may substitute into Eq. (12) to yield

$$\sum_{h \in Z} \Bigg(E_\ominus(h_\mathbf{u}) - E_\ominus(h_\mathbf{v}) + \sum_{h' \in Z, h' < h} \big(E_\ominus(h_\mathbf{u}, h'_\mathbf{u}) - E_\ominus(h_\mathbf{v}, h'_\mathbf{v}) \big) \\ + \sum_{j \in N} \min_{j_s \in R_j} \big(E_\ominus(h_\mathbf{u}, j_s) - E_\ominus(h_\mathbf{v}, j_s) \big) \Bigg) > E_w + I, \tag{13}$$

for each $\mathbf{u} \in U$. Finally, observe that

$$\sum_{h \in Z} \sum_{j \in N} \min_{j_s \in R_j} \big(E_\ominus(h_\mathbf{u}, j_s) - E_\ominus(h_\mathbf{v}, j_s) \big) \leq \sum_{j \in N} \min_{j_s \in R_j} \sum_{h \in Z} \big(E_\ominus(h_\mathbf{u}, j_s) - E_\ominus(h_\mathbf{v}, j_s) \big)$$

and therefore that the left-hand-side of Eq. (13) is less than or equal to that of Eq. (4), for a given $\mathbf{u} \in U$. Therefore Eq. (4) holds for each $\mathbf{u} \in U$, i.e. for each $\mathbf{u}$ specifying a valid conformation of Z with $i_\mathbf{u} = i_r$. Therefore, by Lemma 1, we can prune each conformation $\mathbf{u}$ of Z with $i_\mathbf{u} = i_r$, i.e. no such $\mathbf{u}$ will be in the ensemble of protein conformations within E_w of the GMEC. Thus, there is no way for i_r to be found in this ensemble, and we can prune i_r too. □

A.2 Proof of tuples pruning condition

Theorem 2. *If Algorithm 2 prunes a tuple of RCs* $\mathbf{r}$*, then no protein conformation whose energy is within* E_w *of the GMEC will contain* $\mathbf{r}$.

Proof. The proof is very similar to that of Theorem 1. Suppose the algorithm prunes $\mathbf{r}$ using $\mathbf{t}$, with $M_{\mathbf{r}} = M_{\mathbf{t}} \subseteq Z$. Let $\mathbf{v}$ be as in the algorithm for this pruning. This means that there exists a z-tuple $\mathbf{v}$ of RCs such that $i_{\mathbf{v}} = i_{\mathbf{t}}$ for all $i \in M_{\mathbf{t}}$, $h'_{\mathbf{v}} \in R_{h'}(h_{\mathbf{v}}) \; \forall \; h, h' \in Z$ (i.e. $\mathbf{v}$ specifies a valid conformation of the pruning zone), and

$$\begin{aligned} q_f &= \sum_{h \in Z - M_{\mathbf{r}}} \left(\min_{h_a \in R_h(\mathbf{r})} K(h_a, h_{\mathbf{v}}) \right) + K(\mathbf{r}, \mathbf{t}) \\ = \sum_{h \in Z - M_{\mathbf{r}}} \min_{h_a \in R_h(\mathbf{r})} \Bigg(& E_{\ominus}(h_a) - E_{\ominus}(h_{\mathbf{v}}) + \sum_{h' \in Z, h' < h} \Big(\min_{h'_{a'} \in R_{h'}(h_a)} E_{\ominus}(h_a, h'_{a'}) \\ & - \max_{h'_{b'} \in R_{h'}(h_{\mathbf{v}})} E_{\ominus}(h_{\mathbf{v}}, h'_{b'}) \Big) + \sum_{j \in N} \min_{j_s \in R_j} \big(E_{\ominus}(h_a, j_s) - E_{\ominus}(h_{\mathbf{v}}, j_s) \big) \Bigg) \\ & + K(\mathbf{r}, \mathbf{t}) > E_w + I. \end{aligned} \tag{14}$$

where q_f denotes the value of q when the algorithm is finished. Now, because $h'_{\mathbf{v}} \in R_{h'}(h_{\mathbf{v}})$ for any residues $h, h' \in Z$, $\max_{h'_{b'} \in R_{h'}(h_{\mathbf{v}})} E_{\ominus}(h_{\mathbf{v}}, h'_{b'}) \geq E_{\ominus}(h_{\mathbf{v}}, h'_{\mathbf{v}})$, which may be substituted into Eq. (14) to yield

$$\begin{aligned} \sum_{h \in Z - M_{\mathbf{r}}} \min_{h_a \in R_h(\mathbf{r})} \Bigg(& E_{\ominus}(h_a) - E_{\ominus}(h_{\mathbf{v}}) + \sum_{h' \in Z, h' < h} \Big(\min_{h'_{a'} \in R_{h'}(h_a)} E_{\ominus}(h_a, h'_{a'}) \\ & - E_{\ominus}(h_{\mathbf{v}}, h'_{\mathbf{v}}) \Big) + \sum_{j \in N} \min_{j_s \in R_j} \big(E_{\ominus}(h_a, j_s) - E_{\ominus}(h_{\mathbf{v}}, j_s) \big) \Bigg) \\ & + K(\mathbf{r}, \mathbf{t}) > E_w + I. \end{aligned} \tag{15}$$

The substitution preserves the inequality because the left-hand-side of Eq. (15), which contains the substitution, is greater than or equal to the left-hand-side of Eq. (14). Now, let U be the set of valid conformations of the pruning zone that contain $\mathbf{r}$, so $U = \{\mathbf{u} \mid M_{\mathbf{u}} = Z, h_{\mathbf{u}} \in R_h(\mathbf{r}), h'_{\mathbf{u}} \in R_{h'}(h_{\mathbf{u}}) \; \forall \; h, h' \in Z\}$. Applying the definition of a minimum to the $\min_{h_a \in R_h(\mathbf{r})}$ in Eq. (15), we know

that for any z-tuple of RCs $\mathbf{u} \in U$,

$$\sum_{h \in Z - M_{\mathbf{r}}} \Bigg(E_{\ominus}(h_{\mathbf{u}}) - E_{\ominus}(h_{\mathbf{v}}) + \sum_{h' \in Z, h' < h} \big(\min_{h'_{a'} \in R_{h'}(h_{\mathbf{u}})} E_{\ominus}(h_{\mathbf{u}}, h'_{a'}) - E_{\ominus}(h_{\mathbf{v}}, h'_{\mathbf{v}}) \big) + \sum_{j \in N} \min_{j_s \in R_j} \big(E_{\ominus}(h_{\mathbf{u}}, j_s) - E_{\ominus}(h_{\mathbf{v}}, j_s) \big) \Bigg) + K(\mathbf{r}, \mathbf{t}) > E_w + I. \tag{16}$$

Since $h'_{\mathbf{u}} \in R_{h'}(h_{\mathbf{u}})$ for each $h, h' \in Z$, $\min_{h'_{a'} \in R_{h'}(h_{\mathbf{u}})} E_{\ominus}(h_{\mathbf{u}}, h'_{a'}) \leq E_{\ominus}(h_{\mathbf{u}}, h'_{\mathbf{u}})$, which we may substitute into Eq. (16) to yield

$$\sum_{h \in Z - M_{\mathbf{r}}} \Bigg(E_{\ominus}(h_{\mathbf{u}}) - E_{\ominus}(h_{\mathbf{v}}) + \sum_{h' \in Z, h' < h} \big(E_{\ominus}(h_{\mathbf{u}}, h'_{\mathbf{u}}) - E_{\ominus}(h_{\mathbf{v}}, h'_{\mathbf{v}}) \big) + \sum_{j \in N} \min_{j_s \in R_j} \big(E_{\ominus}(h_{\mathbf{u}}, j_s) - E_{\ominus}(h_{\mathbf{v}}, j_s) \big) \Bigg) + K(\mathbf{r}, \mathbf{t}) > E_w + I, \tag{17}$$

for each $\mathbf{u} \in U$. Similarly, $\min_{h'_{a'} \in R_{h'}(\mathbf{r})} \sum_{i \in M_{\mathbf{r}}} \big(\chi_{h' < i} E_{\ominus}(i_{\mathbf{r}}, h'_{a'}) \big) \leq \sum_{i \in M_{\mathbf{r}}} \big(\chi_{h' < i} E_{\ominus}(i_{\mathbf{r}}, h'_{\mathbf{u}}) \big)$ for each $\mathbf{u} \in U$ and $\max_{h'_{b'} \in R_{h'}(\mathbf{t})} \sum_{i \in M_{\mathbf{r}}} \big(\chi_{h' < i} E_{\ominus}(i_{\mathbf{t}}, h'_{b'}) \big) \geq \sum_{i \in M_{\mathbf{r}}} \big(\chi_{h' < i} E_{\ominus}(i_{\mathbf{t}}, h'_{\mathbf{v}}) \big)$, where $h' \in Z - M_{\mathbf{r}}$, both of which we may substitute into the definition of $K(\mathbf{r}, \mathbf{t})$, Eq. (3):

$$\begin{aligned} K(\mathbf{r}, \mathbf{t}) \leq & \sum_{i \in M_{\mathbf{r}}} \Big(E_{\ominus}(i_{\mathbf{r}}) - E_{\ominus}(i_{\mathbf{t}}) + \sum_{h' \in M_{\mathbf{r}}, h' < i} \big(E_{\ominus}(i_{\mathbf{r}}, h'_{\mathbf{r}}) - E_{\ominus}(i_{\mathbf{t}}, h'_{\mathbf{t}}) \big) \Big) \\ & + \sum_{h' \in Z - M_{\mathbf{r}}} \Big(\sum_{i \in M_{\mathbf{r}}} \big(\chi_{h' < i} E_{\ominus}(i_{\mathbf{r}}, h'_{\mathbf{u}}) \big) - \sum_{i \in M_{\mathbf{r}}} \big(\chi_{h' < i} E_{\ominus}(i_{\mathbf{t}}, h'_{\mathbf{v}}) \big) \Big) \\ & + \sum_{j \in N} \min_{j_s \in R_j} \sum_{i \in M_{\mathbf{r}}} \big(E_{\ominus}(i_{\mathbf{r}}, j_s) - E_{\ominus}(i_{\mathbf{t}}, j_s) \big) \end{aligned} \tag{18}$$

for each $\mathbf{u} \in U$. We may then substitute Eq. (18) into Eq. (17), yielding

$$\sum_{h \in Z} \left(E_{\ominus}(h_{\mathbf{u}}) - E_{\ominus}(h_{\mathbf{v}}) + \sum_{h' \in Z, h' < h} (E_{\ominus}(h_{\mathbf{u}}, h'_{\mathbf{u}}) - E_{\ominus}(h_{\mathbf{v}}, h'_{\mathbf{v}})) \right) + \sum_{h \in Z - M_{\mathbf{r}}} \sum_{j \in N} \min_{j_s \in R_j} \left(E_{\ominus}(h_{\mathbf{u}}, j_s) - E_{\ominus}(h_{\mathbf{v}}, j_s) \right) + \sum_{j \in N} \min_{j_s \in R_j} \sum_{i \in M_{\mathbf{r}}} \left(E_{\ominus}(i_{\mathbf{r}}, j_s) - E_{\ominus}(i_{\mathbf{t}}, j_s) \right) > E_w + I \tag{19}$$

for each $\mathbf{u} \in U$. Finally, observe that

$$\sum_{h \in Z - M_{\mathbf{r}}} \sum_{j \in N} \min_{j_s \in R_j} \left(E_{\ominus}(h_{\mathbf{u}}, j_s) - E_{\ominus}(h_{\mathbf{v}}, j_s) \right) + \sum_{j \in N} \min_{j_s \in R_j} \sum_{i \in M_{\mathbf{r}}} \left(E_{\ominus}(i_{\mathbf{r}}, j_s) - E_{\ominus}(i_{\mathbf{t}}, j_s) \right) \leq \sum_{j \in N} \min_{j_s \in R_j} \sum_{h \in Z} \left(E_{\ominus}(h_{\mathbf{u}}, j_s) - E_{\ominus}(h_{\mathbf{v}}, j_s) \right) \tag{20}$$

and therefore that the left-hand-side of Eq. (19) is less than or equal to that of Eq. (4), for a given $\mathbf{u} \in U$. Therefore Eq. (4) holds for each $\mathbf{u} \in U$, i.e. for each $\mathbf{u}$ specifying a valid conformation of Z with $i_{\mathbf{u}} = i_{\mathbf{r}}$ for each $i \in N$. Therefore, by Lemma 1, we can prune all conformations $\mathbf{u}$ of Z consistent with $\mathbf{r}$, i.e. no such $\mathbf{u}$ will be in the ensemble of protein conformations within E_w of the GMEC; thus, there is no way for $\mathbf{r}$ to be found in this ensemble, and so we can prune $\mathbf{r}$ as well. □

A.3 Time complexity

First, we analyze the complexity of Algorithm 1.

Lemma 2. *Indirect pruning of single RCs for a pruning zone Z runs in $O(z^2r^4 + znr^3 + z^3r^3)$ time.*

Proof. The time complexity of pruning single RCs for a pruning zone is the sum of the time to calculate $K(i_r, i_t)$ using Eq. (2) for each i in Z, and for each ordered pair of RCs (i_r, i_t) at i with

$i_r \neq i_t$, plus the time to evaluate the pruning condition (step 2 of Algorithm 1) for each such pair. At each residue, there are at most r^2 ordered pairs of RCs at the same residue including repeats (i_r, i_r), so there are less than zr^2 pairs of RCs to consider in total.

For a given RC pair (h_a, h_b), it takes constant time to evaluate $E_\ominus(h_a) - E_\ominus(h_b)$. It takes $O(r)$ time to evaluate $\min_{h'_{a'} \in R_{h'}(h_a)} E_\ominus(h_a, h'_{a'})$ or $\max_{h'_{b'} \in R_{h'}(h_b)} E_\ominus(h_b, h'_{b'})$ for some $h' \in Z$, because at most r RCs need to be considered to find the maximum or minimum, so it takes $O(zr)$ time to evaluate $\sum_{h' \in Z, h' < h} \big(\min_{h'_{a'} \in R_{h'}(h_a)} E_\ominus(h_a, h'_{a'}) - \max_{h'_{b'} \in R_{h'}(h_b)} E_\ominus(h_b, h'_{b'}) \big)$. Finally, it takes $O(r)$ time to evaluate $\min_{j_s \in R_j} \big(E_\ominus(h_a, j_s) - E_\ominus(h_b, j_s) \big)$, and therefore it takes $O(nr)$ time to evaluate $\sum_{j \in N} \min_{j_s \in R_j} \big(E_\ominus(h_a, j_s) - E_\ominus(h_b, j_s) \big)$. Thus, it takes $O(1 + zr + nr) = O(nr)$ time to evaluate Eq. (2), and so the computation of $K(i_r, i_t)$ for each of the $O(zr^2)$ pairs (i_r, i_t) takes $O(znr^3)$ time.

Next, for each residue $h \in Z$, evaluating the maximum of a minimum in step 2 of the algorithm will require performing less than r^2 comparisons of terms $K(h_a, h_b)$ and less than zr checks of the condition $h_{\mathbf{v}} \in R_h(j_{\mathbf{v}})$ and so will take $O(r^2 + zr)$ time. Because there are z residues at which this must be performed, the time required for step 2 for a given RC ordered pair (i_r, i_t) is $O(zr^2 + z^2 r)$, and so the total time to evaluate the pruning condition for all $O(zr^2)$ pairs is $O(z^2 r^4 + z^3 r^3)$.

Consequently, the total time required to perform pruning of single RCs for a pruning zone Z is $O(z^2 r^4 + znr^3 + z^3 r^3)$. □

Next, we analyze the complexity of Algorithm 2.

Lemma 3. *Indirect pruning of m-tuples of RCs for a pruning zone Z runs in $O\big(z^m r^{2m+1}(n + z^2 + zr)\big)$ time.*

Proof. Let us first assume that we have already precalculated $K(i_r, i_t)$ for each i in Z, and for each ordered pair of RCs (i_r, i_t) at i with $i_r \neq i_t$; the cost of this precomputation will be considered later.

To prune m-tuples in Z we must consider each of the $\binom{Z}{m} = O(z^m)$ possible m-tuples of residues in Z, and assign a pair of RCs for each residue out of the $O(r^2)$ pairs available, so that we have one tuple $\mathbf{r}$ to prune using another $\mathbf{t}$. This gives us $O(z^m r^{2m})$ pairs of tuples to eval-

uate for pruning. It takes $O(r^2 + zr)$ time to evaluate the maximum of a minimum term for $h \in Z$, and so overall it takes $O(zr^2 + z^2r)$ time to perform step 2 of the algorithm. This leaves the $K(\mathbf{r}, \mathbf{t})$ term, which requires evaluating Eq. (3) once. We require constant time (with respect to n, r, and z) to evaluate $\sum_{i \in M_\mathbf{r}} \Big(E_\ominus(i_\mathbf{r}) - E_\ominus(i_\mathbf{t}) + \sum_{h' \in M_\mathbf{r}, h' < i} \big(E_\ominus(i_\mathbf{r}, h'_\mathbf{r}) - E_\ominus(i_\mathbf{t}, h'_\mathbf{t}) \big) \Big)$. It takes $O(r)$ time to evaluate $\min_{h'_{a'} \in R_{h'}(\mathbf{r})} \sum_{i \in M_\mathbf{r}} \big(\chi_{h'<i} E_\ominus(i_\mathbf{r}, h'_{a'}) \big)$ or $\max_{h'_{b'} \in R_{h'}(\mathbf{t})} \sum_{i \in M_\mathbf{r}} \big(\chi_{h'<i} E_\ominus(i_\mathbf{t}, h'_{b'}) \big)$ for $h' \in Z - M_\mathbf{r}$, so it takes $O(zr)$ time to evaluate the $\sum_{h' \in Z - M_\mathbf{r}} \Big(\min_{h'_{a'} \in R_{h'}(\mathbf{r})} \sum_{i \in M_\mathbf{r}} \big(\chi_{h'<i} E_\ominus(i_\mathbf{r}, h'_{a'}) \big) - \max_{h'_{b'} \in R_{h'}(\mathbf{t})} \sum_{i \in M_\mathbf{r}} \big(\chi_{h'<i} E_\ominus(i_\mathbf{t}, h'_{b'}) \big) \Big)$ term. Finally it takes $O(r)$ time to evaluate $\min_{j_s \in R_j} \sum_{i \in M_\mathbf{r}} \big(E_\ominus(i_\mathbf{r}, j_s) - E_\ominus(i_\mathbf{t}, j_s) \big)$ for $j \in N$, and thus $O(nr)$ time to evaluate $\sum_{j \in N} \min_{j_s \in R_j} \sum_{i \in M_\mathbf{r}} \big(E_\ominus(i_\mathbf{r}, j_s) - E_\ominus(i_\mathbf{t}, j_s) \big)$. Adding these together, the cost of evaluating Eq. (3) is $O(nr + zr + 1) = O(nr)$, and so the cost of trying to prune one m-tuple using another is $O(nr + zr^2 + z^2r)$. Thus the overall cost of trying to prune with all $O(z^m r^{2m})$ pairs of m-tuples is $O\big(z^m r^{2m+1}(n + z^2 + zr)\big)$. For any $m > 1$, this dominates the time required to prune single RCs even with the $K(i_r, i_t)$ precomputation, so the cost of pruning m-tuples is still $O\big(z^m r^{2m+1}(n + z^2 + zr)\big)$ even if the precomputation cost is added. □

B Implementation of DEEPer: Details

B.1 Perturbation implementation and selection details

To flip a proline ring's pucker, the C_γ atom is moved, holding all bond lengths as well as the C_α-C_β-C_γ angle constant. This limits the C_γ position to at most two positions; if there is a position other than the current one (as there essentially always is for a valid conformation), then the flip consists of moving the C_γ to that position. This is similar to the method of Ho et al. [69]. Then the hydrogens are placed to induce two-fold rotational symmetry at each of the four tetrahedral carbons in the sidechain, using the hydrogen-carbon-hydrogen angles from Allen et al. [70]. Unlike other perturbations, this does affect the sidechain dihedrals, but these dihedrals are not continuously-flexible and are not changed by minDEE or by the normal sidechain-dihedral adjustments that

DEEPer inherits from minDEE. This is because the dihedrals cannot be changed without affecting the bond lengths and angles.

After perturbations are applied, the sidechain is moved as a rigid body to put the C_β in its ideal position (as in the sidechain idealization feature [71] in KiNG [72]) and to return the χ_1 dihedral to its value before the perturbation. The latter step is to make sure perturbations commute with sidechain dihedral adjustments and is not applied for proline, so proline flips are not reversed at this stage. Proline is, however, further idealized by imposing an ideal C_δ-N-C_α angle and then placing the C_γ and the hydrogens in the same manner as for a proline flip, except choosing the C_γ position that does not flip the ring pucker. Backbone conformations that do not allow any valid proline ring conformation are assigned infinite energy, preventing them from being considered as GMEC candidates or low-energy ensemble members. In light of these modifications, the DEEPer version of OSPREY now supports mutations to and from proline.

When computing energy lower bounds and final minimized conformations, continuous perturbations are minimized along with sidechain dihedrals using the steepest descent-based local minimization algorithm currently implemented in OSPREY.

We have implemented an automatic perturbation selection module for OSPREY, which generates shears, backrubs, loop closure adjustments, secondary structure adjustments, and proline flips. The set of flexible residues is taken as input, as in iMinDEE. For any set of consecutive flexible residues with suitable secondary structure and the correct number of residues for a given perturbation type, one perturbation of that type is generated. Perturbed backbone conformations are run through a Ramachandran filter with a used-specified cutoff. Ramachandran data from [71] were used, with glycine, proline, and pre-proline residues all treated as special cases. If specified by the user, perturbed backbone conformations are also run through a backbone RMSD filter. The latter rejects a perturbed backbone conformation (and its associated RCs) if it is too similar to another backbone conformation that is being considered, as measured by backbone heavy-atom RMSD. The RMSD filter can widen the diversity of sampled backbone conformations per unit computational cost. Parameter intervals for shears and backrubs can be specified by the user. Proline flips

are placed at any residues where prolines are allowed. This automatic perturbation selector can be applied straightforwardly to any protein system, though sometimes better results might be obtained by manual adjustment of perturbations, particularly when experimental data on alternate backbone conformations is available. For example, crystallographic alternates can be used for this purpose. Another possibility is to use crystal structures of different conformations of the same or similar proteins, such as the different complementarity-determining region loop structures found for antibodies [73]. The automatic perturbation selector can incorporate manually selected perturbations and then select additional perturbations automatically.

B.2 Choice of pruning zones

Making the pruning zone Z be the entire system will ensure that all pruned pairs are contained within Z (i.e. for each pruned pair (i_r, j_s), $i, j \in Z$), which may result in additional pruning relative to previous methods. The time and space cost of indirect pruning for Z scales as z but is expected to be much smaller than that of A* regardless. However, indirect pruning becomes less powerful than Goldstein DEE [40] when few of the RC pairs in the pruning zone are pruned. To avoid this problem, one may construct a much smaller, "minimal" pruning zone by the following protocol:

1. Select a perturbation and put all the residues it affects in Z.

2. While there is a perturbation that affect at least one residue in Z and at least one in N, add all the residues it affects to Z.

If Z is a *minimal pruning zone*, then there will be no parametrically incompatible pairs (i_r, j_s) for $i \in Z$ but $j \in N$, which would lead to a lack of pruning. But there is no proper subset of Z with this property, so removing more residues from Z would be imprudent. In other words, using a minimal pruning zone will take advantage of all the parametrically incompatible pairs resulting from the perturbations. Also, it will make sure that no RCs are impossible to prune just because they have different parameter values than other RCs at the same residue. But it will put no more

residues in Z than necessary to satisfy those two conditions.

Optimal pruning can be obtained by using multiple rounds of indirect pruning, one with the entire system as the pruning zone and one for each possible minimal pruning zone (there are at most as many as minimal pruning zones as there are perturbations); these can be applied in sequence during each pruning cycle along with all the other DEE methods in our lab's OSPREY software [50, 55] (Goldstein singles, pairs, etc.). The cycle is repeated until it does not prune any RCs. Thus, the number of cycles cannot be greater than the total number of RCs at all residues, but in practice it will tend to be far smaller than that. Using both the entire protein as a pruning zone and the minimal pruning zones achieves a balance between the benefits of large and small pruning zones. Because the pruning is provable for any sequence of pruning zones, more could be included, though the amount of additional pruning will likely be much diminished. In fact the first pruning zone used, which in the current implementation is the entire protein, tends to give most or all of the pruning in test runs. Multiple iterations with that first pruning zone are usually needed for pruning to complete, though, just as for other forms of DEE.