

Protein Fragment Swapping: A Method for Asymmetric, Selective Site-directed Recombination

Wei Zheng¹, Karl E. Griswold², and Chris Bailey-Kellogg¹

¹ Department of Computer Science, Dartmouth College
6211 Sudikoff Laboratory, Hanover NH 03755, USA

`wei.zheng@dartmouth.edu`, `cbk@cs.dartmouth.edu`

² Thayer School of Engineering, Dartmouth College
8000 Cummings Hall, Hanover, NH 03755, USA

`karl.e.griswold@dartmouth.edu`

Abstract. This paper presents a new approach to site-directed recombination, swapping combinations of selected discontinuous fragments from a source protein in place of corresponding fragments of a target protein. By being both asymmetric (differentiating source and target) and selective (swapping discontinuous fragments), our method focuses experimental effort on a more restricted portion of sequence space, constructing hybrids that are more likely to have the properties that are the objective of the experiment. Furthermore, since the source and target need to be structurally homologous only locally (rather than overall), our method supports swapping fragments from functionally important regions of a source into a target “scaffold”; e.g., to humanize an exogenous therapeutic protein. A protein fragment swapping plan is defined by the residue position boundaries of the fragments to be swapped; it is assessed by an average potential score over the resulting hybrid library, with singleton and pairwise terms evaluating the importance and fit of the swapped residues. While we prove that it is NP-hard to choose an optimal set of fragments under such a potential score, we develop an integer programming approach, which we call SWAGMER, that works very well in practice. We demonstrate the effectiveness of our method in two types of swapping problem: selective recombination between beta-lactamases and activity swapping between glutathione transferases. We show that the selective recombination approach generates a better plan (in terms of resulting potential score) than a traditional site-directed recombination approach. We also show that in both cases the optimized experiment is significantly better than one that would result from stochastic methods.

1 Introduction

Protein recombination constructs libraries of hybrids by recombining fragments from two or more parents, with the goal of discovering hybrids with beneficial properties such as improved thermostability, activity, or substrate specificity [1–13]. For example, Stemmer demonstrated the development of beta-lactamase hybrids with a 32,000-fold increase in the required minimum inhibitory concentration of the antibiotic cefotaxime [1]. In contrast with mutagenesis techniques, recombination uses amino acid

combinations that already exist in wild-type proteins and thus are likely to produce viable proteins. Site-directed techniques seek to improve the “hit rate” of good hybrids by recombining the parents at selected breakpoint positions, rather than stochastically. For example, Arnold, Mayo, and co-workers showed that selecting breakpoints so as to minimize disruption of interacting amino acid pairs yields beta-lactamase hybrids that are more likely to be stably folded and functional than random ones [4].

Typically site-directed recombination is both exhaustive and symmetric: a combinatorial library of hybrids is constructed from fragments covering all residues (exhaustive) and taken uniformly from all parents (symmetric). However, in many applications it may be desirable to relax these requirements. A *selective* approach may be warranted if the parents have regions that are significantly “gappy” (insertions/deletions) in a sequence alignment or that are significantly different structurally. In such a case much experimental effort may be wasted on constructing and screening a large number of poor quality hybrids, instead of focusing on those that recombine the non-gappy and structurally analogous regions, and thus are more likely to be stably folded and functional. An *asymmetric* approach may be in order if the goal is to swap portions of particular functional importance from one protein into another. One such application is humanization of exogenous therapeutic proteins, where part of a foreign source is swapped into a human protein target that acts as a scaffold (and will not elicit an immune response). Antibodies have long been humanized this way, e.g., combining murine variable regions with human constant regions [14, 15]. An approach for the much more difficult task of humanizing enzymes (which lack the overtly modular nature of antibodies) was recently demonstrated [10, 11], introducing activity from a rat glutathione transferase into a human one.

In order to enable the optimization of asymmetric, selective site-directed recombination experiments, we develop here a new approach that we call *protein fragment swapping* (Fig. 1). We distinguish a *source parent* and a *target parent*, and construct a library that swaps combinations of selected discontinuous fragments *from* the source *to* the target. By swapping from source to target, our approach is asymmetric; by swapping fragments that can be discontinuous, our approach is selective. Furthermore, fragment swapping does not require the parents to be homologous (in sequence or structure) overall, but only requires there to be corresponding regions of the source and target in which we may swap fragments. Thus it directly supports the humanization application discussed in the previous paragraph. Traditional combinatorial site-directed library construction is a special case of fragment swapping, where the swapped fragments must be contiguous. By enabling the protein engineer to define sequence regions of interest, swapping focuses the experimental effort on a smaller portion of sequence space that is believed to be more relevant. Thus it improves the chance of finding beneficial new hybrids in the resulting library.

We develop an algorithm, which we call “SWAGMER” (a word created by swapping part of “frAGMEnt” into “swAPPEr”) for planning protein fragment swapping experiments. The objective is to select, from the corresponding source and target sequence regions of interest, “good” source fragments to be combinatorially swapped in for corresponding target fragments. To assess possible plans, we employ a statistical potential score analogous to those used in combinatorial recombination to help ensure stability of

the resulting hybrids [4, 16–18]. The potential averages over the entire resulting hybrid library a set of singleton and pairwise terms evaluating the importance of the residues and how well they match. While the averages can be computed efficiently (i.e., without enumerating the exponential number of hybrids), we show that the inclusion of pairwise terms leads to an NP-hard optimization problem. To solve the problem in practice, we develop an integer programming approach that represents swapping assignments for the residues by binary variables, and optimizes the potential score for the resulting library. To demonstrate the effectiveness of our approach, we planned experiments for selective recombination between beta-lactamases and for activity swapping between glutathione transferases. In both cases, the optimized plans outperform all randomly-generated plans (as would result from stochastic recombination methods). We further compared the selective plan with an optimized traditional site-directed recombination plan, and show that the swapping library has a better average potential score, increasing the probability of obtaining functional variants.

2 Methods

There are three main steps (Fig. 1) to planning a fragment swapping experiment for a given source and target. We assume here that we are given a single source protein S of length m and target protein T of length n ; the approach can readily be generalized to multiple proteins.

1. Identify a set of *swappable regions*, $R = \{(s_1, t_1, \ell_1), (s_2, t_2, \ell_2), \dots\}$, where $s_i \in [1..m]$ and $t_i \in [1..n]$ are the starting residue positions in the source and target respectively, and ℓ_i is the length of the swappable region such that $s_i + \ell_i \leq s_{i+1}$ and $t_i + \ell_i \leq t_{i+1}$. Thus a swappable region defines corresponding residue substrings; regions can be separated by gaps of different lengths in the two proteins.
2. Define a *potential* ϕ to evaluate a possible swapping plan. We compute the average score over the hybrids in the library, employing position-specific singleton terms $g_i(a)$ and pairwise terms $g_{i,j}(a, b)$ to assess the importance and fit of the swapped residue a at position i and residue pairs a, b at positions i, j with respect to amino acid statistics for related sequences.
3. Select from the swappable regions a set of *fragments* for swapping, $F = \{(r_1, a_1, b_1), (r_2, a_2, b_2), \dots\}$, such that for all i , we have $r_i \in [1..|R|]$; $a_i, b_i \in [1..\ell_{r_i}]$; $l_{min} \leq b_i - a_i + 1 \leq l_{max}$; and for $j > i$, if $r_i = r_j$ then $b_i < a_j$. The minimum and maximum fragment length constraints, l_{min} and l_{max} , control the number of residues participating in the swapping.

Our goal is to optimize the selected fragments:

Fragment swapping problem. *Given swappable regions R and potential score ϕ , find within R a set F of λ fragments maximizing ϕ .*

Modified versions of existing site-directed recombination techniques may be employed to construct the swapping library defined by a set of fragments (step 4 in Fig. 1). We propose to use SPLISO [19] and RoboMix [20], hierarchically assembling hybrids

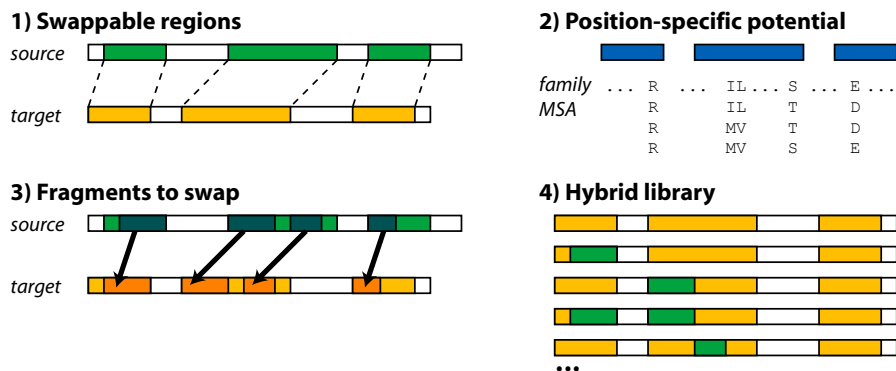


Fig. 1. Overview of fragment swapping method. (1) Identify swappable regions (colored), indicating corresponding portions of the source and target proteins between which fragments may be swapped. The regions may cover most or all of the sequences, or they may be discontinuous. (2) Define a potential score to assess the library resulting from a possible swapping. The example illustrates conservation (R in the first region) and covariation (IL/MV in the second region and SE/TD across the second and third regions) within one of the families (source or target). Hybrids in a possible library can be evaluated for satisfaction of these conservation and covariation constraints. (3) Select fragments (darker colors) within the swappable regions to be swapped from the source into the target, so as to optimize the library potential. (4) Construct a hybrid library by swapping all combinations of the source fragments into the target, replacing its corresponding fragments.

by ligating fragments with short (e.g., 3-nucleotide) overhangs common to both parents, and robotically ensuring that only the desired asymmetric combinations are constructed (swapping source into target but not vice-versa).

2.1 Swappable regions

In combinatorial site-directed protein recombination, parent proteins are typically selected from the same family [4, 16, 21]. It is assumed by homology that the same overall structure is common to the parents and the resulting hybrids. On a more local level, the assumption is that corresponding amino acids in an alignment of the parents are in similar local structural environments, so that the residues in the resulting hybrids will likewise be in favorable environments. The common structural context among parents and hybrids is thereby “factored out” of the planning.

In fragment swapping, we no longer require the source and target to be related or to be similar overall. However, we would still like to ensure that the hybrids maintain the overall structure of the target, and that the swapped source fragments are likely to be placed in suitable local environments in the target. This allows us to focus our optimization efforts on the specific amino acid content (the potential score in the next section). Thus we start with a set of *swappable regions*, pairs of corresponding substrings from the source and target (the first step of Fig. 1).

In cases of sufficient homology, sequence alignment suffices to determine swappable regions. We eliminate the “gappy” parts of the alignment (insertions/deletions) and use the remaining contiguous portions as swappable regions. When structures are available for both source and target, and the structures are similar enough, swappable regions can be found by standard topological structural alignment techniques [22–24]. We keep the portions that structurally align well and eliminate insertions/deletions and portions with poor structural correspondence. In the most challenging cases, global structural alignment yields poor correspondence, but some local regions align well and may serve as swappable regions. Methods for establishing such local structural alignments are beyond the scope of the present work, but may be based on geometric hashing [25] or extension of aligned fragment pairs [23, 24].

For the purposes of planning, we only consider the residues within the swappable regions (the inter-region residues from the target are of course included in library construction). Thus we can re-index the two protein sequences with indices from 1 to $\ell = \sum_i \ell_i$ covering the swappable regions, where the ℓ_i are the lengths of the swappable regions as previously defined. We employ this indexing in the remainder, and use brackets to get residues in S and T ; e.g., $S[3]$ is the third swappable-region residue in S .

2.2 Potential score

Swapping can be seen as making clusters of simultaneous mutations, and our goal is to choose sets of mutations that are in some sense optimal, in that they transfer the desired function without disrupting the current scaffold. As in previous work [4, 17, 8, 18, 26–30], we assume that constraints on amino acid choices required to maintain structure and function are revealed in the sequence record, and devise an objective function seeking to satisfy those constraints. (In fact, related contact potentials have long been the basis for many protein structure prediction techniques [31, 32].) We base the potential function here on the statistical framework from our earlier site-directed recombination work [29], but the planning method can use any potential score of the same form.

We deal here with two types of sequence constraint displayed by a multiply-aligned set of sequences: position-dependent residue conservation and covariation (see again Fig. 1, step 2). For example, if a residue is highly conserved in the source family, it may be important to swap it into the target in order to introduce the desired function. Likewise, if a pair of residues are highly correlated in the source family, it may be necessary to ensure that they are swapped as part of the same fragment, since placing them in different fragments will result in the other combinations less frequently observed in the family. While we do not include in our potential any contribution from residues outside the swappable regions (even by way of pairwise terms with a residue in the swappable regions), the potential can be generalized to do so, or an overall “environment” effect can be incorporated into the singleton terms.

More formally, let us consider conservation and covariation in a multiple sequence alignment \mathcal{S} for the source family. For the singleton terms, we define $s_i(a)$ as the log probability of amino acid type a at residue position i :

$$s_i(a) = \log \frac{|\{P \in \mathcal{S} : P[i] = a\}|}{|\mathcal{S}|} \quad (1)$$

For the pairwise terms, we only consider residue pairs i and j that are in contact in a representative structure for the protein family (assumed common to all, by homology), as contacting pairs have the greatest direct impact on establishing a suitable local environment. We define $s_{i,j}(a, b)$ as the log probability of the pair of amino acid types a and b , vs. what would be expected if they were independent:

$$s_{i,j}(a, b) = \log \frac{|\{P \in \mathcal{S} : P[i] = a \wedge P[j] = b\}|}{|\mathcal{S}|} - s_i(a) - s_j(b) \quad (2)$$

By subtracting the independent terms from the joint term, $s_{i,j}$ contains only the additional information regarding the correlation between the two positions, and we can correctly compute a total score by summing up all the singleton and pairwise terms without “double-counting” the singleton contributions.

We can likewise compute $t_i(a)$ and $t_{i,j}(a, b)$ for the target, based on a multiple sequence alignment and representative structure. We then define the overall constraint on a position or pair of positions as a convex combination of these terms:

$$g_i(a) = \alpha \times s_i(a) + (1 - \alpha) \times t_i(a) \quad (3)$$

$$g_{i,j}(a, b) = \alpha \times s_{i,j}(a, b) + (1 - \alpha) \times t_{i,j}(a, b) \quad (4)$$

The choice of α reflects whether it is more important for the hybrids to satisfy the source constraints (α near 0), the target constraints (α near 1), or both (α in between). Note that the formula readily handles the special case where the source and target are from the same family. Other means of combining the potential are possible; however, we find this one to be both powerful and easy to interpret.

Given a hybrid with sequence P , we can evaluate how well it satisfies the conservation constraints as:

$$\phi(P) = \sum_i g_i(P[i]) + \sum_{i,j} g_{i,j}(P[i], P[j]). \quad (5)$$

Since we would like all hybrids to satisfy the constraints, we evaluate a possible library in terms of the sum of the hybrid scores according to Eq. 5, seeking to maximize the total. More precisely, if we have selected λ fragments, then 2^λ hybrids P_h ($1 \leq h \leq 2^\lambda$) are created, and we seek to maximize:

$$\begin{aligned} \sum_h \phi(P_h) &= \sum_h \left(\sum_i g_i(P_h[i]) + \sum_{i,j} g_{i,j}(P_h[i], P_h[j]) \right) \\ &= \sum_i \sum_h g_i(P_h[i]) + \sum_{i,j} \sum_h g_{i,j}(P_h[i], P_h[j]). \end{aligned} \quad (6)$$

Let us define $\phi_1(i)$ as the average over all hybrids of $g_i(P_h[i])$, and $\phi_2(i, j)$ as the average of $g_{i,j}(P_h[i], P_h[j])$. Then we can rewrite the total potential as

$$\sum_h \phi(P_h) = 2^\lambda \times \left(\sum_i \phi_1(i) + \sum_{i,j} \phi_2(i, j) \right). \quad (7)$$

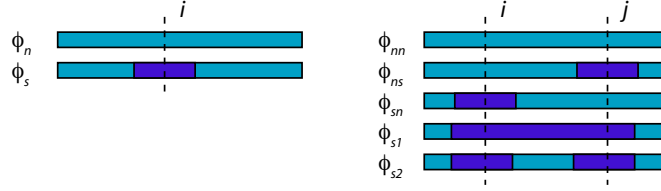


Fig. 2. Swapping patterns for (left) single residue i or (right) pair of residues i, j . Fragments being swapped are shaded darker.

To evaluate the total potential in an efficient planning algorithm we cannot afford to enumerate the exponential number of hybrids in each possible fragment swapping. Fortunately, given the definition of a fragment swapping we can compute average potentials ϕ_1 and ϕ_2 for a given position or pair of positions in constant time, and thus the overall average potential ϕ in at most quadratic time (though in practice the number of pairwise terms is likely to be linear, due to the contact restriction). The key insight is that each residue or residue pair participates in a well-defined pattern of hybrids, depending on the selection of fragments to be swapped. That is, the “projection” of the hybrid library onto a single column or pair of columns can be partitioned into a few cases, each with the same number of hybrids in the overall library as in Fig. 2, and we simply need to average over the cases.

For $\phi_1(i)$ there are two possibilities, depending on whether or not residue i is swapped (Fig. 2, left).

$$\phi_n(i) = g_i(T[i]) \quad (8)$$

$$\phi_s(i) = 1/2 \times (g_i(S[i]) + g_i(T[i])) \quad (9)$$

When residue i is not being swapped (ϕ_n), all the hybrids have the target residue; when it is (ϕ_s), half the hybrids have the source residue and the other half have the target residue.

For $\phi_2(i, j)$ there are five cases (Fig. 2, right): neither i nor j is swapped (ϕ_{nn}), only i is swapped (ϕ_{sn}), only j is swapped (ϕ_{ns}), both are swapped in the same fragment (ϕ_{s1}), or both are swapped in different fragments (ϕ_{s2}).

$$\phi_{nn}(i, j) = g_{i,j}(T[i], T[j]) \quad (10)$$

$$\phi_{sn}(i, j) = 1/2 \times (g_{i,j}(S[i], T[j]) + g_{i,j}(T[i], T[j])) \quad (11)$$

$$\phi_{ns}(i, j) = 1/2 \times (g_{i,j}(T[i], T[j]) + g_{i,j}(T[i], S[j])) \quad (12)$$

$$\phi_{s1}(i, j) = 1/2 \times (g_{i,j}(S[i], S[j]) + g_{i,j}(T[i], T[j])) \quad (13)$$

$$\phi_{s2}(i, j) = 1/4 \times (g_{i,j}(S[i], S[j]) + g_{i,j}(T[i], T[j]) + g_{i,j}(T[i], S[j]) + g_{i,j}(S[i], T[j])) \quad (14)$$

2.3 Fragment selection

Recall that our goal is to select a set of fragments from the swappable regions, so that the average potential score over the resulting hybrid library is maximized. Unfortunately,

we have proved that this optimization problem is NP-hard when using a potential score with pairwise terms. The detailed proof is in an appendix for the interested reader.

Claim. The fragment swapping problem is NP-hard.

Proof sketch. The proof is by reduction from MAX-2SAT. Literals in a 2-CNF formula map to residues in a swapping problem, with a correspondence between a literal being true and a residue being in a swapping fragment. Pairwise swapping potential terms are defined so that maximizing the swapping score results in satisfying each clause and consistently treating (swapping or not) all literals using each variable. \square

Computationally, the fragment swapping problem is somewhat analogous to the threading (sequence-structure alignment) problem, in which secondary structure “fragments” from a template “source” are aligned to the primary sequence for a target, according to a potential score that typically includes both singleton (environment) and pairwise (contact) terms [33–36]. (Like swapping, threading is also NP-hard [37].) The most important difference is that in threading, we know the lengths of the fragments (secondary structure elements) that must be aligned, whereas in fragment swapping, that is part of the optimization. We make use of the analogy in developing an integer programming approach to the fragment swapping problem, since RAPTOR [38] is a very successful threader based on an integer programming formulation. While drawing inspiration from that work, our formulation must employ different variables (since the fragments have unknown lengths), different constraints (to maintain a valid fragment swapping), and of course a different objective function.

In a swapping of λ fragments, conceptually the source and the target (those residues in swappable regions) are partitioned into a total of $2\lambda + 1$ fragments, alternating between $\lambda + 1$ non-swapping fragments and λ swapping fragments. The length of any non-swapping fragment can be 0, yielding adjacent swapping fragments or ensuring that the first or last fragment is swapping rather than non-swapping. We index the fragments from 1 to $2\lambda + 1$, with odd numbers for non-swapping fragments and even numbers for swapping fragments. Let $B = \{\ell_1, \ell_1 + \ell_2, \dots, \sum_{i=1}^{|R|-1} \ell_i\}$ be the indices defining the $|R| - 1$ internal boundaries between the regions (again, using residue indexing for swappable regions, as discussed above). We ensure that no fragment crosses an index in B .

The potential score contributions are determined by the fragments to which single residues belong and the fragment pairs to which residue pairs belong. Thus in order to develop an integer programming approach, we define singleton and pairwise binary variables, $s_{i,f}$ and $p_{i,j,f,g}$, representing the assignment of residues and residue pairs to fragments.

$$s_{i,f} = \begin{cases} 1 & \text{if residue } i \text{ is in fragment } f \\ 0 & \text{otherwise,} \end{cases} \quad (15)$$

$$p_{i,j,f,g} = \begin{cases} 1 & \text{if residue } i \text{ is in fragment } f \text{ and residue } j \text{ is in fragment } g, \\ 0 & \text{otherwise,} \end{cases} \quad (16)$$

where $1 \leq i, j \leq \ell$ and $1 \leq f, g \leq 2\lambda + 1$. If f is even and $s_{i,f} = 1$, then residue i is in the $(f/2)$ th swapping fragment; otherwise it is in a non-swapping fragment. For

efficiency, we only define $p_{i,j,f,g}$ if there is a contact between i and j (so there is a non-zero potential score), and if $i < j$ and $f \leq g$ (to avoid redundancy).

With this representation, our objective function, to optimize the average potential, can be written as:

$$\begin{aligned} \Phi = & \sum_i \sum_{\text{even } f} s_{i,f} \times \phi_s(i) + \sum_i \sum_{\text{odd } f} s_{i,f} \times \phi_n(i) + \sum_{i,j} \sum_{\text{odd } f,g} p_{i,j,f,g} \times \phi_{nn}(i,j) \\ & + \sum_{i,j} \sum_{\text{odd } f, \text{even } g} p_{i,j,f,g} \times \phi_{ns}(i,j) + \sum_{i,j} \sum_{\text{even } f, \text{odd } g} p_{i,j,f,g} \times \phi_{sn}(i,j) \\ & + \sum_{i,j} \sum_{\text{even } f} p_{i,j,f,f} \times \phi_{s1}(i,j) + \sum_{i,j} \sum_{\text{even } f,g; f \neq g} p_{i,j,f,g} \times \phi_{s2}(i,j). \end{aligned} \quad (17)$$

To guarantee the variable assignments yield a valid fragment swapping, we have constraints:

$$\forall i : \sum_f s_{i,f} = 1, \quad (18)$$

$$\forall i, f : s_{i,f} + \sum_{f' < f} s_{i+1,f'} \leq 1, \quad (19)$$

$$\forall \text{even } f : \sum_i s_{i,f} \geq l_{min}, \quad (20)$$

$$\forall \text{even } f : \sum_i s_{i,f} \leq l_{max}, \quad (21)$$

$$\forall i, j, f : \sum_{g \geq f} p_{i,j,f,g} = s_{i,f}, \quad (22)$$

$$\forall i, j, g : \sum_{f \leq g} p_{i,j,f,g} = s_{j,g}, \quad (23)$$

$$\forall \text{even } f \forall i \in B : s_{i,f} + s_{i+1,f} \leq 1. \quad (24)$$

Eq. 18 guarantees a residue can participate in only one fragment. Eq. 19 maintains the sequential order of residues and fragments. Eq. 20 and Eq. 21 enforce the minimum and maximum fragment length constraints. Eq. 22 and Eq. 23 ensure consistent single and pairwise assignments; see the claim below. Eq. 24 guarantees that no swapping fragment crosses the boundary of a swappable region.

Claim. For $i < j$ and $f \leq g$, Eq. 18, Eq. 22 and Eq. 23 guarantee that $p_{i,j,f,g}$ is 1 if and only if $s_{i,f} = 1$ and $s_{j,g} = 1$.

Proof. Assume $p_{i,j,f,g}$ has value 1. By Eq. 22, we have $s_{i,f} \geq 1$. Then by Eq. 18, $s_{i,f}$ must have value 1. Similarly, by Eq. 23 and Eq. 18, we get $s_{j,g} = 1$.

If $s_{i,f} = 1$ and $s_{j,g} = 1$, then Eq. 22 guarantees that there is a $g' \geq f$ such that $p_{i,j,f,g'} = 1$. It must be the case that $g' = g$, because otherwise we would have $s_{j,g'} = 0$ by Eq. 18, since $s_{j,g} = 1$. Then we would have $\sum_{f' \leq g'} p_{i,j,f',g'} = s_{j,g'} = 0$ by Eq. 23, contradicting $p_{i,j,f,g'} = 1$. \square

Claim. Any fragment swapping is a solution to our integer program, and any solution to our integer program defines a fragment swapping.

Proof. The first part is straightforward. In a fragment swapping, a residue is in only one fragment as in Eq. 18. Residue i must be in a fragment with index no larger than the one of residue $i + 1$, satisfying Eq. 19. The length of each swapping fragment is between l_{min} and l_{max} , satisfying Eq. 20 and Eq. 21. By the definition of Eq. 16, the value of $p_{i,j,f,g}$ satisfies Eq. 22 and Eq. 23. Finally, a fragment does not cross swappable region boundaries, so Eq. 24 is satisfied.

Now assume we have a solution to our integer program, and let us construct a fragment swapping. To do so, we must determine the start and end of each swapping fragment, and ensure that the fragment is of the right size and remains within a swappable region. Let us consider even (swapping) fragment number f in the solution. By Eq. 20 and Eq. 21 we have $l_{min} \leq \sum_i s_{i,f} \leq l_{max}$, so f is of the right size. By Eq. 24, its residues do not cross a swappable region boundary. In order to obtain the start and end of f , we must ensure that its residues (i.e., the variables i with $s_{i,f} = 1$) are consecutive. Assume they aren't. Then there are two residues i, j , with $i + 1 < j$, such that $s_{i,f} = 1$, $s_{i+1,f} = 0$ and $s_{j,f} = 1$. Considering residue i , by Eq. 18 and Eq. 19, there is fragment e , with $f < e$, such that $s_{i+1,e} = 1$. Then there must be a residue k , with $i + 1 \leq k < j$, such that k is in a fragment with larger index than that of $k + 1$, since otherwise residue $i + 1$ could not be in a fragment with a larger index than that of residue j . But such a k would contradict Eq. 19. Thus the residues of f must be consecutive, and we can determine the start and end of f by finding the minimum and maximum i with $s_{i,f} = 1$. \square

Thus by maximizing the objective function, we will find the optimal selection of swapping fragments.

As mentioned in the introduction, traditional site-directed recombination between two proteins in a single family is a special case of fragment swapping. We arbitrarily call one parent protein the source and the other one the target. After aligning the sequences by standard techniques, we have a single swappable region of length n including all residues. We add the constraint $\sum_{i, \text{even } f} s_{i,f} = n$, and Eq. 21 is no longer needed. Then the asymmetric swapping will result in a symmetric combinatorial recombination.

3 Results and Discussion

To study the effectiveness of SWAGMER, we applied it to two different types of fragment swapping experiments. First we analyzed, using beta-lactamases, the difference between selective swapping and traditional site-directed recombination. Next we turned to activity swapping for enzyme humanization, using glutathione transferases, and explored planning swaps from rat source to human target.

3.1 Selective swapping of beta-lactamases

Beta-lactamases are enzymes produced by some bacteria; they hydrolyze the beta-lactam found in certain antibiotics (e.g., penicillin). They have been the object of much

chimeragenesis work, including the pioneering site-directed studies of Arnold and colleagues [4, 16]. We have also previously developed experiment planning methods for traditional site-directed recombination and applied them to beta-lactamases [29]. We use here the dataset from our previous study, consisting of 136 beta-lactamases multiply aligned to 263 residues with an average sequence identity of 41.8%, along with the representative 3D structure from *E. coli* TEM-1 beta-lactamase (pdb id 1BTL). We derived the potential score as discussed above; we note that our previous work demonstrated that the potential is predictive of folded and functional hybrids [29]. We used as parents the proteins studied by Arnold, TEM-1 and PSE-4. Because of their highly similar structures and nearly identical disruption profiles [4], we adopted the same potential score for TEM-1 and PSE-4 in fragment selection, arbitrarily choosing TEM-1 as source. Here the objective is to explore selective site-directed recombination compared with traditional recombination covering all residues.

We compared the libraries optimized by SWAGMER to randomly generated plans and to plans optimized by our earlier method [29] for traditional site-directed recombination. We call the traditional approach “exhaustive” as the experiment is defined by selecting breakpoint positions at columns in a multiple sequence alignment, so that recombination necessarily covers the entire sequence rather than focusing on specific fragments. Based on the number of residues in TEM-1 and PSE-4, we set the fragment length constraints to be a minimum of 10 and a maximum of 50. We generated plans with 2, 3, or 4 fragments (yielding a manageable sized library) by SWAGMER and the random approach, and plans with the same number of hybrids by the exhaustive approach (2 swapped fragments corresponds to 1 exhaustive breakpoint, etc.). For the random approach, we generated 10^5 random plans, requiring more than 1 hour for 2 fragments and roughly 2 hours each for 3 and 4 fragments. We implemented SWAGMER using the CBC integer programming solver provided in COIN-OR (<https://projects.coin-or.org/Cbc>). The running times were 32 seconds (2 fragments), 776 seconds (3 fragments), and 3359 seconds (4 fragments).

The top three panels in Fig. 3 summarize the qualities of the resulting plans, in terms of average potential scores. Clearly the optimal plan is much better than would be obtained at random, as would be obtained by stochastic recombination rather than a planned approach. By focusing experimental effort on selected fragments, rather than spreading it out over the entire protein, SWAGMER also significantly outperforms the exhaustive approach. Thus the resulting library better explores this region of sequence space, giving us the opportunity to find hybrids that probably would not be generated under other methods.

Fig. 4(left) illustrates the structure of the swapping plan. It employs minimum-length fragments, perhaps because PSE-4 and TEM-1 are distantly related [8] and thus short fragments are preferable to minimize the disruption introduced by swapping. The plans for the exhaustive approach likewise place breakpoints so as to minimize fragment length (breakpoints are stacked up at either the N- or C-terminus). The swapping fragments selected are all within protein modules identified by profile disruption [8], which it is hypothesized must be maintained in recombination to yield folded and functional hybrids.

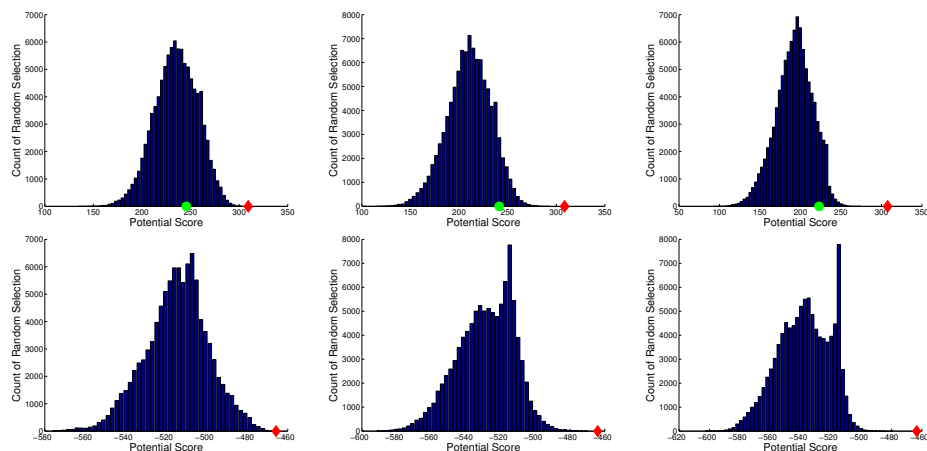


Fig. 3. Average potential scores for generated libraries: (top) beta-lactamases, (bottom) glutathione transferases; (left) 2, (middle) 3, and (right) 4 fragments. The histogram is taken over 10^5 random libraries. The red diamond is the SWAGMER-optimized library. The green circle in the beta-lactamase panels is the optimal library for the “exhaustive” approach.

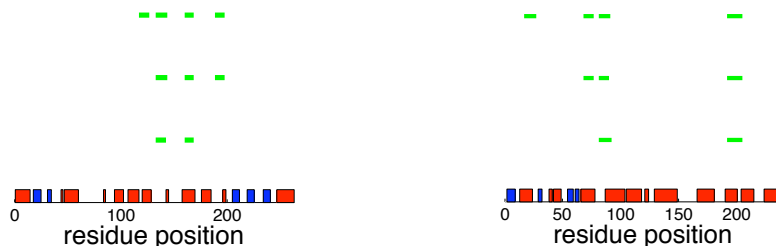


Fig. 4. Swapping plans relative to the reference structures: (left) beta-lactamases, (right) glutathione transferases. Green blocks represent optimal fragment selections for 2, 3, and 4 fragments. Red blocks represent alpha helices and blue blocks represent beta sheets.

3.2 Activity swapping in glutathione transferases

Glutathione transferases (GSTs) are enzymes that help eliminate reactive electrophilic compounds by conjugating them to glutathione. As mentioned in the introduction, Griswold *et al.* recently demonstrated the use of chimeragenesis to swap activity from a rat GST into a human one [10]. They employed stochastic techniques to construct libraries of θ -class GSTs, recombining human GST θ -1-1 (hGSTT1-1) and rat GST θ -2-2 (rGSTT2-2). They identified a hybrid with 83% of the hGSTT1-1 sequence but a swapped-in rat activity. This is a powerful demonstration of the potential for activity swapping, but we show here that optimizing an experiment plan can result in a library with significantly higher average score, while focusing experimental effort on a smaller region in sequence space, thus potentially yielding a much better hit rate.

We started with sequence alignments for the two subclasses (rat and human) of θ -class GSTs, with four sequences each, aligned to 239 residues, with an overall sequence identity of 53%. Given the small number of sequences, we followed our previous sparse data approach [29], augmenting the family statistics with database statistics, thereby introducing an amino acid-specific pseudocount. Since θ -class GSTs have a highly conserved GST 3D fold [10] we used the hGSTT1-1 structure (pdb idb id 2C3N) as the reference structure for both subclasses. We used a weight $\alpha = 0.5$ in Eq. 3 and 4, placing equal importance on maintaining the human scaffold and introducing the rat activity.

The bottom three panels of Fig. 3 show the comparison between SWAGMER-optimized plans and 10^5 random ones, for 2, 3, and 4 fragments. As with beta-lactamases, the average potential of the optimal plans is much better than we would get from stochastic plans. The running times are 5 seconds, 44 seconds, and 1067 seconds for 2, 3, and 4 fragments, respectively. Also as with beta-lactamases, the plans seek small fragments (Fig. 4(right)). This suggests a line for further investigation: balancing maximization of the swapping potential against another metric like diversity, as we have done for traditional exhaustive site-directed recombination [39].

4 Conclusion

We have developed a new general framework for recombination, protein fragment swapping. By swapping only selected discontinuous regions, fragment swapping can focus on functionally important regions in parent sequences, is applicable to parents with heterogeneous structures, and is flexible in the number of residues participating in recombination. Furthermore, the asymmetric role of the source and target parents enables specific construction of libraries seeking to introduce activities from one parent into the other. Our SWAGMER method provides an efficient, effective approach to optimizing fragment swapping experiments.

Acknowledgments. This work was supported in part by an Alfred P. Sloan Fellowship and an NSF CAREER award to CBK (IIS-0444544).

References

1. Stemmer, W.: Rapid evolution of a protein *in vitro* by DNA shuffling. *Nature* **370** (1994) 389–91
2. Ostermeier, M., Shim, J., Benkovic, S.: A combinatorial approach to hybrid enzymes independent of DNA homology. *Nat Biotechnol* **17** (1999) 1205–9
3. Lutz, S., Ostermeier, M., Moore, G., Maranas, C., Benkovic, S.: Creating multiple-crossover DNA libraries independent of sequence identity. *PNAS* **98** (2001) 11248–53
4. Voigt, C., Martinez, C., Wang, Z., Mayo, S., Arnold, F.: Protein building blocks preserved by recombination. *Nat Struct Biol* **9** (2002) 553–8
5. O’Maille, P., Bakhtina, M., Tsai, M.: Structure-based combinatorial protein engineering (SCOPE). *J Mol Biol* **321** (2002) 677–691
6. Aguinaldo, A., Arnold, F.: Staggered extension process (StEP) *in vitro* recombination. *Methods Mol Biol* **231** (2003) 105–10

7. Coco, W.: RACHITT: Gene family shuffling by random chimeragenesis on transient templates. *Methods Mol Biol* **231** (2003) 111–127
8. Otey, C., Silberg, J., Voigt, C., Endelman, J., Bandara, G., Arnold, F.: Functional evolution and structural conservation in chimeric cytochromes P450: calibrating a structure-guided approach. *Chem Biol* **11** (2004) 309–18
9. Castle, L., Siehl, D., Gorton, R., Patten, P., Chen, Y., Bertain, S., Cho, H.J., Duck, N., Wong, J., Liu, D., Lassner, M.: Discovery and directed evolution of a glyphosate tolerance gene. *Science* **304** (2004) 1151–4
10. Griswold, K., Kawarasaki, Y., Ghoneim, N., Benkovic, S., Iverson, B., Georgiou, G.: Evolution of highly active enzymes by homology-independent recombination. *PNAS* **102** (2005) 10082–7
11. Griswold, K., Aiyappan, N., Iverson, B., Georgiou, G.: The evolution of catalytic efficiency and substrate promiscuity in human theta class 1-1 glutathione transferase. *J Mol Biol* **364** (2006) 400–410
12. Taly, V., Urban, P., Truan, G., Pompon, D.: A combinatorial approach to substrate discrimination in the P450 CYP1A subfamily. *Biochim Biophys Acta* **1770** (2006) 446–457
13. Kurtovic, S., Modén, O., Shokeer, A., Mannervik, B.: Structural determinants of glutathione transferases with azathioprine activity identified by DNA shuffling of alpha class members. *J Mol Biol* **375** (2008) 1365–1379
14. Morrison, S., Johnson, M., Herzenberg, L., Oi, V.: Chimeric human antibody molecules: Mouse antigen-binding domains with human constant region domains. *PNAS* **81** (1984) 6851–5
15. Jones, P., Dear, P., Foote, J., Neuberger, M., Winter, G.: Replacing the complementarity-determining regions in a human antibody with those from a mouse. *Nature* **321** (1986) 522–525
16. Meyer, M., Silberg, J., Voigt, C., Endelman, J., Mayo, S., Wang, Z., Arnold, F.: Library analysis of SCHEMA-guided protein recombination. *Protein Sci* **12** (2003) 1686–93
17. Moore, G., Maranas, C.: Identifying residue-residue clashes in protein hybrids by using a second-order mean-field approach. *PNAS* **100** (2003) 5091–6
18. Saraf, M., Horswill, A., Benkovic, S., Maranas, C.: Famclash: A method for ranking the activity of engineered enzymes. *PNAS* **12** (2004) 4142–4147
19. Saftalov, L., Smith, P., Friedman, A., Bailey-Kellogg, C.: Site-directed combinatorial construction of chimeric genes: general method for optimizing assembly of gene fragments. *Proteins* **64** (2006) 629–42
20. Avramova, L., Desai, J., Weaver, S., Friedman, A., Bailey-Kellogg, C.: Robotic hierarchical mixing for the production of combinatorial libraries of proteins and small molecules. *J Comb Chem* **10** (2008) 63–68
21. Otey, C., Landwehr, M., Endelman, J., Hiraga, K., Bloom, J., Arnold, F.: Structure-guided recombination creates an artificial family of cytochromes P450. *PLoS Biol* **4** (2006) e112
22. Holm, L., Sander, C.: Protein structure comparison by alignment of distance matrices. *J Mol Biol* **233** (1993) 123–138
23. Shindyalov, J., Bourne, P.: Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng* **11** (1998) 739–747
24. Ye, Y., Godzik, A.: Flexible structure alignment by chaining aligned fragment pairs allowing twists. *Bioinformatics* **Suppl 2** (2003) ii246–55
25. Nussinov, R., Wolfson, H.: Efficient detection of three-dimensional motifs in biological macromolecules by computer vision techniques. *PNAS* **88** (1992) 10495–9
26. Saraf, M., Gupta, A., Maranas, C.: Design of combinatorial protein libraries of optimal size. *Proteins* **60** (2005) 769–77
27. Russ, W., Lowery, D., Mishra, P., Yaffee, M., Ranganathan, R.: Natural-like function in artificial WW domains. *Nature* **437** (2005) 579–583

28. Socolich, M., Lockless, S., Russ, W., Lee, H., Gardner, K., Ranganathan, R.: Evolutionary information for specifying a protein fold. *Nature* **437** (2005) 512–518
29. Ye, X., Friedman, A., Bailey-Kellogg, C.: Hypergraph model of multi-residue interactions in proteins: sequentially-constrained partitioning algorithms for optimization of site-directed protein recombination. *J Comput Biol* **14** (2007) 777–790 Conference version: Proc. RECOMB, 2006, 15-29.
30. Thomas, J., Ramakrishnan, N., Bailey-Kellogg, C.: Graphical models of residue coupling in protein families. *IEEE/ACM Trans Comput Biol Bioinf* **5** (2008) 183–97
31. Tanaka, S., Scheraga, H.: Medium and long range interaction parameters between amino acids for predicting three dimensional structures of proteins. *Macromolecules* **9** (1976) 945–950
32. Miyazawa, S., Jernigan, R.: Estimation of effective interresidue contact energies from protein crystal structures: Quasi-chemical approximation. *Macromolecules* **18** (1985) 531–552
33. Bowie, J., Luthy, R., Eisenberg, D.: A method to identify protein sequences that fold into a known three-dimensional structure. *Science* **253** (1991) 164–170
34. Jones, D., Taylor, W., Thornton, J.: A new approach to protein fold recognition. *Nature* **358** (1992) 86–89
35. Lathrop, R., Smith, T.: Global optimum protein threading with gapped alignment and empirical pair score functions. *J Mol Biol* **255** (1996) 651–665
36. Godzik, A.: Fold recognition methods. *Methods Biochem Anal* **44** (2003) 525–546
37. Lathrop, R.: The protein threading problem with sequence amino acid interaction preferences is NP-complete. *Protein Eng* **7** (1994) 1059–68
38. Xu, J., Li, M., Kim, D., Xu, Y.: RAPTOR: Optimal protein threading by linear programming. *J Bioinf Comp Biol* **1** (2003) 95–117
39. Zheng, W., Friedman, A., Bailey-Kellogg, C.: Algorithms for joint optimization of stability and diversity in planning combinatorial libraries of chimeric proteins. In: Proc RECOMB. (2008) 300–314

A NP-hardness of Protein Fragment Swapping

We prove the NP-hardness of the protein fragment swapping problem by reduction from MAX-2SAT. For simplicity our construction uses a single swappable region, only a pairwise potential score ϕ_2 , and trivial fragment length constraints $l_{min} = 1$ and $l_{max} = \infty$.

Let $C_1 \wedge C_2 \wedge \dots \wedge C_\tau$ be a boolean formula in 2-CNF with τ clauses. Let N_+ be the number of *pairs* of identical literals, and N_- be the number of pairs of complementary literals.

Let us first define the types of residue positions in the source and target proteins.

- **Clause**: for each clause $C_r = (c_{r,1} \vee c_{r,2})$ with literals $c_{r,1}$ and $c_{r,2}$, add two residues $v_{r,1}$ and $v_{r,2}$ sequentially.
- **Separator**: for each pair $v_{r,p}, v_{r',p'} (p, p' \in \{1, 2\})$ of instances of the same literal in clauses r and r' (i.e., $c_{r,p}, c_{r',p'}$ are the same variable or $c_{r,p}, c_{r',p'}$ are both the negation of the same variable), add two “separator” residues $v_{d,1}$ and $v_{d,2}$ sequentially between $v_{r,2}$ and $v_{r+1,1}$. (Multiple pairs of separator residues may be strung in the region between clauses.)
- **Trivial**: add $2\tau + 2N_+$ trivial residues at the end of the sequence.

The mapping between MAX-2SAT and fragment swapping is: $v_{r,s}$ is in a swapping fragment if and only if $c_{r,s}$ is true ($1 \leq r \leq \tau, s \in \{1, 2\}$).

We need not specify the amino acid sequences for the source and targets, as the swapping problem is defined in terms of the potential. To this end, there are four types of residue pairs contributing to the potential, with $g_{i,j}$ values in Tab. 1 yielding ϕ_2 values in Tab. 2 according to Eq. 9–Eq. 14.

- **Clause**, for each $v_{r,1}, v_{r,2}$ corresponding to a clause $C_r = (c_{r,1} \vee c_{r,2})$
- **Identical**, for each $v_{r,p}, v_{r',p'}$ corresponding to identical literals used in clauses r and r'
- **Complementary**, for each $v_{r,p}, v_{r',p'}$ corresponding to complementary literals used in clauses r and r'
- **Separator**, for each pair $v_{d,1}, v_{d,2}$ of separator residues for the same identical literal

Fig. 5 illustrates one construction.

The construction takes polynomial time. We establish $4\tau + 4N_+$ residues: 2τ for the clauses, $2N_+$ separator residues, and $2\tau + 2N_+$ trivial residues. There are $\tau + 2N_+ + N_-$ terms in the potential: τ for the clauses, N_+ for the identical pairs with a corresponding N_+ for the separator pairs, and N_- for the complementary pairs.

Now we prove a correspondence between the MAX-2SAT solution for $C_1 \wedge C_2 \wedge \dots \wedge C_\tau$ and the optimal fragment swapping for the constructed $2\tau + 2N_+$ swapping fragments. We separate the two directions of the proof.

Table 1. Family statistics for different types of residue pairs.

	$g_{i,j}(T[i], T[j])$	$g_{i,j}(S[i], T[j])$	$g_{i,j}(T[i], S[j])$	$g_{i,j}(S[i], S[j])$
clause	0	2	2	0
identical	1	-1	-1	5
complementary	0	2	2	-4
separator	-3	3	3	-3

Table 2. Average potential scores for different types of residue pairs.

	ϕ_{nn}	ϕ_{ns}	ϕ_{sn}	ϕ_{s1}	ϕ_{s2}
clause	0	1	1	0	1
identical	1	0	0	3	1
complementary	0	1	1	-2	0
separator	-3	0	0	-3	0

Claim. If the MAX-2SAT solution satisfies k clauses, then the fragment swapping solution achieves an average potential score of $k + N_+ + N_-$.

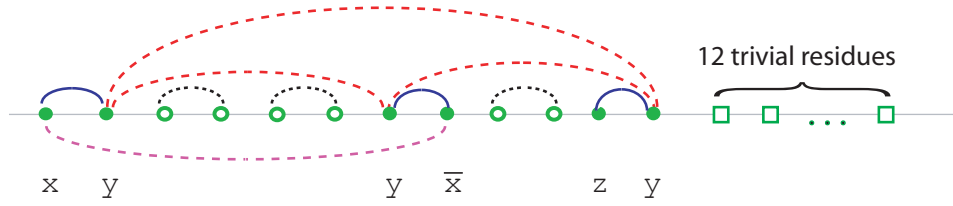


Fig. 5. Residue pairs contributing to the potential ϕ_2 for the MAX-2SAT instance $(x \vee y) \wedge (y \vee \neg x) \wedge (z \vee y)$. Filled dots are residues mapping to literals in the clauses. Empty dots are separator residues. Squares represent trivial residues at the end. There are 3 clause pairs (blue solid), 3 identical pairs (red dashed above), 1 complementary pair (purple dashed below), and 3 separator pairs (black dashed).

Proof. We must show how to select fragments based on the MAX-2SAT solution. For each separator pair, create two single-residue swapping fragments. For each literal, if the literal value is true, create a single-residue fragment for the corresponding residue; otherwise the residue is not in any swapping fragment. Since there are $2\tau + 2N_+$ non-trivial residues, after this step, there are at most $2\tau + 2N_+$ single residue fragments. If the number of fragments is less than $2\tau + 2N_+$, add a sufficient number of single-residue fragments using the trivial residues at the end of the sequence.

Following Tab. 2, we have the following contributions to the potential score:

- clause: 0 for each unsatisfied clause (so that neither residue is in a swapping fragment); 1 for each satisfied clause. Note that ϕ_{s1} cannot happen since we only have single-residue fragments, and each of the remaining possibilities yields 1.
- identical: 1 each. We must have either ϕ_{nn} or ϕ_{s2} , each of which yields 1. ϕ_{s1} cannot happen due to the separator residue pairs between the two identical residues.
- complementary: 1 each. We must have either ϕ_{ns} or ϕ_{sn} , each of which yields 1.
- separator: 0 each

The total is $k + N_+ + N_-$. □

Claim. If the fragment swapping solution achieves an average potential score of $k - N_+ - N_-$, then the MAX-2SAT solution satisfies k clauses.

Proof. We must show how to find an assignment of literals based on the fragment swapping solution. To do this, we show that separator pairs contribute 0 to the potential, while identical and complementary pairs contribute 1 each. Thus there must be $k - N_+ - N_-$ clause pairs contributing 1 each (the only non-zero possibility in Tab. 2). By mapping the swapped residues to literals, we can determine which literals are true and which $k - N_+ - N_-$ clauses are satisfied.

We first prove that each separator pair contributes 0. Assume for contradiction that some separator pair contributes -3 (the only other possibility in Tab. 2). Let us modify the swapping by making two single-residue fragments for the two separator residues, increasing the potential score by 3. If this increases the total number of swapping fragments above $2\tau + 2N_+$, then there must be some swapping fragments in the trivial

residues (since there are only $2\tau + 2N_+$ residues in the main sequence), some of which we can eliminate to leave the total at $2\tau + 2N_+$. The change does not affect the potential contributed by any clause pair. An identical pair can be affected if it involves the same literal as the separator pair and the two residues were previously in the same swapping fragment as the separator residue pair. In that case, the change replaces a single swapping fragment with separate swapping fragments, decreasing the potential by $\phi_{s1} - \phi_{s2} = 2$, which is outweighed by the increase of 3. (If multiple identical pairs are affected, then they are balanced by a corresponding number of separator pairs.) The possible analogous effect on a complementary pair can only be beneficial, yielding a net increase of $\phi_{s2} - \phi_{s1} = 2$. Thus we have increased the total potential, contradicting our assumption that k is the maximum.

Now let us show that identical and complementary pairs contribute 1 each. They must contribute either 0 or 1, since by the above the separator pairs contribute 0 and thus must break up any swapping fragment, eliminating the ϕ_{s1} possibility in Tab. 2. If any pair contributes 0, we can modify the swapping as follows to make them all contribute 1. Let V be the set of residues for all the literals involving a particular variable (either the variable or its negation). Let V_+ be the residues for the variable and V_- for its negation. Let V_{+s} and V_{+n} partition V_+ into residues in swapping fragments and those not in swapping fragments, respectively; and similarly with V_{-s} and V_{-n} . By Tab. 2, each residue pair in $V_{+n} \cup V_{-s}$ contributes 1 (for either the identical or complementary term, as appropriate), and similarly for each residue pair in $V_{+s} \cup V_{-n}$. The remaining residue pairs (one in $V_{+s} \cup V_{-n}$ and one in $V_{+n} \cup V_{-s}$) each contribute 0. Now let us complement the swapping assignment for each residue in $V_{+s} \cup V_{-n}$ —if the residue is in a swapping fragment, shorten or break the fragment to make this residue not in swapping fragments; if it is not, create a single-residue swapping fragment. (As discussed above, we can modify the swapping in the trivial residues to ensure that the total number of swapping fragments is $2\tau + 2N_+$.) Following the above discussion, this change won't affect the potential contributed by separator pairs. It decreases the contribution from clause pairs with residues in V by at most 1 and doesn't affect other clause pairs. Pairs in $V_{+s} \cup V_{-n}$ still contribute 1, but pairs (identical or complementary) between $V_{+s} \cup V_{-n}$ and $V_{+n} \cup V_{-s}$ now contribute 1 instead of 0. The total increase is $|V_{+s} \cup V_{-n}| \times |V_{+n} \cup V_{-s}|$, while the total decrease is at most $|V_{+s} \cup V_{-n}|$. We have $|V_{+s} \cup V_{-n}| \times |V_{+n} \cup V_{-s}| \geq |V_{+s} \cup V_{-n}|$. In this manner, we can change all identical and complementary pairs to contribute 1, which means the swapping fragment assignments of residues for these pairs are consistent. \square