ALGORITHMS FOR SELECTING BREAKPOINT LOCATIONS TO OPTIMIZE DIVERSITY IN PROTEIN ENGINEERING BY SITE-DIRECTED PROTEIN RECOMBINATION

Wei Zheng[1], Xiaoduan Ye[1], Alan M. Friedman[2*], and Chris Bailey-Kellogg[1*]
[1] *Department of Computer Science, Dartmouth College*
[2] *Department of Biological Sciences, Markey Center for Structural Biology, Purdue Cancer Center, and Bindley Bioscience Center, Purdue University*

Protein engineering by site-directed recombination seeks to develop proteins with new or improved function, by accumulating multiple mutations from a set of homologous parent proteins. A library of hybrid proteins is created by recombining the parent proteins at specified breakpoint locations; subsequent screening/selection identifies hybrids with desirable functional characteristics. In order to improve the frequency of generating novel hybrids, this paper develops the first approach to explicitly plan for diversity in site-directed recombination, including metrics for characterizing the diversity of a planned hybrid library and efficient algorithms for optimizing experiments accordingly. The goal is to choose breakpoint locations to sample sequence space as uniformly as possible (which we argue maximizes diversity), under the constraints imposed by the recombination process and the given set of parents. A dynamic programming approach selects optimal breakpoint locations in polynomial time. Application of our method to optimizing breakpoints for an example biosynthetic enzyme, purE, demonstrates the significance of diversity optimization and the effectiveness of our algorithms.

1. INTRODUCTION

Protein engineering aims to create amino acid sequences encoding proteins with desired characteristics, such as improved or novel function. Two contrasting strategies are commonly employed to attempt to improve an existing protein. One approach focuses on redesigning a single sequence towards a new purpose, selecting a small number of mutations to the wild-type[1–5]. Another approach creates libraries of variant proteins to be selected or screened for desired characteristics. The library approach samples a larger portion of the sequence space, accumulating multiple mutations in each library member, increasing both the ability to reveal novel solutions to attaining function, as well as the risk of obtaining non-functional sequences.

Protein engineering by site-directed recombination (Fig. 1) provides one approach for generating libraries of variant proteins. A set of homologous parent genes are recombined at defined breakpoint locations, yielding a combinatorial set of hybrids[6–9]. In contrast to stochastic library construction methods[10–12], site-directed approaches choose breakpoint locations to optimize expected library quality, e.g., predicted disruption[7, 13, 14]. In both cases, the use of recombination enables the creation of protein variants that simultaneously accumulate a relatively large number of "natural" mutations relative to the parent. The mutations have been previously proven compatible with each other and within a similar structural and functional context, and are thus less disruptive than random mutations. Recombination-based approaches, when combined with high-throughput screening and selection, can avoid the need for precise modeling of the biophysical implications of mutations. They employ an essentially "generate-and-test" paradigm. As always, the goal is to bias the "generate" phase to improve the hit rate of the "test" phase.

A library is completely determined by selecting a set of parents and a set of breakpoint locations. To optimize an experiment so as to improve the expected quality of the resulting library, there are essentially two competing goals—we want the resulting proteins to be both viable and novel. Most previous work on planning site-directed recombination experiments has focused on enhancing viability, by seeking to minimize the amount of structural disruption due to recombination[6, 14–17]. However, breakpoints can also be selected so as to enhance novelty by maximizing the diversity of the hybrids. For example, consider choosing one internal breakpoint (in addition to the one at the end) for the three parents in Fig. 1, left. If we put the breakpoint between the

*Contact authors. CBK: 6211 Sudikoff Laboratory, Hanover, NH 03755, USA; cbk@cs.dartmouth.edu. AMF: Lilly Hall, Purdue University, West Lafayette, IN 47907, USA; afried@purdue.edu.

last two residues, all hybrids will be the same as the parents (i.e., a zero-mutation library). To improve the chance of getting novel hybrids, we must choose breakpoints that make hybrids different from each other and/or from the parents (Fig. 1, right).

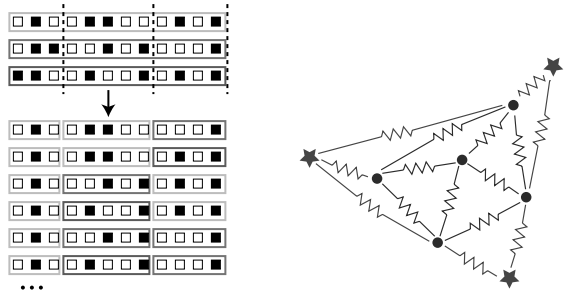


Fig. 1. Diversity optimization in site-directed protein recombination. (Left) Recombination of three parent sequences at a set of three breakpoints (we always include an extra breakpoint at the end of the sequence). A total of $3^3 = 27$ hybrids results, including three sequences equivalent to the parents. (Right) Repulsive spring analogy for library diversity. Hybrids (circles) are defined by parents (stars) and breakpoint locations. In order to sample the sequence space well, we want to choose breakpoint locations to push hybrids away from each other. (For clarity, only some relationships are illustrated.) Since the parents will also appear in the hybrid library, the hybrids are pushed away from them as well. Alternatively, an explicit goal may be to push the hybrids away from the parents as much as possible, so as to maximize the possibility for novel characteristics that are not found in the parents. We capture these two goals as the v_{HH} (hybrid-hybrid) and v_{HP} (hybrid-parent) metrics below, and demonstrate that they are highly correlated as a function of breakpoint location. Note that at all times, the hybrids are restricted to being a combination of the parents.

Diversity has been experimentally demonstrated to be important to obtaining new characteristics. The number of mutations has been correlated with functional change from wild-type in several proteins modified by different methodologies. Hybrid cytochromes P450 with the most altered profiles and greatest activity on a new substrate (allyloxybenzene) were found to have higher effective mutation levels (30–50 mutations among the 460 residues) than the enzymes with similar activities to the parents[16]. A random mutant library of TEM-1 β-lactamase with a minimal mutation load (8.2 mutations/gene) was found to have the highest frequency of clones carrying wild-type or minimally different activity, while a mutant library with maximal mutation load (27.2 mutations/gene) had the highest frequency of clones with improved activity on the normally poor substrate cefotaxime[18]. In a study of single chain Fv antibodies, the greatest affinity improvement was exhibited by libraries of moderate to high mutation levels (3.8–22.5 mutations/gene)[19]. Mutants with significantly higher affinity than the wild-type were well represented within the active fraction of the library population with high mutation levels.

This paper represents the first approach to explicitly plan for diversity in site-directed recombination. We develop metrics for evaluating diversity, in terms of both the differences among hybrids and the differences between hybrids and parents. We develop polynomial-time dynamic programming algorithms to select optimal breakpoint locations for these diversity metrics. We show that the algorithms are effective and significant in optimizing libraries from the purE family of biosynthetic enzymes.

2. METHODS

We are given a set of n parent sequences $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$, forming a multiple sequence alignment with each sequence of length l including residues and gaps. Our goal is to select a set of λ breakpoint locations $X = \{x_1, x_2, \ldots, x_\lambda \mid 1 \leq x_1 < x_2 < ... < x_\lambda = l\}$. For simplicity in notation, we always place the final breakpoint after the final residue position (i.e., $x_\lambda = l$). The breakpoints partition each parent P_a into λ fragments with sequences $P_a[1, x_1], P_a[x_1 + 1, x_2], \ldots, P_a[x_{\lambda-1} + 1, x_\lambda]$, where in general we use $S[r, r']$ to denote the amino acid string from position r to r' in sequence S, and $S[r]$ to denote the single amino acid at position r. A hybrid protein H_i is a concatenation of chosen parental fragments, assembled in the original order. Thus it is also of length l. Then a hybrid library $\mathcal{H}(\mathcal{P}, X) = \{H_1, H_2, \ldots, H_{n^\lambda}\}$ includes all combinations. Our goal is to choose X (such that $|X| = \lambda$ and $x_\lambda = l$) to optimize the diversity of library $\mathcal{H}(\mathcal{P}, X)$, for a set $\mathcal{P}$ of parents.

2.1. Library Diversity

For two amino acid sequences S and S' of length l, we define the *mutation level* $m(S, S')$ as the number of corresponding residues that differ:

$$m(S, S') = \sum_{1 \leq r \leq l} I\{S[r] \neq S'[r]\}, \qquad (1)$$

where indicator function I is 1 when the predicate is true and 0 when it is false. To mitigate the effect of neutral mutations, rather than using literal equality we measure functional relatedness using one of the standard sets of amino acid classes {{C},{F,Y,W},{H,R,K},{N,D,Q,E},{S,T,P,A,G}, {M,I,L,V}}. In either case, a "gap" in the alignment is taken as a distinct amino acid type. Our approach can be used with any similarly-structured metric for mutation level.

While our goal is to optimize library diversity, we show that the choice of parents and number of breakpoints, independent of breakpoint location, determines the mutation level between all pairs of hybrids (Claim 2.1), between one parent and all hybrids (Claim 2.2), and between all hybrids and all parents (Claim 2.3).

Claim 2.1. $\sum_{i=1}^{n^\lambda-1}\sum_{j=i+1}^{n^\lambda} m(H_i,H_j) = n^{2(\lambda-1)} \times \sum_{a=1}^{n-1}\sum_{b=a+1}^{n} m(P_a,P_b)$.

Claim 2.2. $\forall P_a \in \mathcal{P} : \sum_{i=1}^{n^\lambda} m(H_i,P_a) = n^{\lambda-1} \times \sum_{b=1}^{n} m(P_a,P_b)$.

Claim 2.3. $\sum_{a=1}^{n}\sum_{i=1}^{n^\lambda} m(H_i,P_a) = n^{\lambda-1} \times \sum_{a=1}^{n}\sum_{b=1}^{n} m(P_a,P_b)$.

Proof. Consider residue position r, where $1 \le r \le l$. Over the set of n^λ hybrids, there must be $n^{\lambda-1}$ instances of $P_1[r]$, $n^{\lambda-1}$ of $P_2[r]$, ..., and $n^{\lambda-1}$ of $P_n[r]$. Thus we have

$$\sum_{j=1}^{n^\lambda} m(H_j,H_i) = \sum_{r=1}^{l}\sum_{a=1}^{n} n^{\lambda-1} \times I\{P_a[r] \neq H_i[r]\} = n^{\lambda-1} \times \sum_{a=1}^{n} m(P_a,H_i). \quad (2)$$

By extending this to all pairs we have (Claim 2.1):

$$\sum_{i=1}^{n^\lambda-1}\sum_{j=i+1}^{n^\lambda} m(H_i,H_j) = \sum_{r=1}^{l}\sum_{a=1}^{n-1}\sum_{b=a+1}^{n} n^{2(\lambda-1)} \times I\{P_a[r] \neq P_b[r]\} = n^{2(\lambda-1)} \times \sum_{a=1}^{n-1}\sum_{b=a+1}^{n} m(P_a,P_b), \quad (3)$$

and by similarly comparing to a fixed parent we have (Claim 2.2):

$$\sum_{i=1}^{n^\lambda} m(H_i,P_a) = \sum_{r=1}^{l}\sum_{b=1}^{n} n^{\lambda-1} \times I\{P_a[r] \neq P_b[r]\} = n^{\lambda-1} \times \sum_{b=1}^{n} m(P_a,P_b). \quad (4)$$

Claim 2.3 follows immediately from Claim 2.2. □

The right-hand sides of the claims involve the parents but not the hybrids. Thus, surprisingly, the total number of mutations differentiating hybrids from each other and from the parents are independent of breakpoint locations and determined solely by the choice of parents. However, the distribution of the diversity within the library does depend on the breakpoints.

2.2. Metrics for Breakpoint Selection

Intuitively (Fig. 1, right), hybrids sample a sequence space defined by the parents and the breakpoint locations. *A priori*, we don't know what parts of the space are most promising, and thus we seek to generate novel proteins by sampling the space as uniformly as possible, rather than clustering hybrids near each other or near the parents.

More formally, consider one particular hybrid H_i. We want to make other hybrids roughly all as different from H_i; i.e., for the other H_j, the various $m(H_i,H_j)$ should be roughly equal. If we do this for all H_i, then we will also make the H_j different from each other (and not just from one particular H_i). That is, we want to make $m(H_i,H_j)$ relatively uniform, or minimize its deviation:

$$\sqrt{\frac{2}{n^\lambda(n^\lambda-1)} \times \sum_{i=1}^{n^\lambda-1}\sum_{j=i+1}^{n^\lambda} (m(H_i,H_j)-\overline{m})^2}, \quad (5)$$

where $\overline{m}$ is the mean value of $m(H_i,H_j)$.

Expanding the square in Eq. (5) yields an $m(H_i,H_j)^2$ term, a constant $\overline{m}^2$ term, and an $\overline{m} \times m(H_i,H_j)$ term whose sum is constant by Claim 2.1. Thus we need only minimize the $m(H_i,H_j)^2$ term, which we call the "variance." This gives us the first of two diversity optimization targets.

Problem 2.1. (**Hybrid-Hybrid Diversity Optimization**) *Given n parent sequences $\mathcal{P}$ of l residues*

and a positive integer λ, choose a set X of λ breakpoints (with $x_\lambda = l$) to minimize the hybrid-hybrid "variance" *$v_{HH}(X)$ of the resulting library, where*

$$v_{HH}(X) = \sum_{i=1}^{n^\lambda - 1} \sum_{j=i+1}^{n^\lambda} m(H_i, H_j)^2 \tag{6}$$

for $H_i, H_j \in \mathcal{H}(\mathcal{P}, X)$.

In addition to making hybrids different from each other, we also may want to focus on making them different from the parents. Following a similar intuition and argument as above, we obtain a second diversity optimization target:

Problem 2.2. (**Hybrid-Parent Diversity Optimization**) *Given n parent sequences $\mathcal{P}$ of l residues and a positive integer λ, choose a set X of λ breakpoints (where $x_\lambda = l$) to minimize the* hybrid-parent "variance" *$v_{HP}(X)$ of the resulting library, where*

$$v_{HP}(X) = \sum_{i=1}^{n^\lambda} \sum_{a=1}^{n} m(H_i, P_a)^2 \tag{7}$$

for $H_i \in \mathcal{H}(\mathcal{P}, X), P_a \in \mathcal{P}$.

Intuitively (Fig. 1, right), both H-H and H-P diversity optimization will spread hybrids out in sequence space. In fact, we can show that for any set X of λ breakpoints,

$$n^{\lambda-2} < \frac{v_{HH}(X)}{v_{HP}(X)} \leq n^{\lambda-1}. \tag{8}$$

Due to lack of space, we omit the proof, which is an algebraic manipulation of the terms. This relationships means that the two criteria should be highly correlated, as our results below confirm.

2.3. Dynamic Programming for Breakpoint Selection

In order to select an optimal set of breakpoints, we select breakpoints from left to right (N- to C-terminal) in the sequences. We slightly abuse our previous notation, truncating the parents at the last breakpoint selected (consistent with our previous use of the end of the sequence as the final breakpoint). As Fig. 2 illustrates, a hybrid library with breakpoints $X = \{x_1, \ldots, x_{k-1} = r', x_k = r\}$ extends a hybrid library with breakpoints $X' = \{x_1, \ldots, x_{k-1} = r'\}$ by concatenating each of the hybrids with each parent fragment $P_a[r'+1, r]$. Optimal substructure holds, since the best choice for x_k depends only on the best choice for x_{k-1}.

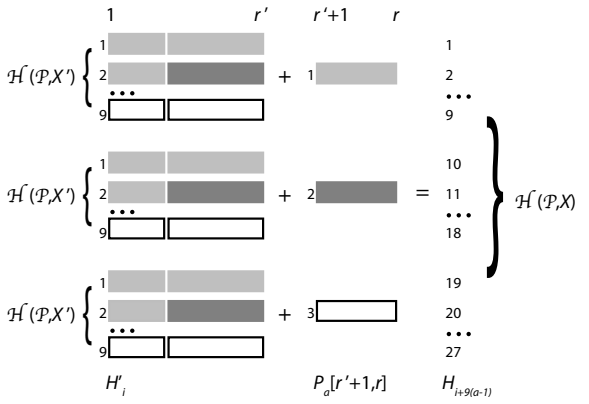


Fig. 2. Library substructure: library $\mathcal{H}(\mathcal{P}, X)$ ending at position r extends library $\mathcal{H}'(\mathcal{P}, X')$ ending at position r' by adding each parent fragment $P_a[r'+1, r]$ to each hybrid H'_i in $\mathcal{H}'(\mathcal{P}, X')$.

H-H Diversity Optimization. We use this insight to devise a dynamic programming recurrence to compute the optimal value of v_{HH} for the kth breakpoint location, based on the optimal values of v_{HH} for the possible $(k-1)$st locations. Define $d_{HH}(r, k)$ to be the minimum value of $v_{HH}(X)$, for any $X = \{x_1, \ldots, x_k = r\}$. Then $d_{HH}(l, \lambda)$ is the optimal value for H-H diversity optimization.

Claim 2.4. *We can compute $d_{HH}(r, k)$ recursively in time $O(\lambda n^2 l^2)$ as*

$$\begin{cases} \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} m(P_a[1, r], P_b[1, r])^2 & \text{if } k = 1, \\ \min_{r'<r} \{n^2 \times d(r', k-1) + e_{HH}(k, r, r')\} & \text{if } k > 1, \end{cases}$$

where e_{HH} is defined in Eq.(10).

Proof. As discussed above, the hybrid library $\mathcal{H}(\mathcal{P}, X)$ is extended from $\mathcal{H}(\mathcal{P}, X')$, where X' is missing the final breakpoint in X. Let us use H_i for the members of $\mathcal{H}(\mathcal{P}, X)$ and H'_i for those of $\mathcal{H}(\mathcal{P}, X')$, and "+" to denote sequence concatenation. Following the structure in Fig. 2, we can separate v_{HH} into terms $\mathcal{H}(\mathcal{P}, X') + P_a[r'+1, r]$ from hybrids in a single "sub-library" sharing the same added fragments, and terms $\mathcal{H}(\mathcal{P}, X') + P_a[r'+1, r]$ and $\mathcal{H}(\mathcal{P}, X') + P_b[r'+1, r]$ between separate "sub-libraries" with distinct added fragments. This gives Eq. (11).

Expanding the second term on the right-hand side in Eq. (11) gives Eq. (12).

By Claim 2.1 for parents with $k-1$ breakpoints (and thus truncated at r'), we have Eq. (13).

We can substitute twice the right-hand side of Eq. (13) into the third term in Eq. (12) (with "twice" to account for summing over all pairs vs. all distinct pairs), noting that the sums over the parents a and b in Eqs. (12) and (13) are independent. We can then substitute the resulting formula back into Eq. (11). Simplification yields Eq. (14), where most terms are collected into e_{HH}, except for the sums of $m(H'_i, H'_j)^2$, including n from the first term in Eq. (11) and twice $\binom{n}{2}$ from the first term in Eq. (12) (with "twice" again due to all vs. all distinct). Because Eq. (14) only depends on r' and not the previous breakpoints,

$$d(r,k) = \min_{r'<r}\{n^2 \times d(r', k-1) + e_{HH}(k,r,r')\}. \quad (9)$$

Computing this recurrence using dynamic programming requires a table of size $\lambda \times l$; filling in each entry requires time $O(n^2)$ to compute e_{HH} and must look back at $O(l)$ previous entries to compute the minimum, for a total time of $O(\lambda n^2 l^2)$. □

$$\begin{aligned} e_{HH}(k,r,r') = {} & 4n^{2(k-2)} \times \sum_{a=1}^{n-1}\sum_{b=a+1}^{n} m(P_a[1,r'], P_b[1,r']) \times \sum_{a=1}^{n-1}\sum_{b=a+1}^{n} m(P_a[r'+1,r], P_b[r'+1,r]) \\ & + n^{2(k-1)} \times \sum_{a=1}^{n-1}\sum_{b=a+1}^{n} m(P_a[r'+1,r], P_b[r'+1,r])^2. \end{aligned} \quad (10)$$

$$\begin{aligned} \sum_{i=1}^{n^k-1}\sum_{j=i+1}^{n^k} m(H_i,H_j)^2 = {} & \sum_{a=1}^{n}\sum_{i=1}^{n^{k-1}-1}\sum_{j=i+1}^{n^{k-1}} m(H'_i + P_a[r'+1,r], H'_j + P_a[r'+1,r])^2 \\ & + \sum_{a=1}^{n-1}\sum_{b=a+1}^{n}\left(\sum_{i=1}^{n^{k-1}}\sum_{j=1}^{n^{k-1}} m(H'_i + P_a[r'+1,r], H'_j + P_b[r'+1,r])^2\right). \end{aligned} \quad (11)$$

$$\begin{aligned} & \sum_{a=1}^{n-1}\sum_{b=a+1}^{n}\left(\sum_{i=1}^{n^{k-1}}\sum_{j=1}^{n^{k-1}} m(H'_i + P_a[r'+1,r], H'_j + P_b[r'+1,r])^2\right) = \\ & \qquad \sum_{a=1}^{n-1}\sum_{b=a+1}^{n}\left(\sum_{i=1}^{n^{k-1}}\sum_{j=1}^{n^{k-1}} m\left(H'_i, H'_j\right)^2\right) \\ & \qquad + \sum_{a=1}^{n-1}\sum_{b=a+1}^{n}\left(\sum_{i=1}^{n^{k-1}}\sum_{j=1}^{n^{k-1}} m\left(P_a[r'+1,r], P_b[r'+1,r]\right)^2\right) \\ & \qquad + \sum_{a=1}^{n-1}\sum_{b=a+1}^{n}\left(\sum_{i=1}^{n^{k-1}}\sum_{j=1}^{n^{k-1}} 2\, m\left(H'_i, H'_j\right) \times m\left(P_a[r'+1,r], P_b[r'+1,r]\right)\right). \end{aligned} \quad (12)$$

$$\sum_{i=1}^{n^{k-1}-1}\sum_{j=i+1}^{n^{k-1}} m(H'_i, H'_j) = nn^{2(k-2)} \times \sum_{a=1}^{n-1}\sum_{b=a+1}^{n} m(P_a[1,r'], P_b[1,r']). \quad (13)$$

$$\sum_{i=1}^{n^k-1}\sum_{j=i+1}^{n^k} m(H_i,H_j)^2 = n^2 \times \sum_{i=1}^{n^{k-1}-1}\sum_{j=i+1}^{n^{k-1}} m(H'_i, H'_j)^2 + e_{HH}(k,r,r'). \quad (14)$$

H-P Diversity Optimization. A similar dynamic programming algorithm to the H-H one above allows us to optimize H-P diversity. Let $d_{HP}(r,k)$ be the minimum value of $v_{HP}(X)$ for any $X = \{x_1, \ldots, x_k = r\}$, so that $d_{HP}(l,\lambda)$ is the optimal value for H-P diversity optimization.

Claim 2.5. *We can compute $d_{HP}(r,k)$ recursively in time $O(\lambda n^2 l^2)$ as*

$$\begin{cases} \sum_{a=1}^{n}\sum_{b=1}^{n} m(P_a[1,r],P_b[1,r])^2 & \text{if } k=1, \\ \min_{r'<r}\{n \times d_{HP}(r',k-1) + e_{HP}(k,r,r')\} & \text{if } k>1, \end{cases}$$

where e_{HP} is defined in Eq. 16.

Proof. The proof is similar to that for H-H diversity. By partitioning the library, we have Eq. (17).

By Claim 2.2 for parents with $k-1$ breakpoints truncated at position r', we have Eq. (18).

Substituting the right-hand side of Eq. (18) into the third term in Eq. (17), and simplifying, we get Eq. (19). Here $e_{HP}(k,r,r')$ also depends only on r' and not the preceding breakpoints, so we have

$$d(r,k) = \min_{r'<r}\{n \times d(r',k-1) + e_{HP}(k,r,r')\}. \quad (15)$$

The table size and time to fill in each entry are the same as with H-H diversity. □

$$e_{HP}(k,r,r') = 2n^{k-2} \times \sum_{a=1}^{n}\left(\sum_{b=1}^{n} m(P_a[1,r'],P_b[1,r']) \times \sum_{b=1}^{n} m(P_a[r'+1,r],P_b[r'+1,r])\right) + n^{k-1} \times \sum_{a=1}^{n}\sum_{b=1}^{n} m(P_a[r'+1,r],P_b[r'+1,r])^2. \quad (16)$$

$$\begin{aligned}\sum_{a=1}^{n}\sum_{i=1}^{n^k} m(H_i,P_a[1,r])^2 &= \sum_{a=1}^{n}\sum_{b=1}^{n}\sum_{i=1}^{n^{k-1}} m(H'_i + P_b[r'+1,r], P_a[1,r'] + P_a[r'+1,r])^2 \\ &= \sum_{a=1}^{n}\sum_{b=1}^{n}\sum_{i=1}^{n^{k-1}} m(H'_i,P_a[1,r'])^2 + \sum_{a=1}^{n}\sum_{b=1}^{n}\sum_{i=1}^{n^{k-1}} m(P_b[r'+1,r],P_a[r'+1,r])^2 \\ &\quad + \sum_{a=1}^{n}\sum_{b=1}^{n}\sum_{i=1}^{n^{k-1}} 2\, m(H'_i,P_a[1,r']) \times m(P_b[r'+1,r],P_a[r'+1,r]) \end{aligned} \quad (17)$$

$$\sum_{i=1}^{n^{k-1}} m(H'_i,P_a[1,r']) = n^{k-2} \times \sum_{b=1}^{n} m(P_a[1,r'],P_b[1,r']). \quad (18)$$

$$\sum_{a=1}^{n}\sum_{i=1}^{n^k} m(H_i,P_a[1,r])^2 = n \times \sum_{a=1}^{n}\sum_{i=1}^{n^{k-1}} m(H'_i,P_a[1,r'])^2 + e_{HP}(k,r,r'). \quad (19)$$

3. RESULTS AND DISCUSSION

The orthologous proteins of the purE family (COG 41 and pfam 731) form a valuable target for engineering a diverse hybrid library. The small (generally about 120 residue) purE sequences, which form either a single protein or a single domain in a fusion protein, catalyze steps in the *de novo* synthesis of purines. While clear orthologs, purE proteins carry out substantially different enzymatic activities in different organisms: in eubacteria, fungi and plants (as well as probably most archaebacteria), the purE product functions as a mutase in the second step of a two-step reaction, while in metazoans and methanogenic archaebacteria, the purE product functions as a carboxylase in a single-step reaction that yields the same product[20, 21]. A genetic system allows selection *in vivo* for both the catalytic mechanism and different levels of enzymatic activity.

In order to uncover explanations for the striking divergence of function (mutase vs. carboxylase activity) within homologous sequences, we sought to evenly partition the sequence space, bridging the two "islands." To establish a set of purE parents, we performed standard sequence search and alignment techniques, and eliminated columns not mapped to

the structure of *E. coli* purE (PDB id: 1qcz) and eliminated sequences with more than 20% gaps. This yielded a diverse set of 367 sequences of 162 residues each, including 28 of the rarer class of metazoans and methanogens with inferred carboxylase activity. The average pairwise sequence identity (under the classes of Sec. 2.1) is 65.8%.

We first chose three diverse parent sequences from the purE family: P_1 from the eubacterium *Escherichia coli*, P_2 from the vertebrate chicken (*Gallus gallus*) and P_3 from the methanogenic archaebacterium *Methanothermobacter thermautotrophicus*. The mutation levels among these three parent sequences are $m(P_1, P_2) = 94$, $m(P_1, P_3) = 65$ and $m(P_2, P_3) = 85$. We applied our algorithms to choose a set of 4, 5, 6 and 7 internal breakpoints (Fig. 3).

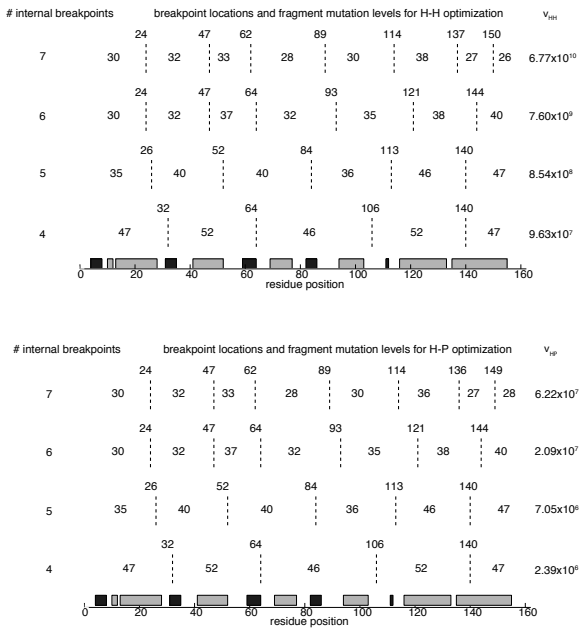


Fig. 3. Breakpoint locations for three purE proteins, under (top) H-H and (bottom) H-P diversity optimization. The sequence is labeled with residue indices, with α-helices shown with light boxes and β-sheets with dark ones, according to the crystal structure of *E. coli* purE (PDB id: 1qcz). Numbers above the dashed lines indicate the positions of breakpoints. Numbers within the fragments give the sum of the intra-fragment mutation levels between all pairs of parents.

For 4, 5, and 6 internal breakpoints, both H-H and H-P optimization yield the same breakpoint locations. For 7 internal breakpoints, the locations only differ by a few residues for the last two breakpoints. As the mutation levels show, in seeking to make hybrids distributed uniformly in the sequence space, breakpoint selection optimization equalizes the contributions to diversity from the fragments.

To show that it is not likely to generate equivalent diversity by chance, we chose 10000 random sets of four internal breakpoints. The distributions of v_{HH} and v_{HP} for these random sets are plotted in Fig. 4.

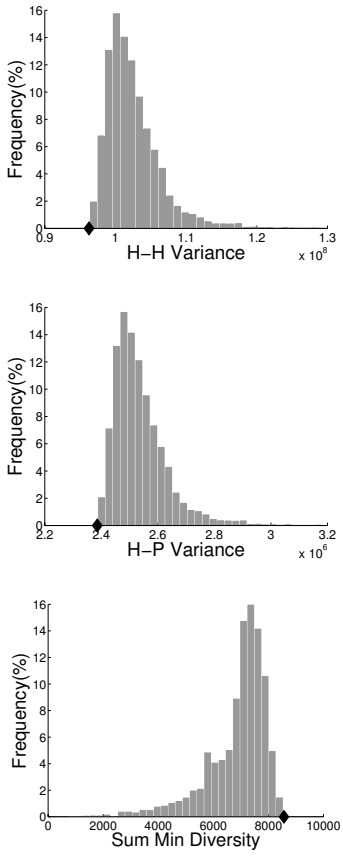


Fig. 4. Distribution of diversity values for random breakpoint selection compared with dynamic programming optimization. The x-axis indicates different diversity values. The y-axis indicates the frequencies of the diversity value among 10000 random sets of four internal breakpoints. Dark diamonds indicate diversity values for breakpoints selected by our algorithm: 9.63×10^7 for H-H, 2.39×10^6 for H-P, and 8565 for sum-min (using the H-P breakpoints).

The breakpoints selected by our algorithms are better than any random selection. For comparison, we also calculated the "sum-min" diversity metric $\sum_{i=1}^{n^\lambda} \min_a m(H_i, P_a)$ used by Arnold and colleagues[13]. Currently no efficient algorithm has

been found to directly maximize sum-min diversity, but our H-H and H-P optimization algorithms also apparently do a good job of optimizing it; no random breakpoint selection was found to do better.

As we proved in Claims 2.1–2.3, the choice of parent sequences determines the total number of mutations. We also expect it to affect library diversity, since the choice of parents defines the available sequence space (we can only recombine the parents). To test the effect of parent diversity on optimization of library diversity, we randomly chose 1000 three-member purE parent sets. For each set, we selected optimized breakpoints with our algorithms, and calculated the three diversity values as above (using the H-P breakpoints for calculation of sum-min diversity). For each parent set, we also calculated the means of the three diversity metrics over 1000 random sets of four internal breakpoints. Fig. 5 plots the additive difference between values under our optimized breakpoint sets vs. mean values for random breakpoint sets. As the total mutation level of the parents increases, so does the improvement of our breakpoints over random. Presumably, more parent diversity provides more opportunity to explicitly optimize library diversity.

As shown by the ratio analysis of v_{HH} and v_{HP} in Eq. (8) and confirmed empirically in Fig. 3, hybrid-parent diversity optimization is highly correlated with hybrid-hybrid diversity optimization. It also appears to be highly correlated with the sum-min diversity of Arnold and co-workers. Fig. 6(a,c) shows the relationship among these values, using the same random breakpoint selections as in Fig. 4. Optimization for hybrid-parent diversity also achieves good diversity according to the other two metrics. Fig. 6(b,d) shows that the correlation remains extremely high (R near 1 and -1) over the random parent sets and random breakpoint sets used in Fig. 5. These correlations allow us to do just one polynomial-time diversity optimization, achieving three goals simultaneously.

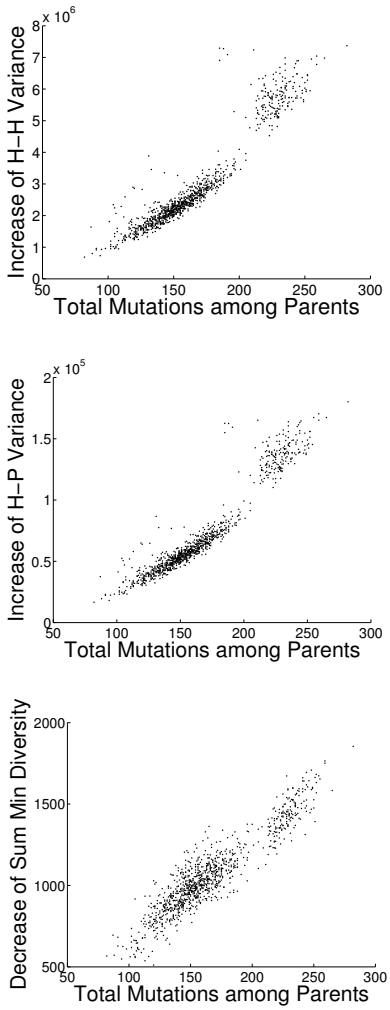
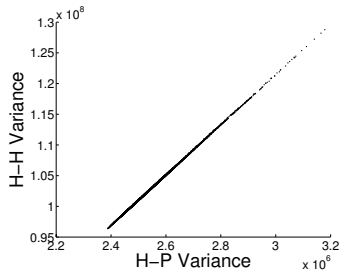
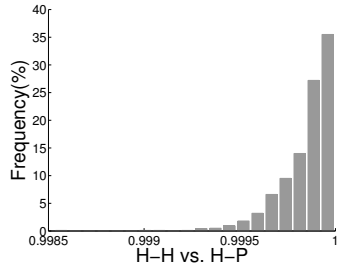


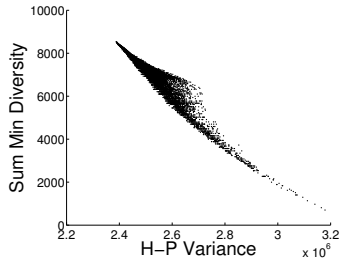
Fig. 5. Effect of parent selection on diversity optimization. The x-axis indicates the total number of mutations between pairs of purE parents in 1000 randomly chosen three-parent plans. The y-axis indicates, for each parent choice, the improvement in diversity from 1000 random plans to the optimized plan (larger y values indicate more improvement). For H-H and H-P, improvement is measured as the mean random plan value minus the value of our plan; for sum-min, improvement is the value of our plan minus the mean random plan value.



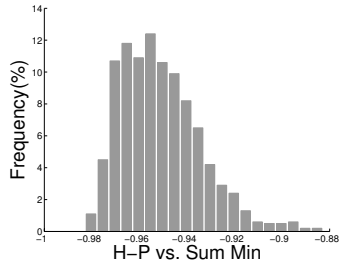
(a)



(b)



(c)



(d)

Fig. 6. Relationship among three diversity metrics. (a,c): Correlation over random four-breakpoint sets with the fixed three-parent set of Fig. 4. The x-axis indicates H-P variance (v_{HP}), the y-axis indicates H-H variance (v_{HH}) or sum-min diversity, respectively. (b,d): Histogram of correlation coefficients of diversity metrics for random sets of four internal breakpoints with the same random parent sets as Fig. 5. Note that the histograms are focused on a small region very near 1 and -1, respectively.

4. CONCLUSION

While diversity in hybrid libraries is the key to finding novel function, library design has instead previously focused on reducing the fraction of non-viable hybrids. Diversity has been a side-effect, rather than an explicit optimization target. In this initial approach to optimizing diversity, we showed here that the total number of mutations in a library is fixed by the choice of parents, but that their distribution among hybrids can be optimized so that the hybrids broadly sample sequence space. Our metrics and algorithms enable efficient selection of breakpoint locations to optimize diversity. In practical applications, a suitable combination of diversity and viability will be desired. Since the dynamic programming approach here has a similar structure to algorithms for minimizing disruption[13, 14], it might be possible to optimize for a desired trade-off between these two competing goals. We likewise anticipate integrating knowledge of important residues (e.g., targeting an active site), via appropriate weights. Finally, since the parents define the searchable sequence space and the total possible diversity, the importance of parent selection is reemphasized.

ACKNOWLEDGMENTS

This work was supported in part by an NSF CAREER award to CBK (IIS-0444544) and a grant from NSF SEIII (IIS-0502801) to CBK, AMF, and Bruce Craig.

References

1. B. Kuhlman, G. Dantas, G.C. Ireton, G. Varani, B.L. Stoddard, and D. Baker. Design of a novel globular protein fold with atomic-level accuracy. *Science*, 302(5649):1364–8, 2003.
2. L.L. Looger, M.A. Dwyer, J.J. Smith, and H.W. Hellinga. Computational design of receptor and sensor proteins with novel functions. *Nature*, 423(6936):185–90, 2003.
3. R.H. Lilien, B.W. Stevens, A.C. Anderson, and B.R. Donald. A novel ensemble-based scoring and search algorithm for protein redesign and its application to modify the substrate specificity of the gramicidin synthetase A phenylalanine adenylation enzyme. *J. Comput. Biol.*, 12(6):740–61, 2005.
4. J. Li, Z. Yi, M.C. Laskowski, M. Laskowski Jr., and C. Bailey-Kellogg. Analysis of sequence-reactivity space for protein-protein interactions. *Proteins*, 58(3):661–71, 2005.

5. I. Georgiev, R.H. Lilien, and B.R. Donald. A novel 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. In *Proc. RECOMB*, pages 530–45, 2006.
6. C.A. Voigt, C. Martinez, Z.G. Wang, S.L. Mayo, and F.H. Arnold. Protein building blocks preserved by recombination. *Nat. Struct. Biol.*, 9(7):553–8, 2002.
7. M.M. Meyer, J.J. Silberg, C.A. Voigt, J.B. Endelman, S.L. Mayo, Z.G. Wang, and F.H. Arnold. Library analysis of SCHEMA-guided protein recombination. *Protein Sci.*, 12:1686–93, 2003.
8. C.R. Otey, M. Landwehr, J.B. Endelman, K. Hiraga, J.D. Bloom, and F.H. Arnold. Structure-guided recombination creates an artificial family of cytochromes P450. *PLoS Biol.*, 4(5):e112, 2006.
9. L. Saftalov, P.A. Smith, A.M. Friedman, and C. Bailey-Kellogg. Site-directed combinatorial construction of chimaeric genes: general method for optimizing assembly of gene fragments. *Proteins*, 64(3):629–42, 2006.
10. W.P. Stemmer. Rapid evolution of a protein *in vitro* by DNA shuffling. *Nature*, 370(6488):389–91, 1994.
11. A.M. Aguinaldo and F.H. Arnold. Staggered extension process (StEP) *in vitro* recombination. *Methods Mol. Biol.*, 231:105–10, 2003.
12. W.M. Coco. RACHITT: Gene family shuffling by random chimeragenesis on transient templates. *Methods Mol. Biol.*, 231:111–127, 2003.
13. J.B. Endelman, J.J. Silberg, Z.G. Wang, and F.H. Arnold. Site-directed protein recombination as a shortest-path problem. *Protein Eng. Des. Sel.*, 17:589–594, 2004.
14. X. Ye, A.M. Friedman, and C. Bailey-Kellogg. Hypergraph model of multi-residue interactions in proteins: sequentially-constrained partitioning algorithms for optimization of site-directed protein recombination. *J. Comput. Biol.*, in press, 2007. Conference version: Proc. RECOMB, 2006, pp. 15-29.
15. G.L. Moore and C.D. Maranas. Identifying residue-residue clashes in protein hybrids by using a second-order mean-field approach. *PNAS*, 100(9):5091–6, 2003.
16. C.R. Otey, J.J. Silberg, C.A. Voigt, J.B. Endelman, G. Bandara, and F.H. Arnold. Functional evolution and structural conservation in chimeric cytochromes p450: calibrating a structure-guided approach. *Chem. Biol.*, 11(3):309–18, 2004.
17. M. C. Saraf, A. Gupta, and C.D. Maranas. Design of combinatorial protein libraries of optimal size. *Proteins*, 60(4):769–77, 2005.
18. M. Zaccolo and E. Gherardi. The effect of high-frequency random mutagenesis on *in vitro* protein evolution: a study on TEM-1 beta-lactamase. *J. Mol. Biol.*, 285:775–83, 1999.
19. P.S. Daugherty, G. Chen, B.L. Iverson, and G. Georgiou. Quantitative analysis of the effect of the mutation frequency on the affinity maturation of single chain Fv antibodies. *PNAS*, 97:2029–34, 2000.
20. S.M. Firestine, S.W. Poon, E.J. Mueller, J. Stubbe, and V.J. Davisson. Reactions catalyzed by 5-aminoimidazole ribonucleotide carboxylases from Escherichia coli and Gallus gallus: a case for divergent catalytic mechanisms. *Biochemistry*, 33:11927–34, 1994.
21. J. Thomas et al. in preparation.