

# Site-Directed Combinatorial Construction of Chimaeric Genes: General Method for Optimizing Assembly of Gene Fragments

Liz Saftalov\* Peter A. Smith† Alan M. Friedman†§ Chris Bailey-Kellogg‡ §

## Running Title: Specific Planned Ligation of Short Overhangs

### Abstract

Site-directed construction of chimaeric genes by *in vitro* recombination “mixes-and-matches” precise building blocks from multiple parent proteins, generating libraries of hybrids to be tested for structure-function relationships and/or screened for favorable properties and novel enzymatic activities. A direct annealing and ligation method can construct chimaeric genes without requiring sequence identity between parents, except for the short ( $\approx 3$  nt) sequences of the fragment overhangs used for specific ligation. Careful planning of the assembly process is necessary, though, in order to ensure effective construction of desired fragment assemblies and to avoid undesired assemblies (e.g., repetition of fragments, fragments out of order).

We develop algorithms for specific planned ligation (SPLISO) that efficiently explore possible assembly plans, varying the fragment overhangs and the order of ligation steps in the assembly pathway. While there is a combinatorial explosion in the number of possible assembly plans as the number of breakpoints and parent genes increases, we employ a dynamic programming approach to find globally optimal ones in low-order polynomial time (in practice, taking only seconds for basic assembly plans). We demonstrate the effectiveness of our algorithms in planning the assembly of hybrid libraries, under a variety of experimental options and restrictions, including flexibility in the position and amino acid sequence of breakpoints. Our method promises to enable more effective application of site-directed recombination to protein investigation and engineering.

**Keywords:** Protein engineering; experiment planning and optimization; protein modularity; recombinant gene assembly; structure-function relationships

---

\*Department of Computer Science, Purdue University.

†Department of Biological Sciences and the Purdue Cancer Center, Purdue University.

‡Department of Computer Science, Dartmouth College.

§Contact authors. AMF: Lilly Hall, Purdue University, West Lafayette, IN 47907, USA; phone: 765-494-5911; fax: 765-496-1189; email: afried@purdue.edu. CBK: 6211 Sudikoff Laboratory, Hanover, NH 03755, USA; phone: 603-646-3385; fax: 603-646-1672; email: cbk@cs.dartmouth.edu.

# 1 Introduction

Chimaeric genes are valuable for understanding protein structure and function and for creating variants with improved properties and novel functions. Many methods for making chimaeric genes have been developed, including DNA shuffling [1], ITCHY [2], SCRATCHY [3], StEP [4], RACHITT [5], SHIPREC [6], SCOPE [7], and RM-PCR [8], and SISDC [9]. Most of these methods rely on stochastic joining events. Protein engineering might be improved if the sites of recombination (breakpoints) could be selected to yield the greatest fraction of stable and active chimaeras [10, 11]. Site-directed recombination would ease subsequent screening and selection, especially valuable for the large number of protein engineering projects where simple selections are not available. Experimental testing of specific hypotheses about the modular composition of proteins [10, 12, 13] also requires making site-directed chimaeras, as do investigations into the interconnections and correlations between structural features that are separated in amino acid sequence [14, 15].

In contrast to site-directed recombination, many stochastic methods (including DNA shuffling, StEP, and RACHITT) rely on annealing of parental DNA sequences, but many protein families useful for the construction of chimaeric genes are too distantly related for effective and unbiased annealing [16]. While ITCHY, SCRATCHY, and SHIPREC do not rely on annealing, they also do not allow precise control over the site of recombination. In addition, the separate steps for each crossover required by these methods and the generation of only a minority of in-frame combinations at each site make these methods most suitable for chimaeras having only one or a small number of crossovers.

Several non-stochastic methods for making chimaeric libraries are available, but these also have limitations. Splicing by overlap extension (SOEing) [17] requires substantial regions of identical sequence in the overlap between the two parents to prime the PCR reactions. The requirement for substantial overlap limits the experimental possibilities and may require undesired amino acid substitutions to create identical sequences in diverse proteins. Separate reactions are also required for each pair of parents unless the same long overlap is to be enforced on all progeny. The SCOPE and RM-PCR methods do allow recombination at specific sites without enforcing identical sequences between parents. However, separate steps are required for each crossover and each pair of parents in these methods, either to make a chimaeric fragment for SCOPE or a dimer template for RM-PCR. The requirements of these methods as originally published limits the ability to make more diverse libraries with larger numbers of parents and crossovers. These methods could be extended if it were possible to make chimaeric priming oligos for SCOPE or dimer templates for RM-PCR that randomly combine all parents. Another non-stochastic method, SISDC [9], does allow random recombination of all parents, requiring overhangs of identical sequence at the breakpoints. SISDC and the proposed extensions to SCOPE and RM-PCR are in fact all potential applications for the general method described here.

We are focused on making practicable an alternative strategy for generating diverse libraries with less experimental effort. Previously demonstrated in the generation of beta lactamase chimaeras [18], this method is based on well-studied biochemical principles of DNA annealing and ligation. In the specific ligation method, synthetic or PCR-generated fragments from diverse parents but bearing the same short specific overhangs are annealed and ligated. Limitations arise only from the requirements for specific annealing and ligation during assembly. The short overhangs must be selected so that all parents share the same sequence at the overhang for a single breakpoint, thus allowing recombination among all of them in a single experiment. The directionality of assembly is predominantly determined by the specificity of the overhang sequences in annealing and ligation. However, the ability of diverse short overhangs to prevent incorrect assembly is limited, and the fragments may need to be assembled in a specific sequence of ligation steps in order to generate only the desired products. Thus novel computational methodology is required to identify

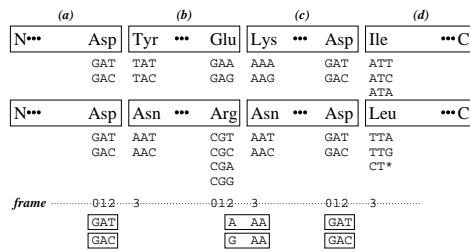
a suitable assembly pathway and set of overhangs and thereby support effective use of this technique. With its many advantages, site-directed construction via annealing and ligation would be considered more often with such methodology available to rapidly design experiments that incorporate large numbers of parents and crossovers.

We present here a generalized method for site-directed combinatorial construction of chimaeric genes. In the Specific Planned LIgation of Short Overhangs (SPLISO) method, we computationally optimize both the overhangs and the assembly pathway for a previously defined set of parents and recombination breakpoints (Fig. 1). Manual determination of overhangs and assembly pathway for such ligations has been conducted and experimentally verified for recombining fourteen protein fragments from two closely related parents [18] and for joining up to six genes coding for resistance to different antibiotics [19]. The manual approach rapidly becomes more difficult, though, with increased experimental complexity. For a typical case that we will present, the total number of possible combinations of overhangs and assembly pathways is greater than  $10^{16}$ . Our general method for determining overhangs and order of assembly employs dynamic programming to efficiently determine globally optimal solutions from this combinatorial explosion. In solving this larger problem, we also solve limitations present in the SCOPE and RM-PCR methods by providing means for constructing unbiased pools of randomly joined primers and dimer templates.

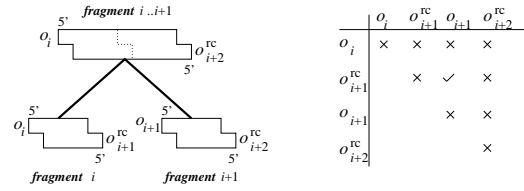
Combinatorial diversity in a hybrid library results from employing multiple parents and multiple breakpoints. Increasing library diversity by employing more breakpoints requires more complex experimental plans. Our method has no arbitrary limit on complexity; more complex assemblies are readily planned through an increased number of ligation steps, each one optimizing the number of fragments that are brought together. Since sequences in a protein family are more variable than structure [20], homologs with divergent sequences are often available to contribute effective diversity to the construction of chimaeric libraries. However, employing additional parents restricts the use of potential overhangs (which all parents must share) and thus indirectly the possibilities for assembly, particularly as the parents become more diverse. We also propose mechanisms for incorporating additional degrees of freedom, in the form of conservative substitutions and small shifts in breakpoint location, in order to overcome the restrictions on overhang choices. (In contrast, any pair of parents too distantly related will not recombine at all in the stochastic methods.) Our method promotes diversity through efficient planning of the overhangs and the assembly pathway, and through the controlled addition of degrees of freedom.

A major factor in the ability to employ greater numbers of breakpoints is the number of fragments assembled in a single step. This number is limited experimentally by the ability to accurately ligate the fragments and ensure their non-biased assembly. We study here the case of binary assemblies (two fragments ligated together at a step) as well as the more general multi-way assemblies (two or more fragments at a step). The number of fragments ligatable in a single assembly step potentially depends on the success of the present method; optimal selection of overhangs as described here so that they are maximally non-complementary (except where complementarity is desired) should generally allow assembling a large number of fragments in a single step. In a well-designed test case, Tsuge et al. [19] demonstrated efficient ligation of six fragments into a linear product using three nucleotide 3' overhangs. Even using more stringent restrictions on overhang sequences, we show that there are thousands of possible six-fragment assemblies (but none with seven or more). Employing multi-way ligation, conservative substitutions, and small shifts in breakpoint location, we demonstrate that efficient assembly of highly diverse libraries (up to nine divergent parents in a protein family and incorporating thirteen breakpoints, generating  $> 2 \times 10^{13}$  precisely directed chimaeras) can be readily planned.

Algorithm 1: Identify admissible nucleotide overhangs



Algorithm 2: Evaluate overhangs in an assembly step



Algorithm 3: Optimize tree structure and overhang selection

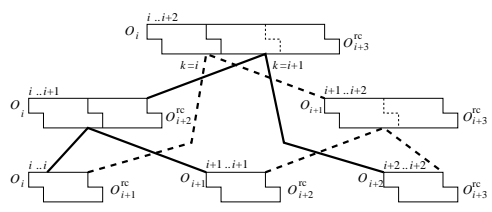


Figure 1: Overview of chimaeric gene assembly by specific planned ligation of short overhangs (SPLISO). (1) Overhangs common to all parent sequences are identified. For three-nucleotide overhangs, a breakpoint can be defined by a codon from either of the adjacent amino acids, as in (a)/(b), or spanning parts of the two codons, as in (b)/(c). Two adjacent fragments can be ligated when the second has the overhang nucleotide triplet at its 5' end on one strand and the first has the complement on the other strand. (2) An assembly step ligates fragments  $i$  and  $i + 1$ , based on the complementary overhangs  $o_{i+1}^{rc}$  and  $o_{i+1}$  (see section **Admissible Fragment Overhangs** for notation). Such a step is evaluated for the likelihood of undesired ligations (marked with  $\times$  symbols in the table) between all pairs of overhangs other than the desired one (marked with a  $\checkmark$  symbol). Note that overhang  $o_{i+1}$  is no longer accessible in the assembled fragment (as shown with dotted lines), allowing the same or similar overhangs to be used in subsequent assembly steps. (3) An assembly tree structure and specific overhangs for the fragments are selected so that each assembly step has the desired ligation as the unique biochemically feasible result. Shown is the case of binary assembly of a fragment consisting of fragments  $i$  through  $i + 2$ . Two choices are available for the breakpoint  $k$  being ligated in the top node. When  $k = i$  (dashed lines), fragment  $i$  is combined with the fragment assembled from  $i + 1$  and  $i + 2$ ; when  $k = i + 1$  (solid lines), the fragment assembled from  $i$  and  $i + 1$  is combined with fragment  $i + 2$ . The selected assembly pathway must admit choices for the overhangs that satisfy Algorithms 1 (are admissible) and 2 (yield only desired ligations). For example, if this assembly involved fragments (a), (b), and (c) illustrated above in Algorithm 1, the  $k = i$  tree (dashed lines) joining (b) and (c) first would fail Algorithm 2 when joining (a) to the (b)+(c) product, as the potential downstream overhangs for both (a) and (b)+(c) are the same at two out of three nucleotides. The  $k = i + 1$  tree (solid lines) joining (a) and (b) first, and then (c), avoids such problems by hiding the overhang (stepped solid line) between fragments  $i$  and  $i + 1$  in the earlier step. Note that substituting Ile for Leu at the start of fragment (d) in the second parent sequence would allow use of an upstream overhang such as ATC for (d) and a complementary 3'-TAG-5' downstream for (c). These overhangs would then permit a three-way assembly of (a), (b), and (c) in a single step.

## 2 Materials and Methods

Computationally, the specific ligation approach to constructing chimaeric genes can be formulated as a tree-structured set of assembly steps with a fixed ordering of leaves. The leaves of the tree represent the *original fragments*, the set of DNA molecules synthesized prior to assembly. They are defined by the  $n$  specified breakpoint locations. Each internal node represents an assembly of two or more fragments, producing a larger *assembled fragment*. Assembled fragments are themselves assembled into successively larger fragments, ultimately producing the completely assembled gene at the root of the tree. In practice, assembly steps (ligation of fragments) at the same level in the tree would generally be conducted in separate parallel reactions.

In general, our method (Fig. 1) optimizes an assembly plan for a maximal set of usable parents by considering choices for the degrees of freedom in the assembly design. These degrees of freedom are the nucleotide overhangs for the fragments and the tree structure ordering the assembly steps. Given a list of breakpoint locations for the parent protein sequences, we must first find for each fragment the set of nucleotide overhangs common to the parents so that all may be ligated without bias and without change in sequence (Algorithm 1). We will then select a single overhang for each fragment, along with a tree structure defining the assembly pathway. The selected tree and overhangs must support correct assembly of the desired fragments at each node while avoiding spurious assembly of other possible combinations (Algorithm 2). A dynamic programming algorithm determines a globally optimal assembly order, along with suitable overhangs, by percolating up through successively larger assemblies the effects of local decisions for individual assembly steps (Algorithm 3).

The following sections deal with each of these three algorithms; a final section discusses the incorporation of additional degrees of freedom (conservative substitution and small shifts of breakpoint location) in order to enable assembly of more diverse families. Throughout, algorithms are described for binary assembly; the generalization to multi-way assembly is straightforward.

### 2.1 Admissible Fragment Overhangs (Algorithm 1)

Each fragment has a single-stranded overhang to allow for specific ligation. We consider here the case of three-nucleotide 5' overhangs, so as to exploit the ability of the SapI restriction enzyme to generate such overhangs in fragments amplified by PCR with primers containing a suitably located SapI recognition sequence (5'-GCTCTTCN/NNNGAAGAGC-3'). Three nucleotides is generally long enough for specific ligation, yet short enough that the overhang nucleotides, which must be the same for all parents, can accommodate some diversity of protein sequences at the breakpoint (see Results, Fig. 2). The use of a 5' overhang allows selection of terminal overhang bases which are on the more discriminatory 3' side of the ligation site [21, 22, 23, 24, 25]. Alternatively, 3' overhangs of three nucleotides or five nucleotides can be generated by BsaXI or BaeI, respectively, and 3' overhangs of any length (with some restrictions on sequence) can be generated from PCR products by incorporating uracil into the primer and digesting the products with uracil DNA glycosylase [26] and endonuclease VIII [27] (USER friendly cloning, New England Biolabs). All of these methods provide the option of PCR amplifying intermediate fragments to allow a complete assembly if yields from earlier ligation steps are low. Overhangs of any length and orientation (5' or 3') can be generated synthetically, although with the loss of the ability to reamplify intermediate fragments. In any case, the choice of overhang length and orientation is not essential for our algorithms. Future versions of our program could even allow overhangs of different length and orientation for different fragments, automatically making the best choices from this enlarged universe.

In the interest of sequence preservation at the chimaeric breakpoints, only a small number of overhangs

---

**Algorithm 1:** Identifying admissible overhangs.

---

**Input:**  $\mathcal{S}$ : set of multiply-aligned parent sequences;  $B$ : sequence of breakpoint positions in the MSA

**Output:**  $\{O_1, O_2, \dots, O_n\}$ : set  $O_i$  has admissible overhangs for fragment  $i$

```
1 for  $i$  (fragment index) from 1 to  $n$  do
2    $O_i \leftarrow \emptyset$ 
3   for  $o \in$  three-nucleotide 5' overhangs do
4     for  $f \in \{0, 1, 2, 3\}$  (frame) do
5       for  $s \in \mathcal{S}$  (parent sequence) do
6         for  $l \in$  codons for residue left of  $B_i$  in  $s$  do
7           for  $r \in$  codons for residue right of  $B_i$  in  $s$  do
8              $p \leftarrow$  concatenation of  $l$  and  $r$ 
9             if  $o$  is  $f..f + 2$  substring of  $p$  then
10              note that parent  $s$  has overhang  $o$  in frame  $f$  using codons  $(l, r)$ 
11          if all parents have overhang  $o$  in frame  $f$  then
12            add  $o$  to  $O_i$ 
```

---

are appropriate for a given fragment. At each breakpoint, an overhang nucleic acid sequence is first sought that will result in no amino acid sequence alterations. That is, all assembled genes (both recombinant and not) preserve the complete sequences of parental fragments at and between all breakpoints. As a less desirable alternative to preserving the original sequence, we can allow amino acids at the breakpoint to change to that of another parent, or more generally to allow conservative (or increasingly diverse) changes to the amino acid sequence (see below). We note that codon changes (whether preserving or changing amino acids) are currently localized to the breakpoints. Future versions of our program will allow the user to either avoid undesirable codons at the breakpoints or simultaneously optimize overhang selection and codon usage for protein expression over the entire gene.

Each fragment has an upstream overhang that is an independent factor in the experiment. Since all parents must have the same overhang at a given breakpoint, we can refer to a fragment and its upstream overhang without regard to parent, and we denote fragment  $i$ 's upstream overhang by  $o_i$ . There is also a downstream overhang, but it is dependent on the upstream fragment of the next fragment; it is the reverse complement,  $o_{i+1}^{rc}$ . All sequences are written 5' to 3', so that, for example, if  $o_{i+1}$  is GAT, then  $o_{i+1}^{rc}$  is ATC. Our description here focuses on the internal fragments; some of the conditions below can be ignored for the N-terminal and C-terminal fragments which have only one overhang.

As already discussed, the overhangs to be employed in an assembly plan must be capable of reproducing all parent sequences, as well as generating all possible chimaeric combinations. We call such overhangs *admissible*, and determine overhang admissibility as an initial step prior to consideration of any particular assembly tree structure. We note that overhangs are described by their nucleotide sequences, whereas parental and chimaeric sequences are described by their amino acids, and we take advantage of redundancy in the genetic code in computing admissibility. Since parents will generally have residues with similar properties at each position when aligned, the correlation between residue properties and codons [28, 29] in fact aids the search for overhangs that preserve the parental fragment sequences.

Algorithm 1 summarizes the process for the determination of the sets  $O_i$  of admissible overhangs for each fragment  $i$ , from a given multiple sequence alignment (MSA) and breakpoint positions. A breakpoint defines a split between a particular pair of amino acids in each parent. All possible codons for those amino

---

**Algorithm 2:** Calculating legality and overhang score of an assembly step.

---

**Input:**  $o_i, o_{i+1}, o_{i+2}$ : overhangs of consecutive fragments to be assembled**Output:**  $s$ : overhang score or “illegal”

```
1  $s \leftarrow 0$ 
2 foreach undesired ligation do
3   let  $a$  and  $b$  stand for the overhangs involved
4    $m \leftarrow 0$  (number of matches)
5   for  $t$  from 0 to 2 (base-pair index) do
6     if  $a[t]$  and  $b[2 - t]$  are complementary then
7       increment  $m$ 
8       if  $m > 1$  then return “illegal”
9       if  $t = 1$  (internal complementarity) then
10        increment  $s$  by  $0.1 * (\# \text{ H-bonds})$ 
11      else
12        increment  $s$  by  $(\# \text{ H-bonds})$ 
```

---

acids are explored in the search for overhangs (lines 6 and 7). In the nucleic acid sequence surrounding the breakpoint, an overhang can be defined in one of several possible *overhang frames* (analogous to, but distinct from reading frame, which is fixed throughout). Overhang frames define the tuple of nucleotides on the ends of the fragments (Fig. 1, algorithm 1). In the case of three-nucleotide overhangs (line 9), the overhang can cover exactly the codon of the first amino acid (frame 0), the codon of the second amino acid (frame 3), or span nucleotides from the first and second codons (frames 1 and 2). An overhang in a particular frame is admissible if all parents can have the same nucleotide tuple in the same frame (lines 10 and 11). The identification of codons and frame yielding a particular overhang is maintained in a table so that gene sequences may be produced for the generation of PCR primers or for complete synthesis.

## 2.2 Evaluating an Assembly Step (Algorithm 2)

Suppose we are given the overhangs for fragments to be assembled in some step in an assembly plan. How can we evaluate the feasibility and quality of the specified assembly? With a pair of fragments  $i$  and  $i + 1$ , there are four overhangs involved and a total of ten possible ligations (Fig. 1, algorithm 2). We must guard against the possibility of ligation between all pairs except the desired  $o_{i+1}^{\text{rc}}$  and  $o_{i+1}$ . For example, we must make sure that one molecule of fragment  $i$  cannot ligate with another by either the  $o_i$  or  $o_{i+1}^{\text{rc}}$  overhang (as would result from their being palindromes or approximate palindromes), and similarly for fragment  $i + 1$ . Furthermore we must ensure that fragment  $i$  cannot ligate on the downstream side of fragment  $i + 1$  or in the wrong orientation with it. Although not independent, we “double count” the cases of  $o_{i+1}^{\text{rc}}$  with itself, and  $o_{i+1}$  with itself, even though they reflect the same sequence choices, in order to capture the increased likelihood of undesirable ligations that arise independently from each case. As assembly proceeds, an overhang may also appear in more than one ligation step. Each ligation step is counted independently, consistent with an overhang’s appearance in separate ligation reactions.

Algorithm 2 calculates the *overhang score* of a binary assembly step involving four total (three independent) overhangs on both ends of the two fragments (Fig. 1, algorithm 2). The overhang score for an individual assembly step is the sum of the pairwise scores for unwanted ligations at that step; it reflects the total likelihood of incorrect assembly at that step. For each unwanted pairwise ligation, we sum up a complementarity score between nucleotide partners. Here, 0 is best, indicating totally different overhangs.

---

**Algorithm 3:** Optimizing a binary assembly plan, selecting tree structure and associated overhangs.

---

**Input:**  $\{O_1, O_2, \dots, O_n\}$ : set  $O_i$  has admissible overhangs for original fragment  $i$  (from Algorithm 1)  
**Output:** Matrix  $m$ : entry  $m_{i,j}$  has a map  $O_i \times O_{j+1} \rightarrow (\mathbb{N}, \mathbb{N}, [0, \infty))$  giving score (height, # ligations, overhang) for assembling fragment  $i..j$  with specified external overhangs

- 1 create  $m$  with all entries  $m_{i,j}(o_i, o_{j+1}) = \infty$   
(fill  $m$  for leaves)
- 2 **for**  $i$  (original fragment) from 0 to  $n$  **do**
- 3     **for**  $o_i \in O_i$  **do**
- 4         **for**  $o_{i+1} \in O_{i+1}$  **do**
- 5              $m_{i,i}(o_i, o_{i+1}) \leftarrow (0, 0, 0)$
- (fill  $m$  for successively longer assembled fragments)
- 6 **for**  $b$  (# breakpoints in assembled fragment) from 1 to  $n - 1$  **do**
- 7     **for**  $i$  (start of assembled fragment) from 0 to  $n - b - 1$  **do**
- 8          $j \leftarrow i + b$  (end of assembled fragment)
- 9         **for**  $o_i \in O_i$  **do**
- 10             **for**  $o_{j+1} \in O_{j+1}$  **do**
- 11                 **for**  $k$  (end of left subfrag) from  $i$  to  $j - 1$  **do**
- 12                     **for**  $o_{k+1} \in O_{k+1}$  **do**  
                       (compute height  $h$ , # ligations  $l$ , and overhang score  $v$  from values for subfragments)
- 13                          $(h_1, l_1, v_1) \leftarrow m_{i,k}(o_i, o_{k+1})$
- 14                          $(h_2, l_2, v_2) \leftarrow m_{k+1,j}(o_{k+1}, o_{j+1})$
- 15                          $h \leftarrow \max\{h_1, h_2\} + 1$
- 16                          $l \leftarrow l_1 + l_2 + 1$
- 17                          $v \leftarrow v_1 + v_2 + \text{Alg2}(o_i, o_{k+1}, o_{j+1})$
- 18                         **if**  $(h, l, v) < m_{i,j}(o_i, o_{j+1})$  **then**
- 19                              $m_{i,j}(o_i, o_{j+1}) \leftarrow (h, l, v)$

---

There are many ways of evaluating the likelihood of annealing and subsequent ligation and thus assigning scores [21, 22, 23, 24, 25]. In practice, we disallow any assembly step in which there is Watson-Crick base complementarity at more than one position for any undesired ligation (line 8); we call other steps *legal*. In the interest of conservative planning, we have made these criteria more stringent than those described by Tsuge *et al.* [19] in their six-way ligation to form linear concatemers.

Whenever there is single position complementarity, we employ a simple H-bond counting approach for the Watson-Crick base pairs, so that G–C pairs contribute 3 and A–T pairs contribute 2. DNA ligase is much more likely to be inhibited by mismatches on the 3' side of the ligation site than at the penultimate 3' position. Depending on the ligase chosen, initial rates of ligation range from three- to twenty-fold lower for mismatches at the 3' end [25]. Since both ends of the overhang are on the 3' side of the ligation site for one of the strands, complementarity on either end is more to be avoided than internal complementarity, and we downweight internal complementarity 10-fold (lines 9–12).

### 2.3 Optimizing Assembly Tree Structure and Overhang Selection (Algorithm 3)

Given a set of admissible overhangs at breakpoints (Algorithm 1), the experiment planning algorithm must determine an optimal order (tree structure) of ligations and optimal overhang choices that yield legal assem-

bly steps (Algorithm 2). In an ideal binary assembly, chimaeric proteins can be constructed using a complete binary tree. A complete tree assures the fewest levels (parallel ligations), thereby minimizing experimental effort. However, as the diversity of sequences and number of parents increases, the number of admissible overhangs decreases, and it becomes less likely that a complete tree can be developed. Furthermore, complete trees become even less likely in the expansion to multi-way assemblies.

The optimization target includes the height of the tree, the number of internal nodes, and the overhang score. All three factors could be optimized simultaneously, with decisions made by a weighting scheme indicating the relative importance of the different factors. However, we find it useful practically to optimize primarily for tree height, with number of internal nodes and overhang score providing additional guidance in the case of ties. We consider tree height (number of sequential ligations) to be most important, because tree height determines the time required to complete assembly. In addition, each successive ligation at less than 100 percent yield reduces the experimental material and increases the likelihood that extra effort will be required for reamplification. The total number of internal nodes (total number of ligation reactions) is next most important to us, as it represents the experimental effort. For strictly binary assemblies, the total number of nodes is fixed ( $n - 1$  assembly steps for  $n$  fragments), but when multi-way assemblies are considered, optimizing both height and number of ligations produces more efficient experimental plans. Finally, overhang score is important in evaluating the relative quality of ligation efficiency in different plans. By enforcing a baseline standard, as we do in disallowing overhangs with more than one complementary base, we avoid assemblies for which the risk of an unwanted ligation is too great and ensure that each ligation is likely to succeed. Optimizing primarily for overhang score, however, yields unbalanced trees where typically only one fragment is added per node and whose height then approaches the number of fragments.

Rather than enumerating the entire combinatorial set of possibilities for overhang choices and tree structures, the experimental procedure allows us to adopt a dynamic programming algorithm that focuses on only a small part of the entire problem context at each assembly step (Fig. 1, algorithm 3). Each assembly step is conducted in a separate ligation reaction, so the only possible interface between the assembled fragment and others in the tree is at the external overhangs. That is, the overhangs that were used for ligation of previously assembled fragments are now internal and no longer matter. Consequently, we can make a local decision about an assembly step without considering all the combinatorial possibilities for the subassemblies. As illustrated in Fig. 1, the tree structure at a single assembly step is defined by which breakpoint is being ligated. We score the assembled fragment using each possible pair of external overhangs, so that we will later be able to consider further assembly with any choices of overhangs for the neighboring fragments.

More precisely (Algorithm 3), consider an assembled fragment that is composed of original fragments  $i$  through  $j$  (we write this  $i..j$ ). The assembly of this fragment ligates two fragments that, for some breakpoint  $k$  ( $i \leq k < j$ ), are composed respectively of fragments  $i$  through  $k$  and  $k + 1$  through  $j$  (defining the left child and right child, respectively). Optimal substructure for dynamic programming holds, since the best assembly for  $i..j$  must build from the best assemblies for  $i..k$  and  $k + 1..j$  (else we would construct  $i..k$  and  $k + 1..j$  by following some other, more efficient plan). For each pair of external overhangs (lines 9 and 10) for fragment  $i..j$ , we determine the breakpoint  $k$  and internal overhang  $o_{k+1}$  (lines 11 and 12) that provide the best score. We calculate the scores for these choices bottom-up in a dynamic programming style (lines 6–8), considering increasingly larger assemblies. We compute the score for an assembled fragment, for a particular pair of external overhangs and any available choice for the internal overhang, from the score for the assembly step and the previously-computed scores for the sub-fragments (lines 13–17). We fill in matrix  $m$ , indexed by initial and final fragment numbers  $i$  and  $j$ , to map a pair of external overhangs  $o_i$  and  $o_{j+1}$  to the best possible score for assembling that fragment using those overhangs (lines 18 and 19).

As discussed above, the score consists of terms for height, number of ligations, and overhang score. We compare scores lexicographically (line 18), so our algorithm can only guarantee optimality of the overall assembly's height. However, our results (see below) show that the "advice" provided by the other terms tends to push them toward optimal as well. In the Discussion we mention the trade-offs vs. optimizing all terms simultaneously.

We note that in the final assembly step, there are no external overhangs at the N and C termini, so we simply determine the best internal overhang. In practice, back-pointers are preserved to enable reconstruction of the assembly. Through the course of the algorithm, each assembled fragment is tested with all possible internal divisions, and each division uses loops over the three overhang sets, so each assembled fragment requires time  $O(no^3)$ . Since we consider all  $O(n^2)$  possible assembled fragments, the total time complexity is  $O(n^3o^3)$ .

## 2.4 Enabling Planning for More Diverse Libraries: Incorporating Additional Degrees of Freedom

Given the most expansive set of parent sequences and breakpoints, it is not always possible to find a set of admissible overhangs or a valid assembly for them. In such a case, we must adjust the desired goals by either changing the parent sequences or the breakpoints. Failure to find a set of admissible overhangs can be fixed by reducing the set of parent sequences or changing the breakpoints, while failure to find a valid assembly can be fixed in the same way. Rather than complicating the selection of the best experimental plan with an additional set of factors that must be weighed and combined, we handle these adjustments with "outer loops." The outer loops enable the user to run Algorithm 3 with increasingly drastic changes until the best compromise is achieved between eliminating parents and altering the preferred breakpoints.

Since we always search first for admissible overhangs, some multiple sequence alignment and breakpoint combinations fail at this step. In this case, the program selects subsets of parent sequences that yield admissible overhangs. Reducing the number of parents will yield more overhang possibilities at each position, but naturally this will also reduce the diversity of the chimaeric library, so we seek to identify the maximal workable subset. The outer loop algorithm for selecting subsets starts with the entire set of parent sequences; upon failure, it runs separately with each subset leaving out one parent; in subsequent iterations, it takes increasingly smaller subsets. It eventually reports those maximal parent subsets that yield admissible overhangs at all breakpoints. As an option, the user can specify particularly desirable parents that must be included in a useful subset. Wherever this method does not yield a sufficient number of parents, our program identifies additional parents that can be added to the finished plan. These additional parents skip one or more of the lowest level ligations for which they cannot use the overhang employed by the other parents.

A possible additional or alternative approach to the goal of including more parents is to shift breakpoints and/or make amino acid substitutions. These degrees of freedom are controlled by user-defined absolute restrictions. Breakpoint shifting restricts the breakpoints to moving only over a range defined by the user for each breakpoint. Although small shifts in breakpoint location are not predicted to make large differences in the ability of recombinants to make functional chimaeras [10, 18], the user can still minimize this effect by allowing small shifts at first followed by larger ones in succeeding iterations. Amino acid substitution first restricts substitutions only to residues at the same position in another parent. It should be noted that changes of this kind result in absolutely no change in SCHEMA score [10]. More broadly, general conservative substitutions, applied uniformly to all breakpoints, can be chosen. General substitutions are subject to a maximum on the allowed drasticness, with increasingly drastic substitutions potentially allowed by the user in successive iterations. Since the effects of mutation are hard to determine however, we expect that users will allow changes only to other parental sequences or to the most conservative general substitutions.

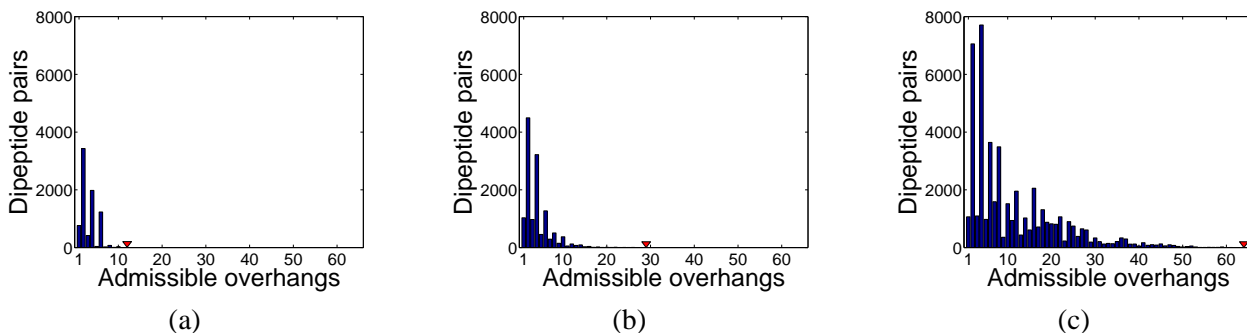


Figure 2: Analysis of the availability of admissible three-nucleotide overhangs. All possible pairs of dipeptides spanning breakpoints for two parents were examined for admissible overhangs (as defined in Materials and Methods) that would allow for recombination. In (a) admissible overhangs were determined using only frame 0 (the frame of the first amino acid) and frame 3 (the frame of the second amino acid). In all, 8,000 dipeptide pairs out of 80,200 can be used for recombination in either frame. These 8,000 occur when both parents share either the first or second amino acid. Panel (b) shows the expansion in both dipeptide pairs with possible recombination breakpoints and the overhangs admissible for them when frames 1 and 2 (mixing nucleotides from both amino acids) are included. Here, a total of 13,250 dipeptide pairs can be used for recombination in at least one of the four possible frames. Panel (c) shows that conservative substitution further expands the set of recombinable dipeptide pairs. Now 47,908 dipeptide pairs have admissible overhangs in one of the frames. The triangles indicate the maximum number of admissible overhangs for any dipeptide pair: 12 for (a), 29 for (b), and 64 for (c).

Choices of breakpoint and substitution satisfying those restrictions are also penalized according to drasticness in the scoring function. Penalties are summed over subassemblies as with overhang score, and are considered secondarily to the height, number of ligations, and overhang score. A shift penalty is assigned to each shift, based on the user’s estimation of anticipated structural disruption. A substitution penalty is evaluated by the difference in score between retaining the wild-type (parent) sequence and making the mutation, according to a user-defined substitution matrix.

### 3 Results

The incorporation of diverse sequences into libraries of chimaeric genes requires that all parents share overhang sequences long enough to ensure efficient ligation of the intended fragments and that the set of overhangs be sufficiently diverse in sequence to prevent coupling in undesirable combinations. To investigate the first requirement of sharing potential sequences for ligation we conducted a survey of possible overhangs among all pairs of amino acids that might span a breakpoint. For simplicity, we assumed a recombination experiment of just two parents, and considered all possible overhangs including those that are approximate palindromes. For each parent there are 400 possible amino acid pairs (dipeptides) that could span a breakpoint. For two parents, there are then 160,000 pairs of dipeptides, although some of these represent the same amino acid combinations. While four hundred of the pairs have the same dipeptide in both parents and have to be retained, half of the 159,600 remaining pairs of dipeptides are redundant (e.g., YK in parent one with FR in parent two, and FR in parent one with YK in parent two). Of the 80,200 unique dipeptide pairs we first determined how many could be ligated with admissible three-nucleotide overhangs in frame 0 (the

frame covering the first amino acid codon) or frame 3 (the frame covering the second amino acid codon). Fig. 2(a) shows a histogram of the number of dipeptide pairs that could be recombined using one or more overhang sequences. Eight thousand or 10% of the unique dipeptide pairs could be recombined using three nucleotide overhangs consisting of frame 0 or 3.

Of course, it is also possible to use overhangs that mix nucleotides from the two codons. Using these two frames, we can cover an additional 5,250 dipeptide pairs. A histogram of admissible overhangs using all four frames is shown in Fig. 2(b). In total, 13,250 dipeptide pairs or approximately 16.5% of the unique dipeptide pairs can be recombined with three nucleotide overhangs in all possible frames. In practice the likelihood that a breakpoint can be recombined will be much higher than this in the formation of libraries using families of related sequences. Using related sequences, amino acids on either side of the breakpoint will be more likely to be shared among parents, and the conservative changes found within families are likely to have codons that are more similar than the random combination assumed here. Nonetheless, the relatively small fraction of admissible overhangs points out the possible need for additional mechanisms for generating useful overhangs by conservative substitution and small shifts to the breakpoint position.

Because of the correlation between amino acid properties and codons [28, 29], it might be thought that not much could be gained by conservative substitution. To evaluate the utility of conservative substitution, we allowed the amino acids of the 80,200 unique pairs to be substituted according to a conservative rule for general substitutions (BLOSUM-62 score change at most 4). As Fig. 2(c) illustrates, conservative substitution nearly quadruples the number of recombinable dipeptide pairs, to 47,908 (about 59.7% of the unique pairs). This greatly increases the number of overhangs potentially usable for constructing assembly plans. Note that by employing conservative substitution and using overhangs spanning codons, the dipeptide pair Val-Ser / Val-Ser potentially admits all 64 possible 3-nucleotide overhangs.

The second requirement in the selection of overhangs is sufficient sequence diversity between overhangs for different fragments to ensure only the correct assembly without undesirable cross-ligation. To investigate this requirement we evaluated the possible combinations of three-nucleotide overhangs in ligations of varying order. Using our criteria stated in Materials and Methods for avoiding incorrect ligation, we first note that of the 64 possible nucleotide triplets, 16 are partially palindromic (e.g., ACT) and thus not suitable for use in ligation. Working with the remaining 48, we found that 67% of all the overhang pairs are sufficiently diverse to satisfy our criteria and avoid incorrect ligation of two fragments in a single step. As the number of overhangs ligated in a single step increases, the number of legal ordered combinations also increases to a maximum at a six-way assembly, but their fraction of the total combinations simultaneously decreases (Fig. 3). No combinations of three-nucleotide overhangs were judged to allow accurate assembly using more than six overhangs. Our example assemblies below employ steps (nodes) that use from three to six overhangs to ligate up to six fragments, while the biochemical feasibility of efficiently ligating six fragments into a linear product has been demonstrated [19].

In a test of their profile SCHEMA theory, Meyer *et al.* [18] constructed a chimaeric library with 13 breakpoints and two parents from the beta-lactamase family and tested lactamase activity in the chimaeras. Their breakpoints and assembly tree were described, but not their overhangs. A straightforward variant of Algorithm 3 follows a fixed tree structure and propagates, bottom-up, scores for feasible overhang choices at the assembly nodes. This method reveals optimal overhang choices for the published tree and breakpoints (Fig. 4). Our dynamic programming planning algorithm further optimizes overhangs and paths to permit multi-way ligations. This optimization allows an alternative assembly (Fig. 5) with a tree height of two (our primary optimization target) using only four ligations total (our secondary target). This plan is substantially more efficient than that previously described, which required a tree height of three and nine ligations.

PurE, an essential enzyme in purine biosynthesis, presents a planning target with a large number of

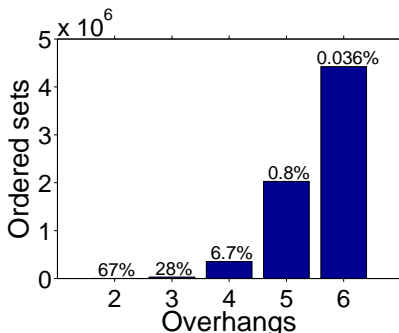


Figure 3: Single-step assembly possibilities employing the sequence diversity in three-nucleotide overhangs. There are 48 overhangs that are legal on their own (not partially palindromic). For  $k$  overhangs to be employed in ligation at a single step, the number of legally ligatable ordered (from upstream to downstream) combinations of overhangs was computed. So that the number of overhangs and the number of fragments is the same, we have assumed that the final fragment is at the C terminus. The percentage of the total number of possible ordered sets ( $48^k$ ) is shown above the bar.

bacterial homologs. To test the ability of our program to extract maximal sets of parents for recombination, we performed a SCHEMA analysis [10] with the available webserver, using PurE sequences from *Sulfolobus solfataricus* and *Methanobrevibacter smithii* and the structure of *E. coli* PurE (PDB id 1QCZ). Based upon the SCHEMA minima, we chose breakpoints in the multiple sequence alignment corresponding to residues Glu40, His75, Met110, and Gln135 in the *E. coli* sequence numbering. These breakpoints yield a similar breakpoint density in the 169-residue PurE to that of the thirteen breakpoints in the 263-residue lactamase. Our program first evaluates which family members can be used, selecting either the maximal subsets or those maximal subsets that use certain required parents. In the case of PurE, our planning mechanism identified several maximal subsets, the largest of which allowed 9 parents to be recombined in a single-step assembly (Fig. 6). Note that even in the strictly conserved second breakpoint, it selects an overhang that spans the two codons, in order to achieve a legal assembly with an optimal overhang score for the five-way ligation.

To test the application of our method to including more diverse parents than in either case thus far, we first constructed an intentionally diverse MSA of beta-lactamase genes for organisms listed in Fig. 7(a). A phylogenetic tree of that alignment is shown in Fig. 7(b). To provide a stringent test of our algorithm's ability to plan the recombination of diverse parents we established a minimal difference criterion, indicated by the vertical line in Fig. 7(b). To eliminate potential parents that are too close in sequence, we chose only one representative from each subfamily whose lineage crosses the vertical line. This selected subset was aligned (see Supplementary Material), a tree constructed (Fig. 7(c)), and analyzed for pairwise sequence identity, as shown. In our planning tests below, we use the same breakpoints as previously, except for the possibility of shifting.

More extensive plans, using additional diverse parents from the lactamase MSA, result in significantly more restrictions on trees and overhangs. Our program's ability to first evaluate parental subsets allows the user to balance the desired number and range of parents versus controlled alteration of breakpoint location and sequence. In the lactamases, our method shows that no assembly is possible for sets of parents including TEM-1, PSE-4, and any additional diverse lactamase, when using fixed sequences and breakpoints.

In many cases the exact location of the breakpoint is not critical and could be shifted by a small number of amino acids, presumably without harm to the experiment (for example, see the broad minima in the

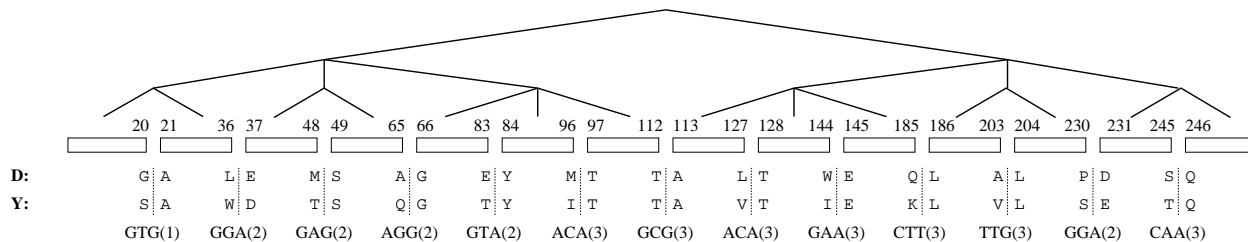


Figure 4: Potential overhangs optimized by our program for beta-lactamase parents TEM-1 (D) and PSE-4 (Y), with breakpoints and assembly fixed as published by Meyer *et al.* [18] (overhangs were not originally described therein). Our selected overhang and the corresponding frame is indicated below each breakpoint. Note that the residue numbers in our figure differ from theirs due to the use of a different sequence alignment.

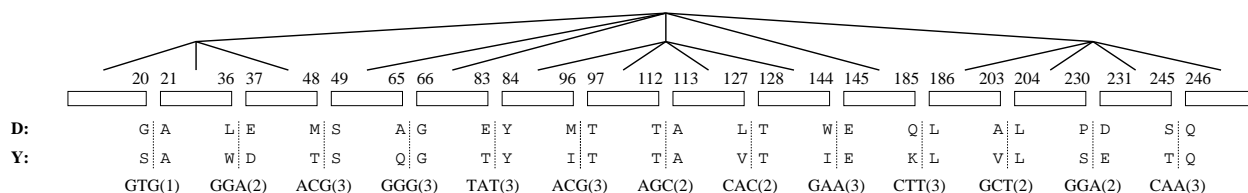


Figure 5: An alternative multi-way experimental plan for TEM-1(D)/PSE-4(Y) selected by our program when optimizing overhangs and assembly, scoring for tree height, number of ligations, and assembly score, as described in Materials and Methods. This plan requires fewer ligations (nodes) and parallel sets of ligations (levels) than the original plan (Fig. 4).

SCHEMA profiles for TEM-1 and PSE-4 [10]). Since the protein families involved in recombination often employ different residues at a breakpoint position, conservative substitution of these residues provides additional degrees of freedom for experiment planning. Conservative sequence changes may be useful either as an alternative to breakpoint shifting or in combination with it. Users can set restrictions for both degrees of freedom, allowing the program to report back larger maximal subsets of parents with admissible overhangs for recombination. Within these restrictions scoring terms reflecting the total extent of change further limit subsequent assembly planning in keeping with the primary criteria (see Materials and Methods).

Our analysis above demonstrated that breakpoint shifting and/or sequence substitution would be required to add additional parents beyond TEM-1 and PSE-4. Restricting breakpoint shifting to plus or minus three residues, only one additional parent (*Rhodobacter capsulatus*) could be incorporated in the plan. Likewise, when allowing changes in BLOSUM-62 score up to 4, the same additional parent could be incorporated.

Using both shifts and substitutions enables planning for a much larger, diverse family. Several maximal subfamilies that include TEM-1 and PSE-4 and up to seven other parents were identified by combining these same shift and substitution limits: {D, T, Q, R, X, G, C, Y, A}; {D, T, Q, R, X, G, M, C, Y}; {D, Q, R, T, M, C, Y, A}; {D, W, T, Q, R, X, M, Y}; {D, U, W, Q, C, Y, A}; {D, U, Q, R, C, Y, A}; {D, W, Q, R, C, Y, A}; {D, U, W, T, Q, Y}; {D, U, Q, R, M, Y}. Fig. 8 shows an assembly and overhangs for the first subfamily. The optimal tree employs 12 shifted breakpoints and 20 substitutions in one or more parents. This extensive combination of substitution and breakpoint shifting allows the assembly of chimaeras from a much larger number of diverse parents.

Our algorithm employs dynamic programming to select optimal solutions from an enormous number

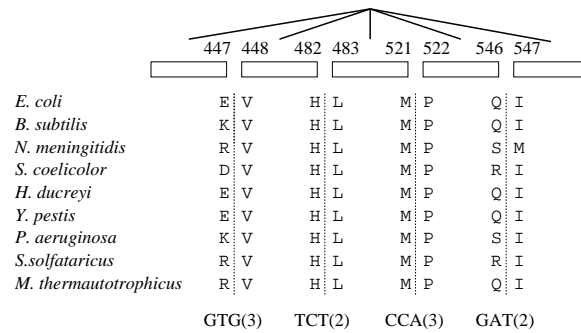


Figure 6: Experimental plan for recombining a maximal subset of PurE genes at breakpoints defined by SCHEMA minima. Our mechanism determined a single-step assembly plan, selecting among the possible overhangs and minimizing tree height, number of ligations, and assembly overhang score as described in Materials and Methods.

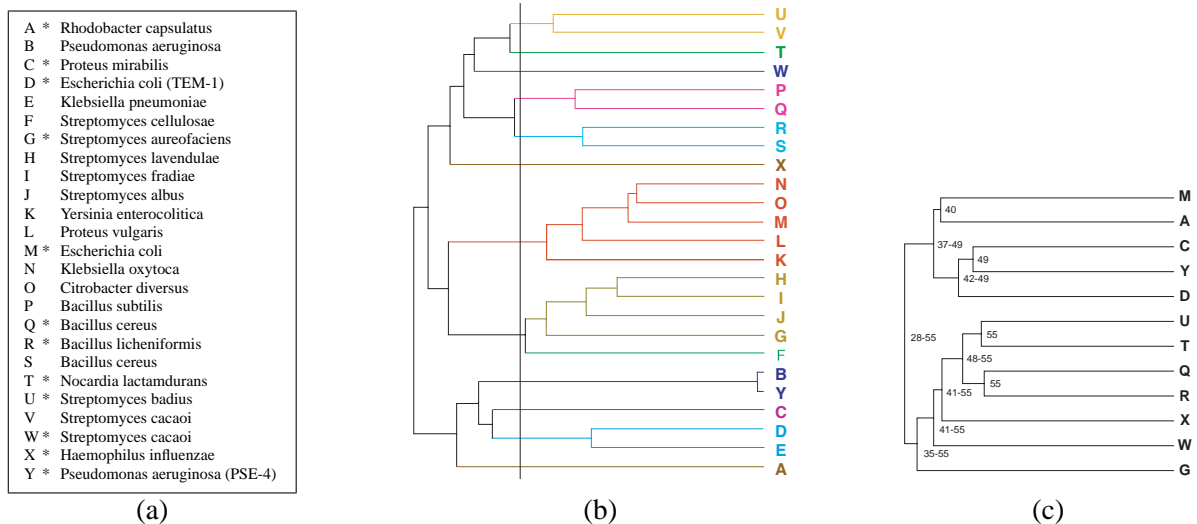


Figure 7: Sequence relationships among members of the beta-lactamase family. (a) Species of beta-lactamase family homologs employed in the multiple sequence alignment and assemblies, and their letter codes. (b) Phylogenetic tree for selected members of the beta-lactamase family. Vertical line indicates similarity threshold. (c) Phylogenetic tree for diverse subset of (b). The range of pairwise sequence identity for all members branching from a node is indicated.

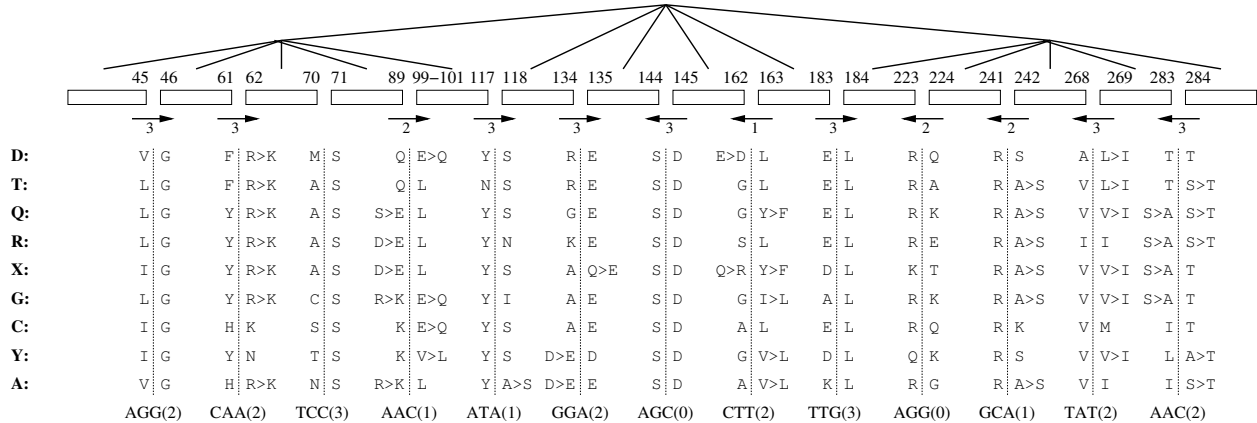


Figure 8: Experimental plan for recombining the maximal subset of beta-lactamase genes that includes TEM-1 and PSE-4, under both conservative amino acid substitutions and shifting the breakpoint location. To increase diversity by incorporating additional parents beyond TEM-1 and PSE-4, both general amino acid substitution up to a difference of four in the BLOSUM-62 table and shifting each breakpoint location up to three positions were allowed. A maximal subset of nine parents was identified during the search for sets of mutually consistent, admissible overhangs. A multi-way assembly tree was then constructed, selecting among the possible overhangs and minimizing tree height, number of ligations, and assembly overhang, substitution and shift scores as described in Materials and Methods.

of possible assembly plans. The number of possible assembly plans (considering choices of overhangs and tree structures) grows very quickly with the number of breakpoints. Let number  $K_i$  represent the number of three-nucleotide overhangs admissible at fragment  $i$ , typically substantially fewer than the absolute bound of  $4^3 = 64$ . For example in the two-parent TEM-1/PSE-4 beta-lactamase case (Fig. 5), there are 13 breakpoints with respectively  $\langle 6, 1, 5, 4, 4, 3, 9, 6, 2, 8, 9, 2, 6 \rangle$  admissible overhangs to be considered. There are a total of  $\prod_i K_i$  combinations of  $n$  overhangs; for TEM-1/PSE-4 there are 134,369,280 combinations. The number of tree structures also grows very quickly. Establishing a tree structure essentially amounts to inserting balanced parentheses in a list, and in fact our algorithm is similar to that for computing optimal parenthesizations [30]. The number of binary tree structures (binary parenthesizations) with  $n$  leaves is given by the Catalan number  $C_{n-1}$ , where  $C_n = \frac{(2n)!}{(n+1)!n!}$ . There are 14 possible binary trees for 5 leaves, 4862 for 10 leaves, and 742,900 for the 14-fragment (13-breakpoint) TEM-1/PSE-4 case. If we allow multi-way assemblies, the count follows the “super-Catalan” numbers, with 45 possibilities for 5 leaves, more than 103,000 for 10, and more than 71 million for 14 [31, 32]. The total number of assembly plans to be considered is thus the product of the number of overhang combinations and the number of tree structures; for TEM-1/PSE-4 the total is about  $10^{14}$  for binary assemblies and  $10^{16}$  for multi-way assemblies. Thus the plan shown in Fig. 5 represents one optimal solution out of  $10^{16}$  possibilities. Thus automated selection of assembly paths and overhangs provides substantial advantages in evaluating the combinatorial possibilities of the problem.

While the number of trees is thus clearly too large to consider exhaustively, good plans might still be frequent enough to be identified without recourse to our algorithm. (The usefulness of our algorithms would not be so great if it were possible to find a good enough plan without them.) To test this possibility, we exhaustively enumerated all binary assembly plans for the first eight TEM-1/PSE-4 breakpoints (as many as was feasible in a reasonable time). There are a total of  $(6 \times 1 \times 5 \times 4 \times 4 \times 3 \times 9 \times 6) \times 1430 = 111,196,800$

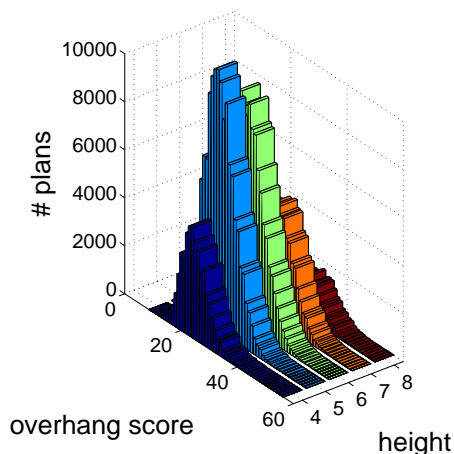


Figure 9: Histogram of height and overhang scores of legal binary assembly plans for assembling the first eight TEM-1/PSE-4 breakpoints. No plan is better in height, and the vast majority of plans are significantly worse in overhang score than the one identified by our algorithm (not shown, height of 4, overhang score of 13.8). Note that the legal plans shown here represent only 0.4% of the total combinations of tree structures and overhang selections.

possible assembly plans, of which only 447,928 (0.4%) are legal. Fig. 9 shows the distribution of scores of the legal plans. Of these, only 47,428 (11%) achieve the optimal height of 4. The majority require more levels — 137,688 require 5 levels, 142,088 require 6 levels, 89,300 require 7 levels, and 31,424 require 8 levels. Thus a person is unlikely to randomly find a legal plan, and even less likely to find an optimal one, without applying these (or similar) algorithms. In addition to guaranteeing optimality of height, our mechanism also does quite well in finding a plan with a small overhang score. The overhang score of our selected binary plan (not shown) is 13.8, close to the best (9.6) and better than all but 94 (0.2%) of the 4-level plans. Thus even a legal, height-optimal experiment discovered without our planner is likely to have poorer specificity of ligations, and consequently greater potential for experimental difficulties.

Our dynamic programming algorithm requires time polynomial in the number of breakpoints and their admissible overhangs ( $O(n^3 o^3)$  for  $n$  fragments and  $o$  overhangs). The effect of number of overhang choices on the computational complexity is apparent in the increasingly complex planning problems studied here. Our program required less than a second for the no-shifting/no-substitution plan with the two beta-lactamases in Fig. 5 (a total of  $10^8$  admissible overhang combinations), but 20 hours for substitution and shifting with the larger family in Fig. 8 ( $10^{17}$  combinations). (All runs were performed on a Linux box with a 3.2 GHz Pentium 4 processor, with Java code compiled by the Sun standard edition software development kit.)

## 4 Discussion

Protein engineering has evolved in several distinct directions. One approach seeks *de novo* design of proteins by optimizing sequence for a given structure [33], sampling structures for a given sequence [34], or both [35, 36]. Another approach focuses on structure-based design or redesign of particular regions, employing various methods, notably variations on dead-end elimination [37, 38, 39]. This method has also

achieved good results in splicing functions onto previously known cores [40, 41]. Protein engineering based on the generation of combinatorial libraries using stochastic or site-directed breakpoints followed by screening has also proven an effective method for the generation of novel activities [1, 2, 3, 4, 5, 6, 7, 8, 10, 17, 18]. Combinatorial libraries have previously been limited by lack of diversity in the parents, inability to direct the sites of recombination towards the greatest effect, and computational difficulties in planning for large numbers of breakpoints. Our algorithms address these limitations by providing a mechanism for planning site-directed combinatorial construction using short overhangs for annealing and ligation. By incorporating suitable degrees of freedom, these assemblies can be used to build complete combinatorial libraries from diverse parents, recombined at user-selected breakpoints. While we focus on the generation of large combinatorial libraries with multiple breakpoints by SPLISO, running our algorithm with just a single breakpoint allows the generation of oligos for SCOPE and dimer templates for RM-PCR.

At the present state of the art, construction of chimaeric genes by any method may involve ten or more recombination breakpoints from as many parents, yielding potential libraries with on the order of  $10^{11}$  members. The libraries designed here have more than  $10^{13}$  members arising from parents that have no more than 55% pairwise sequence identity. Engineering of novel properties and activities may require the generation of still greater diversity, though. Incorporation of additional parents can aid in achieving such diversity. Our program can also insert additional diverse parents into an existing plan, allowing them to skip breakpoints for which they do not support the corresponding overhang. More generally, the supported parents could be considered throughout experiment planning in future implementations; e.g., the dynamic programming algorithm could be extended to propagate the set of supported parents, and a scoring function could evaluate the size of the supported set (and consider the number of skipped breakpoints) when selecting among alternative tree structures and overhangs.

As an additional source of diversity, multiple rounds of recombination and selection [1, 42, 43] can also be accomplished in our method. Using PCR and either restriction enzymes or the USER friendly method, fragments with either the original breakpoints or new ones can be recovered in their approximate relative amounts in the selected pool. These fragments can be generated with suitably designed ends and reassembled as our planning algorithm directs. Using a nested set of PCR primers or synthetic fragments, our method also allows the addition or deletion of codons at the breakpoints, generating variable linkage diversity which proved valuable in the generation of chimaeras from the X-family of DNA polymerases [7].

Although the ligation steps in our assembly pathways may involve up to six fragments and appear quite complex, the work of Tsuge *et al.* [19] demonstrates the viability of such multi-way ligations for generating linear products. In ligating five fragments, each containing a resistance gene, plus a sixth vector fragment, Tsuge *et al.* found that 98 percent of cells transformed from the ligation mix possess the five resistances that would arise from accurately ligated products. The biochemical efficiency of such ligation was also demonstrated by the production of linear concatemers as much as five-fold longer than a single set of six fragments. We note that Tsuge *et al.* conducted their six-way ligation with several overhangs that would not pass our criteria of having only one Watson-Crick base pair per overhang. Assuming that all their overhangs were legal by our definition, the total hydrogen bond-based overhang score for their six-way assembly step according to our metric would be 81.0. By way of comparison, our computer planned six-way assembly step in Fig. 5 has a better score of only 41.8.

Different efficiencies of ligation with different overhangs may bias the ligation process. Under their ligation conditions, Tsuge *et al.* found that overhangs employing two AT and one GC base pair were desirable. Our method could readily incorporate such additional restrictions, while still optimizing overhangs under these restrictions. Finally, it is worth noting that avoiding bias in ligation may also require employing precisely equimolar amounts of all fragments [19].

We have decided to optimize primarily on tree height, as it represents the experimental time. Secondary factors consisting of number of ligations and overhang score are used locally (i.e., at individual nodes) to break ties. Certainly other methods of scoring are possible and may prove advantageous as experience develops. For example, our method could employ as its scoring term a weighted combination of criteria, but this requires a more sophisticated understanding of the experimental cost and alterations in efficacy due to these factors. A variation on the dynamic programming algorithm could guarantee optimality of all three terms, by extending the matrix to include indices for the height and number of ligations. Thus we would compute the optimal overhang score for each fragment, using each pair of external overhangs, and assembled for each height and number of ligations. This would scale up the computational complexity by two additional factors, and we expect it to be practically unsuitable for complex plans testing many overhangs. Our results (Fig. 9) show that we can avoid this expense and still do very well at reducing the number of ligations and the overhang score. To reduce the “brittleness” of identifying a single optimum with respect to the current (or any) scoring model, the algorithm could readily be extended to compute a set of alternative trees with optimal or near-optimal score. At each step, it would keep a list of choices that yield a number of the best scores; at the end, it would back-chain through these best choices. The user would then be able to choose among the output trees.

Various refinements to the individual scoring components might also be advantageous. A more refined substitution score might model all the amino acid changes that would occur in the library as a result of the substitution within the context of the known structural environment of that residue. We can also refine the breakpoint shift penalty with criteria reflecting expected reductions in the resulting modularity. Finally, the overhang score was developed from a general knowledge of ligation efficiencies and fidelity [21, 22, 23, 24, 25]. Nonetheless, any overhang score is only an attempt to predict an actual ligation efficiency, which may vary for reasons that are not yet obvious. As experience develops, the overhang score could be improved to assure efficiency, fidelity, and lack of bias in the ligation process.

We have developed a method that allows planning of site-directed recombination experiments under varying degrees of freedom in the choice of parent genes, breakpoint locations, amino acid substitutions, and assembly pathway. Our dynamic programming approach is computationally efficient and yields globally optimal plans. We have demonstrated that even diverse members of a protein family can be recombined under suitable combinations of substitution and shifting of the breakpoint positions. By enabling site-directed recombination from diverse families, the SPLISO method extends the repertoire and specificity of protein engineering beyond that available to stochastic methods. It should be a boon for investigating the modular nature of proteins and for protein engineering based upon rational selection of modular units.

## **5 Supporting Materials**

Our algorithm has been implemented in platform-independent Java code. The software can be freely obtained for academic use by request from the authors.

## **6 Acknowledgments**

We gratefully acknowledge support for this work from the following sources: Liz Saftalov, a Ruzicka Summer Research Fellowship from the College of Science at Purdue University; Peter Smith, an undergraduate research fellowship from an undergraduate initiative grant from the Howard Hughes Medical Institute to the Department of Biological Sciences at Purdue University; Chris Bailey-Kellogg, an NSF CAREER award (IIS-0444544); and Chris Bailey-Kellogg and Alan M. Friedman, a grant from NSF SEIII (IIS-0502801).

We thank V. Jo Davisson and Vishal Nashine for sequence alignment and SCHEMA minima of the PurE family. We also thank an anonymous reviewer for the idea of supplementing experiment plans with parents that skip some ligation steps.

## References

- [1] Stemmer WPC. Rapid evolution of a protein *in vitro* by DNA shuffling. *Nature* 1994; 370:389–391.
- [2] Ostermeier M, Shim JH, Benkovic SJ. A combinatorial approach to hybrid enzymes independent of DNA homology. *Nature Biotech* 1999; 17:1205–9.
- [3] Lutz S, Ostermeier M, Moore GL, Maranas CD, Benkovic SJ. Creating multiple-crossover DNA libraries independent of sequence identity. *PNAS* 2001; 98:11248–53.
- [4] Aguinaldo AM, Arnold FH. Staggered extension process (StEP) *in vitro* recombination. *Methods Mol Biol* 2003; 231:105–110.
- [5] Coco WM. RACHITT: Gene family shuffling by random chimeragenesis on transient templates. *Methods Mol Biol* 2003; 231:111–127.
- [6] Sieber V, Martinez CA, Arnold FH. Libraries of hybrid proteins from distantly related sequences. *Nat Biotechnol* 2001; 19:456–60.
- [7] O’Maille PE, Bakhtina M, Tsai MD. Structure-based combinatorial protein engineering (SCOPE). *J Mol Biol* 2002; 321:677–691.
- [8] Tsuji T, Onimaru M, Yanagawa H. Random multi-recombinant PCR for the construction of combinatorial protein libraries. *Nucleic Acids Res* 2001; 29:E97.
- [9] Hiraga K, Arnold FH. General method for sequence-independent site-directed chimeragenesis. *J Mol Biol* 2003; 330:287–296.
- [10] Voigt CA, Martinez C, Wang ZG, Mayo SL, Arnold FH. Protein building blocks preserved by recombination. *Nature Struct Biol* 2002; 9:553–558.
- [11] Endelman JB, Silberg JJ, Wang ZG, Arnold FH. Site-directed protein recombination as a shortest-path problem. *Protein Eng Des Sel* 2004; 17:589–94.
- [12] Tsai CJ, Maizel Jr JV, Nussinov R. Anatomy of protein structures: visualizing how a one-dimensional protein chain folds into a three-dimensional shape. *PNAS* 2000; 97:12038–43.
- [13] Ye X, Friedman AM, Bailey-Kellogg C. Hypergraph model of multi-residue interactions in proteins: Sequentially-constrained partitioning algorithms for optimization of site-directed protein recombination. In: 10th International Conference on Research in Computational Molecular Biology (RECOMB). To appear.
- [14] Lockless SW, Ranganathan R. Evolutionarily conserved pathways of energetic connectivity in protein families. *Science* 1999; 286:295–299.
- [15] Thomas J, Ramakrishnan N, Bailey-Kellogg C. Graphical models of residue coupling in protein families. In: 5th ACM SIGKDD Workshop on Data Mining in Bioinformatics (BIOKDD).

- [16] Joern JM, Meinhold P, Arnold FH. Analysis of shuffled gene libraries. *J Mol Biol* 2002; 316:643–56.
- [17] Horton RM, Hunt HD, Ho SN, Pullen JK, Pease LR. Engineering hybrid genes without the use of restriction enzymes: gene splicing by overlap extension. *Gene* 1989; 77:61–68.
- [18] Meyer MM, Silberg JJ, Voigt CA, Endelman JB, Mayo SL, Wang ZG, Arnold FH. Library analysis of SCHEMA-guided protein recombination. *Protein Sci* 2003; 12:1686–93.
- [19] Tsuge K, Matsui K, Itaya M. One step assembly of multiple DNA fragments with a designed order and orientation in *Bacillus subtilis* plasmid. *Nucleic Acids Res* 2003; 31:e133.
- [20] Chothia C, Lesk AM. The relation between the divergence of sequence and structure in proteins. *EMBO J* 1996; 5:823–6.
- [21] Wu DY, Wallace RB. Specificity of the nick-closing activity of bacteriophage T4 DNA ligase. *Gene* 1989; 76:245–54.
- [22] Husain I, Tomkinson AE, Burkhart WA, Moyer MB, Ramos W, Mackey ZB, Besterman JM, Chen J. Purification and characterization of DNA ligase III from bovine testes. Homology with DNA ligase II and vaccinia DNA ligase. *J Biol Chem* 1995; 270:9683–90.
- [23] Shuman S. Vaccinia virus DNA ligase: specificity, fidelity, and inhibition. *Biochemistry* 1995; 34:16138–47.
- [24] Luo J, Bergstrom DE, Barany F. Improving the fidelity of *Thermus thermophilus* DNA ligase. *Nucleic Acids Res* 1996; 24:3071–8.
- [25] Tong J, Cao W, Barany F. Biochemical properties of a high fidelity DNA ligase from *Thermus* species AK16D. *Nucleic Acids Res* 1999; 27:788–94.
- [26] Lindahl T, Ljungquist S, Siegert W, Nyberg B, Sperens B. DNA N-glycosidases: properties of uracil-DNA glycosidase from *Escherichia coli*. *J Biol Chem* 1977; 252:3286–94.
- [27] Melamede RJ, Hatahet Z, Kow YW, Ide H, Wallace SS. Isolation and characterization of endonuclease VIII from *Escherichia coli*. *Biochemistry* 1994; 33:1255–64.
- [28] Sjolstrom M, Wold S. A multivariate study of the relationship between the genetic code and the physical-chemical properties of amino acids. *J Mol Evol* 1985; 22:272–277.
- [29] Trinquier G, Sanejouand YH. Which effective property of amino acids is best preserved by the genetic code? *Protein Eng* 1998; 11:153–169.
- [30] Cormen TH, Leiserson CE, Rivest RL, Stein C. *Introduction to Algorithms*. MIT Press, 2 edition, 2001.
- [31] Weisstein EW, et al. Mathworld — a Wolfram web resource. <http://mathworld.wolfram.com/>, 2004.
- [32] Sloane NJA. The on-line encyclopedia of integer sequences. <http://www.research.att.com/~njas/sequences/>, 2004.

- [33] Reina J, Lacroix E, Hobson SD, Fernandez-Ballester G, Rybin V, Schwab MS, Serrano L, Gonzalez C. Computer-aided design of a PDZ domain to recognize new target sequences. *Nat Struct Biol* 2002; 9:621–7.
- [34] Harbury PB, Piecs JJ, Tidor B, Alber T, Kim PS. High-resolution protein design with backbone freedom. *Science* 1998; 282:1462–7.
- [35] Desjarlais JR, Handel TM. De novo design of the hydrophobic core of proteins. *Protein Sci* 1995; 4:2006–18.
- [36] Kuhlman B, Dantas G, Ireton GC, Varani G, Stoddard B, Baker D. Design of a novel globular protein fold with atomic-level accuracy. *Science* 2003; 302:1364–8.
- [37] Desmet J, Maeyer M, Hazes B, Lasters I. The dead-end elimination theorem and its use in protein side-chain positioning. *Nature* 1992; 356:539–542.
- [38] Dahiyat BI, Mayo SL. Protein design automation. *Protein Sci* 1996; 5:895–903.
- [39] Looger LL, Hellinga HW. Generalized dead-end elimination algorithms make large-scale protein side-chain structure prediction tractable: implications for protein design and structural genomics. *J Mol Biol* 2001; 307:429–45.
- [40] Dahiyat BI, Mayo SL. De novo protein design: fully automated sequence selection. *Science* 1997; 278:82–7.
- [41] Looger L, Dwyer M, Smith J, Hellinga H. Computational design of receptor and sensor proteins with novel functions. *Nature* 2003; 423:185–190.
- [42] Zhang JH, Dawes G, Stemmer WP. Directed evolution of a fucosidase from a galactosidase by DNA shuffling and screening. *PNAS* 1997; 94:4504–4509.
- [43] Rothman SC, Kirsch JF. How does an enzyme evolved in vitro compare to naturally occurring homologs possessing the targeted function? Tyrosine aminotransferase from aspartate aminotransferase. *J Mol Biol* 2003; 327:593–608.