Optimization Algorithms for Site-directed Protein Recombination Experiment Planning

Wei Zheng

Dartmouth College

Follow this and additional works at: https://digitalcommons.dartmouth.edu/dissertations

Part of the Computer Sciences Commons

Recommended Citation
https://digitalcommons.dartmouth.edu/dissertations/29

This Thesis (Ph.D.) is brought to you for free and open access by the Theses and Dissertations at Dartmouth Digital Commons. It has been accepted for inclusion in Dartmouth College Ph.D Dissertations by an authorized administrator of Dartmouth Digital Commons. For more information, please contact dartmouthdigitalcommons@groups.dartmouth.edu.
Optimization Algorithms for Site-directed Protein Recombination Experiment Planning

A Thesis
Submitted to the Faculty
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in
Computer Science by
Wei Zheng
DARTMOUTH COLLEGE
Hanover, New Hampshire
June 2010

Examining Committee:

__________________________
(chair) Chris Bailey-Kellogg

__________________________
Devin Balkcom

__________________________
Lisa K. Fleischer

__________________________
Karl E. Griswold

__________________________
Brian W. Pogue, PhD.
Dean of Graduate Studies

Dartmouth Computer Science Technical Report TR2010-672
Abstract

Site-directed protein recombination produces improved and novel protein variants by recombining sequence fragments from parent proteins. The resulting hybrids accumulate multiple mutations that have been evolutionarily accepted together. Subsequent screening or selection identifies hybrids with desirable characteristics. In order to increase the “hit rate” of good variants, this thesis develops experiment planning algorithms to optimize protein recombination experiments. First, to improve the frequency of generating novel hybrids, a metric is developed to assess the diversity among hybrids and parent proteins. Dynamic programming algorithms are then created to optimize the selection of breakpoint locations according to this metric. Second, the trade-off between diversity and stability in recombination experiment planning is studied, recognizing that diversity requires changes from parent proteins, which may also disrupt important residue interactions necessary for protein stability. Accordingly, methods based on dynamic programming are developed to provide combined optimization of diversity and stability, finding optimal breakpoints such that no other experiment plan has better performance in both aspects simultaneously. Third, in order to support protein recombination with heterogeneous structures and focus on functionally important regions, a general framework for protein fragment swapping is developed. Differentiating source and target parents, and swappable regions within them, fragment swapping enables asymmetric, selective site-directed recombination. Two applications of protein fragment swapping are studied. In order to generate hybrids inheriting functionalities from both source and target proteins by fragment swapping, a method based
on integer programming selects optimal swapping fragments to maximize the predicted stability and activity of hybrids in the resulting library. In another application, human source protein fragments are swapped into therapeutic exogenous target protein to minimize the occurrence of peptides that trigger immune response. A dynamic programming method is developed to optimize fragment selection for both humanity and functionality, resulting in therapeutically active variants with decreased immunogenicity.
Acknowledgments

In China, there’s an old saying “Of a hundred miles journey, the last ten miles is the most difficult”. If I were just by myself, I wouldn’t have been able to finish the last ten miles of this thesis work. Fortunately I received help and support from many people, which made my dream come true.

First and foremost I would like to express my deep and sincere gratitude to my Ph.D. advisor, Prof. Chris Bailey-Kellogg. I will always be thankful for his wisdom, knowledge and insight that strengthened this thesis work significantly. From his patience and insistence, I came to know that the procedure is more important than the result. Because of his continuous encouragement and inspiration, my self-confidence was gradually built up. I really appreciate his understanding and kindness during my Ph.D. studies. It has been a great honor to work with him and this experience will be a significant presence in my life forever.

I wish to express my sincere thanks to Prof. Alan M. Friedman and Prof. Karl E. Griswold for their careful guidance, kind help and stimulating discussions in our collaboration.

I would like to thank the rest of my thesis committee, Prof. Devin Balkcom and Prof. Lisa K. Fleischer, for their understanding, support and insightful comments.
I am grateful to all friends in CBK lab. This lab is a source of friendship as well as good advice and collaboration. The members that I have had the pleasure to work with and want to thank are: Dr. Shobha Potluri, Dr. Xiaoduan Ye, Dr. John Thomas, Dr. Fei Xiong, Dr. Jason Vertrees, Himanshu Chandola, Bornika Ghosh, Andrew Scott Parker, Lu He and Tuobin Wang.

I would like to thank my parents Kailiang Zheng and Deyun Wang. They bore me, raised me, supported me, and loved me. They taught me by their example to be a person with integrity and kindness.

I also wish to thank my parents-in-law Beihai Tan and Hexiu Jiang, for their understanding and support during the thesis work.

Last but not the least, I want to thank my wife Wenfang, and our child Andrew. Your patience, love and encouragement have upheld me, particularly in those many days in which I spent more time with my computer than with you. It would take more than a thesis to express my deep love for you.
## Contents

1  Introduction                               1

2  Algorithms for Selecting Breakpoint Locations to Optimize Diversity in Protein Engineering by Site-directed Protein Recombination  9

   2.1  Introduction .......................................................... 10

   2.2  Methods .................................................................. 13

      2.2.1  Library Diversity ......................................................... 14

      2.2.2  Metrics for Breakpoint Selection ................................. 16

      2.2.3  Dynamic Programming for Breakpoint Selection ............... 17

   2.3  Results and Discussion ................................. 22


   3.1  Introduction .......................................................... 29

   3.2  Methods ............................................................... 34

      3.2.1  Metrics ............................................................. 35
3.2.2 Finding Undominated Breakpoint Sets on the Convex Hull 36
3.2.3 Finding Locally Undominated Breakpoint Sets in Concavities 41
3.3 Optimality Guarantees 43
3.4 Results and Discussion 46
  3.4.1 PurE Family 46
  3.4.2 Beta-lactamase Family 54

4 Protein fragment swapping: A method for asymmetric, selective site-directed recombination 58
  4.1 Introduction 59
  4.2 Methods 62
    4.2.1 Swappable regions 64
    4.2.2 Potential score 66
    4.2.3 Fragment selection 70
  4.3 Results and Discussion 75
    4.3.1 Selective swapping of beta-lactamases 75
    4.3.2 Activity swapping in glutathione transferases 80
    4.3.3 Activity swapping in purEs 83
  4.4 Conclusion 84
  4.5 NP-hardness of Protein Fragment Swapping 86

5 Protein Deimmunization by Fragment Swapping 92
  5.1 Introduction 93
5.2 Methods ................................................................. 97
   5.2.1 Epitope Score ....................................................... 97
   5.2.2 Conservation Score .............................................. 99
   5.2.3 Optimization of Fragment Swapping Libraries ................. 101
   5.2.4 Swapping Optimization by Dynamic Programming ............. 104
5.3 Results ................................................................. 106
   5.3.1 Deimmunization of mouse anti-CD30 antibody AC10 VL .......... 106
   5.3.2 Deimmunization of glutathione transferases ..................... 108
5.4 Conclusion ........................................................... 114

6 Summary and Future Work ............................................ 115
   6.1 Summary ............................................................. 115
       6.1.1 Optimization in Symmetric and Exhaustive Protein Recombination 116
       6.1.2 Optimization in Asymmetric and Selective Protein Recombination 117
   6.2 Future Work ......................................................... 118
       6.2.1 Sub-optimal Experiment Planning .............................. 118
       6.2.2 Enhancements of Protein Deimmunization ...................... 119
       6.2.3 Recombination Experiment Data Interpretation ................ 121
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Comparison of STAVERSITY with RAND and IMPLICIT for three different parent sets in purE family and from 2 to 10 breakpoints.</td>
<td>51</td>
</tr>
<tr>
<td>3.2</td>
<td>Comparison of STAVERSITY with RAND and IMPLICIT for parents TEM-1 and PSE-4, with 2 to 10 breakpoints.</td>
<td>56</td>
</tr>
<tr>
<td>4.1</td>
<td>Family statistics for different types of residue pairs.</td>
<td>88</td>
</tr>
<tr>
<td>4.2</td>
<td>Average potential scores for different types of residue pairs.</td>
<td>88</td>
</tr>
</tbody>
</table>
List of Figures

1.1 Symmetric and exhaustive site-directed protein recombination with two parent protein and 2 breakpoints, generating a library of 8 hybrids. 4

1.2 Asymmetric and selective site-directed protein recombination, protein fragment swapping with 3 fragments, generating a library of 8 hybrids. 5

2.1 Diversity optimization in site-directed protein recombination to maximize the possibility for novel characteristics that are not found in the parent proteins. 12

2.2 Optimal substructure for diversity variance minimization by dynamic programming. 18

2.3 Optimal breakpoint selections by diversity optimization for Escherichia coli, Gallus gallus and Methanothermobacter thermautotrophicus in purE family. 24

2.4 Distribution of diversity variance for random breakpoint selections compared with optimal selection. 25

2.5 Effects of parent protein diversity on diversity optimization. 27
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.6</td>
<td>Relationship among diversity metrics: H-P diversity variance, H-H diversity variance and sum-min mutation level.</td>
<td>27</td>
</tr>
<tr>
<td>3.1</td>
<td>Stability and diversity are two competing criteria for selecting breakpoint locations for site-directed recombination.</td>
<td>33</td>
</tr>
<tr>
<td>3.2</td>
<td>Diversity variance and perturbation for completely enumerated 2-breakpoint sets for three purE proteins <em>E. coli</em>, <em>G. gallus</em>, and <em>M. thermautotrophicus</em>.</td>
<td>33</td>
</tr>
<tr>
<td>3.3</td>
<td>Procedure of finding undominated breakpoint sets on the convex hull.</td>
<td>38</td>
</tr>
<tr>
<td>3.4</td>
<td>Algorithm for finding undominated breakpoint sets on the convex hull.</td>
<td>39</td>
</tr>
<tr>
<td>3.5</td>
<td>Algorithm for finding undominated breakpoint sets in concavities.</td>
<td>43</td>
</tr>
<tr>
<td>3.6</td>
<td>Optimality guarantee for breakpoint sets found by <strong>STAVERSITY</strong>.</td>
<td>45</td>
</tr>
<tr>
<td>3.7</td>
<td>Breakpoint sets found by <strong>STAVERSITY</strong> for purE parent proteins <em>Escherichia coli</em>, <em>Gallus gallus</em> and <em>Methanothermobacter thermautotrophicus</em>.</td>
<td>49</td>
</tr>
<tr>
<td>3.8</td>
<td>Optimality guarantees for breakpoint sets found by <strong>STAVERSITY</strong> for purE parent proteins <em>Escherichia coli</em>, <em>Gallus gallus</em> and <em>Methanothermobacter thermautotrophicus</em></td>
<td>53</td>
</tr>
<tr>
<td>3.9</td>
<td>Breakpoint sets found by <strong>STAVERSITY</strong> for beta-lactamases parent proteins TEM-1 and PSE-4.</td>
<td>55</td>
</tr>
<tr>
<td>3.10</td>
<td>Optimality guarantees for breakpoint sets found by <strong>STAVERSITY</strong> for beta-lactamases parent proteins TEM-1 and PSE-4.</td>
<td>57</td>
</tr>
<tr>
<td>4.1</td>
<td>Overview of protein fragment swapping method.</td>
<td>64</td>
</tr>
</tbody>
</table>
4.2 Patterns of potential score contribution in swapping from single residue and pair of residues. ................................. 70

4.3 Comparison of average potential score of libraries constructed by optimal fragment selection and random selections. ............................. 77

4.4 For proteins from beta-lactamases, glutathione transferases and purE family, optimal swapping plans relative to the reference structures. ....... 79

4.5 Three-fragment swapping optimization for beta-lactamases trading off potential score vs. diversity. ................................................. 80

4.6 Three-fragment swapping optimization for glutathione transferases trading off potential score vs. diversity. ................................. 81

4.7 Three-fragment swapping optimization for glutathione transferases with different values of weight $\alpha$ for source and target potential contribution. ... 82

4.8 Three-fragment swapping optimization for purEs trading off potential score vs. diversity. ................................................................. 85

4.9 Three-fragment swapping optimization for purEs with different values of weight $\alpha$ for source and target potential contribution. ............... 85

4.10 Optimization of fragment swapping with pair-wise terms is NP-hard through a reduction from MAX-2SAT. ........................................... 88

5.1 Deimmunization by fragment swapping for aligned target and source sequences. ................................................................. 95

5.2 Encoding of residue swapping states. ................................................. 103
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3</td>
<td>Optimal substructure of fragment swapping optimization by dynamic programming.</td>
<td>104</td>
</tr>
<tr>
<td>5.4</td>
<td>Fragment swapping optimization for deimmunization of mouse anti-CD30 antibody AC10 VL.</td>
<td>109</td>
</tr>
<tr>
<td>5.5</td>
<td>Conservation and epitope scores for hybrids constructed by optimal fragment selection, wild-type AC10 VL, and deimmunized variants in [46].</td>
<td>110</td>
</tr>
<tr>
<td>5.6</td>
<td>Deimmunization of θ-class glutathione transferases rGSTT2-2 by fragment swapping from hGSTT1-1.</td>
<td>112</td>
</tr>
<tr>
<td>5.7</td>
<td>Conservation and epitope scores for hybrids constructed by optimal fragment selection, along with rGSTT2-2 and SCR23.</td>
<td>113</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

Even after millions of years of evolution, existing proteins represent only an extremely small fraction of the possible amino acid sequences. Protein engineering aims to create new amino acid sequences encoding proteins with desired characteristics, such as improved or novel function. Protein engineering generally adopts two strategies, rational design and directed evolution.

In rational design, scientists use detailed knowledge of the structure and function of a protein to make desired changes (such as site-directed mutagenesis) or design new proteins from scratch (protein design) [29, 6, 13, 30]. However, such detailed knowledge is often unavailable, and models of the relationships among sequence, structure and function are poor, particularly regarding the prediction of the effects of sequence mutations on structure and function.

Protein engineering by directed evolution mimics the two mechanisms of natural evolution. Random mutagenesis generates diversity and is followed by a selection regime to pick out variants with the desired qualities [98, 73]. Further rounds of mutation and selection can then be applied to improve or alter the characteristics of the engineered variants. However the additional diversity generated by an increasing number of random mutations comes at
a significant cost in the fraction of functional variants, which declines exponentially with the number of random mutations [73, 31].

Recombination is another approach for directed evolution. It can incorporate mutations without significantly sacrificing the fraction of folded variants. Recombination of proteins sharing the same fold employs only mutations that are compatible with the backbone characteristics of the fold. Thus mutations introduced by recombination are less likely to disturb the protein structure than are random mutations. DNA shuffling is a method for the reassembly of genes from their random DNA fragments by repeated cycles of annealing, and has been used to generate proteins with properties that are due to the action of several mutations working synergistically [85, 11]. However, DNA shuffling experiments produce variants that are usually not significantly different from their parents [3] and parental genes are recovered more frequently from the reaction as the number of potential crossover sites decrease, resulting in fewer DNA fragments for reassembly. Homology-independent techniques have been developed that can recombine sequences of any identity [67, 66, 51, 81, 37]. However these techniques often result in a population of hybrids that are mostly unfolded [54, 65], since they incorporate a significant number of frame shifts, deletions, and insertions due to the random generation of recombination sites in the DNA sequence. Furthermore, additional diversity introduced by recombining distantly related sequences comes with a decrease in the fraction of folded variants as recombining more diverse sequences results in an increased number of potentially deleterious pairwise interactions [54, 65, 15].

Site-directed protein recombination [92, 54, 68, 76] seeks to circumvent many of the
problems in both homologous recombination and homology-independent recombination methods by choosing specific sequence fragments of the parental genes, and reassembling these fragments combinatorially to create a library of variants. Here the key point is how to choose fragments for recombination. Several different computational energy functions have been developed to evaluate in silico the probability of hybrids being folded and functional [88, 55, 52, 82, 38, 22, 96]. Based on such predictions, site-directed protein recombination can be planned computationally to improve the efficiency and effectiveness of the experiments, hopefully improving the quality of the constructed hybrid library [92, 56, 17, 96].

This thesis studies computational experiment planning in site-directed protein recombination, considering three significant objectives of protein engineering: “novelty”, “stability” and “immunogenicity”. To maximize the probability of generating protein variants as expected, optimization algorithms are developed according to two types of site-directed protein recombination: symmetric, exhaustive recombination and asymmetric, selective recombination.

In symmetric, exhaustive site-directed recombination, it is assumed by homology that parent proteins share common structure. For recombination, parent proteins contribute sequence fragments equally, including all residues in parent sequences as shown in Fig. 1.1. For optimization in experiment planning, beginning with novelty, we first developed algorithms to optimize diversity among hybrids and between hybrids and parents. As stability and diversity constitute two competing aspects in protein engineering, we then developed methods to provide combined optimization for both aspects.
Fig. 1.1: Symmetric and exhaustive site-directed protein recombination with two parent protein and 2 breakpoints, generating a library of 8 hybrids.

In order to enable recombination of parents without overall structural homology and focus on interested parent regions, we then developed a general framework for a new approach to site-directed protein recombination — protein fragment swapping. Parents contribute differently in roles of source and target proteins, in which the target protein is taken as a framework for fragment swapping, and only residues in swappable regions participate in recombination (Fig. 1.2). Two applications of protein fragment swapping are developed and studied. In functionality recombination by fragment swapping, the goal is to generate protein variants inheriting different functionalities from both source and target proteins. A method based on integer programming is developed to select optimal fragments, maximizing a weighted potential score capturing the sequence constraints of both source and target. In another application of fragment swapping, protein humanization, fragments from a human source protein are selected and swapped into an exogenous therapeutic protein. In optimizing swapping fragment selection, our objective is to minimize epitope occurrences in the resulting hybrid library, thereby reducing their immunogenicity while preserving functionally and structurally important amino acids. A dynamic programming method is developed to simultaneously optimize both of these two aspects. Following are brief descriptions for these studies; the details are explained in Chapter 2, Chapter 3, Chapter 4 and
Chapter 5, and previous publications [103, 99, 101].

![Diagram of asymmetric and selective site-directed protein recombination](image)

**Fig. 1.2:** Asymmetric and selective site-directed protein recombination, protein fragment swapping with 3 fragments, generating a library of 8 hybrids.

**Diversity Optimization for Hybrid Library**  In previous works of symmetric, exhaustive site-directed recombination, breakpoint selection algorithms have been provided to optimize library stability according to different energy functions [56, 69, 96]. However, they are mainly focused on maximization of the stability of hybrids generated; the novelty of the resulting hybrid library is much less explored. We defined *diversity variance* metrics to evaluate novelty in terms of both the diversity among hybrids and the diversity between hybrids and parents. By optimizing the diversity of the hybrid library, we aim to distribute hybrids uniformly in protein sequence space and increase the probability to obtain novel protein variants. We developed polynomial-time dynamic programming algorithms to select optimal breakpoint locations, minimizing diversity variance.

**Combined Optimization of Stability and Diversity**  In experiment planning for protein recombination, diversity and stability are two competing aspects: diversity requires changes from parent proteins, which may disrupt important residue interactions necessary for protein stability. Optimization of only one of these aspects will consequently overlook the other [17, 53, 99]. In practical recombination experiments, both stability and diversity
should be considered in order to obtain folded protein variants with improved functions. We developed STAVERSITY, the first method to explicitly optimize both stability and diversity in planning site-directed recombination experiments. Based on dynamic programming algorithms, STAVERSITY finds the optimal breakpoint sets, for which no other breakpoint set is better in both stability and diversity. The identified breakpoint sets enable a protein engineer to understand and account for the trade-offs between these two key considerations in combinatorial recombination.

**Functionality Recombination by Protein Fragment Swapping** In traditional experiments of site-directed protein recombination, parent proteins share a common structure and contribute combinatorial blocks equally in library construction. However some applications in protein engineering may require recombination among parent proteins without structural homology. Accordingly, we developed a new method of site-directed protein recombination — protein fragment swapping. Protein fragment swapping implements recombination within swappable regions of parent proteins, which can be from locally aligned parent structures or functionally important sequence regions. Instead of equal contributions, parent proteins in fragment swapping are treated differently as “source” and “target” proteins, with the objective of introducing functionality from the source parent into the target parent. Protein fragment swapping can focus the experimental effort on a smaller portion of sequence space and have a better chance of finding beneficial new hybrids in the resulting library. It provides a general method for site-directed hybrid library construction, of which traditional combinatorial recombination is in fact a special case.
In functionality recombination by fragment swapping, protein hybrids are constructed to inherit different activities from source and target proteins. By optimal fragment selection, the objective is to maximize a weighted combination of source and target potential score contribution according to different experiment considerations. We proved that potential score maximization is NP-hard when a pair-wise potential score is included. In order to find optimal swapping fragments, we developed a method based on integer programming. The significance of optimization in functionality recombination is demonstrated by comparison to random fragment selections and hybrids constructed by symmetric, exhaustive recombination.

**Therapeutic Protein Deimmunization by Fragment Swapping** Therapeutic proteins may be recognized as exogenous agents by the immune system, triggering immune responses and causing loss of effectiveness. As human sequence fragments tend to be less immunogenic (or not immunogenic at all), they are selected and swapped into corresponding fragments in therapeutic proteins, in order to minimize the occurrence of epitopes, small peptide sequence fragments that help to cause an immune system response. At the same time, functionally and structurally important amino acids must be maintained so as to preserve stability and desired properties. Accordingly a dynamic programming algorithm is developed to optimize these two aspects simultaneously in fragment selection, providing optimal trade-offs under different experiment requirements. Experiment planning for practical exogenous proteins demonstrates the efficiency and effectiveness of therapeutic protein deimmunization by fragment swapping. In comparison with protein variants gen-
erated in previous work, deimmunized hybrids constructed by optimal fragment swapping improve in both immunogenicity and activity, providing improved candidates in therapeutic protein deimmunization.
2. ALGORITHMS FOR SELECTING BREAKPOINT LOCATIONS TO OPTIMIZE DIVERSITY IN PROTEIN ENGINEERING BY SITE-DIRECTED PROTEIN RECOMBINATION

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. This work has been published in [103].
2.1 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 [41, 50, 49, 48, 20]. 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. 2.1) 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 [92, 54, 68, 76]. In contrast to stochastic library construction methods [85, 1, 10], site-directed approaches choose breakpoint locations to optimize expected library quality, e.g., predicted disruption [54, 17, 96]. 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 biophys-
ical 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 [92, 56, 69, 77, 96]. 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. 2.1, left. If we put the breakpoint between the 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. 2.1, right).

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 [69]. A random mutant library of TEM-1 β-lactamase with a minimal mutation load (8.2 mutations/gene)
Fig. 2.1: 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.
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 [98]. 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) [12]. 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.

2.2 Methods

We are given a set of \( n \) parent sequences \( P = \{P_1, P_2, \ldots, P_n\} \), forming a multiple sequence alignment with each sequence of length \( l \) including residues and gaps. Our goal is to select a set of \( \lambda \) breakpoint locations \( X = \{x_1, x_2, \ldots, x_\lambda \mid 1 \leq x_1 < x_2 < \ldots < x_\lambda = l\} \). 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 \( \lambda \) 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 $\mathcal{H}(P, X) = \{H_1, H_2, \ldots, H_n\}$ includes all combinations. Our goal is to choose $X$ (such that $|X| = \lambda$ and $x_\lambda = l$) to optimize the diversity of library $\mathcal{H}(P, X)$, for a set $P$ of parents.

### 2.2.1 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:

$$m(S, S') = \sum_{1 \leq r \leq l} I\{S[r] \neq S'[r]\}, \quad (2.1)$$

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

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 \( 1 \leq r \leq l \). Over the set of \( n^\lambda \) hybrids, there must be \( n^{\lambda-1} \) instances of \( P_1[r], n^{\lambda-1} \) of \( P_2[r], \ldots, \) and \( n^{\lambda-1} \) of \( P_n[r] \). Thus we have

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

\[
= n^{\lambda-1} \times \sum_{a=1}^{n} m(P_a, H_i).
\]  \hspace{1cm} \text{ (2.2)}

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

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

\[
= n^{2(\lambda-1)} \times \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} m(P_a, P_b),
\]  \hspace{1cm} \text{ (2.3)}

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

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

\[
= n^{\lambda-1} \times \sum_{b=1}^{n} m(P_a, P_b).
\]  \hspace{1cm} \text{ (2.4)}

Claim 2.3 follows immediately from Claim 2.2. \( \square \)

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.
2.2.2 Metrics for Breakpoint Selection

Intuitively (Fig. 2.1, 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:

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

(2.5)

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

Expanding the square in Eq. (2.5) yields an $m(H_i, H_j)^2$ term, a constant $\bar{m}^2$ term, and an $\bar{m} \times m(H_i, H_j)$ 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 $\lambda$, choose a set $X$ of $\lambda$ breakpoints (with $x_\lambda = l$) to minimize the hybrid-hybrid “variance” $v_{HH}(X)$ of the resulting library, where

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

(2.6)
for \( H_i, H_j \in \mathcal{H}(P, X) \).

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 \( P \) of \( l \) residues and a positive integer \( \lambda \), choose a set \( X \) of \( \lambda \) breakpoints (where \( x_\lambda = l \)) to minimize the hybrid-parent “variance” \( v_{HP}(X) \) of the resulting library, where

\[
v_{HP}(X) = \sum_{i=1}^{n} \sum_{a=1}^{n} m(H_i, P_a)^2
\]

(2.7)

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

Intuitively (Fig. 2.1, 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 \( \lambda \) breakpoints,

\[
n^{\lambda - 2} < \frac{v_{HH}(X)}{v_{HP}(X)} \leq n^{\lambda - 1},
\]

(2.8)

which is an algebraic manipulation of the terms. This relationships means that the two criteria should be highly correlated, as our results below confirm.

### 2.2.3 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. 2.2 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 \( x_{k-1} \).

\[
H(P, X') \{ \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} m(P_a[1, r], P_b[1, r])^2 \} \quad \text{if } k = 1,
\]
\[
\min_{r' < r} \{ n^2 \times d(r', k - 1) + e_{HH}(k, r, r') \} \quad \text{if } k > 1,
\]

**H-H Diversity Optimization.** We use this insight to devise a dynamic programming recurrence to compute the optimal value of \( v_{HH} \) for the \( k \)th 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

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

Fig. 2.2: Library substructure: library \( \mathcal{H}(P, X) \) ending at position \( r \) extends library \( \mathcal{H}'(P, X') \) ending at position \( r' \) by adding each parent fragment \( P_a[r'+1, r] \) to each hybrid \( H'_i \) in \( \mathcal{H}'(P, X') \).
where \( e_{HH} \) is defined in Eq. (2.10).

**Proof:** As discussed above, the hybrid library \( \mathcal{H}(P, X) \) is extended from \( \mathcal{H}(P, X') \), where \( X' \) is missing the final breakpoint in \( X \). Let us use \( H_i \) for the members of \( \mathcal{H}(P, X) \) and \( H'_i \) for those of \( \mathcal{H}(P, X') \), and “+” to denote sequence concatenation. Following the structure in Fig. 2.2, we can separate \( v_{HH} \) into terms \( \mathcal{H}(P, X')+P_a[r'+1, r] \) from hybrids in a single “sub-library” sharing the same added fragments, and terms \( \mathcal{H}(P, X') + P_a[r'+1, r] \) and \( \mathcal{H}(P, X')+P_b[r'+1, r] \) between separate “sub-libraries” with distinct added fragments. This gives Eq. (2.11).

Expanding the second term on the right-hand side in Eq. (2.11) gives Eq. (2.12).

By Claim 2.1 for parents with \( k-1 \) breakpoints (and thus truncated at \( r' \)), we have Eq. (2.13).

We can substitute twice the right-hand side of Eq. (2.13) into the third term in Eq. (2.12) (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. (2.12) and (2.13) are independent. We can then substitute the resulting formula back into Eq. (2.11). Simplification yields Eq. (2.14), 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. (2.11) and twice \( \binom{n}{2} \) from the first term in Eq. (2.12) (with “twice” again due to all vs. all distinct). Because Eq. (2.14) only depends on \( r' \) and not the previous breakpoints,

\[
d(r, k) = \min_{r'<r} \{ n^2 \times d(r', k-1) + e_{HH}(k, r, r') \}. \tag{2.9}
\]

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 $O(\lambda n^2 t^2)$.

\begin{align*}
e_{HH}(k, r, r') &= 4n^{2(k-2)} \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} m(P_a[1, r'], P_b[1, r']) \\
&\quad \times \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} m(P_a[r' + 1, r], P_b[r' + 1, r]) \\
&\quad + n^{2(k-1)} \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} m(P_a[r' + 1, r], P_b[r' + 1, r])^2 \quad (2.10)
\end{align*}

\begin{align*}
\sum_{i=1}^{n} \sum_{j=i+1}^{n} m(H_i, H_j)^2 &= \\
&= \sum_{a=1}^{n} \sum_{i=1}^{n} \sum_{j=i+1}^{n} m(H_i' + P_a[r' + 1, r], H_j' + P_a[r' + 1, r])^2 \\
&\quad + \sum_{a=1}^{n} \sum_{b=a+1}^{n} \left( \sum_{i=1}^{n} \sum_{j=1}^{n} m(H_i' + P_a[r' + 1, r], H_j' + P_b[r' + 1, r])^2 \right) \quad (2.11)
\end{align*}

\begin{align*}
\sum_{a=1}^{n} \sum_{b=a+1}^{n} \left( \sum_{i=1}^{n} \sum_{j=1}^{n} m(H_i' + P_a[r' + 1, r], H_j' + P_b[r' + 1, r])^2 \right) &= \\
&= \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} \left( \sum_{i=1}^{n} \sum_{j=1}^{n} m(H_i', H_j')^2 \right) \\
&\quad + \sum_{a=1}^{n} \sum_{b=a+1}^{n} \left( \sum_{i=1}^{n} \sum_{j=1}^{n} m(P_a[r' + 1, r], P_b[r' + 1, r])^2 \right) \\
&\quad + \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} \left( \sum_{i=1}^{n} \sum_{j=1}^{n} 2m(H_i', H_j') \times m(P_a[r' + 1, r], P_b[r' + 1, r]) \right) \quad (2.12)
\end{align*}

\begin{align*}
\sum_{i=1}^{n} \sum_{j=i+1}^{n} m(H_i', H_j') &= \sum_{i=1}^{n-1} \sum_{b=a+1}^{n} \left( \sum_{a=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{n} m(P_a[1, r'], P_b[1, r']) \right). \quad (2.13)
\end{align*}
\[ n^{k-1} \sum_{i=1}^{n} \sum_{j=i+1}^{n} m(H_i, H_j)^2 = n^2 \times \sum_{i=1}^{n^{k-1}-1} \sum_{j=i+1}^{n^{k-1}} m(H'_i, H'_j)^2 + e_{HH}(k, r, r'). \quad (2.14) \]

**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

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

where \( e_{HP} \) is defined in Eq. 2.16.

**Proof:** The proof is similar to that for H-H diversity. By partitioning the library, we have Eq. (2.17).

By Claim 2.2 for parents with \( k - 1 \) breakpoints truncated at position \( r' \), we have Eq. (2.18).

Substituting the right-hand side of Eq. (2.18) into the third term in Eq. (2.17), and simplifying, we get Eq. (2.19). Here \( e_{HP}(k, r, r') \) also depends only on \( r' \) and not the preceding breakpoints, so we have

\[ d(r, k) = \min_{r' < r} \{ n \times d(r', k - 1) + e_{HP}(k, r, r') \}. \quad (2.15) \]

The table size and time to fill in each entry are the same as with H-H diversity. \( \square \)
\begin{align*}
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.
\end{align*}

(2.16)

\begin{align*}
\sum_{a=1}^{n} \sum_{i=1}^{n} m(H_i, P_a[1, r])^2 \\
= \sum_{a=1}^{n} \sum_{b=1}^{n} \sum_{i=1}^{n} (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} m(H_i', P_a[1, r'])^2 \\
+ \sum_{a=1}^{n} \sum_{b=1}^{n} \sum_{i=1}^{n} m(P_b[r'+1, r], P_a[r'+1, r])^2 \\
+ \sum_{a=1}^{n} \sum_{b=1}^{n} \sum_{i=1}^{n} 2m(H_i', P_a[1, r']) \times m(P_b[r'+1, r], P_a[r'+1, r]).
\end{align*}

(2.17)

\begin{align*}
\sum_{i=1}^{n} m(H_i', P_a[1, r']) &= n^{k-2} \times \sum_{b=1}^{n} m(P_a[1, r'], P_b[1, r']). \\
\sum_{a=1}^{n} \sum_{i=1}^{n} m(H_i, P_a[1, r])^2 &= n \times \sum_{a=1}^{n} \sum_{i=1}^{n} m(H_i', P_a[1, r'])^2 + e_{HP}(k, r, r').
\end{align*}

(2.18)

\(2.19\)

\textbf{2.3 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 \textit{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 [18, 89]. 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. 2.2.2) 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. 2.3).

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
Fig. 2.3: 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 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 ran-
dom sets are plotted in Fig. 2.4.

![Fig. 2.4](image)

Fig. 2.4: Distribution of diversity variance for random breakpoint selection compared with opti-
mal breakpoint selection. 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 di-
amonds indicate diversity values for breakpoints selected by our algorithm: $9.63 \times 10^7$ for H-H,
$2.39 \times 10^6$ 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 [17]. 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. 2.5 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. (2.8) and confirmed empirically in Fig. 2.3, 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. 2.6(a,c) shows the relationship among these values, using the same random breakpoint selections as in Fig. 2.4. Optimization for hybrid-parent diversity also achieves good diversity according to the other two metrics. Fig. 2.6(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. 2.5. These correlations allow us to do just one polynomial-time diversity optimization, achieving three goals simultaneously.
Fig. 2.5: Effect of parent protein diversity 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.

Fig. 2.6: Relationship among diversity metrics: H-P diversity variance, H-H diversity variance and sum-min mutation level. (a,c): Correlation over random four-breakpoint sets with the fixed three-parent set of Fig. 2.4. 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. 2.5. Note that the histograms are focused on a small region very near 1 and −1, respectively.
In engineering protein variants by constructing and screening combinatorial libraries of chimeric proteins, two complementary and competing goals are desired: the new proteins must be similar enough to the evolutionarily-selected wild-type proteins to be stably folded, and they must be different enough to display functional variation. We present here the first method, \textsc{Staversity}, to simultaneously optimize stability and diversity in selecting sets of breakpoint locations for site-directed recombination. Our goal is to uncover all “undominated” breakpoint sets, for which no other breakpoint set is better in both factors. Our first algorithm finds the undominated sets serving as the vertices of the lower envelope of the two-dimensional (stability and diversity) convex hull containing all possible breakpoint sets. Our second algorithm identifies additional breakpoint sets in the concavities that are either undominated or dominated only by undiscovered breakpoint sets within a distance bound computed by the algorithm. Both algorithms are efficient, requiring only time polynomial in the numbers of residues and breakpoints, while characterizing a space defined by an exponential number of possible breakpoint sets. We applied \textsc{Staversity} to iden-
tify breakpoint plans for different sets of parent proteins taken from the purE family and the beta-lactamase family. The quality of our plans is guaranteed by the calculation of the lower bound for optimal plans. Our plans dominate most of the plans found by other possible approaches, random sampling or explicit optimization for stability with implicit optimization for diversity. The identified breakpoint sets provide a compact representation of good plans, enabling a protein engineer to understand and account for the trade-offs between two key considerations in combinatorial chimeragenesis. This work has been published in [100].

3.1 Introduction

Protein engineering by site-directed recombination (Fig. 3.1) generates libraries of hybrid proteins (or “chimeras”) by mimicking the mixing and inheritance that occur in natural reproduction. A set of homologous parent genes are recombined at defined breakpoint locations, yielding a combinatorial set of hybrids [92, 54, 68, 76]. In contrast to stochastic library construction methods [85, 1, 10], site-directed approaches explicitly choose breakpoint locations to optimize expected library quality, e.g., predicted disruption [54, 17, 96] or library diversity [103]. In contrast to mutagenesis, the mutations introduced by site-directed recombination are known to be compatible with each other in parent proteins with a similar structural context (due to homology), and are thus expected to be less disruptive. Without requiring precise modeling or prediction of the effects of mutation, site-directed recombination can produce variant proteins with improved properties and activities [54, 68, 53, 43].

There are two competing goals in recombination experiment planning. We want the
resulting hybrids to be stably folded, which is easiest to achieve if they are just like wild-type proteins. At the same time, we want the hybrids to have different activity, which of course requires that they be different from wild-type. By construction, site-directed recombination preserves single-position conservation statistics, since each residue position in the hybrid library is simply taken from one of the parents, and all parents are equally represented within the combinatorial library. Thus evaluation of stability typically focuses on correlation statistics between interacting residues [92, 56, 69, 77, 96]. The key insight (middle of Fig. 3.1) is that recombination “perturbs” the distributions of amino acid types for interacting residues, thereby potentially disrupting the interactions underlying stable folding. Models of residue correlation have been shown to capture important information in a number of applications, including prediction of free energy changes caused by hydrophobic core mutations [8], prediction and recognition of native-like protein structure [40], and functional classification of members of protein families [89]. Pairwise [92] and higher-order [96] models have been used in algorithms to plan site-directed recombination experiments minimizing perturbation (and thereby maximizing expected stability), and have led to the creation of variant proteins with improved or novel activities [54, 68, 53, 43].

In addition to stable hybrids, we also want a diverse hybrid library in order to obtain hybrids with improved or novel activities. Under various methodologies and on a number of systems, including cytochromes P450 [69], beta-lactamases [98], and single chain Fv antibodies [12], functional change from wild-type has been correlated with the number of mutations in protein variants. Earlier work on site-directed recombination optimizes for stability while indirectly forcing diversity by constraining the minimum fragment length [69].
Our recent work developed the first approach to explicitly optimize for diversity, by finding breakpoint locations that sample protein sequence space relatively uniformly [103], as shown in the right of Fig. 3.1. However, since diversity competes with stability, it is desirable to explicitly consider both criteria simultaneously.

This work presents Staversity (a hybrid word with both “stability” and “diversity”), the first method to explicitly optimize both stability and diversity in planning site-directed recombination experiments. A set of breakpoints defines a hybrid library that can be evaluated by metrics we call “perturbation” and “diversity variance” (see again Fig. 3.1). We seek to minimize perturbation as a way to ensure stable hybrids, and we seek to minimize diversity variance as a way to evenly spread out hybrids in sequence space. Using diversity variance and perturbation values as two dimensions, we can consider possible breakpoint sets as points in a two-dimensional space (Fig. 3.2). Since it is difficult for an experimenter to decide a priori upon the “best” combination of these two incommensurate factors, our methods provide insights into the trade-offs by finding undominated sets of breakpoints—those for which no other set of breakpoints is better for both factors (“Pareto optimal”, in economics jargon). Our goal is to find the undominated sets efficiently, without explicitly enumerating the exponential number of possible plans.

The problem of finding optimal trade-offs between competing desired criteria is a common one. For example, in considering how to segment records in a large shared database, Eisner and Severance studied the optimal trade-off between the cost of storage and the benefit of retrieval [16]. The goal was assumed to be either a linear or non-linear combination of cost and benefit, and parametric analysis was applied to find the optimal trade-off under
all possible parameter values. In computational biology, such ideas are also at the heart of a parametric approach to sequence alignment, e.g., trading off match scores and gap penalties [93]. A comprehensive analysis of parametric sequence alignment [27] showed that for both global and local alignment, the number of parametric regions to be considered is bounded, and that fast algorithms [26] can be employed to perform the alignment.

In the present case, there is no underlying notion of an optimal trade-off; instead, we want to provide the experimenter with an overview of all possibilities worth considering (because they are undominated). We prove that the results of convex optimization (breakpoint sets on the lower convex hull) are undominated, and develop the natural polynomial-time algorithm to find those breakpoint sets (similar to the parametric analysis in [16, 93, 27]). We also develop a polynomial-time algorithm to uncover many of the breakpoint sets in the concavities which are either undominated or can be shown by the algorithm to be within a small distance of any undiscovered set that would dominate them.

We present planning results for cases with from 2 to 10 breakpoints and three different sets of parents from the purE family of biosynthetic enzymes that we are currently studying by site-directed recombination. We also present results for two parents from the beta-lactamase family, the subject of previous recombination experiments. Overall, our plans can be proved to be quite good—the average normalized distance between our plans and the lower bound on optimal plans is around 2 percent. Other possible methods (either sampling breakpoint sets randomly, or explicitly optimizing for stability while implicitly optimizing for diversity) don’t do nearly as well—on average for each parent set, our plans dominate 60–90% of those.
Fig. 3.1: Stability and diversity are two competing criteria for selecting breakpoint locations for site-directed recombination. Given a family of homologous proteins (here just a cartoon fragment of a multiple sequence alignment), we select a small number of parents to be recombined at specific locations, generating a library of hybrids. Middle: recombination may disrupt the previously observed correlations between amino acid types for interacting residues, here at positions 1 and 8, thereby possibly affecting stability. Our *perturbation* metric evaluates the overall change in correlation statistics, as an indicator of stability. Right: the resulting library may have more or less diversity; the choice of breakpoint location on the top yields hybrids that are identical to the parents, while that on the bottom results in sequence space being sampled relatively uniformly, with an equal number of mutations between each hybrid and each parent. Our *diversity variance* metric evaluates the overall differences in numbers of mutations between different parent-hybrid pairs, as an indicator of diversity.

Fig. 3.2: Diversity variance (*x*-axis) and perturbation (*y*-axis) for completely enumerated 2-breakpoint sets for three purE proteins *E. coli*, *G. gallus*, and *M. thermautotrophicus*. Blue dots are all possible breakpoint sets (for larger numbers of breakpoints, we would not be able to enumerate them all); red diamonds (to the lower left, minimizing one objective for a fixed value of the other) are the undominated ones.
3.2 Methods

Let $P = \{P_1, P_2, \ldots, P_n\}$ represent a multiple sequence alignment of $n$ parent proteins, with each sequence of length $l$ including residues and gaps. A recombination experiment with $\lambda$ breakpoints is defined by a set of breakpoint locations $X = \{x_1, x_2, \ldots, x_\lambda\}$, $1 \leq x_1 < x_2 < \ldots < x_\lambda < l$. The breakpoints partition each parent $P_a$ into $\lambda + 1$ fragments with sequences $P_a[1, x_1], P_a[x_1 + 1, x_2], \ldots, P_a[x_\lambda, l]$, where in general we use $P_a[r, r']$ to denote the amino acid string from position $r$ to $r'$ in sequence $P_a$, and $P_a[r]$ 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 $\mathcal{H}(P, X) = \{H_1, H_2, \ldots, H_{n^\lambda+1}\}$ includes all combinations.

Given a breakpoint set $X$, we can evaluate the perturbation and diversity variance of the resulting hybrid library with metrics $v_p(X)$ and $v_d(X)$ (see below). We assume, without loss of generality, that both $v_d$ and $v_p$ are to be minimized. For two breakpoint sets $X$ and $X'$, if $v_p(X) \leq v_p(X')$ and $v_d(X) \leq v_d(X')$, and one of the two inequalities is strict, we say that $X'$ is dominated by $X$. Let $\mathcal{X}_\lambda$ be the set of all possible $\lambda$-breakpoint sets. If for some breakpoint set $X$ there is no $X' \in \mathcal{X}_\lambda$ that dominates it, we say that $X$ is undominated. If $X$ is not dominated by any $X' \in \mathcal{X}_\lambda'$ for some subset $\mathcal{X}_\lambda' \subset \mathcal{X}_\lambda$ of possible breakpoint sets, we say that $X$ is locally undominated. Our goal is then:

**Goal:** Given parent proteins $\mathcal{P}$ and number of breakpoints $\lambda$, find the set $\mathcal{U}_\lambda$ of undominated $\lambda$-breakpoint sets.

Once undominated breakpoint sets have been computed, the experimenter can readily
evaluate trade-offs between diversity variance and perturbation. For example, the minimal perturbation experiment $X_*$ for a given maximum diversity variance threshold $\theta_d$ is readily found as $X_* = \arg \min_{X \in U : v_d(X) \leq \theta_d} v_p(X) = \arg \max_{X \in U : v_d(X) \leq \theta_d} v_d(X)$. If desired, appropriate data structures can be established to efficiently support such queries.

### 3.2.1 Metrics

To evaluate diversity and perturbation, we adopt here the metrics from our previous work [96, 103]. However, the method presented below is generic enough to support other metrics, including the perturbation scores of [92] or [56].

Perturbation $v_p$ is computed according the hypergraph model of pairwise and higher-order interactions, developed to characterize stability of hybrid libraries [96]. A hyperedge $e$ is defined for each set of residue positions that are in mutual contact. A “hyperresidue” $R$ represents a tuple of amino acids for the residues. An edge-specific potential score $\Phi_e(R)$ is calculated for each hyperresidue for each hyperedge, based on occurrence statistics in a multiple sequence alignment of the specific protein family, as well as in proteins in general. The potential score captures the degree of “hyperconservation” for the edge—how important it appears to be to preserve the combination of amino acid types. Then, given a set of parent proteins $\mathcal{P}$ and a breakpoint set $X$ defining a hybrid library $\mathcal{H}$, we can compute the perturbation as the difference in amino acid distributions (see again Fig. 3.1), weighted by the potentials:

$$v_p(X) = \sum_e \left( \frac{f_{e,\mathcal{P}}(R)}{|\mathcal{P}|} \cdot \Phi_e(R) \right) - \sum_e \left( \frac{f_{e,\mathcal{H}}(R)}{|\mathcal{H}|} \cdot \Phi_e(R) \right),$$

(3.1)
where $f_{e,P}(R)$ and $f_{e,H}(R)$ are the number of occurrences of $R$ at $e$ in the parent proteins and hybrid library, respectively.

We introduced the idea of evaluating diversity in a library according to the variance in the number of mutations between each hybrid–parent pair [103], as illustrated in the right of Fig. 3.1. (Hybrid–hybrid diversity variance can likewise be calculated, and is highly correlated with the hybrid–parent metric.) We have shown that the total number of mutations is a constant determined only by the parents, but that by assessing the squared-differences in the numbers, we are optimizing for a relatively uniform sampling of sequence space. We use here the average diversity variance, the original metric divided by the number of hybrids (of course these have the same minima):

$$v_d(X) = 1/n^{\lambda+2} \sum_{a=1}^{n} \sum_{i=1}^{n^{\lambda+1}} m(H_i, P_a)^2$$  \hspace{1cm} (3.2)

where $m(H_i, P_a) = \sum_{1 \leq r \leq t} I\{H_i[r] \neq P_a[r]\}$ is the number of positions at which hybrid $H_i$ and parent $P_a$ have different residues. To ignore conservative substitutions, we test “equality” according to 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\}\}.

### 3.2.2 Finding Undominated Breakpoint Sets on the Convex Hull

Based on $v_p$ and $v_d$, we equate breakpoint set $X$ with its location in the two-dimensional space with axes for perturbation and diversity variance. If breakpoint set $X'$ is dominated by $X$, then for any line passing through $X$ with a negative slope, $X'$ must be above the line. Thus we know $X$ is undominated if we can find a negative-slope line through it such
that all other breakpoint sets are on or above the line.

This insight leads us to the basis for our first algorithm (see Fig. 3.3), which constructs the lower convex hull connecting the breakpoint sets $X_p$ and $X_d$ minimizing perturbation and diversity variance alone, respectively. For simplicity, we assume throughout the paper that there is a unique $X_p$ minimizing perturbation and a unique $X_d$ minimizing diversity. The extensions to handle non-unique minima are straightforward, requiring us to consider inequalities that aren’t strict.

**Claim 3.1** Let $X_p$ and $X_d$ be the (unique) breakpoint sets minimizing perturbation alone and diversity variance alone, respectively. Then any breakpoint set $X$ on the lower envelope of the convex hull of all breakpoint sets, below the line connecting $X_p$ and $X_d$, is undominated.

**Proof:** Consider such an $X$, and let $X'$ be an adjacent breakpoint set on the convex hull ($X'$ could be $X_p$ or $X_d$). By the definition of convex hull, all other breakpoint sets must be on one side of the line connecting $X$ and $X'$. In fact, they must be above the line since otherwise $X_p$ or $X_d$ would be below the line, contradicting the definition. The line connecting $X_p$ and $X_d$ must have a negative slope. Otherwise, since $v_d(X_d) < v_d(X_p)$, we would also have $v_p(X_d) < v_p(X_p)$, contradicting $X_p$’s optimality. It similarly follows that $v_d(X_d) < v_d(X) < v_d(X_p)$ and that the line connecting $X$ and $X'$ has a negative slope. Thus, all breakpoint sets lie on or above a negative-slope line through $X$, so $X$ is undominated. □

Of course we want to find the breakpoint sets on the convex hull without enumerating
Fig. 3.3: Finding undominated breakpoint sets on the convex hull, for the completely enumerated 2-breakpoint test system of Fig. 3.1. In practice, we would not enumerate all the points within the hull. (top) Undominated breakpoint set $X$ below and farthest from the line connecting $X_p$ and $X_d$, the sets optimizing perturbation alone and diversity variance alone. (middle) Undominated breakpoint set between $X_d$ and $X$. (bottom) All undominated breakpoint sets on the hull.
the exponential number of breakpoint sets inside the hull. Our algorithm, illustrated in
Fig. 3.3 and described in Fig. 3.4, is similar to the quickhull algorithm [7] but efficiently
finds the hull points without knowing the interior points. The algorithm starts with \( X_p \) and
\( X_d \), and recursively finds hull points between an existing pair of hull points. The key is
finding the intermediate hull breakpoint set \( X \) below and farthest from the line connecting
hull breakpoint sets \( X_1 \) and \( X_2 \). (The same method can find the initial \( X_p \) and \( X_d \) as special
cases.) Let \( \alpha = v_p(X_1) - v_p(X_2) \) and \( \beta = v_d(X_1) - v_d(X_2) \), so that \( \alpha/\beta \) is the slope of
the line connecting \( X_1 \) and \( X_2 \). For the \( X \) we seek, all other breakpoint sets must be above
the line passing through \( X \) with slope \( \alpha/\beta \). Thus \( X \) is the breakpoint set minimizing the
value of \( \alpha v_p(X) + \beta v_d(X) \).

```
initialize Q to be an empty queue
enqueue \((X_p, X_d)\) into Q
\( B_H \leftarrow \{X_p, X_d\} \)
repeat
  dequeue from Q one pair of breakpoint sets \((X_1, X_2)\) \((v_d(X_1) < v_d(X_2))\)
  find breakpoint set \( X \) below and farthest from the line connecting \( X_1 \) and \( X_2 \)
  (by dynamic programming, Eq. 3.3)
  if \( X \neq X_1 \) and \( X \neq X_2 \)
    \( B_H \leftarrow B_H + \{X\} \)
    enqueue into Q either \((X_1, X)\) or \((X, X_2)\)
  end if
until Q is empty
return \( B_H \)
```

Fig. 3.4: Algorithm for finding undominated breakpoint sets on the convex hull.

To find \( X \), we adopt the dynamic programming frameworks from our earlier meth-
ods for perturbation alone and diversity variance alone, to handle convex combinations.
The idea is to add breakpoints one-by-one from left to right in the sequence (N- to C-
terminus), at each point considering the change to $\alpha v_p + \beta v_d$ for this breakpoint given previous breakpoints. Optimal substructure holds since a hybrid library with breakpoints $X_k = \{x_1, \ldots, x_{k-1} = r', x_k = r\}$ extends a hybrid library with breakpoints $X_{k-1} = \{x_1, \ldots, x_{k-1} = r'\}$ by concatenating each of the hybrids with each parent fragment $P_a[r'+1, r]$. The best choice for $x_k$ depends only on the best choice for $x_{k-1}$.

Let $d_{pd}(r, k)$ be the optimal value for the linear combination $\alpha v_p + \beta v_d$ with $k$ breakpoints, with the last breakpoint at residue position $r$. The structure of the recurrence to compute $d_{pd}(r, k)$ is as follows:

$$d_{pd}(r, k) = \begin{cases} 
C_{pd}(r) & \text{if } k = 1, \\
\min_{r < r'} \{d_{pd}(r', k - 1) + \Delta d_{pd}(r', r)\} & \text{if } k > 1.
\end{cases} \tag{3.3}$$

where $C_{pd}$ is the initialization value for only one breakpoint and $\Delta d_{pd}(r', r)$ is the increment when one more breakpoint is put after residue position $r$. Straightforward algebraic manipulations to derive $C_{pd}$ and $\Delta d_{pd}(r', r)$ have been omitted; the resulting formulas are as follows:

$$C_{pd}(r) = \alpha v_p(\{r\}) + \frac{\beta}{n^2} \cdot \sum_{a=1}^{n} \sum_{b=1}^{n} m(P_a[1, r], P_b[1, r])^2, \tag{3.4}$$

$$\Delta d_{pd}(r', r) = \alpha (v_p(\{r', r\}) - v_p(\{r\})) + \frac{\beta}{n^2} \cdot \sum_{a=1}^{n} \sum_{b=1}^{n} m(P_a[r' + 1, r], P_b[r' + 1, r])^2 + \frac{\beta}{n^3} \cdot \sum_{a=1}^{n} \left( \sum_{b=1}^{n} m(P_a[1, r'], P_b[1, r']) \cdot \sum_{b=1}^{n} m(P_a[r' + 1, r], P_b[r' + 1, r]) \right) \tag{3.5}$$

To compute this recurrence by dynamic programming requires a table of size $\lambda l$ (recall that $\lambda$ is the number of breakpoints and $l$ is the sequence length) and each entry depends on $O(l)$ previous entries in computing the minimum. Based on previous derivations [96, 103], the complexity of calculating $\Delta d_{pd}(r', r)$ (done in a preprocessing step, for look up during
the dynamic programming) includes $O(l^2E)$ for the increment in perturbation (where $E$ is the set of hyperedges, $l^2$ is for residue pair $(r', r)$) and $O(n^2l^2)$ for the increment in diversity variance (where we have $n$ sequences). Thus the complexity for dynamic programming is $O(l^2E + n^2l^2 + \lambda l^2)$. We run this algorithm once to find each undominated breakpoint set on the convex hull, so to compute the whole set $B_H$ requires $O(B_H(l^2E + n^2l^2 + \lambda l^2))$—polynomial in each of the input variables and output size.

3.2.3 Finding Locally Undominated Breakpoint Sets in Concavities

The algorithm of Fig. 3.4 finds all undominated breakpoint sets on the convex hull, but as Fig. 3.2 illustrates, many undominated breakpoint sets (45/59 in that example) lie in the concavities. Since our underlying dynamic programming framework (Eq. 3.3) is limited to convex combinations of $v_p$ and $v_d$, in order to use it we must focus on smaller regions whose convex hulls intersect the concavities. We can then find breakpoint sets that are locally undominated with respect to the various regions. While these breakpoint sets are not necessarily undominated globally, in the next section we develop an approach to evaluate their optimality.

Let us consider how to constrain our optimization to regions within the perturbation-diversity space. Consider the effect of moving from a breakpoint set $X$ with breakpoint $i$ fixed to residue position $r$, to breakpoint set $X'$ with $i$ fixed to $r + 1$. The contribution to the perturbation score $v_p$ is changed only for those edges incident on $r + 1$. If we assume a constant degree in the contact graph (since physically each residue can only contact a
limited number of other residues), then the expected change in perturbation, \( |v_p(X') - v_p(X)| \), is bounded by a constant fraction of the overall perturbation range. Similarly, the contribution to diversity variance \( v_d \) is changed only for the fragments from position \( X_{i-1} \) to \( X_i \) and from \( X_i \) to \( X_{i+1} \). While we omit the details, which aren’t essential here, it follows that the expected difference \( |v_d(X') - v_d(X)| \) is bounded by a linear function of the total number of mutations in those fragments, a small amount compared to the range of diversity variance. Thus each time we advance a single breakpoint location, we take a small step in perturbation-diversity space.

Based on this insight, our algorithm for exploring the concavities iterates over all possible (breakpoint, position) pairs as in Fig. 3.5. (For a position to be “possible” for a breakpoint, it must leave enough room in the sequence for the preceding and following breakpoints.) With a breakpoint fixed to a position, we apply a variant of our dynamic programming framework (Eq. 3.3), changing \( v_p \) and \( v_d \) appropriately to account for the fixed breakpoint. After obtaining the locally undominated breakpoint sets for each (breakpoint, location) pair, we take the union of the sets and eliminate those that are dominated. The dynamic programming framework is used to find each point on the local lower convex hull. Thus to find a multiset (including duplicates) \( B_L \) of locally undominated breakpoint sets and do dominance checking for \( D_H \), the total complexity is \( O(B_L(I E + n^2 l^2 + \lambda l^2) + D_H^2) \); more efficient dominance checking for \( D_H \) can be achieved by sorting in perturbation and diversity variance. By fixing more breakpoint locations (e.g., pairs), we would explore the concavities even better, but of course at increased cost. We could also consider variations, such as sampling positions rather than trying each one. However, our results show that
fixing each breakpoint at each position is fast enough and yields high quality results.

\begin{center}
\begin{tabular}{|l|}
\hline
initialize $D_H$ to $\emptyset$
\hline
for $x_i \in X$
\hline
for each possible residue position $r$
\hline
fix $x_i$ at $r$; will optimize positions for other breakpoints
\hline
find local undominated breakpoint sets $B_L(i, r)$ using algorithm in Fig. 3.4
\hline
remove breakpoint sets from $D_H$ dominated by breakpoint sets in $B_L(i, r)$
\hline
insert into $D_H$ breakpoint sets from $B_L(i, r)$ not dominated by any breakpoint set in $D_H$
\hline
end for
\hline
end for
\hline
return $D_H$
\end{tabular}
\end{center}

Fig. 3.5: Algorithm for finding undominated breakpoint sets in concavities.

### 3.3 Optimality Guarantees

Suppose that for a particular experiment, we want to find the optimal breakpoint set $X$ minimizing perturbation such that $v_d(X) \leq \theta_d$, for some diversity variance threshold $\theta_d$ (minimizing diversity subject to a perturbation threshold can be handled similarly). In our concavity-exploring algorithm, when we fix breakpoint $i$ to be at residue position $r$, we obtain a “local” convex hull. This hull may have a lower convex chord $X_1X_2$ (connecting consecutive points on the lower hull) that intersects the $\theta_d$ line (i.e., the vertical line $v_d = \theta_d$) at some point with perturbation value $p$ (see Fig. 3.6). We represent this convex chord as $c = (i, r, p)$. We can use the set $C$ of all convex chords that intersect the $\theta_d$ line over the various local hulls, to bound the best possible perturbation for $X$.

**Claim 3.2** Given diversity variance threshold $\theta_d$ and set $C$ of all convex chords intersecting the line for $\theta_d$, let $S_i = \{(r, p) \ | \ (i, r, p) \in C\}$ ($1 \leq i \leq \lambda$) and let $T = \sum_{i=1}^{\lambda} |S_i|$...
\{\{(r_1, p_1), (r_2, p_2), \ldots, (r_\lambda, p_\lambda)\} \in S_1 \times S_2 \times \ldots \times S_\lambda \mid r_1 < r_2 < \ldots r_\lambda}\) If \(T\) is empty, the experiment plan provided by \textsc{Staversity} is optimal. Otherwise, the undiscovered optimal undominated breakpoint set \(X_o = \{x_{o,1}, x_{o,2}, \ldots, x_{o,\lambda}\}\) has \(v_p(X_o) \geq \min_{T \in T} \max_{(r, p) \in T} p\).

**Proof:** Suppose \(X_o\) is the optimal undominated breakpoint set and is not found by \textsc{Staversity}. Then for each local hull fixing breakpoint \(i\) at residue position \(x_{o,i}\), \(X_o\) is not found, implying that it is inside the hull and above a lower convex chord. Let \(X_1\) and \(X_2\) be the breakpoint sets at the left and right ends, respectively, of the convex chord below \(X_o\) (i.e., \(v_d(X_1) < v_d(X_o) < v_d(X_2)\)), as in the top of Fig. 3.6. Since the line through \(X_1\) and \(X_2\) has a negative slope (as in the proof in Sec. 3.2.2), we have \(v_p(X_1) > v_p(X_2)\) and \(v_p(X_o) > v_p(X_2)\). Thus the line of \(\theta_d\) must intersect the convex chord \(X_1X_2\), since otherwise either \(v_d(X_1) > \theta_d\) and \(v_d(X_o) > \theta_d\) (contradicting its satisfaction of the threshold), or \(v_d(X_2) < \theta_d\) and \(v_p(X_2) < v_p(X_o)\) (contradicting its optimality). Furthermore, the perturbation value of the intersection is less than \(v_p(X_o)\) as the convex chord has a negative slope and \(v_d(X_o) \leq \theta_d\).

Thus, if \(X_o\) exists, for each local hull fixing breakpoint \(i\) at residue position \(x_{o,i}\), we have a convex chord \(c = (i, x_{o,i}, p_i)\) such that \(c\) is below \(X_o\) and intersects the line of \(\theta_d\) at perturbation \(p_i\). So \(v_p(X_o) > \max \{p_i\}\) and \(\{(x_{o,1}, p_1), (x_{o,2}, p_2), \ldots, (x_{o,\lambda}, p_\lambda)\} \in T\).

As a result, if \(T\) is empty, no \(X_o\) exists, and the plan is optimal. Otherwise, following the argument above, we can bound the perturbation for any missed \(X_o\) by \(v_p(X_o) \geq \min_{T \in T} \max_{(r, p) \in T} p\). \(\square\)
This claim suggests an approach for computing the perturbation bound: consider convex chords for the local hulls in order of perturbation, moving up the line of $\theta_d$. When we have found a set of chords, one for each breakpoint, such that the corresponding breakpoint locations are in increasing order, then we have the best possible perturbation value. The bottom of Fig. 3.6 gives an example.

Fig. 3.6: Optimality guarantee for breakpoint sets found by STAVERSITY. (Left) The dashed line representing a diversity threshold $\theta_d$ intersects convex chords below $X_o$. (Right) Bounding the optimal perturbation for a 2-breakpoint set with respect to the $\theta_d$ threshold. Each (breakpoint, position) pair generates one convex chord (blue line segment) intersecting the $\theta_d$ line. The lower bound for perturbation is the maximum perturbation from a set of convex chords for a consistent breakpoint set, here $x_1 = 6$ and $x_2 = 8$.

To efficiently compute the lower bound of $v_p(X_o)$, we develop another dynamic programming algorithm. Let $T_{k, \gamma}$ be the valid breakpoint sets from $S_i$ with breakpoint $k$ ($\leq \lambda$) at position $\gamma$, i.e., $T_{k, \gamma} = \{(r_1, p_1), (r_2, p_2), \ldots, (r_k, p_k) \in S_1 \times S_2 \times \ldots \times S_k \mid r_1 < r_2 < \ldots < r_k = \gamma\}$. We can then define the minimum perturbation with breakpoint $k$ at position $\gamma$ as:

$$d_e(\gamma, k) = \min_{T \in T_{k, \gamma}} \max_{(r, p) \in T} p. \quad (3.6)$$
If $T_{k,\gamma}$ is empty, then the $k$-breakpoint set cannot be constructed, and $d_e(\gamma, k) = \infty$. Otherwise, we can form a $k$-breakpoint set by extending a valid $(k-1)$-breakpoint set ending at residue position $\tau < \gamma$. Optimal substructure holds, and to compute the perturbation we have the recurrence:

$$d_e(\gamma, k) = \begin{cases} 
\max \{ p, \min_{\tau < \gamma} d_e(\tau, k-1) \} & \text{if } \exists (r, p) \in S_k \text{ with } r = \gamma, \\
\infty & \text{otherwise.}
\end{cases} \quad (3.7)$$

And we get the lower bound of $v_p(X_o)$ from the final column:

$$v_p(X_o) \geq \min_{T \in T_\lambda} \max_{(r, p) \in T} \{ p \} = \min_{\gamma} d_e(\gamma, \lambda). \quad (3.8)$$

If $\min_{\gamma} d_e(\gamma, \lambda)$ is larger than the perturbation value of the experiment plan provided by Staversity, $X_o$ does not exist and the experiment plan provided by Staversity must be optimal.

In the dynamic programming of Eq. 3.7, the table is of size $\lambda l$, and each entry depends on $O(l)$ previous entries in computing the minimum, for a total complexity of $O(\lambda l^2)$. The preprocessing to put the chords into the $S_i$ buckets and order them within the buckets can be done in linear time, since we have small ranges of integers ($[1, \lambda]$ and $[1, l]$, respectively).

### 3.4 Results and Discussion

#### 3.4.1 PurE Family

We have been studying by site-directed recombination homologous proteins of the purE family (COG 41 and pfam 731), which catalyze steps in the de novo synthesis of purines.
While clearly homologous, purE proteins carry out substantially different enzymatic activities in different organisms: in eubacteria, fungi and plants (as well as probably most archaeabacteria), the purE product functions as a mutase in the second step of a two-step reaction, while in metazoans and methanogenic archaeabacteria, the purE product functions as a carboxylase in a single-step reaction that yields the same product [18]. This striking difference in activity makes the purE family a valuable target in protein engineering—by exploring sequence space through site-directed recombination, we seek to find the features of the “boundaries” enclosing the distinct activities.

To identify a set of possible purE parents, we created a multiple sequence alignment of the purE family, then eliminated columns not mapped to the structure of \textit{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 is 65.8%. We selected three parent sets, each consisting of three purE parents with varying diversity—medium diversity (average identity 55%): \textit{Escherichia coli}, \textit{Gallus gallus}, \textit{Methanothermobacter thermautotrophicus}; high diversity (31%): \textit{Drosophila melanogaster}, \textit{Bdellovibrio bacteriovorus}, \textit{Treponema denticola}; low diversity (80%): \textit{Gibberella zeae}, \textit{Magnaporthe grisea}, \textit{Saccharomyces cerevisiae}.

For each parent set, and for 2 to 10 breakpoints, we applied \textsc{Staversity} to find breakpoint sets. On average, it took around 5 minutes for 2 breakpoints. The running time increased according to the number of breakpoints, and it took around 2 hours for 10 breakpoints.
Fig. 3.7 illustrates the breakpoint sets found by STAVERSITY for the medium-diversity parent set. As expected, perturbation is minimized (resulting in the breakpoint sets plotted at large $y$ values) when breakpoints are selected at the ends of the sequence, which makes hybrid proteins as similar as possible to the parents. When breakpoints are selected more evenly (breakpoint sets plotted at small $y$ values), perturbation is increased as more hyper-edges are disrupted, while diversity variance is decreased as hybrid proteins are distributed more uniformly around the parents in sequence space. These figures demonstrate the utility of STAVERSITY in finding a wide range of breakpoint sets, and helping the protein engineer understand the trade-offs between the two key considerations in choosing breakpoint locations.

To assess the completeness of STAVERSITY, we enumerated for the medium diversity parents all 2-breakpoint sets (plotted in Fig. 3.2) and all 3-breakpoint sets, deeming it impractical to enumerate plans with more breakpoints. STAVERSITY finds 55 of the 59 undominated 2-breakpoint sets in the enumeration and 77 of the 115 undominated 3-breakpoint sets. In both cases, the breakpoint sets that STAVERSITY missed were quite close to others that it found. For missed set $X$ and found sets $B$, we compute the distance as $\min_{X' \in B : v_d(X') < v_d(X)} v_p(X') - v_p(X)$, divided by the range of perturbation values over all breakpoint sets. The average value for the 4 missed 2-breakpoint sets is 0.5%, as is that for the 38 missed 3-breakpoint sets.

Based on RASPP [17], the only other method available for optimizing stability and diversity does so implicitly while optimizing perturbation. The lengths of the fragments to be recombined are constrained to lie between minimum and maximum values; a perturbation-
Fig. 3.7: Breakpoint sets (top: 2, middle: 6, and bottom: 10 breakpoints) found by STAVERSITY for medium-diversity purE parents. Each horizontal line corresponds to one breakpoint set, with marks at the $x$ coordinates for the residue indices. Along the $x$-axis the sequence is illustrated with red boxes for $\alpha$-helices and blue ones for $\beta$-sheets, according to the crystal structure of *E. coli* purE (PDB id: 1qcz). With increasing $y$ value, the breakpoint sets decrease in perturbation and increase in diversity variance, thereby moving from lower right to upper left along the curve in the 2D perturbation / diversity variance space.
optimal library is generated for each minimum-maximum pair. This restriction does provide some sampling of various levels of diversity, since larger fragments generally lead to greater diversity. For comparison, we implemented a version of this approach (called IMPLICIT below) using our metrics and returning only the locally undominated breakpoint sets (i.e., not dominated by any others in the set). For a baseline for comparison, we also applied a simple random selection method (called RAND below), in which we randomly sample sets of breakpoints and return the locally undominated ones.

Tab. 3.1 summarizes the results on the different tests. To put the methods on a relatively equal footing and avoid saturation by random sampling (which happens with small numbers of breakpoints), the number of random samples for each test case was set as the total number of breakpoint sets found by STAVERSITY in the local convex hulls. (We also tested a large number of random samples; see below.) In each table, the rows of “STAVERSITY”, “RAND” and “IMPLICIT” give the numbers of breakpoint sets found by each method. The rows of the form “STAVERSITY dom. RAND” give the percentage of breakpoint sets found by the second method that are dominated by those found by the first method (not counting the breakpoint sets common to both methods).

On average, for the medium diversity parents, RAND finds only 28 percent as many breakpoint sets as STAVERSITY does, and 96 percent of the RAND ones are dominated by STAVERSITY ones. IMPLICIT finds 45 percent as many, of which 71 percent are dominated. RAND performs similarly badly on the high diversity parents, finding 32 percent as many with 98 percent dominated, and improves a little on the low diversity set, at 48 percent with 88 percent dominated. IMPLICIT improves a little on the high diversity set, finding
Tab. 3.1: Comparison of STAVERSITY with RAND and IMPLICIT for three different parent sets in purE family and from 2 to 10 breakpoints.

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>55</td>
<td>89</td>
<td>112</td>
<td>134</td>
<td>180</td>
<td>215</td>
<td>229</td>
<td>263</td>
<td>284</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>44</td>
<td>39</td>
<td>36</td>
<td>45</td>
<td>37</td>
<td>37</td>
<td>24</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>43</td>
<td>63</td>
<td>68</td>
<td>78</td>
<td>71</td>
<td>65</td>
<td>61</td>
<td>47</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>72.2%</td>
<td>90.9%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td></td>
<td>14.0%</td>
<td>41.3%</td>
<td>58.8%</td>
<td>76.9%</td>
<td>85.9%</td>
<td>92.3%</td>
<td>90.2%</td>
<td>87.2%</td>
<td>89.1%</td>
</tr>
<tr>
<td></td>
<td>0.0%</td>
<td>3.4%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td></td>
<td>0.0%</td>
<td>9.0%</td>
<td>2.7%</td>
<td>0.7%</td>
<td>0.0%</td>
<td>0.5%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td></td>
<td>79</td>
<td>104</td>
<td>160</td>
<td>179</td>
<td>230</td>
<td>263</td>
<td>300</td>
<td>352</td>
<td>369</td>
</tr>
<tr>
<td></td>
<td>63</td>
<td>62</td>
<td>55</td>
<td>55</td>
<td>49</td>
<td>50</td>
<td>52</td>
<td>43</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>86</td>
<td>82</td>
<td>106</td>
<td>107</td>
<td>106</td>
<td>120</td>
<td>98</td>
<td>122</td>
<td>109</td>
</tr>
<tr>
<td></td>
<td>82.5%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td></td>
<td>4.7%</td>
<td>23.2%</td>
<td>49.1%</td>
<td>55.1%</td>
<td>70.8%</td>
<td>83.3%</td>
<td>81.6%</td>
<td>85.2%</td>
<td>85.3%</td>
</tr>
<tr>
<td></td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td></td>
<td>0.0%</td>
<td>1.9%</td>
<td>0.6%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>26</td>
<td>34</td>
<td>38</td>
<td>62</td>
<td>68</td>
<td>79</td>
<td>87</td>
<td>106</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>12</td>
<td>21</td>
<td>25</td>
<td>25</td>
<td>29</td>
<td>29</td>
<td>31</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>6</td>
<td>10</td>
<td>14</td>
<td>17</td>
<td>20</td>
<td>23</td>
<td>24</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>35.7%</td>
<td>83.3%</td>
<td>81.0%</td>
<td>100.0%</td>
<td>96.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td></td>
<td>33.3%</td>
<td>33.3%</td>
<td>70.0%</td>
<td>78.6%</td>
<td>82.4%</td>
<td>85.0%</td>
<td>87.0%</td>
<td>87.5%</td>
<td>88.9%</td>
</tr>
<tr>
<td></td>
<td>5.0%</td>
<td>0.0%</td>
<td>5.9%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td></td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
</tbody>
</table>
56 percent with 60 percent dominated, but for the low diversity parents finds only 29 percent with 72 percent dominated. STAVERSITY always finds more and better breakpoint sets than RAND and IMPLICIT. One possible explanation for the variation of IMPLICIT’s performance with parent diversity level is that when parents become more diverse (in the limit, being entirely different), the fragment length (the implicit diversity control) is increasingly important for generating diversity. When parents become less diverse, longer fragment length does not necessarily mean more diversity, so the impact of fragment length on diversity is not so significant. The performance of random selection is clearly subject to the curse of dimensionality.

We tried using a large number of samples (> $10^7$) for the different parent sets with 10 breakpoints. Even with this large number of samples, STAVERSITY still significantly outperforms random selection. For the medium diversity parents, RAND finds only 83 breakpoints, all of which are dominated by STAVERSITY ones, while with the high diversity parents, it finds only 117, again all dominated. It does relatively better for the low diversity parents, finding 101 breakpoints of which 94% are dominated, and it dominates 2 of the STAVERSITY breakpoints.

In addition to finding more, better breakpoint sets, STAVERSITY can provide optimality guarantees. To evaluate how close our results are to optimal perturbation-diversity trade-offs, we tested 100 diversity variance thresholds $\theta_d$ (selected evenly within the range of $v_d(X)$ excluding the minimum and the maximum values of $v_d(X)$) for the bound on the optimal perturbation value. As Fig. 3.8 illustrates, our plans are very close to optimal. Quantitatively, we can compute the average distance between the vector of perturbation
Fig. 3.8: Optimality guarantees for the medium-diversity purE parents: breakpoint sets found by STAVERITY (green diamonds) compared to lower bounds on optimal perturbation for 100 different diversity variance values (red crosses), for (top) 2, (middle) 6, and (bottom) 10 internal breakpoint sets.
bounds and corresponding actual perturbation values, normalized by the range of perturbation values as above. For the medium diversity parents, the difference is 0.9% for 2 breakpoints, 1.0% for 6, and 0.9% for 10. The results are similar for the high diversity set (1.2%, 1.4%, and 1.3%) and low diversity set (3.4%, 1.1%, and 1.4%). The one outlier, 3.4% for low-diversity parents with 2 breakpoints, appears to arise from a relatively large “jump” in the diversity variance between a pair of adjacent STAVERSITY-identified breakpoint sets. This can happen due to an inadequately-explored concave region.

3.4.2 Beta-lactamase Family

To further evaluate the effectiveness of our method, we planned recombination experiments for proteins from the beta-lactamase family, as previous site-directed recombination experiments have employed beta-lactamase parents TEM-1 and PSE-4 [54, 17, 53]. To calculate perturbation of hybrid proteins, we used a previously-studied multiply-aligned set of 136 beta-lactamases including TEM-1 and PSE-4 [96]. We planned 2–10 breakpoint sets for parent proteins TEM-1 and PSE-4, which have 263 residues and sequence identity of 41.8%.

Fig. 3.9 illustrates the breakpoint sets identified by STAVERSITY, varying the trade-off between perturbation and diversity variance. Once again we see the movement from spread-out breakpoints (better diversity) to breakpoints closer to the ends (better perturbation).

Tab. 3.2 compares the results across the three planning methods. On average, RAND finds only 27 percent as many breakpoint sets as STAVERSITY does, and more than 95
Fig. 3.9: Breakpoint sets (top: 2, middle: 6, and bottom: 10 breakpoints) found by STAVERSITY for TEM-1 and PSE-4 parents, using the representation described in Fig. 3.7 with the reference crystal structure of TEM-1 (PDB id: 1btl).
percent of the RAND ones are dominated by STAVERSITY ones. IMPLICIT finds 48 percent as many, of which more than 58 percent are dominated. For the case of 10 breakpoints, we also performed a large-scale random breakpoint selection (> $10^7$ samples). This yields only 31 percent as many breakpoint sets as STAVERSITY found, all dominated by STAVERSITY ones.

Tab. 3.2: Comparison of STAVERSITY with RAND and IMPLICIT for parents TEM-1 and PSE-4, with 2 to 10 breakpoints.

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>STAVERSITY</td>
<td>50</td>
<td>72</td>
<td>115</td>
<td>132</td>
<td>157</td>
<td>174</td>
<td>193</td>
<td>215</td>
<td>229</td>
</tr>
<tr>
<td>RAND</td>
<td>44</td>
<td>24</td>
<td>28</td>
<td>27</td>
<td>23</td>
<td>28</td>
<td>35</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>IMPLICIT</td>
<td>36</td>
<td>48</td>
<td>56</td>
<td>62</td>
<td>65</td>
<td>71</td>
<td>79</td>
<td>81</td>
<td>81</td>
</tr>
<tr>
<td>STAVERSITY dom. RAND</td>
<td>63.6%</td>
<td>91.7%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td>STAVERSITY dom. IMPLICIT</td>
<td>36.1%</td>
<td>37.5%</td>
<td>42.9%</td>
<td>58.1%</td>
<td>58.5%</td>
<td>70.4%</td>
<td>67.1%</td>
<td>74.1%</td>
<td>81.5%</td>
</tr>
<tr>
<td>RAND dom. STAVERSITY</td>
<td>6.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>IMPLICIT dom. STAVERSITY</td>
<td>2.0%</td>
<td>1.4%</td>
<td>2.6%</td>
<td>0.8%</td>
<td>1.3%</td>
<td>0.6%</td>
<td>0.5%</td>
<td>1.9%</td>
<td>1.3%</td>
</tr>
</tbody>
</table>

To evaluate optimality guarantees, we computed the bound on the perturbation value for 100 diversity variance thresholds $\theta_d$ (Fig. 3.10) as we did for the parent sets in the purE family. The average normalized distances are 3.7% for 2 breakpoints, 1.5% for 6 and 1.7% for 10. Once again, the loosest bound of 3.7% appears to be due to a diversity variance gap for a concave region.
Fig. 3.10: Optimality guarantees for TEM-1 and PSE-4: breakpoint sets found by STAVERSITY (green diamonds) compared to lower bounds on optimal perturbation for 100 different diversity variance values (red crosses), for (top) 2, (middle) 6, and (bottom) 10 internal breakpoint sets.
4. PROTEIN FRAGMENT SWAPPING: A METHOD FOR ASYMMETRIC, SELECTIVE SITE-DIRECTED RECOMBINATION

This chapter presents a new approach to site-directed recombination, swapping combinations of selected discontiguous fragments from a source protein in place of corresponding fragments of a target protein. By being both asymmetric (differentiating source and target) and selective (swapping discontiguous fragments), our method focuses experimental effort on a more restricted portion of sequence space, constructing hybrids that are more likely to have the properties that are the objective of the experiment. Furthermore, since the source and target need to be structurally homologous only locally (rather than overall), our method supports swapping fragments from functionally important regions of a source into a target “scaffold”; e.g., to humanize an exogenous therapeutic protein. A protein fragment swapping plan is defined by the residue position boundaries of the fragments to be swapped; it is assessed by an average potential score over the resulting hybrid library, with singleton and pairwise terms evaluating the importance and fit of the swapped residues. While we prove that it is NP-hard to choose an optimal set of fragments under such a potential score, we develop an integer programming approach, which we call SWAGMER, that works very
well in practice. We demonstrate the effectiveness of our method in two types of swapping problem: selective recombination between beta-lactamases and activity swapping between glutathione transferases. We show that the selective recombination approach generates a better plan (in terms of resulting potential score) than a traditional site-directed recombination approach. We also show that in both cases the optimized experiment is significantly better than one that one would result from stochastic methods. This work has been published in [102].

4.1 Introduction

Protein recombination constructs libraries of hybrids by recombining fragments from two or more parents, with the goal of discovering hybrids with beneficial properties such as improved thermostability, activity, or substrate specificity (e.g., [85, 67, 51, 92, 64, 1, 10, 69, 9, 25, 24, 87, 42]). For example, Stemmer demonstrated the development of beta-lactamase hybrids with a 32,000-fold increase in the required minimum inhibitory concentration of the antibiotic cefotaxime [85]. In contrast with mutagenesis techniques, recombination uses amino acid combinations that already exist in wild-type proteins and thus are likely to produce viable proteins. Site-directed techniques seek to to improve the “hit rate” of good hybrids by recombining the parents at selected breakpoint positions, rather than stochastically. For example, Arnold, Mayo, and co-workers showed that selecting breakpoints so as to minimize disruption of interacting amino acid pairs yields beta-lactamase hybrids that are more likely to be stably folded and functional than random ones [92].
Typically site-directed recombination is both exhaustive and symmetric: a combinatorial library of hybrids is constructed from fragments covering all residues (exhaustive) and taken uniformly from all parents (symmetric). However, in many applications it may be desirable to relax these requirements. A selective approach may be warranted if the parent proteins have regions that are significantly “gappy” (insertions/deletions) in a sequence alignment or that are significantly different structurally. In such a case much experimental effort may be wasted on constructing and screening a large number of poor quality hybrids, instead of focusing on those that recombine the non-gappy and structurally analogous regions, and thus are more likely to be stably folded and functional. An asymmetric approach may be in order if the goal is to swap portions of particular functional importance from one protein into another. One such application is introduction of exogenous therapeutic protein activities, where part of a foreign source is swapped into a human protein target that acts as a scaffold (and will not elicit an immune response). Antibodies have long been humanized this way, e.g., combining murine variable regions with human constant regions [57, 34]. An approach for the much more difficult task of humanizing enzymes (which lack the overtly modular nature of antibodies) was recently demonstrated [25, 24], introducing activity from a rat glutathione transferase into a human one.

In order to enable the optimization of asymmetric, selective site-directed recombination experiments, we develop here a new approach that we call protein fragment swapping (Fig. 4.1). We distinguish a source parent and a target parent, and construct a library that swaps combinations of discontiguous fragments from the source to the target. By swapping from source to target, our approach is asymmetric; by swapping fragments that can be dis-
contiguous, our approach is selective. Furthermore, fragment swapping does not require the parents to be homologous (in sequence or structure) overall, but only requires there to be corresponding regions of the source and target in which we may swap fragments. Thus it directly supports the humanization application discussed in the previous paragraph. Traditional combinatorial site-directed library construction is a special case of fragment swapping, where the swapped fragments must be contiguous. By enabling the protein engineer to define sequence regions of interest, swapping focuses the experimental effort on a smaller portion of sequence space that is believed to be more relevant. Thus it improves the chance of finding beneficial new hybrids in the resulting library.

We develop an algorithm, which we call SWAGMER (swapping part of “fragment” into “swapper”) for planning protein fragment swapping experiments. The objective is to select, from the corresponding source and target sequence regions of interest, “good” source fragments to be combinatorially swapped in for corresponding target fragments. To assess possible plans, we employ a statistical potential score analogous to those used in combinatorial recombination to help ensure stability of the resulting hybrids [92, 54, 56, 78]. The potential score averages over the entire resulting hybrid library for a set of singleton and pairwise terms evaluating the importance of the residues and how well they match. While the averages can be computed efficiently (i.e., without enumerating the exponential number of hybrids), we show that the inclusion of pairwise terms leads to an NP-hard optimization problem. To solve the problem in practice, we develop an integer programming approach that represents swapping assignments for the residues by binary variables, and optimizes the sum potential score for all hybrids in the resulting library. To demonstrate the effec-
tiveness of our approach, we planned experiments for selective recombination between beta-lactamases, activity swapping between glutathione transferases and activity swapping between carboxylases and mutases in the purE family. In all cases, the optimized plans outperform all randomly-generated plans (as would result from stochastic recombination methods). We also compared the selective plan with an optimized traditional site-directed recombination plan, and show that the swapping library has a better average potential score, increasing the probability of obtaining functional variants. Finally, we extended our objective function to incorporate the diversity of the resulting libraries in addition to their overall potential scores, and show that a small number of plans capture the key trade-offs between these two important criteria.

4.2 Methods

There are three main steps (Fig. 4.1) to planning a fragment swapping experiment for a given source and target. We assume here that we are given a single source protein $S$ of length $m$ and target protein $T$ of length $n$; the approach can readily be generalized to multiple proteins.

1. Identify a set of swappable regions, $R = \{(s_1, t_1, \ell_1), (s_2, t_2, \ell_2), \ldots\}$. For all $i$, $s_i$ and $t_i$ are the beginning residues of swappable region $i$ in the source and target respectively, $\ell_i$ is the length of swappable region $i$. We have $s_i \in [1..m]$; $t_i \in [1..n]$; $s_i + \ell_i \leq s_{i+1}$; and $t_i + \ell_i \leq t_{i+1}$.

2. Define a potential score function $\phi$ to evaluate a possible swapping plan. We com-
pute the average score over the hybrids in the library, employing position-specific singleton terms \( g_i(a) \) and pairwise terms \( g_{i,j}(a, b) \) to assess the importance and fit of the swapped residues \( a \) at position \( i \) and residue pairs \( a, b \) at positions \( i, j \) with respect to amino acid statistics for related sequences.

3. Select a set of swapping fragments \( F = \{(r_1, a_1, b_1), (r_2, a_2, b_2), \ldots\} \) from the swap-pable regions such that for all \( i \), we have \( r_i \in [1..|R|] \); \( a_i, b_i \in [1..l_{r_i}] \); \( l_{\min} \leq b_i - a_i + 1 \leq l_{\max} \); and for \( j > i \), if \( r_i = r_j \) then \( b_i < a_j \). The minimum and maximum fragment length constraints, \( l_{\min} \) and \( l_{\max} \), control the number of residues participating in the swapping.

Our goal is to optimize the selected fragments:

**Fragment swapping problem.** *Given swappable regions \( R \) and potential score function \( \phi \), find within \( R \) a set \( F \) of \( \lambda \) fragments maximizing the average potential score for all hybrids constructed.*

Modified versions of existing site-directed recombination techniques may be employed to construct the swapping library defined by a set of fragments (step 4 in Fig. 4.1). We propose to use SPLISO [76] and RoboMix [4], hierarchically assembling hybrids by ligating fragments with short (e.g., 3-nucleotide) overhangs common to both parents, and robotically ensuring that only the desired asymmetric combinations are constructed (swapping source into target but not vice-versa).
Fig. 4.1: Overview of fragment swapping method. (1) Identify swappable regions (colored), indicating corresponding portions of the source and target proteins between which fragments may be swapped. The regions may cover most or all of the sequences, or they may be discontiguous. (2) Define a potential score to assess the library resulting from a possible swapping. The example illustrates conservation (R in the first region) and covariation (IL/MV in the second region and SE/TD across the second and third regions) within one of the families (source or target). Hybrids in a possible library can be evaluated for satisfaction of these conservation and covariation constraints. (3) Select fragments (darker colors) within the swappable regions to be swapped from the source into the target, so as to optimize the library potential. (4) Construct a hybrid library by swapping all combinations of the source fragments into the target, replacing its corresponding fragments.

4.2.1 Swappable regions

In combinatorial site-directed protein recombination, parent proteins are typically selected from the same protein family [92, 54, 68]. It is assumed by homology that the same overall structure is common to the parents and the resulting hybrids. On a more local level, the assumption is that corresponding amino acids in an alignment of the parents are in similar local structural environments, so that the residues in the resulting hybrids will likewise be in favorable environments. The common structural context among parents and hybrids is thereby “factored out” of the planning.

In fragment swapping, we no longer require the source and target to be related or to be similar overall. However, we would still like to ensure that the hybrids maintain the overall
structure of the target, and that the swapped source fragments are likely to be placed in suitable local environments in the target. This allows us to focus our optimization efforts on the specific amino acid content (the potential score in the next section). Thus we start with a set of swappable regions, pairs of corresponding substrings from the source and target (the first step of Fig. 4.1).

In cases of sufficient homology, sequence alignment suffices to determine swappable regions. We eliminate the “gappy” parts of the alignment (insertions/deletions) and use the remaining contiguous portions as swappable regions. When structures are available for both source and target, and the structures are similar enough, swappable regions can be found by standard topological structural alignment techniques [32, 80, 97]. We keep the portions that structurally align well and eliminate insertions/deletions and portions with poor structural correspondence. In the most challenging cases, global structural alignment yields poor correspondence, but some local regions align well and may serve as swappable regions. Methods for establishing such local structural alignments are beyond the scope of the present work, but may be based on geometric hashing [61] or extension of aligned fragment pairs [80, 97].

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

Swapping can be seen as making clusters of simultaneous mutations, and our goal is to choose sets of mutations that are in some sense optimal, in that they transfer the desired function without disrupting the current scaffold. As in previous work [92, 56, 69, 78, 77, 74, 84, 96, 90], we assume that constraints on amino acid choices required to maintain structure and function are revealed in the sequence record, and devise an objective function seeking to satisfy those constraints. (In fact, related contact potentials have long been the basis for many protein structure prediction techniques [88, 55].) We base the potential function here on the statistical framework from our earlier site-directed recombination work [96], but the planning method can use any potential score of the same form.

We deal here with two types of sequence constraint displayed by a multiply-aligned set of sequences: position-dependent single residue conservation and pair-wise covariation (see again Fig. 4.1, step 2). For example, if a residue is highly conserved in the source family, it may be important to swap it into the target in order to introduce the desired function. Likewise, if a pair of residues are highly correlated in the source family, it may be necessary to ensure that they are swapped as part of the same fragment, since placing them in different fragments will result in the other combinations less frequently observed in the family. While we do not include in our potential any contribution from residues outside the swappable regions (even by way of pairwise terms with residue in the swappable regions), the potential can be generalized to do so, or an overall “environment” effect can be incorporated into the singleton terms.
More formally, let us consider conservation and covariation in a multiple sequence alignment $S$ for the source protein family. For the singleton terms, we define $s_i(a)$ as the log probability of amino acid type $a$ at residue position $i$ in the family:

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

(4.1)

For the pairwise terms, we only consider residue pairs $i$ and $j$ that are in contact in a representative structure for the protein family (assumed common to all, by homology), as contacting pairs have the greatest direct impact on establishing a suitable local environment.

We define $s_{i,j}(a,b)$ as the log probability of the pair of amino acid types $a$ and $b$, vs. what would be expected if they were independent:

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

(4.2)

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

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

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

(4.3)

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

(4.4)

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

Given a hybrid sequence $P$, we can evaluate how well it satisfies the sequence constraints considering both source and target protein families as:

$$\phi(P) = \sum_i g_i(P[i]) + \sum_{i,j} g_{i,j}(P[i], P[j]). \quad (4.5)$$

The higher the value of $\phi(P)$, the more likely that hybrid $P$ will be folded and have the desired function. Since we would like all hybrids to satisfy the constraints, we evaluate a possible library in terms of the sum of the hybrid scores according to Eq. 4.5, seeking to maximize the total. More precisely, if we have selected $\lambda$ fragments, then $2^\lambda$ hybrids $P_h$ ($1 \leq h \leq 2^\lambda$) are created, and we seek to maximize:

$$\sum_h \phi(P_h) = \sum_h \left( \sum_i g_i(P_h[i]) + \sum_{i,j} g_{i,j}(P_h[i], P_h[j]) \right) = \sum_i \sum_h g_i(P_h[i]) + \sum_{i,j} \sum_h g_{i,j}(P_h[i], P_h[j]). \quad (4.6)$$

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

$$\sum_h \phi(P_h) = 2^\lambda \times \left( \sum_i \phi_1(i) + \sum_{i,j} \phi_2(i, j) \right). \quad (4.7)$$

In order to develop an efficient planning algorithm, we cannot afford to enumerate the exponential number of hybrids for a possible fragment swapping, in order to evaluate the total potential. Fortunately, given the definition of a fragment swapping we can compute average potentials $\phi_1$ and $\phi_2$ for a given position or pair of positions in constant time, and
thus the overall average potential $\phi$ in at most quadratic time (though in practice the number of pairwise terms is likely to be linear, due to the contact restriction). The key insight is that each residue or reside pair participates in a well-defined pattern of hybrids, depending on the selection of fragments to be swapped. That is, the “projection” of the hybrid library onto a single column or pair of columns can be partitioned into a few cases, each with the same number of hybrids in the overall library as in Fig. 4.2, and we simply need to average over the cases.

For $\phi_1(i)$ there are two possibilities, depending on whether or not residue $i$ is swapped (Fig. 4.2, left).

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

$$\phi_s(i) = \frac{1}{2} \times (g_i(S[i]) + g_i(T[i]))$$

When residue $i$ is not being swapped ($\phi_n$), all the hybrids have the target residue; when it is ($\phi_s$), half the hybrids have the source residue and the other half have the target residue.

For $\phi_2(i, j)$ there are five cases (Fig. 4.2, right): neither $i$ nor $j$ is swapped ($\phi_{nn}$), only $i$ is swapped ($\phi_{sn}$), only $j$ is swapped ($\phi_{ns}$), both are swapped in the same fragment ($\phi_{s1}$), or both are swapped in different fragments ($\phi_{s2}$).

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

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

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

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

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

(4.14)

Fig. 4.2: Patterns of potential score contribution in swapping from single residue \(i\) (left) or pair of residues \(i, j\) (right). Fragments being swapped are shaded darker.

4.2.3 Fragment selection

Recall that our goal is to select a set of fragments from the swappable regions, so that the average potential score over the resulting hybrid library is maximized. Unfortunately, we have proved that this optimization problem is NP-hard when using a potential score with pairwise terms. The detailed proof is in an appendix for the interested reader.

**Claim 4.1** The fragment swapping problem is NP-hard.

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

Computationally, the fragment swapping problem is somewhat analogous to the threading (sequence-structure alignment) problem, in which secondary structure “fragments”
from a template “source” are aligned to the primary sequence for a target, according to a potential score that typically includes both singleton (environment) and pairwise (contact) terms [5, 33, 44, 22]. (Like swapping, threading is also NP-hard [45].) The most important difference is that in threading, we know the lengths of the fragments (secondary structure elements) that must be aligned, whereas in fragment swapping, that is part of the optimization. We make use of that analogy in developing an integer programming approach to the fragment swapping problem, since RAPTOR [95] is a very successful thresher based on an integer programming formulation. While drawing inspiration from that work, our formulation must employ different variables (since the fragments have unknown lengths), different constraints (to maintain a valid fragment swapping), and of course a different objective function.

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

71
The potential score contributions are determined by the fragments to which single residues belong and the fragment pairs to which residue pairs belong. Thus in order to develop an integer programming approach, we define singleton and pairwise binary variables, \( s_{i,f} \) and \( p_{i,j,f,g} \), representing the assignment of residues and residue pairs to fragments.

\[
\begin{align*}
  s_{i,f} &= \begin{cases} 
  1 & \text{if residue } i \text{ is in fragment } f \\
  0 & \text{otherwise}, 
  \end{cases} \\
  p_{i,j,f,g} &= \begin{cases} 
  1 & \text{if residue } i \text{ is in fragment } f \text{ and residue } j \text{ is in fragment } g, \\
  0 & \text{otherwise}, 
  \end{cases}
\end{align*}
\]

where \( 1 \leq i, j \leq \ell \) and \( 1 \leq f, g \leq 2\lambda + 1 \). If \( f \) is even and \( s_{i,f} = 1 \), then residue \( i \) is in the \((-f/2)\)th swapping fragment; otherwise it is in a non-swapping fragment. For efficiency, we only define \( p_{i,j,f,g} \) if there is a contact between \( i \) and \( j \) (so there is a non-zero potential score), and if \( i < j \) and \( f \leq g \) (to avoid redundancy).

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

\[
\Phi = \sum_i \sum_{\text{even } f} s_{i,f} \times \phi_s(i) + \sum_i \sum_{\text{odd } f} s_{i,f} \times \phi_n(i) + \sum_{i,j} \sum_{\text{odd } f,g} p_{i,j,f,g} \times \phi_{nn}(i,j) + \sum_{i,j} \sum_{\text{odd } f, \text{even } g} p_{i,j,f,g} \times \phi_{ns}(i,j) + \sum_{i,j} \sum_{\text{even } f,g \neq g} p_{i,j,f,g} \times \phi_{s2}(i,j).
\]

(4.17)

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

\[
\forall i : \sum_{1 \leq f \leq 2\lambda + 1} s_{i,f} = 1, \quad (4.18)
\]
\[
\forall i, f: \ s_{i,f} + \sum_{f' < f} s_{i+1,f'} \leq 1, \quad (4.19)
\]

\[
\forall \text{even } f: \sum_i s_{i,f} \geq l_{\text{min}}, \quad (4.20)
\]

\[
\forall \text{even } f: \sum_i s_{i,f} \leq l_{\text{max}}, \quad (4.21)
\]

\[
\forall i, j, f \text{ and } i < j: \sum_{g \geq f} p_{i,j,f,g} = s_{i,f}, \quad (4.22)
\]

\[
\forall i, j, g \text{ and } i < j: \sum_{f \leq g} p_{i,j,f,g} = s_{j,g}, \quad (4.23)
\]

\[
\forall \text{even } f \forall i \in B: \ s_{i,f} + s_{i+1,f} \leq 1. \quad (4.24)
\]

Eq. 4.18 guarantees a residue can participate in only one fragment. Eq. 4.19 maintains the sequential order of residues and fragments. Eq. 4.20 and Eq. 4.21 enforce the minimum and maximum fragment length constraints. Eq. 4.22 and Eq. 4.23 ensure consistent single and pairwise assignments; see Claim 4.2 below. Eq. 4.24 guarantees that no swapping fragment crosses the boundary of a swappable region.

**Claim 4.2** For \( i < j \) and \( f \leq g \), Eq. 4.18, Eq. 4.22 and Eq. 4.23 guarantee that \( p_{i,j,f,g} \) is 1 if and only if \( s_{i,f} = 1 \) and \( s_{j,g} = 1 \).

**Proof:** Assume \( p_{i,j,f,g} \) has value 1. By Eq. 4.22, we have \( s_{i,f} \geq 1 \). Then by Eq. 4.18, \( s_{i,f} \) must have value 1. Similarly, by Eq. 4.23 and Eq. 4.18, we get \( s_{j,g} = 1 \).

If \( s_{i,f} = 1 \) and \( s_{j,g} = 1 \), then Eq. 4.22 guarantees that there is a \( g' \geq f \) such that \( p_{i,j,f,g'} = 1 \). It must be the case that \( g' = g \), because otherwise we would have \( s_{j,g'} = 0 \) by Eq. 4.18, since \( s_{j,g} = 1 \). Then we would have \( \sum_{f' \leq g'} p_{i,j,f',g'} = s_{j,g'} = 0 \) by Eq. 4.23, contradicting \( p_{i,j,f,g'} = 1 \). \( \Box \)
Claim 4.3 Any fragment swapping is a solution to our integer program, and any solution to our integer program defines a fragment swapping.

Proof: The first part is straightforward. In a fragment swapping, a residue is in only one fragment as in Eq. 4.18. Residue \( i \) must be in a fragment with index no larger than the one of residue \( i + 1 \), satisfying Eq. 4.19. The length of each swapping fragment is between \( l_{\min} \) and \( l_{\max} \), satisfying Eq. 4.20 and Eq. 4.21. By the definition of Eq. 4.16, the value of \( p_{i,j,f,g} \) satisfies Eq. 4.22 and Eq. 4.23. Finally, a fragment does not cross swappable region boundaries, so Eq. 4.24 is satisfied.

Now assume we have a solution to our integer program, and let us construct a fragment swapping. To do so, we must determine the start and end of each swapping fragment, and ensure that the fragment is of the right size and remains within a swappable region. Let us consider even (swapping) fragment number \( f \) in the solution. By Eq. 4.20 and Eq. 4.21 we have \( l_{\min} \leq \sum_i s_{i,f} \leq l_{\max} \), so \( f \) is of the right size. By Eq. 4.24, its residues do not cross a swappable region boundary. In order to obtain the start and end of \( f \), we must ensure that its residues (i.e., the variables \( i \) with \( s_{i,f} = 1 \)) are consecutive. Assume they aren’t. Then there are two residues \( i, j \), with \( i + 1 < j \), such that \( s_{i,f} = 1, s_{i+1,f} = 0 \) and \( s_{j,f} = 1 \). Considering residue \( i \), by Eq. 4.18 and Eq. 4.19, there is fragment \( e \), with \( f < e \), such that \( s_{i+1,e} = 1 \). Then there must be a residue \( k \), with \( i + 1 \leq k < j \), such that \( k \) is in a fragment with larger index than that of \( k + 1 \), since otherwise residue \( i + 1 \) could not be in a fragment with a larger index than that of residue \( j \). But such a \( k \) would contradict Eq. 4.19. Thus the residues of \( f \) must be consecutive, and we can determine the start and end of \( f \) by finding the minimum and maximum \( i \) with \( s_{i,f} = 1 \). □
Thus by maximizing the objective function, we will find the optimal selection of swapping fragments.

As mentioned in the introduction, traditional site-directed recombination between two proteins in a single family is a special case of fragment swapping. We arbitrarily call one parent protein the source and the other one the target. After aligning the sequences by standard techniques, we have a single swappable region of length $n$ including all residues. We add the constraint $\sum_{i, \text{even}} f s_{i,f} = n$, and Eq. 4.21 is no longer needed. Then the asymmetric swapping will result in a symmetric combinatorial recombination.

4.3 Results and Discussion

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

4.3.1 Selective swapping of beta-lactamases

Beta-lactamases are enzymes produced by some bacteria; they hydrolyze the beta-lactamases found in certain antibiotics (e.g., penicillin). They have been the object of much chimera-
genesis work, including the pioneering site-directed studies of Arnold and colleagues [92, 54]. We have also previously developed experiment planning methods for traditional site-directed recombination and applied them to beta-lactamases [96]. We use here the dataset from our previous study, consisting of 136 beta-lactamases multiply aligned to 263 residues with an average sequence identity of 41.8%, along with the representative 3D structure from *E. coli* TEM-1 beta-lactamase (pdb id 1BTL). We derived the potential score as discussed above; we note that our previous work demonstrated that the potential is predictive of folded and functional hybrids [96]. We used as parents the proteins studied by Arnold, TEM-1 and PSE-4, arbitrarily choosing TEM-1 as source.

We compared the libraries optimized by SWAGMER to randomly generated plans and to plans optimized by our earlier method [96], which we call here “exhaustive” as it covers the entire sequence rather than focusing on specific fragments. Based on the number of residues in TEM-1 and PSE-4, we set the fragment length constraints to be a minimum of 10 and a maximum of 50. We generated plans with 2, 3, or 4 fragments (yielding a manageable sized library) by SWAGMER and the random approach, and plans with the same number of hybrids by the exhaustive approach (2 swapped fragments corresponds to 1 exhaustive breakpoint, etc.). For the random approach, we generated $10^5$ random plans, requiring more than 1 hour for 2 fragments and roughly 2 hours each for 3 and 4 fragments. We implemented SWAGMER using the CBC integer programming solver provided in COIN-OR (https://projects.coin-or.org/Cbc). The running times were 32 seconds (2 fragments), 776 seconds (3 fragments), and 3359 seconds (4 fragments).

The top three panels in Fig. 4.3 summarize the qualities of the resulting plans, in terms...
of average potential scores. Clearly the optimal plan is much better than would be obtained at random, as would be obtained by stochastic recombination rather than a planned approach. By focusing experimental effort on selected fragments, rather than spreading it out over the entire protein, SWAGMER also significantly outperforms the exhaustive approach. Thus the resulting library better explores this region of sequence space, giving us the opportunity to find hybrids that probably would not be generated under other methods.

Fig. 4.3: Comparison of average potential score of libraries constructed by optimal fragment selection and random selections: (top) beta-lactamases, (middle) glutathione transferases, (bottom) purEs; (left) 2, (middle) 3, and (right) 4 fragments. The histogram is taken over $10^5$ random libraries. The red diamond is the SWAGMER-optimized library. The green circle in the beta-lactamase panels is the optimal library for the “exhaustive” approach.

Fig. 4.4(left) illustrates the structure of the swapping plan. It employs minimum-length fragments, perhaps because PSE-4 and TEM-1 are distantly-related beta-lactamases [69]
and thus short fragments are preferable to minimize the disruption introduced by swapping. The exhaustive plans likewise place breakpoints so as to minimize fragment length (breakpoints are stacked up at either the N- or C-terminus). The swapping fragments selected are all within protein modules identified by profile disruption [69], which it is hypothesized must be maintained in recombination to yield folded and functional hybrids.

While it is natural to minimize fragment length in order to maintain the amino acid statistics from the target protein, by swapping too little, we are unlikely to achieve our overarching goal of developing novel proteins combining the “good” parts from both parents. We have previously explored optimizing metrics for both stability and diversity in site-directed recombination, in order to balance these two complementary (and competing) goals [99]. To explore the relationship between these criteria in the new context of fragment swapping, we extended our objective function to include a diversity metric:

\[
\Phi' = w \times (\Phi - \Phi_{\text{min}})/(\Phi_{\text{max}} - \Phi_{\text{min}}) + (1 - w) \times (D - D_{\text{min}})/(D_{\text{max}} - D_{\text{min}})
\] (4.25)

where \(0 \leq w \leq 1\), \(\Phi\) is the average potential score over the library (Eq. 4.17), \(D\) is the average mutation level over the library (below), and we normalize \(\Phi\) and \(D\) according to their ranges in order to render the weight \(w\) interpretable. Note that larger is “better” for both aspects.

\[
D = 1/2^\lambda \times \sum_{h=1}^{2^\lambda} d(P_h, T)
\] (4.26)

where \(\lambda\) is the number of fragments selected for swapping and \(d(P_h, T)\) is the mutation level (number of different amino acids) between hybrid \(P_h\) and target protein \(T\).

Fig. 4.5 demonstrates the effect of \(w\) on the optimal 3-fragment plans. At the extreme
of large $w$, we focus on maximizing the potential score, and thus obtain small fragments generating relatively little diversity. As $w$ decreases, the diversity term contributes more and thus more residues from the source protein are introduced. At the same time, the average potential score is decreased and it is more likely that the hybrids have disrupted important residue interactions. At the extreme of small $w$, we focus on generating diversity and obtain large fragments whose amino acid content is not as consistent with that of the target family. The plans are fairly stable across ranges of $w$, with a couple of discrete jumps. This implies that although diversity and potential score are two competing criteria, a small set of plans capture the key trade-offs, as we have also seen in traditional site-directed recombination [99]. That is, while there are certainly other plans trading off these two criteria, they are not as good as these in one measure or the other, and these plans represent the Pareto optimal points in the 2D potential-diversity space.

Fig. 4.4: Swapping plans relative to the reference structures: (left) beta-lactamases, (middle) glutathione transferases, (right) purEs. Green blocks represent optimal fragment selections for 2, 3, and 4 fragments. Red blocks represent alpha helices and blue blocks represent beta sheets.
4.3.2 Activity swapping in glutathione transferases

Glutathione transferases (GSTs) are enzymes that help eliminate reactive electrophilic compounds by conjugating them to glutathione. As mentioned in the introduction, Griswold et al. recently demonstrated the use of chimeragenesis to swap activity from a rat GST into a human one [25]. They employed stochastic techniques to construct libraries of $\theta$-class GSTs, recombining human GST $\theta$-1-1 (hGSTT1-1) and rat GST $\theta$-2-2 (rGSTT2-2). They identified a hybrid with 83% of the hGSTT1-1 sequence but a swapped-in rat activity. This is a powerful demonstration of the potential for activity swapping, but we show here that optimizing an experiment plan can result in a library with significantly higher average score, while focusing experimental effort on a smaller region in sequence space, thus potentially yielding a much better hit rate.

We started with sequence alignments for the two subclasses (rat and human) of $\theta$-class GSTs, with four sequences each, aligned to 239 residues, with an overall sequence identity of 53%. Given the small number of sequences, we followed our previous sparse data ap-
proach [96], augmenting the family statistics with database statistics, thereby introducing an amino acid-specific pseudocount. Since \( \theta \)-class GSTs have a highly conserved GST 3D fold [25] we used the hGSTT1-1 structure (pdb idb id 2C3N) as the reference structure for both subclasses. We used a weight \( \alpha = 0.5 \) in Eq. 4.3 and 4.4, placing equal importance on maintaining the human scaffold and introducing the rat activity.

The middle three panels of Fig. 4.3 show the comparison between SWAGMER-optimized plans and \( 10^5 \) random ones, for 2, 3, and 4 fragments. As with beta-lactamases, the average potential of the optimal plans is much better than we would get from stochastic plans. The running times are 5 seconds, 44 seconds, and 1067 seconds for 2, 3, and 4 fragments, respectively. Also as with beta-lactamases, the plans seek small fragments (Fig. 4.4(middle)). We see similar trends in trading off potential score and diversity (Fig. 4.6): small fragments when emphasizing potential score, transitioning to large fragments for more diversity, with a small number of plans dominating the space.

![Graph](image)

**Fig. 4.6:** Three-fragment swapping optimization for glutathione transferases trading off potential score vs. diversity, with potential score weight \( w \) decreasing from 1 to 0 by 0.1 from top to bottom. (left) Plans relative to the hGSTT1-1 reference structure. (right) Average library potential score \( \Phi \) and diversity \( D \).

The above plans all used a weight \( \alpha = 0.5 \) in Eq. 4.3 and 4.4, so that maintaining the
human scaffold and introducing the rat activity are equally important. To explore the effect of the weight $\alpha$, we optimized 3-fragment plans with different relative contributions from source (rGSTT2-2) and target (hGSTT1-1) potential scores. We normalized the source and target potentials to the $[0, 1]$ range, based on their minimum and maximum values. As with our potential-diversity study, this renders the weight (here $\alpha$) directly interpretable.

Fig. 4.7 shows the resulting plans. When $\alpha$ is small, the focus is on the target potential, so small fragments are selected to yield hybrids relatively similar to it. Furthermore, the fragments are in the G site (residue position $\leq 77$), where the two classes have a high identity [25]. As $\alpha$ increases, the source potential is more important, yielding fragments that are larger and in the H site (residue position $\geq 89$), where the two classes have lower identity. The resulting average source $\Phi_s$ and target $\Phi_t$ potential score contributions change accordingly (Fig. 4.7 (right)). We again see a few basic plans dominating, with discrete transitions between them. This could be due to a few very important conservation terms, so that planning devolves to a transition among choices for those residues.

### Fig. 4.7: Three-fragment swapping optimization for glutathione transferases with different values of weight $\alpha$ for source and target potential contribution, increasing from 0 to 1 by 0.1 from top to bottom. (left) Swapping plans relative to the hGSTT1-1 reference structure. (right) Average source $\Phi_s$ and target $\Phi_t$ potential scores over the libraries.

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$\Phi_s$</th>
<th>$\Phi_t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>536.7</td>
<td>66.7</td>
</tr>
<tr>
<td>0.1</td>
<td>536.7</td>
<td>66.7</td>
</tr>
<tr>
<td>0.2</td>
<td>536.7</td>
<td>66.7</td>
</tr>
<tr>
<td>0.4</td>
<td>510.3</td>
<td>46.1</td>
</tr>
<tr>
<td>0.5</td>
<td>340.4</td>
<td>162.9</td>
</tr>
<tr>
<td>0.6</td>
<td>319.4</td>
<td>194.8</td>
</tr>
<tr>
<td>0.7</td>
<td>319.4</td>
<td>194.8</td>
</tr>
<tr>
<td>0.8</td>
<td>312.1</td>
<td>229.9</td>
</tr>
<tr>
<td>0.9</td>
<td>311.5</td>
<td>234.8</td>
</tr>
<tr>
<td>1</td>
<td>311.5</td>
<td>234.8</td>
</tr>
</tbody>
</table>
4.3.3 Activity swapping in purEs

Proteins in the purE family catalyze steps in the *de novo* synthesis of purines. While clear homologs, their activities fall into two quite different classes: mutases in a two-step reaction, in eubacteria, fungi and plants (as well as probably most archeabacteria); and carboxylases in a single-step reaction that yields the same product, in metazoans and methanogenic archeabacteria [18]. We have previously explored recombining different sets of parents of these two activities, in order to find the sequence features of the “boundaries” enclosing the distinct activities [99]. Here we consider more directly finding possible ways to swap from one class to the other.

We used our previously collected sequence alignment [99] of 347 mutases and 26 carboxylases, aligned to 162 residues, mapped to the structure of *E. coli* purE (PDB id: 1qcz), and with gappy sequences (> 20% gaps) eliminated. We selected the *Gallus gallus* sequence as a carboxylase source and the *E. coli* sequence as a mutase target. We decreased the maximum fragment length to 30 because the sequence is not long enough for four 50-residue fragments (the maximum we used for the previous families).

We compared optimal fragment swapping plans (at the default $\alpha = 0.5$) with $10^5$ random ones, for 2, 3, and 4 fragments (requiring 6, 38, and 1005 seconds to optimize, respectively). We again found the optimized ones to be superior to what one could hope to stumble upon with stochastic methods (bottom panels of Fig. 4.3).

Fig. 4.4(right) shows the 2-, 3-, and 4-fragment plans, which contrast with those for beta-lactamases and GSTs by selecting non-minimal lengths. Further insights are gained
by exploring the relationship of potential score and diversity, with different weights $w$ in the combined objective function (Eq. 4.25). As Fig. 4.8 shows, consistent with our initial results for 2, 3, and 4 fragments, the fragments tend to be large and don’t significantly change over the range of weights. One possible reason is that, in contrast with beta-lactamases and glutathione transferases, the range of diversity values here is smaller, so it is acceptable to swap more residues. At the same time, the changes in the average potential score and diversity values (right of Fig. 4.8) show that there are still variations in the underlying objective function, with these representative plans dominating the space.

Fig. 4.9 illustrates the effect of balancing the source and target contributions (weight $\alpha$). Here we see trends similar to those in the other two families, with small fragments dominating target-oriented plans and large ones dominated source-oriented ones.

The patterns of transitions in plans for all three case studies suggests a line for further investigation: using the small set of discretely different plans as an investigative tool to uncover and verify the patterns of conservation accounting for the relationship between sequence, structure, and function.

### 4.4 Conclusion

We have developed a new general framework for recombination, protein fragment swapping. By swapping only selected discontiguous regions, fragment swapping can focus on functionally important regions in parent sequences, is applicable to parents with heterogeneous structures, and is flexible in the number of residues participating in recombination.
Fig. 4.8: Three-fragment swapping optimization for purEs trading off potential score vs. diversity, with potential score weight $w$ decreasing from 1 to 0 by 0.1 from top to bottom. (left) Plans relative to the reference structure from *E. coli* purE. (right) Average library potential score $\Phi$ and diversity $D$.

<table>
<thead>
<tr>
<th>$w$</th>
<th>$\Phi$</th>
<th>$D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-614</td>
<td>21.5</td>
</tr>
<tr>
<td>0.9</td>
<td>-615.9</td>
<td>24</td>
</tr>
<tr>
<td>0.8</td>
<td>-615.9</td>
<td>24</td>
</tr>
<tr>
<td>0.7</td>
<td>-641.4</td>
<td>31</td>
</tr>
<tr>
<td>0.6</td>
<td>-644.9</td>
<td>31.5</td>
</tr>
<tr>
<td>0.5</td>
<td>-644.9</td>
<td>31.5</td>
</tr>
<tr>
<td>0.4</td>
<td>-644.9</td>
<td>31.5</td>
</tr>
<tr>
<td>0.3</td>
<td>-644.9</td>
<td>31.5</td>
</tr>
<tr>
<td>0.2</td>
<td>-692</td>
<td>32.5</td>
</tr>
<tr>
<td>0.1</td>
<td>-721.3</td>
<td>32.5</td>
</tr>
</tbody>
</table>

Fig. 4.9: Three-fragment swapping optimization for purEs with different values of weight $\alpha$ for source and target potential contribution, increasing from 0 to 1 by 0.1 from top to bottom. (left) Swapping plans relative to the reference structure from *E. coli* purE. (right) Average source $\Phi_s$ and target $\Phi_t$ potential scores over the libraries.

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$\Phi_s$</th>
<th>$\Phi_t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-716.9</td>
<td>-25.7</td>
</tr>
<tr>
<td>0.1</td>
<td>-716.9</td>
<td>-25.7</td>
</tr>
<tr>
<td>0.2</td>
<td>-690.3</td>
<td>-29.2</td>
</tr>
<tr>
<td>0.3</td>
<td>-632.4</td>
<td>-43.8</td>
</tr>
<tr>
<td>0.4</td>
<td>-601.6</td>
<td>-55.4</td>
</tr>
<tr>
<td>0.5</td>
<td>-479.8</td>
<td>-134.3</td>
</tr>
<tr>
<td>0.6</td>
<td>-479.8</td>
<td>-134.3</td>
</tr>
<tr>
<td>0.7</td>
<td>-377.7</td>
<td>-263.7</td>
</tr>
<tr>
<td>0.8</td>
<td>-377.7</td>
<td>-263.7</td>
</tr>
<tr>
<td>0.9</td>
<td>-377.7</td>
<td>-263.7</td>
</tr>
<tr>
<td>1</td>
<td>-377.3</td>
<td>-267.7</td>
</tr>
</tbody>
</table>
Furthermore, the asymmetric role of the source and target parents enables specific construction of libraries seeking to introduce activities from one parent into the other. Our SWAGMER method provides an efficient, effective approach to optimizing fragment swapping experiments.

4.5 NP-hardness of Protein Fragment Swapping

We prove the NP-hardness of the protein fragment swapping problem by reduction from MAX-2SAT. For simplicity our construction uses a single swappable region, only a pairwise potential score $\phi_2$, and trivial fragment length constraints $l_{\text{min}} = 1$ and $l_{\text{max}} = \infty$.

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

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

- **Clause**: for each clause $C_r = (c_{r,1} \lor c_{r,2})$ with literals $c_{r,1}$ and $c_{r,2}$, add two residues $v_{r,1}$ and $v_{r,2}$ sequentially.

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

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

We need not specify the amino acid sequences for the source and targets, as the swapping problem is defined in terms of the potential. To this end, there are four types of residue pairs contributing to the potential, with \(g_{i,j}\) values in Tab. 4.1 yielding \(\phi_2\) values in Tab. 4.2 according to Eq. 4.9–Eq. 4.14.

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

Fig. 4.10 illustrates one construction.

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

Now we prove a correspondence between the MAX-2SAT solution for \(C_1 \land C_2 \land \ldots \land C_\tau\) and the optimal fragment swapping for the constructed \(2\tau + 2N_+\) swapping fragments. We separate the two directions of the proof.
\begin{table}
\centering
\begin{tabular}{|c|c|c|c|c|}
\hline
 & \(g_{i,j}(T[i], T[j])\) & \(g_{i,j}(S[i], T[j])\) & \(g_{i,j}(T[i], S[j])\) & \(g_{i,j}(S[i], S[j])\) \\
\hline
clause & 0 & 2 & 2 & 0 \\
identical & 1 & -1 & -1 & 5 \\
complementary & 0 & 2 & 2 & -4 \\
separator & -3 & 3 & 3 & -3 \\
\hline
\end{tabular}
\caption{Family statistics for different types of residue pairs.}
\label{tab:family_statistics}
\end{table}

\begin{table}
\centering
\begin{tabular}{|c|c|c|c|c|}
\hline
 & \(\phi_{nn}\) & \(\phi_{ns}\) & \(\phi_{sn}\) & \(\phi_{s1}\) & \(\phi_{s2}\) \\
\hline
clause & 0 & 1 & 1 & 0 & 1 \\
identical & 1 & 0 & 0 & 3 & 1 \\
complementary & 0 & 1 & 1 & -2 & 0 \\
separator & -3 & 0 & 0 & -3 & 0 \\
\hline
\end{tabular}
\caption{Average potential scores for different types of residue pairs.}
\label{tab:average_potential_scores}
\end{table}

\begin{figure}
\centering
\includegraphics[width=\textwidth]{fig4_10.png}
\caption{Residue pairs contributing to the potential \(\phi_2\) for the MAX-2SAT instance \((x \lor y) \land (y \lor \neg x) \land (z \lor y)\). Filled dots are residues mapping to literals in the clauses. Empty dots are separator residues. Squares represent trivial residues at the end. There are 3 clause pairs (blue solid), 3 identical pairs (red dashed above), 1 complementary pair (purple dashed below), and 3 separator pairs (black dashed).}
\label{fig:residue_pairs}
\end{figure}
Claim 4.4 If the MAX-2SAT solution satisfies $k$ clauses, then the fragment swapping solution achieves an average potential score of $k + N_+ + N_-$. 

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

Following Tab. 4.2, we have the following contributions to the potential score:

- clause: 0 for each unsatisfied clause (so that neither residue is in a swapping fragment); 1 for each satisfied clause. Note that $\phi_{s1}$ cannot happen since we only have single-residue fragments, and each of the remaining possibilities yields 1.

- identical: 1 each. We must have either $\phi_{nn}$ or $\phi_{s2}$, each of which yields 1. $\phi_{s1}$ cannot happen due to the separator residue pairs between the two identical residues.

- complementary: 1 each. We must have either $\phi_{ns}$ or $\phi_{sn}$, each of which yields 1.

- separator: 0 each

The total is $k + N_+ + N_-$. □

Claim 4.5 If the fragment swapping solution achieves an average potential score of $k - N_+ - N_-$, then the MAX-2SAT solution satisfies $k$ clauses.
**Proof:** We must show how to find an assignment of literals based on the fragment swapping solution. To do this, we show that separator pairs contribute 0 to the potential, while identical and complementary pairs contribute 1 each. Thus there must be $k - N_+ - N_-$ clause pairs contributing 1 each (the only non-zero possibility in Tab. 4.2). By mapping the swapped residues to literals, we can determine which literals are true and which $k - N_+ - N_-$ clauses are satisfied.

We first prove that each separator pair contributes 0. Assume for contradiction that some separator pair contributes $-3$ (the only other possibility in Tab. 4.2). Let us modify the swapping by making two single-residue fragments for the two separator residues, increasing the potential score by 3. If this increases the total number of swapping fragments above $2\tau + 2N_+$, then there must be some swapping fragments in the trivial residues (since there are only $2\tau + 2N_+$ residues in the main sequence), some of which we can eliminate to leave the total at $2\tau + 2N_+$. The change does not affect the potential contributed by any clause pair. An identical pair can be affected if it involves the same literal as the separator pair and the two residues were previously in the same swapping fragment as the separator residue pair. In that case, the change replaces a single swapping fragment with separate swapping fragments, decreasing the potential by $\phi_{s1} - \phi_{s2} = 2$, which is outweighed by the increase of 3. (If multiple identical pairs are affected, then they are balanced by a corresponding number of separator pairs.) The possible analogous effect on a complementary pair can only be beneficial, yielding a net increase of $\phi_{s2} - \phi_{s1} = 2$. Thus we have increased the total potential, contradicting our assumption that $k$ is the maximum.

Now let us show that identical and complementary pairs contribute 1 each. They must
contribute either 0 or 1, since by the above the separator pairs contribute 0 and thus must break up any swapping fragment, eliminating the $\phi_{s1}$ possibility in Tab. 4.2. If any pair contributes 0, we can modify the swapping as follows to make they all contribute 1. Let $V$ be the set of residues for all the literals involving a particular variable (either the variable or its negation). Let $V_+$ be the residues for the variable and $V_-$ for its negation. Let $V_{+s}$ and $V_{+n}$ partition $V_+$ into residues in swapping fragments and those not in swapping fragments, respectively; and similarly with $V_{-s}$ and $V_{-n}$. By Tab. 4.2, each residue pair in $V_{+n} \cup V_{-s}$ contributes 1 (for either the identical or complementary term, as appropriate), and similarly for each residue pair in $V_{+s} \cup V_{-n}$. The remaining residue pairs (one in $V_{+s} \cup V_{-n}$ and one in $V_{+n} \cup V_{-s}$) each contribute 0. Now let us complement the swapping assignment for each residue in $V_{+s} \cup V_{-n}$—if the residue is in a swapping fragment, shorten or break the fragment to make this residue not in swapping fragments; if it is not, create a single-residue swapping fragment. (As discussed above, we can modify the swapping in the trivial residues to ensure that the total number of swapping fragments is $2\tau + 2N_+$.)

Following the above discussion, this change won’t affect the potential contributed by separator pairs. It decreases the contribution from clause pairs with residues in $V$ by at most 1 and doesn’t affect other clause pairs. Pairs in $V_{+s} \cup V_{-n}$ still contribute 1, but pairs (identical or complementary) between $V_{+s} \cup V_{-n}$ and $V_{+n} \cup V_{-s}$ now contribute 1 instead of 0. The total increase is $|V_{+s} \cup V_{-n}| \times |V_{+n} \cup V_{-s}|$, while the total decrease is at most $|V_{+s} \cup V_{-n}|$. We have $|V_{+s} \cup V_{-n}| \times |V_{+n} \cup V_{-s}| \geq |V_{+s} \cup V_{-n}|$. In this manner, we can change all identical and complementary pairs to contribute 1, which means the swapping fragment assignments of residues for these pairs are consistent.
5. PROTEIN DEIMMUNIZATION BY FRAGMENT SWAPPING

Therapeutic proteins have revolutionized the treatment of a number of diseases. Unfortunately, most therapeutic proteins have been shown to induce an immune response, causing loss of efficacy and undesirable side effects. One key challenge in therapeutic protein engineering is to decrease immunogenicity while simultaneously preserving therapeutic activity. This chapter focuses on one aspect of immunogenicity, the recognition of epitopes by class II MHC molecules to initiate the T-cell response. We develop a dynamic programming algorithm that identifies sequence fragments to be combinatorially swapped from a human source protein into a therapeutic target protein in order to delete predicted epitopes while maintaining predicted activity. Since immunogenicity reduction and activity preservation are competing aspects, we optimize linear combinations of epitope scores and conservation scores, in order to characterize the trade-offs. To ensure overall maintenance of stability and activity, fragment swapping is constrained by an alignment between the source and target, based on sequence or structural similarity. We demonstrate the effectiveness of our method in case studies on both an antibody (mouse anti-CD30 antibody AC10) and an enzyme (glutathione transferase). Compared with variants in previous studies, hybrids generated based by fragment swapping have better performance in both epitope
and conservation scores. At the same time, they properly incorporate key determinants of immunogenicity reduction and activity preservation identified in the experimental studies.

5.1 Introduction

Deimmunization of exogenous proteins to generate therapeutic agents with low immunogenicity poses a significant challenge in protein engineering. Engineered in the laboratory for pharmaceutical use, therapeutic proteins have been used in the treatment of a number of diseases [59, 62, 63, 19]. However, together with the clinical benefits, most therapeutic proteins have been shown to induce an immune response [72, 75], which causes the loss of clinical efficacy along with undesirable side effects. Theoretically, any foreign protein introduced into the human body may trigger an immune response. There are many factors affecting protein immunogenicity, such as structural features [23, 36], sequence variation and genetic characteristics [2, 79], and so forth. Here we focus on sequence variation between exogenous proteins and native human proteins. Intuitively, the more exogenous proteins differ from native human proteins, the more immunogenic they are likely to be.

The immune response is initiated by activation of helper T-cells that recognize antigens bound to Class II MHC molecules. Human leukocyte antigen-group DR (HLA-DR), the predominant isotype of the human class II major histocompatibility complex (MHC), plays a central role in helper T-cell selection and activation. Proteins of HLA-DR bind peptide fragments, or epitopes, derived from protein antigens, and display them on the surface of antigen-presenting cells for interaction with antigen-specific receptors of T lymphocytes.
Thus, to deimmunize exogenous proteins, a key step is to identify immunogenic peptide fragments and introduce mutations that will delete the epitopes, thereby decreasing the possibility of binding with the MHC II for activation of helper T-cells.

Various approaches have been developed by which therapeutic proteins are modified to reduce their immunogenicity. Deimmunization of antibodies has been extensively studied because of their common structural organization, with two heavy chains and two light chains. In [57, 34, 91], chimeric antibody variants are constructed by recombination of fragments from exogenous proteins and human sequences. Deimmunization by chimera construction can introduce a large number of mutations into an exogenous protein. However, by generating chimeric variants stochastically [57], only a small fraction of the chimera library is likely to retain high activity and be deimmunized. CDR grafting [34] is an improvement of the stochastic method, evaluating global sequence identity between exogenous and human proteins to evaluate immunogenicity. But practically, the immune response is triggered by the recognition of epitopes, short linear peptides with a small number of residues. Maximizing global identity in CDR grafting places unnecessary restrictions on protein engineering and limits the sampling of the critical amino acid diversity.

Another approach to protein deimmunization is to make site-specific mutations to delete epitopes [28, 35, 47, 71]. This approach requires a precise understanding of the relationship between structure and activity, as well as immunogenicity, in order to select critical residues that can be mutated to simultaneously disrupt MHC II binding while preserving activity. Unfortunately it is hard to maintain stability and activity with an increasing mutational load; some studies have shown that the fraction of functional mutants declines exponentially with
the number of mutations [12, 31]. As a result, the sequence space that can be explored by site-specific mutations is limited.

We develop here a general method for protein deimmunization by aligned fragment swapping from human native sequences into exogenous proteins based on sequence or structural homology (Fig. 5.1). Presumably peptides from human proteins exhibit low immunogenicity. By replacing peptides in exogenous proteins with appropriate human sequence fragments, chimeric variants can be generated with reduced immunogenicity. At the same time, by preserving important residues in the exogenous proteins, the desired activities can be maintained in the chimeras. An alignment based on sequence or structural similarity makes it more likely to preserve important amino acids when swapping fragments from human to exogenous proteins. Since immunogenicity and activity are two competing aspects to the deimmunization goal, we evaluate a swapping plan by two different metrics, epitope score and conservation score. A dynamic programming algorithm seeks to optimize a specified linear combination of these scores. By changing weights in the...
combination, the trade-offs between these aspects may be explored, and optimal swapping fragments may be selected according to different experimental requirements.

Different from traditional recombination, deimmunization by fragment swapping is an asymmetric recombination similar to SWAGMER (Chapter 4) [101]. An exogenous protein is taken as the target protein for fragment swapping, while a human sequence is the source protein, providing fragments to be swapped into the exogenous framework. Different from [57], deimmunization by fragment swapping constructs chimeras in a directed way to optimize both immunogenicity and functionality. Though similar to SWAGMER, our goal here is different, and thus a different algorithm is required. In SWAGMER, there are two conservation scores, one for the target and one for source proteins; these metrics are of the same format. Here, the optimization metrics have different origins and our objective is to optimize both of them simultaneously.

We demonstrate the effectiveness of deimmunization by fragment swapping in two very different case studies of proteins previously targeted for humanization: mouse anti-CD30 antibody AC10 [46] and rat glutathione transferase [25]. We varied the weights on epitope and conservation scores, to study the change in optimized fragments as the focus moved between reduction of immunogenicity and maintenance of activity. We compared the resulting specified hybrid libraries with variants that had previously been experimentally tested, and found that our hybrids are predicted to have better epitope and conservation scores. We found significant consistencies between our hybrids and the previous variants, including specific mutations and fragments that were experimentally found to be important for activity and immunogenicity. Our optimized hybrid libraries provide promising new
5.2 Methods

The objective of deimmunization by fragment swapping is to replace immunogenic peptides in therapeutic proteins with human sequence fragments (Fig. 5.1), seeking to minimize immunogenicity while maintaining therapeutic activity. In this approach, deimmunization is evaluated by a 9-mer epitope score, predicting the occurrence of T-cell epitopes in a protein sequence. The higher the epitope score, the more likely that the protein sequence will elicit an immune response. To keep the desired therapeutic activity, it is necessary to maintain important amino acids in the exogenous target; we evaluate that by a conservation score calculated from a family of evolutionarily related proteins. The higher the conservation score, the higher the probability that therapeutic activity will be preserved in the resulting hybrids.

5.2.1 Epitope Score

To evaluate the immunogenicity of a protein sequence, we employ T-cell epitope predictors based on human leukocyte antigen group DR (HLA-DR) of MHC II proteins. HLA-DR is the predominant isotype, and these proteins play a central role in helper T-cell selection and activation. HLA-DR proteins have a recognition groove in which they bind short peptide fragments processed from protein antigens [21]. The recognition groove consists of
pockets enabling energetically favorable interactions with specific side-chains of peptides approximately 9 residues in length [86]. The pocket specificity can be characterized functionally by substituting the corresponding peptide ligand position with all natural amino acid residues and quantifying their effects on binding. Sturniolo et al. [86] measured the binding affinity of the MHC II binding groove on a limited set of alleles, and incorporated the resulting “pocket profiles” in the TEPITOPE tool for epitope prediction. To recognize a 9-mer peptide as epitope with pocket profiles, the sum of position-specific weights for each residue in the 9-mer provides a score that is compared against a threshold to determine whether or not the peptide is in a given percentage of the best-recognized peptides. In an analysis of TEPITOPE predictions against HLA-DR selected and nonselected peptide repertoires, TEPITOPE was shown to achieve a prediction accuracy of 80% with a false positive rate less than 5%. Later, Singh and Raghava extended TEPITOPE to build the ProPred tool [83]. ProPred has been applied in several different studies with significant results [14, 70, 39, 58], facilitating the rapid identification of potential vaccine targets that were then experimentally characterized in detail. In this work, we applied the ProPred scoring matrices with a 10% threshold; however, our optimization algorithm is generic enough to support other metrics like SMM-Align [60].

There are over 50 different HLA-DR alleles and each one has its own pocket profile. As in an earlier study [71], we considered the eight most common alleles (DRB1*0101, DRB1*0301, DRB1*0401, DRB1*0701, DRB1*0801, DRB1*1101, DRB1*1301, and DRB1*1501), which represent the majority of human populations world-wide. Thus we compute the epitope score of a 9-mer \( m \) as the fraction of these 8 alleles predicted to rec-
ognize it (at the given threshold):

\[ e(m) = \sum_{i=1}^{8} e_i(m) / 8, \]

(5.1)

where indicator function \( e_i(m) \) is 1 if the corresponding allele recognizes 9-mer \( m \) and 0 if not.

We can define an epitope score for an entire \( n \)-residue protein sequence \( P \) by summing the epitope score over all its 9-mers

\[ e(P) = \sum_{i=1}^{n-8} e(P[i, i+8]). \]

(5.2)

5.2.2 Conservation Score

Fragment swapping introduces multiple simultaneous mutations into the target sequence. We must be careful that these mutations do not result in an unfolded or unfunctional hybrid. To evaluate hybrid quality from the perspective of stability and functionality, we adopt a standard position-dependent residue conservation score from our earlier work [96, 99, 101].

Statistical analysis of amino acid frequencies in protein family provides a metric to evaluate the possible effects of amino acid substitutions.

Given a multiply-aligned set of sequences of a protein family \( \mathcal{F} \), conservation score \( \phi_r(a) \) is defined as the log probability of amino acid type \( a \) at residue position \( r \) for all sequences \( P \) in \( \mathcal{F} \):

\[ \phi_r(a) = \log \frac{|\{ P \in \mathcal{F} : P[r] = a \}|}{|\mathcal{F}|}. \]

(5.3)

In fragment swapping optimization, \( \mathcal{F} \) is the protein family of the target protein or a multiple sequence alignment of functionally similar proteins including the target protein.
In addition to conservation information in a closely related family, we can also estimate the general frequency of amino acids in a larger database $\mathcal{D}$ of protein sequences. Database score $\phi_\mathcal{D}(a)$ is the log probability of amino acid type $a$ among the amino acids in the database sequences.

$$\phi_\mathcal{D}(a) = \log \frac{\text{number of } a \text{ in } \mathcal{D}}{\text{number of amino acids in } \mathcal{D}}. \quad (5.4)$$

Estimation of probabilities from frequencies is valid only if the frequencies are large. Though frequencies in the protein family are more valuable, as they capture important evolutionary information about the target protein, bias may also be introduced with a relatively small family size. To make the conservation score evaluation robust and family-specific, we adopted the method in [96] to combine family-specific information together with the background frequencies in the database:

$$\phi_r(a) = w \times \phi_r(a) + (1 - w) \times \phi_\mathcal{D}(a), \quad (5.5)$$

where $w (0 \leq w \leq 1)$ is the weight to control contributions of family-specific and database information. By changing $w$, we can obtain a probability distribution that is close to the overall database distribution for a small family and also approximates the family distribution for a large one. We adopted $w = 0.9$ in the results presented below.

We can compute the total conservation score for an entire $n$-residue protein sequence $P$ by summing over its residue positions:

$$\phi(P) = \sum_{r=1}^{n} \phi_r(P[r]). \quad (5.6)$$

In addition to single-position conservation, it is also desirable to account for amino acid interactions that may underlie protein folding and function. However, we have proved that
optimization of fragment swapping with a pair-wise potential is NP-hard [101]. While the integer programming approach developed there was effective in practice, it does not readily scale up to incorporate the large number of constraints and variables that would be required to evaluate epitope scores. Thus we consider only single-position conservation in order to make the optimization tractable.

5.2.3 Optimization of Fragment Swapping Libraries

A set of swapping fragments defines a library of hybrids, to be constructed by combinatorially mixing and matching the source and target swapping fragments within the target scaffold. We evaluate the overall quality of a library in terms of the quality of each of its hybrids. Our optimization objective is thus to minimize the average epitope score and maximize the average conservation score, taken over the hybrids in the library:

Given aligned source protein $S$ (human protein sequence) and target protein $T$ (exogenous therapeutic protein sequence) of $n$ residues, select $\lambda$ swapping fragments to maximize the linear combination of average conservation and epitope score of the resulting hybrid library, evaluated as

$$\frac{\alpha}{2^\lambda} \times \sum_{i=1}^{2^\lambda} \sum_{r=1}^{n} \phi_r(H_i[r]) \quad \text{and} \quad \frac{1 - \alpha}{2^\lambda} \times \sum_{i=1}^{2^\lambda} \sum_{r=1}^{n-8} e(H_i[r..r+8]).$$  \hspace{1cm} (5.7)

Here $H_i$ is a hybrid generated by fragment swapping, $\phi_r(H_i[r])$ is the conservation score contribution from residue $r$ of $H_i$, $e(H_i[r..r+8])$ is the epitope score contribution.
from 9-mer \((r, r + 1, ..., r + 8)\) of \(H_i\), and \(\alpha\) (where \(0 \leq \alpha \leq 1\)) is the weight to control the relative contributions from epitope and conservation scores.

To compute this objective function, it is not necessary to actually enumerate all the hybrids in the defined library. Instead, the average epitope score and average conservation score can be computed directly from the specification of a swapping plan. The contribution to the conservation score from a residue position depends simply on whether or not it is in a swapping fragment. If it is, then its contribution is the average of the conservation scores for the source and target proteins at that position; if it isn’t, then its contribution is simply that of the target. For example, in Fig. 5.2, residue \(r\) is in a swapping fragment, so both source amino acid “T” and target amino acid “N” will occur in equal proportions in the library. On the other hand, for residue \(r’\), only target amino acid “T” will be represented in the hybrid library. For the average epitope score calculation of a 9-mer, we need to know which of its residues are in swapping fragments and which aren’t, and for those in swapping fragments, we need to know which are in the same swapping fragment. For example, in Fig. 5.2, the two swapping fragments are sequentially contiguous. To calculate epitope score of the 9-mer \((r, r + 1, ..., r + 8)\) covering both of these two fragments, we must know the boundary between these two swapping fragments. Given such a specification of swapping fragments, we know which 9-mers will be in the library, and then can average their epitope scores.

To specify the residue swapping states and thus efficiently compute average conservation and epitope score contributions (without enumerating hybrids), we define a set \(W = \{0, 1, 2\}\) of 3 values to represent the swapping state for each residue. Then the
swapping state $w$ for a residue $r$ is interpreted as:

\[
  w = \begin{cases} 
    0 & r \text{ is not in any swapping fragment} \\
    1 & r \text{ is in the same swapping fragment as } r - 1 \\
    2 & r \text{ is the beginning residue of a swapping fragment}
  \end{cases}
\] (5.8)

Then the overall fragment swapping can be represented as a sequence $(w_1, w_2, \ldots, w_n) \in \mathcal{W}^n$. To guarantee that the individual residue states are consistent, $w_1 \neq 1$, and if $w_r = 0$ then $w_{r+1} \neq 1$.

Then in the hybrid library constructed by fragment swapping, the average conservation contribution from residue position $r$ with state value $w$ is:

\[
  I_\phi(r, w_r) = \begin{cases} 
    \phi_r(T[r]) & w_r = 0 \\
    (\phi_r(T[r]) + \phi_r(S[r]))/2 & \text{otherwise}
  \end{cases}
\] (5.9)

and the average epitope contribution from 9-mer $(r, r+1, \ldots, r+8)$ with 9 swapping states
\[ W = (w_r, w_{r+1}, \ldots, w_{r+8}) \in \mathcal{W}^9 \] is:

\[ I_e(r, W) = \sum_{m \in M} e(m) / |M|, \quad (5.10) \]

where \( M \) is the set of amino acid occurrences in 9-mer \((r, r+1, \ldots, r+8)\), which is decided by \( S[r..r+8], T[r..r+8], \) and \( W \).

### 5.2.4 Swapping Optimization by Dynamic Programming

To find an optimal fragment selection, it is not practical to enumerate all possible fragment selections and score each. Here, we develop an efficient dynamic programming algorithm that takes advantage of the optimal substructure in the problem (Fig. 5.3): the library of optimal score for a 9-mer from residue \( r \) with swapping state \( W \) must extend the library of optimal score for a 9-mer from residue \( r-1 \) with swapping state \( W' \) (properly overlapping \( W \)) with an appropriate score increment.

![Fig. 5.3: Optimal substructure of fragment swapping optimization by dynamic programming.](image)

For dynamic programming, let us recursively define \( A(r, W) \) as the optimal value for...
fragment selection through 9-mer \((r, r+1, \ldots, r+8)\). \(W = (w_r, w_{r+1}, \ldots, w_{r+8})\) gives the swapping states for the residues in this 9-mer. Then we have:

\[
A(r, W) = \begin{cases} 
\alpha \times \sum_{1 \leq i \leq 9} I_\phi(i, w_i) - (1 - \alpha) \times I_e(1, (w_1, w_2, \ldots, w_9)), & \text{if } r = 1, \\
\min_{w_{r-1} \in W} \left( A(r - 1, w_{r-1} \cdot W_{1..8}) \alpha \times I_\phi(r + 8, w_{r+8}) - (1 - \alpha) \times I_e(r, W) \right), & \text{otherwise.}
\end{cases}
\]

where \(\cdot\) represents concatenation and \(W_{1..8}\) represents the first eight elements of \(W\) as \((w_r, w_{r+1}, \ldots, w_{r+7})\). So \(w_{r-1} \cdot W_{1..8}\) specifies the swapping state of the 9-mer beginning from residue \(r - 1\). As discussed in the previous section, knowing the swapping states is sufficient to compute the score contributions.

As swapping fragments are sequentially contiguous residues, here insertion and deletion in source and target sequence alignment are not considered. Our dynamic programming algorithm can be easily extended to accommodate insertion and deletion with some penalties in the score calculation.

To compute this recurrence by dynamic programming requires a table of size \(\lambda \times (n - 8)\) (recall that \(\lambda\) is the number of swapping fragments and \(n\) is the sequence length). Each entry depends on at most 3 previous entries for optimal value (\(w_{r-1}\) has 3 possibilities in Eq. 5.11). The increments for epitope and conservation score can be pre-calculated. \(W\) for each 9-mer has 6765 cases, so epitope score increment calculation has complexity \(O(6765n) = O(n)\). Conservation score increment only has 2 cases for each residue, so it is \(O(n)\) for the whole sequence. Finally the total complexity is \(O(\lambda n + n)\).
5.3 Results

We studied the performance of protein deimmunization by fragment swapping in two case studies of exogenous proteins that had previously been humanized by incorporating either mutations or fragments from homologous human proteins: mouse anti-CD30 antibody AC10 VL [46] and rat glutathione transferase [25]. In general, hybrids designed computationally by our optimization method score better than the earlier variants under both epitope and conservation scores, while also displaying striking consistency with those variants in some key choices of residues and fragments.

5.3.1 Deimmunization of mouse anti-CD30 antibody AC10 VL

Antibody (Ab) therapeutics have demonstrated significant clinical efficacy in a range of applications. In antibody engineering, molecular biologists alter the amino acid sequence of antibody molecules to improve their characteristics. As a first step in Ab immunogenicity reduction, chimeric antibodies are constructed by linking murine variable and human constant regions [57]. The chimeras must then be deimmunized. Lazar et al. [47] developed a method for antibody deimmunization by introducing separate amino acid mutations to maximize similarity between an antibody target and aligned human proteins. The deimmunization objective is to maximize the human string content (HSC), which evaluates similarity between the therapeutic variant and human proteins, summed over overlapping windows of the sequence (analogous to the peptides employed in epitope score calculation). Recently Lazar et al. applied their method to generate full-length variant antibodies.
optimized for human string content [46].

To evaluate the effectiveness of fragment swapping, we study one of the antibody targets: anti-CD30 antibody AC10 VL domain. In [46], 39 variants of AC10 VL were generated by HSC-optimizing mutations from human germline sequences. These 39 variants were classified into 7 clusters based on their mutation distances. For comparison, we selected 13 sequences covering all clusters.

Lazar et al. employed residue masks to prohibit mutations from positions and regions that were deemed most responsible for antigen affinity. This is a basic approach to preserving the therapeutic activity of the variants. We employed their masks to prohibit swapping at those same positions; our conservation score also seeks to minimize structural and functional perturbation throughout the protein.

We used the structural model of the mouse AC10 variable region in [46] as the target protein and human germline Vlk sequence Vlk_1-5 as the source protein. We used PSI-BLAST to obtain a multiple sequence alignment including the mouse antibody AC10. After removing redundant proteins with sequence identity > 90%, we had 103 sequences aligned to 111 positions, which we used to calculate the conservation score.

We optimized 2, 3 and 4-fragment swapping plans for AC10 VL and Vlk_1-5. Similar to [99], we considered different values of $\alpha$ (the relative weight between conservation and epitope scores) so as to find undominated fragment selections—those for which no other fragment selection is better in both epitope and conservation scores. Fig. 5.4 (left) shows the optimal fragment selections. Compared with variants constructed in [46], human mutations introduced from Vlk_1-5 are also covered by the fragments. As weight $\alpha$
for conservation score is decreased from 1 to 0, swapping fragments are selected in regions where AC10 VL and Vlk_1-5 are more different (residue positions \(< 20\) and \(> 80\)), seeking to minimize epitope score. This is consistent with the idea of deimmunization by fragment swapping that amino acid fragments in human protein sequences are more acceptable by immune system.

The average conservation and epitope scores for hybrid libraries constructed by optimal fragment selections are plotted in Fig. 5.4 (right). From bottom left to top right in each line, the weight \(\alpha\) is increased from 0 to 1, changing the focus from epitope score to conservation score accordingly. The trade-offs between the average epitope and conservation scores demonstrates their competing relationship in optimization. Fig. 5.5 explores the performance of individual hybrids in these libraries. Compared with the target sequence AC10 VL and 13 deimmunized variants constructed in [46], fragment swapping optimization generated hybrid libraries with improvement in both conservation and epitope scores.

5.3.2 Deimmunization of glutathione transferases

Glutathione transferases (GSTs) are enzymes that help eliminate reactive electrophilic compounds by conjugating them to glutathione. Griswold et al. [25] recently demonstrated the use of chimeragenesis to swap activity from a rat GST into a human one. They employed stochastic techniques to construct libraries of \(\theta\)-class GSTs by swapping activity from rat GST \(\theta\)-2-2 (rGSTT2-2) to human GST \(\theta\)-1-1 (hGSTT1-1).

Here we try to generate GST variants from another perspective, as deimmunization of
Fig. 5.4: Fragment swapping optimization for deimmunization of mouse anti-CD30 antibody AC10 VL. Top row: 2 fragments; middle row: 3 fragments; bottom row: 4 fragments. Left: selected fragments, with $\alpha$ decreasing from 1 to 0 from top to bottom. Blue circles specify positions for residue masks applied in [46]. Red stars in the first row specify residue positions of mutations introduced from human germline sequence Vlk,1-5 in AC10 VL variants generated in [46]. Right: epitope and conservation scores, averaged over the libraries (cyan circles), for Lazar et al. variants (red stars), and for the wild-type (green diamonds).
Fig. 5.5: Conservation and epitope scores for hybrids constructed by optimal fragment selection (blue points), wild-type AC10 VL (green diamonds) and deimmunized variants (red stars) in [46]. Top: 2 fragments; middle: 3 fragments; bottom: 4 fragments.
rGSTT2-2 by swapping fragments from hGSTT1-1. We adopted sequence alignments for the two subclasses (rat and human) of \( \theta \)-class GSTs, with four sequences each, aligned to 239 residues. Conservation score evaluation is based on GST \( \theta \)-2-2, which includes rGSTT2-2.

Fig. 5.6 shows the optimized fragments and library scores, while Fig. 5.7 details the individual hybrids. As optimization focus is changed from conservation to epitope scores (from top to bottom in Fig. 5.6 (left)), larger fragments are selected in order to introduce more mutations from human GST \( \theta \)-1-1. Similar optimization trade-offs between epitope and conservation scores are also observed (Fig. 5.6 (right)).

Chimera SCR23 generated in [25] is constructed by two fragments from hGSTT1-1 (residues 1–87 and 154–239) and one fragment from rGSTT2-2 (residues 88–153). Here we have a very interesting observation. In Fig. 5.6, when epitope score becomes the optimization focus, fragments selected are very similar to SCR23. In particular, the last two optimal fragment selections for 2-fragment swapping (same as SCR23), \( S_{10} \) and \( S_{11} \), are 1–86 and 123–228, and 1–82 and 123–227 respectively, which are very similar to SCR23 constructing fragments. In addition, as specified in [25], residue position 234 plays an important role in substrate selectivity. Here all optimal fragments selected consistently avoided this residue position, preserving amino acid conservation there. Potentially these optimal selections can provide promising candidates to generate humanized functional variants.
Fig. 5.6: Deimmunization of rGSTT2-2 by fragment swapping from hGSTT1-1. Top row: 2 fragments; middle row: 3 fragments; bottom row: 4 fragments. Left: selected fragments, with $\alpha$ decreasing from 1 to 0 from top to bottom. Red blocks represent alpha helices and blue blocks represent beta sheets in the 3D structure of rGSTT2-2. Magenta lines in the first row show human fragment introduced from hGSTT1-1 in [25] to generate protein variant SCR23. Right: epitope and conservation scores, averaged over the libraries (cyan circles), for SCR23 (red stars), and for the wild-type (green diamonds).
Fig. 5.7: Conservation and epitope scores for hybrids constructed by optimal fragment selection (blue points), along with rGSTT2-2 (green diamond) and SCR23 (red star). Top: 2 fragments; middle: 3 fragments; bottom: 4 fragments.
5.4 Conclusion

We have developed a new method for protein deimmunization by fragment swapping. Combining conservation and epitope scores in optimal fragment selection, fragment swapping substitutes peptides in exogenous proteins with appropriate human sequence fragments, resulting in low inherent immunogenicity. At the same time, functionally important regions in the exogenous proteins are preserved. Compared with results in previous studies, our method can efficiently generate protein variants with better scores under metrics for immunogenicity and activity, providing promising candidates for experimental evaluation.
6. SUMMARY AND FUTURE WORK

6.1 Summary

To maximize the hit rate of generating folded and functional protein variants in protein engineering, this thesis developed effective criteria and efficient algorithms to optimize experiment planning in site-directed protein recombination. Algorithms were developed to plan experiments for different scenarios: generating novel functionalities from homologous parent proteins by symmetric, exhaustive recombination; inheriting different properties from heterogeneous source and target proteins by asymmetric, selective recombination; and minimizing immunogenicity of therapeutic proteins by incorporating fragments from human proteins. These algorithms considered the interactions among protein sequence, structure and function and addressed recombination experiment planning from the perspectives of novelty, stability and immunogenicity. The overall computational goals were to efficiently find plans optimizing the predicted quality of protein variants constructed, according to these various criteria.
6.1.1 Optimization in Symmetric and Exhaustive Protein Recombination

In symmetric and exhaustive protein recombination, we developed the first approach to explicitly optimize for diversity in experiment planning. Based on the observation that there are a fixed total number of mutations over the hybrid library, but that it is desirable to spread out these mutations relatively uniformly within the library, we defined diversity variance criteria for diversity optimization. Dynamic programming algorithms were developed to find optimal sets of breakpoints according to hybrid-hybrid and hybrid-parent diversity variance. A study of hybrid-hybrid and hybrid-parent diversity variance showed that these two criteria are correlated.

Since novelty and stability have a complementary and competing relationship in site-directed protein recombination, we developed an approach for joint optimization of these two aspects. To evaluate both novelty and stability in considering the performance of a recombination plan, we focused on undominated breakpoint sets, those for which no other set has better performance in both diversity variance (novelty metric) and perturbation (stability metric). Our dynamic programming method STAVERSITY seeks to minimize a weighted combination of these two metrics, finding all undominated breakpoint sets on the convex hull of all possible sets (without enumerating the other breakpoint sets). With an extension of STAVERSITY, we can also find some undominated breakpoint sets in the concavities, and we provide a method to evaluate the performance difference between those and any that were possibly missed. By characterizing the optimal trade-offs between novelty and stability, our approach provides promising experiment plans to satisfy different experimen-
6.1.2 Optimization in Asymmetric and Selective Protein Recombination

In contrast to symmetric and exhaustive protein recombination, we developed the first mechanism for asymmetric, selective protein recombination, which we call protein fragment swapping. Protein fragment swapping differentiates parent roles as source and target proteins, taking the target protein as a framework for fragment swapping, into which fragments from the source are incorporated. Only residues in predefined swappable regions may be selected for swapping, so that fragment swapping can deal with the parent proteins having gappy regions in a sequence or structural alignment. In this thesis, protein fragment swapping is studied in two applications: functionality recombination and therapeutic protein deimmunization.

The objective of functionality recombination is to inherit different properties from the source and the target. The quality of the library constructed is determined by the swapping fragments selected. To optimize the library quality, a weighted combination of potential scores considering the parents is optimized. We proved that the resulting optimization problem is NP-hard when a pair-wise potential score is considered. However, based on the patterns for potential score contributions from single and pair-wise terms, we developed an integer programming method that is effective in practice at finding optimal sets of fragments. By changing the weight on the potential score contributions, optimization can be achieved with different focus on source and target.
In therapeutic protein deimmunization, the objective is to decrease immunogenicity while simultaneously preserving the therapeutic activity of the therapeutic protein. By using the therapeutic protein as the target and a human protein as the source, human sequence fragments can be swapped into the therapeutic protein in order to obtain variants with reduced immunogenicity. At the same time, the human fragments must preserve structurally and functionally important amino acids of the target. To optimize these two aspects simultaneously, we defined a weighted combination of epitope score (immunogenicity metric) and potential score (therapeutic activity metric) for experiment optimization. We developed a dynamic programming method to maximize this weighted score combination. Case study tests with antibodies and glutathione transferases demonstrated the effectiveness of the approach. Compared with protein variants previously constructed, deimmunization by fragment swapping can generate hybrids with better predicted performance according to both epitope and potential score. We also found consistency between the optimal fragments selected by our algorithms and variants experimentally evaluated in previous work.

### 6.2 Future Work

#### 6.2.1 Sub-optimal Experiment Planning

This thesis has focused on algorithms to find optimal experiment plans according to different quality metrics. Practically, sub-optimal experiment plans may also be worth considering. Generating sub-optimal plans provides more choices that may be subsequently evalu-
ated for additional experimental requirements. Furthermore, while by definition the overall quality of a sub-optimal library is less than that of an optimal library, the quality metrics are based on aggregate evaluations of all the hybrids. The performance among the individual hybrids may vary substantially. In fact, individual hybrids in a sub-optimal library may have comparable or even better performance than those in an optimal library. Finally, finding sub-optimal plans can give us insight of the effects of different experimental choices on stability and functionality, which may help to find important amino acid constraints. For example, in Fig. 3.7 and Fig. 3.9 for joint optimization, Fig. 5.4, and Fig. 5.6 for protein deimmunization, transitions among the optimal experiment plans are sometimes dramatic. With sub-optimal planning, we may find intermediate steps among those transitions.

In [94], a dynamic programming algorithm was developed to find sub-optimal solutions by near-optimal backtracking, which can be incorporated into our dynamic programming optimization algorithms to find sub-optimal experiment planning. Similarly we can also incorporate suboptimal techniques for integer programming in experiment optimization of protein fragment swapping.

6.2.2 Enhancements of Protein Deimmunization

In Chapter 5, we developed the criteria and methods for protein deimmunization by human protein fragment swapping. To make the optimization tractable, fragment swapping was restricted to aligned source and target proteins, considering only single residue conservation. Though the practical significance of this method was demonstrated in experiment
planning case studies, enhancements can make this approach more flexible and potentially more effective.

First, a pair-wise potential score could be introduced in the optimization. In Chapter 4 we proved that potential score optimization for fragment swapping is NP-hard with pair-wise terms included, and thus developed an integer programming method. To apply integer programming to protein deimmunization swapping, the chief difficulty is accounting for the 9-mer epitope scores. In Chapter 5, we designed an encoding to represent the relative position between each 9-mer and the swapping fragments. For each 9-mer, its relative position has more than 6000 cases. In integer programming, variables would need to be defined for all these cases, and more importantly, constraints would need to be defined for all the relative position variables. The problem sizes may be beyond the scope of standard integer programming solvers. One option we can try is to relax the constraints for 9-mer relative position variables, to reduce the number of constraints. Alternatively, special-purpose algorithms for solving the system may be able to take advantage of the particular structure of the constraints.

Second, in the current approach to deimmunization by fragment swapping, the source and target proteins must be aligned sequentially or structurally, and there is only one source human protein providing swapping fragments. To make fragment swapping more flexible, it would be beneficial to do protein deimmunization by swapping fragments selected from multiple human proteins that need not even be aligned.
6.2.3 Recombination Experiment Data Interpretation

Experiment optimization for protein recombination actually should be an iterative procedure, in which initial experiment planning is just the starting point. After the designed hybrid library is constructed and data are collected for the hybrids, we may be able to get feedback to guide further rounds of experiment planning. In particular, we can adjust both the potential scores and the overall goals in optimizing the recombination experiments. The key point in data interpretation is to explore the interactions among sequence, structure and function, including interacting residue positions, structurally and functionally important amino acids, immunogenic and non-immunogenic regions, etc.

Depending on properties of the metrics of interest, the data for the hybrid library can be interpreted in different ways. For interacting residue positions, we must consider how to combine interacting residues found in constructed protein variants with those found in the original protein family. For data interpretation of amino acid conservation, we may adopt a different method, maximum \textit{a posteriori} probability estimation, starting from a prior distribution and perturbing it with information from the folded and functional protein variants. In general, protein variants constructed in recombination experiments provide us with important information regarding constraints for beneficial protein sequences, which should be explored and applied in further experiment planning.
Bibliography


specific and promiscuous hla ligand database using dna microarrays and virtual hla

[87] V. Taly, P. Urban, G. Truan, and D. Pompon. A combinatorial approach to substrate
discrimination in the P450 CYP1A subfamily. Biochim Biophys Acta, 1770:446–

[88] S. Tanaka and H.A. Scheraga. Medium and long range interaction parameters be-
tween amino acids for predicting three dimensional structures of proteins. Macro-

coupling in protein families. IEEE/ACM Transactions on Computational Biology

coupling in protein families. IEEE/ACM Transactions on Computational Biology

[91] N. Tsurushita and M. Vasquez. Humanization of monoclonal antibodies. in: Honjo,


method for asymmetric, selective site-directed recombination. In Proc. RECOMB,

[102] W. Zheng, K.E. Griswold, and C. Bailey-Kellogg. Protein fragment swapping:
A method for asymmetric, selective site-directed recombination. J. Comp. Biol.,

breakpoint locations to optimize diversity in protein engineering by site-directed