GGAssembler: Economical Design of Gene Libraries with Precise Control over Mutations

Authors: Shlomo Yakir Hoch, Jonathan Yaacov Weinstein, Ravit Netzer, Karen Hakeny, Sarel Jacob Fleishman

Abstract

Golden Gate (GG) assembly is an effective tool for the seamless assembly of genes from DNA fragments. Its use for generating combinatorial mutagenic libraries is much less developed. We describe a new method called GGAssembler for designing gene libraries that encode combinatorial mutations through GG gene assembly. GGAssembler uses a graph-theoretical approach to find the most cost-effective way to assemble diversity from DNA fragments. We demonstrate the accuracy, efficiency, low bias, and cost-effectiveness of GGAssembler by encoding hundreds of thousands of variants of camelid single-domain antibodies. We envision that GGAssembler will advance protein science and engineering by enabling the construction of diverse protein libraries with precise control over mutations. Source code is freely available at: https://github.com/Fleishman-Lab/GGAssembler.

Introduction

Over the past decade, high-throughput screening of protein variant libraries and deep sequencing have revolutionized protein characterization, engineering, and design[1,2]. Most current approaches introduce random or systematic single-point mutations[3], but large changes in protein activity and stability often require multiple simultaneous mutations[4], calling for methods that introduce combinatorial mutations. Such multipoint combinatorial libraries can be designed by analyzing the results of previous screens[5–7] or based on atomistic design calculations[8–10]. Whereas approaches for encoding single-point mutation libraries have greatly advanced in control and cost, approaches that encode combinatorial-mutation libraries are much less developed[3].

We recently described methods for the computational design of active-site libraries that generated thousands of diverse and active variants[9,11]. In these designed libraries, diversity was encoded throughout the gene, including, in one case, through insertions and deletions[11]. Unlike many previous libraries, these were completely defined by atomistic design calculations without allowing any random mutations. To construct such complex yet completely defined variant libraries, we considered different cloning strategies that would be general, economical, enable a high level of control over the location and identity of mutations and the ratio of mutations to the wild-type parental gene, and exhibit low representation bias among the variants. Given these considerations, we focused on Golden

Gate (GG) gene assembly[12,13]. GG assembly employs Type IIs restriction enzymes that cleave outside their DNA recognition sequence to generate fragments with overhangs that can be programmed to ligate specific oligos into full-length genes[12,13]. By targeting the overhangs to immutable amino acid positions, a combinatorial assembly of genetic diversity can be generated by freely ligating DNA oligos processed by Type IIs restriction enzymes and a ligase.

In principle, GG assembly enables constructing scarless DNA of any size from small fragments[14–19]. High-fidelity GG assembly, however, is governed by a number of factors: the nonspecific (star) activity of the restriction enzyme[20], the sensitivity of the ligase to mismatches[21], and the propensity of DNA overhangs to anneal under different reaction conditions. These considerations limit GG assembly to a small set of well-studied Type IIs enzymes, reaction conditions, and, critically, preferred overhangs. The limitation on the number and identity of overhangs forced researchers to develop iterative DNA assembly methods that use a small set of overhangs[14–19], restricting the scope of GG assembly. Recent high-throughput screening and DNA sequencing studies have examined the results of GG assembly under different conditions and have expanded the set of allowed overhangs. Based on these developments, dozens of fragments have been shown to assemble successfully[22,23], opening the way to assembling diverse libraries from many fragments. Despite these advances, however, these strategies do not optimize the cost of the DNA required to build the library.

Here, we present a new method, called GGAssembler, for the economical design of DNA fragments that assemble through GG assembly to generate genes that encode arbitrarily chosen diversity with minimal representation bias. GGAssembler uses a graph-theoretical strategy to find the most economical approach to build precisely the desired diversity from DNA fragments. In a companion paper, we demonstrated how GGAssembler encodes millions of designed multipoint variants in GFP[9]. From this library, we isolated thousands of active-site variants from these libraries, demonstrating how atomistic design calculations can massively sample the sequence space of functional variants and how GGAssembler provides an economical and reliable workflow to achieve such diversification.Here, we demonstrate that this approach can accurately, efficiently, and economically encode complex multipoint mutation libraries of camelid single-domain (vHH) antibodies comprising hundreds of thousands of variants. GGAssembler is freely available at https://github.com/Fleishman-Lab/GGAssembler.

Results

A general method to design combinatorial diversity

We developed GGAssembler to provide the most economical approach to building a library while allowing nearly complete freedom to decide on mutation types, positions, and the size of the library. GGAssembler accordingly provides: 1) control over the GG assembly, ensuring a successful assembly each time, and 2) Minimize the cost of DNA fragments. To control GG assembly, we leverage empirically derived overhang ligation efficiency and fidelity[22–24], defined as the frequency of ligations of a given overhang divided by highest frequency and as the fraction of correct ligations divided by the total fraction of ligations for a given overhang[24], respectively. Based on the empirically determined efficiency and fidelity, GGAssembler estimates the probability that any set of overhangs would assemble successfully. The cost associated with each DNA fragment is the number of nucleotides (nts) required to produce all the variability encompassed in that fragment. For example, the cost associated with a DNA fragment of 20 nucleotides with a single mutated amino acid position that requires two codons would be 40 nucleotides (the length of the fragment times the variability encoded). Adding an extra codon would increase the cost to 60 nucleotides, while adding another mutated position would multiply the cost by the number of codons in the second position. GGAssembler leverages the fact that extending diversity by using a degenerate codon that encompasses more desired identities carries no additional cost to optimize the cost of the library.

The GGAssembler workflow consists of three building blocks (**Fig. 1**). We start by defining the DNA encoding the desired diversity and apply an optional step of codon compression which encodes the diversity as degenerate codons using nucleotide combinations[25]. Next, we run a graph-theoretical cost optimization algorithm and provide the user with several alternatives that vary the number of overhangs, cost, or other criteria. To run our cost optimization algorithm, we build a graph in which nodes represent restriction-enzyme cleavage points, and edges represent the DNA fragment defined by the two cleavage points.

To construct the graph, we start with a user-defined DNA sequence of the target gene prior to diversification, a list of desired mutations, the Type IIs restriction enzyme, the minimum required level of ligation efficiency, and a set of criteria that define acceptable DNA fragments; for example: the minimal and maximal lengths of a fragment. We test which nucleotides may serve as potential cleavage points given the restriction enzyme provided. To serve as a cleavage point, the DNA region encompassing the nucleotide and the overhang created by the restriction enzyme must be immutable (not encoding diversity), and the overhang must exhibit at least the level of ligation efficiency set by the user. Nucleotides that fulfill these requirements are added as nodes to the graph with their associated overhang. An edge connects each pair of nodes if the two overhangs would not

ligate to one another. We weight each edge by the number of nucleotides (proportional to cost) required to produce the variability in the DNA fragment it represents.

We then search for the shortest-path solution that encompasses all diversified DNA positions. Our construction ensures that any shortest-path solution represents the minimum cost in terms of DNA material. Critically, however, non-adjacent overhangs could assemble to generate undesired assembly products. To ensure that we obtain only desired assembly products, we employ color-coding[26] and apply a colorful shortest-path algorithm that only generates paths in which each color appears at most once (**Fig. 1**). By color-coding nodes with overhangs that ligate in the same color, we ensure any colorful shortest-path solution would not include identically colored nodes and therefore overhangs that ligate. This scheme elevates the fidelity of an overhang above the user-defined threshold and results in minimal cost solutions that assemble correctly.

Applying GGAssembler results in segmenting the target gene sequence into fragments that encode diversity (variable) and ones devoid of diversity (constant). The output is provided in a table that specifies the variable fragments and the sequences of the constant fragments that can be custom synthesized or amplified from a preexisting gene.

We next introduce a streamlined wet-lab protocol for assembling a library generated by GGAssembler. To maintain cost-effectiveness, we amplify constant fragments with primers flanked by the appropriate restriction-enzyme recognition site and all variable DNA oligos with a single 3’ primer, filling the reverse strand. By adjusting the concentration of each fragment, we control biases in the library on the variant or mutation level. As the last step, we follow established GG assembly protocols.

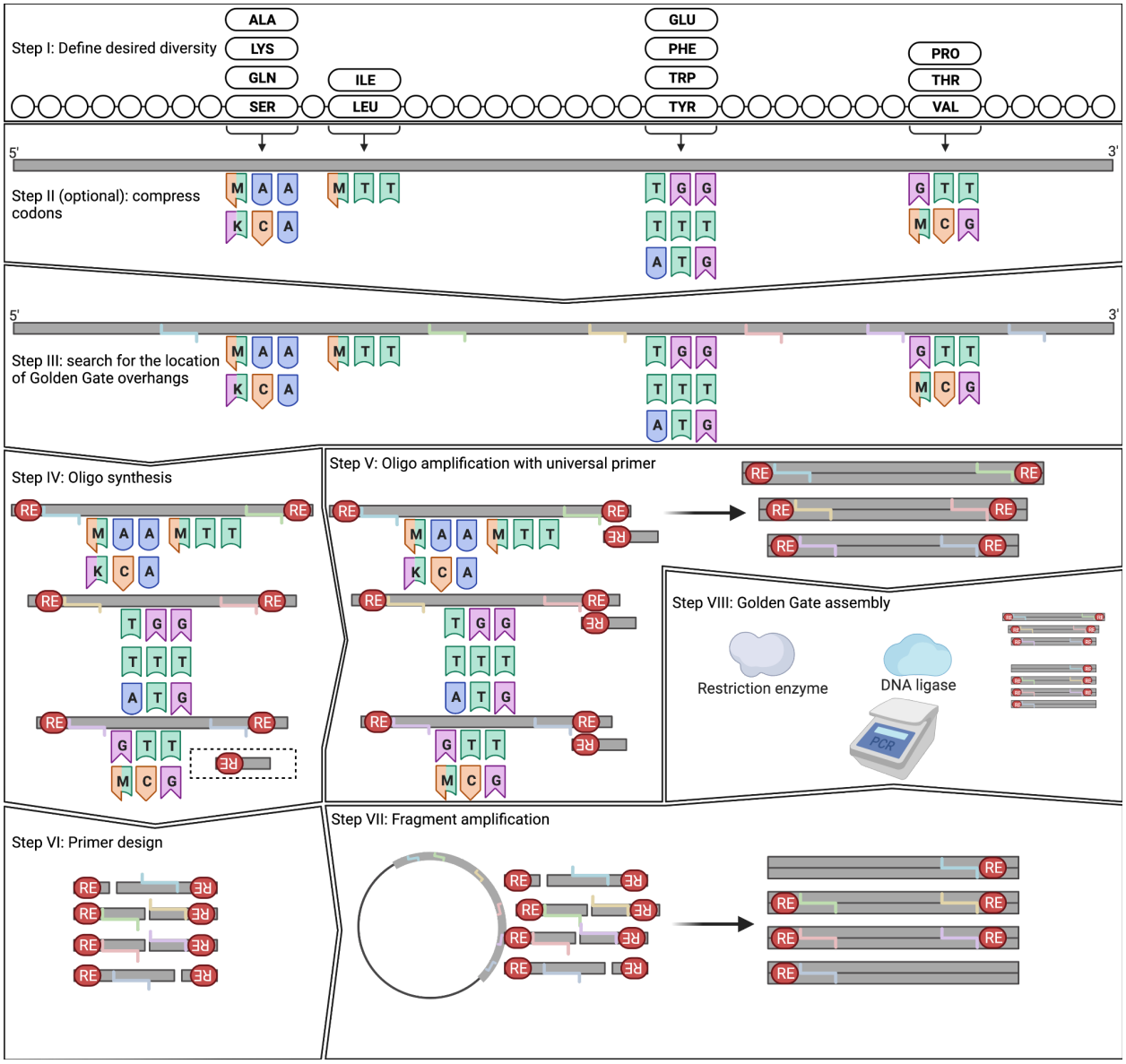

Figure 1. GGAssembler workflow composed of three main building blocks, 1) defining the desired diversity, 2) computational steps (steps II and III), and 3) lab work (steps IV-VII). Step IV: Order variable segments as DNA oligos and a universal primer (shown in dashed lines) to amplify them.

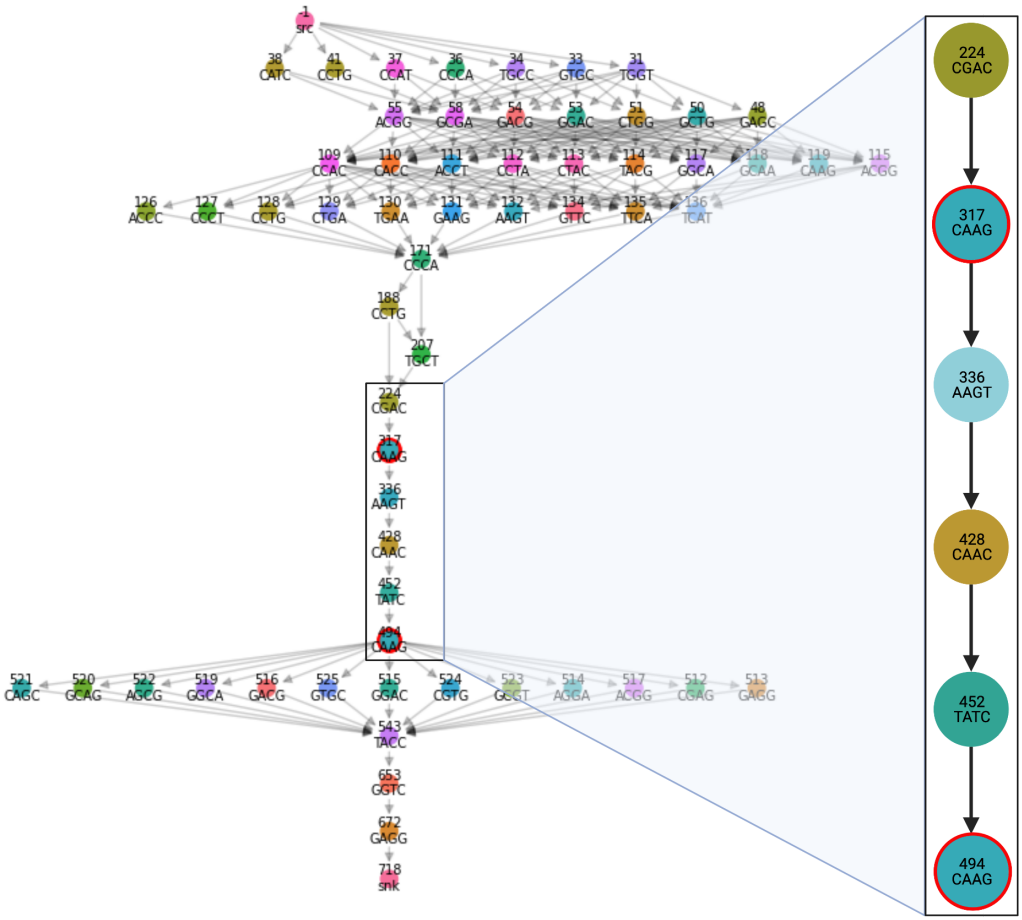

Figure 2. Showing all shortest paths in a designed GFP library[9]. Vertices marked by cleavage point location and overhang are colored identically if the overhangs ligate. Two vertices that have the same overhang (CAAG) are marked with red circles (positions 317 and 494). Using any of the shortest-path solutions would result in misassembly as the overhang at 317 would ligate to 494, eliminating the intervening segment. When a colorful search is used, all the solutions shown on this plot are eliminated as illegal, forcing the search to find alternative paths that do not misassemble.

Here, we verify the accuracy of GGAssembler by applying the method to construct two complex camelid antibody (vHH) libraries. Unlike GFP, vHHs are small, allowing us to analyze the accuracy and representation bias of the library through Illumina deep sequencing. A detailed cost analysis of similarly sized libraries shows the entire reaction cost is less than $1,700, of which 97% are in DNA-associated costs, translating to 0.45¢ per variant. In many practical applications, one may wish to experiment with the assembly procedure or generate libraries that focus diversity in different regions. In such cases, parts

of the assembly reaction can be reused, drastically reducing the cost of additional assemblies.

High fidelity and low representation bias in assembly

Using the htFuncLib approach that generated the GFP libraries[9], we designed two camelid antibody (vHH) libraries, vHH1 and vHH2. Briefly, htFuncLib uses Rosetta atomistic design calculations and a machine-learning method to select mutations that can combine freely to generate low-energy multipoint mutation libraries. Such libraries are likely to be enriched in stable, well-folded, and potentially functional protein variants for high-throughput screening. In two cases, we showed that such libraries could be used in one-shot shot high-throughput screening to isolate thousands of unique and highly diverse designs[9,11]. Applied to the vHHs, htFuncLib resulted in 55,296 and 233,280 designs for the two libraries, respectively, in 11 and 13 positions across the complementarity-determining regions of the antibodies. Following design, we applied the GGAssembler process, including codon compression. The process resulted in a biased assembly in which designs do not appear with equal proportions. After assembly, we split the resulting DNA material and subjected it to next-generation sequencing (NGS) immediately after gene assembly without being transformed or cloned into cells to eliminate any source of biological bias. We also transformed the library into bacterial cells to assess the number of in-frame constructs. Sanger sequencing of 8 and 16 colonies revealed that 75% and 81% of sequences were in-frame and matched the designs of vHH1 and vHH2, respectively.

To assess the quality of our assembly, as a stringent test, we discarded any read that was not completely identical to one of the designed variants across the variable segments, allowing for mismatches and deletion only in the constant segments. Even at this high stringency, 96% and 93% of designed variants were uniquely identified in vHH1 and vHH2, respectively. We analyzed the difference between the expected and observed design counts, finding they closely matched (Fig. 3C and D). Only two variants exhibited more than a tenfold count difference in vHH1 (out of >55,000 variants) and approximately 1,300 of vHH2 (out of >230,000 variants; 0.6%), and in both libraries, these discrepancies were at most 22 fold (Fig 3A and B).

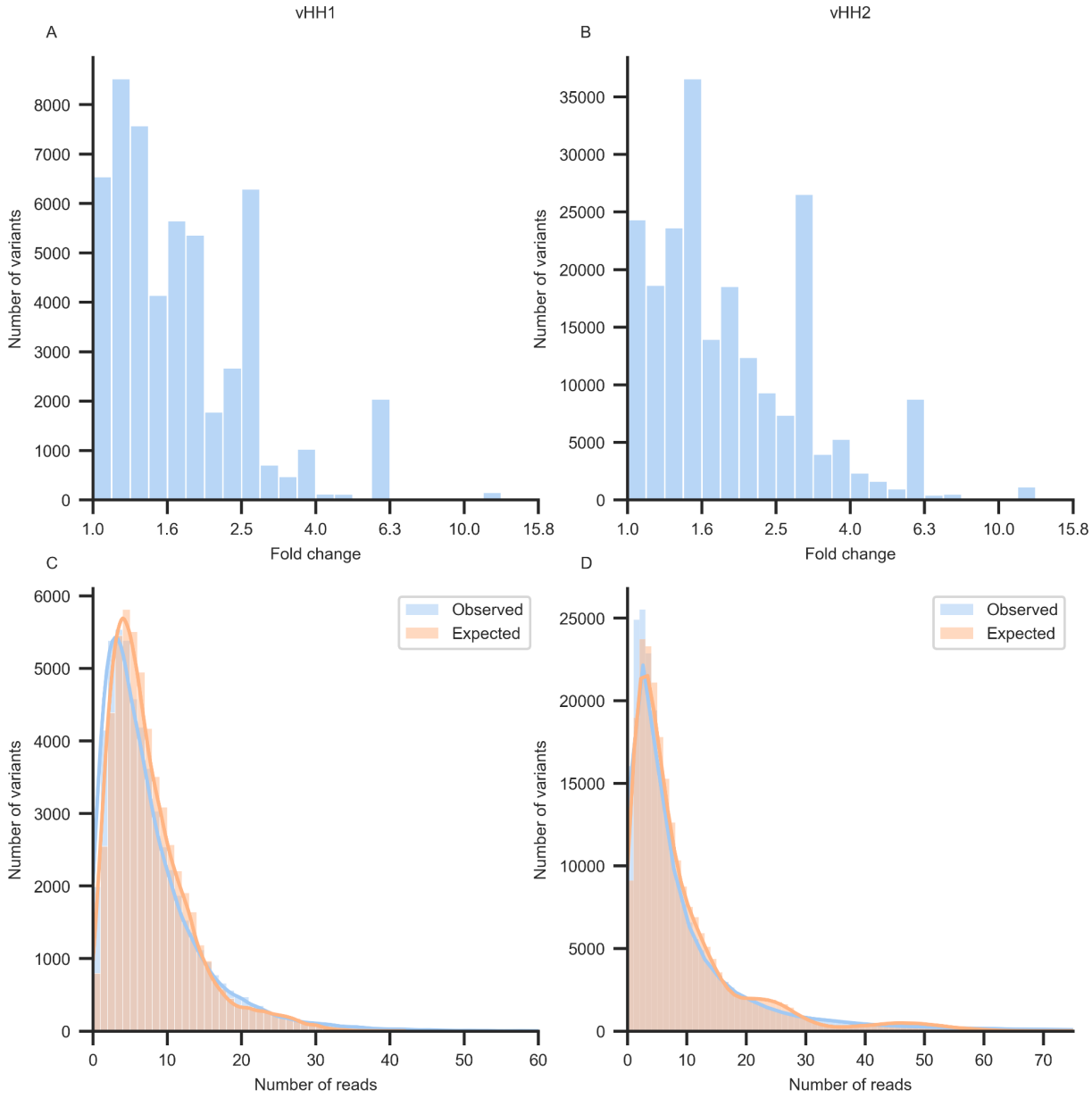

Figure 3. Deep sequencing analysis of libraries vHH1 (panels A and C) and vHH2 (panels B and D). (A, B) Order of magnitude difference between observed and expected counts. (C, D) Distribution of the number of reads, assuming a Poisson distribution (orange) against the observed number of reads.

Discussion

New approaches to probe the mutational tolerance of proteins experimentally and computationally are fueling interest in methods for efficient synthesis of combinatorial mutation libraries. GGAssembler exploits the fact that typically, the desired diversity is clustered in contiguous epitopes (*e.g.*, antibody CDRs or TIM-barrel active-site loops) that

are separated by conserved regions. This clustering lends itself to breaking the assembly reaction into constant fragments that can be synthesized or amplified from a preexisting gene and variable regions that can be encoded by short oligos that comprise degenerate codons. GGAssembler uses codon compression and chooses a set of empirically determined high-fidelity overhangs to minimize costs and ensure accurate assembly even in complex libraries. In the two camelid antibody examples provided here and the companion paper on GFP, correct assemblies were made from 9 and 20 parts, respectively. To our knowledge, this study represents the first demonstrations of successful one-pot GG library assembly with many parts. GGAssembler could be used with DNA oligo pool technology, increasing the complexity and diversity of designed libraries even further. We envision GGAssembler enabling the efficient and economical assembly of libraries of antibodies, enzymes, pathogenic antigens and other functional proteins to study the mutational tolerance of a variety of proteins and enable one-shot optimization of function.

Methods

Library design

Codon compression

For each diversified AA position, we codon compressed the required diversity by generating all ambiguous codons that encode a portion of the required diversity using CodonGenie[27] and formulating the compression as the famous set cover problem. We have implemented DLX[25] to solve the exact cover problem, which results in every amino acid identity appearing exactly once.

DNA segmentation

As described in the results, we model all possible segmentations as a directed graph with each vertex representing a cleavage point and overhang, and edges representing DNA segments between a pair of cleavage points weighted by the number of nucleotides required to produce that segment. To find segmentations that will assemble correctly, we employ color coding[26], coloring vertices with the same color if empirical results[22–24] indicate their overhangs would ligate. Next, we run a colorful version of Dijkstra's shortest path algorithm[28], which considers extending a path only if the added vertex color does not already appear in the path. The result is a DNA segmentation and set of overhangs that ensures high-fidelity assembly.

Deep Sequencing

The Weizmann Institute's Israel National Center for Personalized Medicine (INCPM) prepared the amplicon libraries from 20 ng of DNA, as previously described[29]. The libraries were verified by Qubit (Thermo fisher scientific) and TapeStation (Agilent) and sequenced using a paired-end V3 600 cycles Illumina kit (Illumina MS-102-3003) on an Illumina Miseq. We preprocessed Fastq sequences using the BBTools software suite[30], and pairwise aligned the results to constant DNA segments using Parasail[31]. We further analyzed matches using Python[32–37] filtering sequences which do not exactly match variable DNA segments.

Acknowledgements

Shlomo Yakir Hoch thanks David Peleg for helpful discussions and comments. Figures 1 and 2 were created with BioRender.com. The research was supported by the Volkswagen Foundation (94747), the Israel Science Foundation (1844), the European Research Council through a Consolidator Award (815379), the Dr. Barry Sherman Institute for Medicinal Chemistry, and a donation in memory of Sam Switzer.

References

(1) Wójcik, M.; Telzerow, A.; Quax, W. J.; Boersma, Y. L. High-Throughput Screening in Protein Engineering: Recent Advances And Future Perspectives. *Int. J. Mol. Sci.* **2015**, *16* (10), 24918–24945.
(2) Markel, U.; Essani, K. D.; Besirlioglu, V.; Schiffels, J.; Streit, W. R.; Schwaneberg, U. Advances in Ultrahigh-Throughput Screening for Directed Enzyme Evolution. *Chem. Soc. Rev.* **2020**, *49* (1), 233–262.
(3) Alejaldre, L.; Pelletier, J. N.; Quaglia, D. Methods for Enzyme Library Creation: Which One Will You Choose? *Bioessays* **2021**, *43* (8), 2100052.
(4) Khersonsky, O.; Lipsh, R.; Avizemer, Z.; Ashani, Y.; Goldsmith, M.; Leader, H.; Dym, O.; Rogotner, S.; Trudeau, D. L.; Prilusky, J.; Amengual-Rigo, P.; Guallar, V.; Tawfik, D. S.; Fleishman, S. J. Automated Design of Efficient and Functionally Diverse Enzyme Repertoires. *Mol. Cell* **2018**, *72* (1), 178–186.e5.
(5) Whitehead, T. A.; Baker, D.; Fleishman, S. J. Computational Design of Novel Protein Binders and Experimental Affinity Maturation. *Methods Enzymol.* **2013**, *523*, 1–19.
(6) Whitehead, T. A.; Chevalier, A.; Song, Y.; Dreyfus, C.; Fleishman, S. J.; De Mattos, C.; Myers, C. A.; Kamisetty, H.; Blair, P.; Wilson, I. A.; Baker, D. Optimization of Affinity, Specificity and Function of Designed Influenza Inhibitors Using Deep Sequencing. *Nat. Biotechnol.* **2012**, *30* (6), 543–548.
(7) Fowler, D. M.; Araya, C. L.; Fleishman, S. J.; Kellogg, E. H.; Stephany, J. J.; Baker, D.; Fields, S. High-Resolution Mapping of Protein Sequence-Function Relationships. *Nat. Methods* **2010**, *7* (9), 741–746.
(8) Guntas, G.; Purbeck, C.; Kuhlman, B. Engineering a Protein--Protein Interface Using a Computationally Designed Library. *Proceedings of the National Academy of Sciences* **2010**, *107* (45), 19296–19301.
(9) Weinstein, J. Y.; Martí-Gómez, C.; Lipsh-Sokolik, R.; Hoch, S. Y.; Liebermann, D.; Nevo, R.; Weissman, H.; Petrovich-Kopitman, E.; Margulies, D.; Ivankov, D.; McCandlish, D.; Fleishman, S. J. Designed Active-Site Library Reveals Thousands of Functional GFP Variants. *bioRxiv*, 2023, 2022.10.11.511732. https://doi.org/10.1101/2022.10.11.511732.
(10) Treynor, T. P.; Vizcarra, C. L.; Nedelcu, D.; Mayo, S. L. Computationally Designed Libraries of Fluorescent Proteins Evaluated by Preservation and Diversity of Function. *Proc. Natl. Acad. Sci. U. S. A.* **2007**, *104* (1), 48–53.
(11) Lipsh-Sokolik, R.; Khersonsky, O.; Schröder, S. P.; de Boer, C.; Hoch, S.-Y.; Davies, G. J.; Overkleeft, H. S.; Fleishman, S. J. Combinatorial Assembly and Design of Enzymes.

Science **2023**, *379* (6628), 195–201.
(12) Engler, C.; Kandzia, R.; Marillonnet, S. A One Pot, One Step, Precision Cloning Method with High Throughput Capability. *PLoS One* **2008**, *3* (11), e3647.
(13) Engler, C.; Gruetzner, R.; Kandzia, R.; Marillonnet, S. Golden Gate Shuffling: A One-Pot DNA Shuffling Method Based on Type IIs Restriction Enzymes. *PLoS One* **2009**, *4* (5), e5553.
(14) Andreou, A. I.; Nakayama, N. Mobius Assembly: A Versatile Golden-Gate Framework towards Universal DNA Assembly. *PLoS One* **2018**, *13* (1), e0189892.
(15) Püllmann, P.; Ulpinnis, C.; Marillonnet, S.; Gruetzner, R.; Neumann, S.; Weissenborn, M. J. Golden Mutagenesis: An Efficient Multi-Site-Saturation Mutagenesis Approach by Golden Gate Cloning with Automated Primer Design. *Sci. Rep.* **2019**, *9* (1), 10932.
(16) Sarrion-Perdigones, A.; Falconi, E. E.; Zandalinas, S. I.; Juárez, P.; Fernández-del-Carmen, A.; Granell, A.; Orzaez, D. GoldenBraid: An Iterative Cloning System For Standardized Assembly of Reusable Genetic Modules. *PLoS One* **2011**, *6* (7), e21622.
(17) Sarrion-Perdigones, A.; Vazquez-Vilar, M.; Palací, J.; Castelijns, B.; Forment, J.; Ziarsolo, P.; Blanca, J.; Granell, A.; Orzaez, D. GoldenBraid 2.0: A Comprehensive DNA Assembly Framework for Plant Synthetic Biology. *Plant Physiol.* **2013**, *162* (3), 1618–1631.
(18) Vazquez-Vilar, M.; Quijano-Rubio, A.; Fernandez-Del-Carmen, A.; Sarrion-Perdigones, A.; Ochoa-Fernandez, R.; Ziarsolo, P.; Blanca, J.; Granell, A.; Orzaez, D. GB3.0: A Platform for Plant Bio-Design That Connects Functional DNA Elements with Associated Biological Data. *Nucleic Acids Res.* **2017**, *45* (4), 2196–2209.
(19) Weber, E.; Engler, C.; Gruetzner, R.; Werner, S.; Marillonnet, S. A Modular Cloning System for Standardized Assembly of Multigene Constructs. *PLoS One* **2011**, *6* (2), e16765.
(20) Mayer, H. Optimization of the EcoRI*-Activity of EcoRI Endonuclease. *FEBS Lett.* **1978**, *90* (2), 341–344.
(21) Lohman, G. J. S.; Bauer, R. J.; Nichols, N. M.; Mazzola, L.; Bybee, J.; Rivizzigno, D.; Cantin, E.; Evans, T. C., Jr. A High-Throughput Assay for the Comprehensive Profiling of DNA Ligase Fidelity. *Nucleic Acids Res.* **2016**, *44* (2), e14.
(22) Potapov, V.; Ong, J. L.; Kucera, R. B.; Langhorst, B. W.; Bilotti, K.; Pryor, J. M.; Cantor, E. J.; Canton, B.; Knight, T. F.; Evans, T. C.; Lohman, G. J. S. Comprehensive Profiling of Four Base Overhang Ligation Fidelity by T4 DNA Ligase and Application to DNA Assembly. *ACS Synth. Biol.* **2018**, *7* (11), 2665–2674.
(23) Pryor, J. M.; Potapov, V.; Kucera, R. B.; Bilotti, K.; Cantor, E. J.; Lohman, G. J. S. Enabling One-Pot Golden Gate Assemblies of Unprecedented Complexity Using Data-Optimized Assembly Design. *PLoS One* **2020**, *15* (9), e0238592.
(24) Potapov, V.; Ong, J. L.; Langhorst, B. W.; Bilotti, K.; Cahoon, D.; Canton, B.; Knight, T. F.; Evans, T. C., Jr; Lohman, G. J. S. A Single-Molecule Sequencing Assay for the Comprehensive Profiling of T4 DNA Ligase Fidelity and Bias during DNA End-Joining. *Nucleic Acids Res.* **2018**, *46* (13), e79.
(25) Knuth, D. E. Dancing Links. *arXiv:cs/0011047* **2000**.
(26) Alon, N.; Yuster, R.; Zwick, U. Color-Coding. *J. ACM* **1995**, *42* (4), 844–856.
(27) Swainston, N.; Currin, A.; Green, L.; Breitling, R.; Day, P. J.; Kell, D. B. CodonGenie:

Optimised Ambiguous Codon Design Tools. *PeerJ Computer Science* **2017**, *3*, e120.
(28) Dijkstra, E. W. A Note on Two Problems in Connexion with Graphs. *Numer. Math.* **1959**, *1* (1), 269–271.
(29) Blecher-Gonen, R.; Barnett-Itzhaki, Z.; Jaitin, D.; Amann-Zalcenstein, D.; Lara-Astiaso, D.; Amit, I. High-Throughput Chromatin Immunoprecipitation for Genome-Wide Mapping of in Vivo Protein-DNA Interactions and Epigenomic States. *Nat. Protoc.* **2013**, *8* (3), 539–554.
(30) Bushnell, B.; Rood, J.; Singer, E. BBMerge - Accurate Paired Shotgun Read Merging via Overlap. *PLoS One* **2017**, *12* (10), e0185056.
(31) Daily, J. Parasail: SIMD C Library for Global, Semi-Global, and Local Pairwise Sequence Alignments. *BMC Bioinformatics* **2016**, *17*, 81.
(32) McKinney, W. Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference*; SciPy, 2010. https://doi.org/10.25080/majora-92bf1922-00a.
(33) The pandas development team. *Pandas-Dev/pandas: Pandas*; 2023. https://doi.org/10.5281/zenodo.7857418.
(34) Harris, C. R.; Millman, K. J.; van der Walt, S. J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N. J.; Kern, R.; Picus, M.; Hoyer, S.; van Kerkwijk, M. H.; Brett, M.; Haldane, A.; Del Río, J. F.; Wiebe, M.; Peterson, P.; Gérard-Marchant, P.; Sheppard, K.; Reddy, T.; Weckesser, W.; Abbasi, H.; Gohlke, C.; Oliphant, T. E. Array Programming with NumPy. *Nature* **2020**, *585* (7825), 357–362.
(35) Van Der Walt, S.; Colbert, S. C.; Varoquaux, G. The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science and Engineering* **2011**, *13* (2), 22–30.
(36) Cock, P. J. A.; Antao, T.; Chang, J. T.; Chapman, B. A.; Cox, C. J.; Dalke, A.; Friedberg, I.; Hamelryck, T.; Kauff, F.; Wilczynski, B.; De Hoon, M. J. L. Biopython: Freely Available Python Tools for Computational Molecular Biology and Bioinformatics. *Bioinformatics* **2009**, *25* (11), 1422–1423.
(37) Granger, B. E.; Pérez, F. Jupyter: Thinking and Storytelling With Code and Data. *Comput. Sci. Eng.* **2021**, *23* (2), 7–14.