Acquiring Enzyme Sequence-Fitness Data at Scale Toward Predictive Methods for Enzyme Engineering

Thesis by

Kadina Elizabeth Johnston

In Partial Fulfillment of the Requirements for
the Degree of
Doctor of Philosophy

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2024

(Defended September 18, 2023)

Kadina Elizabeth Johnston

ORCID: 0000-0002-2214-3534

ACKNOWLEDGEMENTS

Without great friends, colleagues, and advisors, a PhD can be a grueling journey, but I was lucky enough to have a bountiful supply of these people in my life through my five years at Caltech. Although I cannot possibly put all of my thankfulness into words, I hope that I can say some of it. First of all, thank you to my advisor, Frances. You are an inspiration to me, and I have taught me so much about what it is to be a scientist. You have taught me to be confident in what I know, and to prioritize an ability to communicate that science, especially to those outside my field. I will fondly remember both my time spent in your lab as well as the times we spent out of the lab, from dinners at the Rath to hiking in the western Sierras on group retreat.

Next, thank you to my committee: Prof. Yisong Yue, Prof. Steve Mayo, and Prof. Justin Bois. Yisong, it was a privilege to meet with you as often as I did, and I appreciate all of the time you gave to both me and to the Arnold lab as a whole. With your insights we have been able to explore the niche between protein engineering and machine learning, and it broadened my education immensely over my time here. Steve, I had a wonderful time rotating in your lab my first year at Caltech, and I truly appreciated the mentorship and kindness you provided during that rotation. And finally, Justin, your courses showed me a side of biology that I hardly knew existed before coming to Caltech, providing me the tools I needed to make my PhD project a reality.

For those who have been lucky enough to be a part of it, the Arnold lab is a special place, and I have countless lab members to thank, but none more so than Sabine. Sabine, we really could not do science without you in the Arnold lab. You support us, you keep us sane, and you keep us safe. You have been someone I can rely on, and your strength is immense—it has been there for me when my own has faltered. Also, thank you to my very first mentor in the lab, Zach, who taught me everything he knew about where the worlds of protein engineering and machine learning collided.

I have also made countless friendships in the Arnold lab, from those who were there to welcome me warmly into the lab, to those I will leave behind. Ella, whose excitement about science and life during recruitment got me excited to come to Caltech, Anders, who I bonded with over our shared UW-Madison upbringing, and Nat who knew both how to fix everything and made the spiciest comp cells. You were all role models for the type of lab mate I wanted to be. Bruce and Patrick, I still miss our weekly Friday coffees at ML subgroup as we talked about our projects and destressed from the week. Working on evSeq with you was a highlight of my PhD. Thank you also to the current ML subgroup members, especially Francesca, Jason, and Yueming, who I have also been able to collaborate with during my time at Caltech, and my past mentees Grace and Marianne. I look forward to seeing what you all accomplish. Thank you also to all of the past and present members of Office 335. Zach, David, Patrick, Shilong, Jae, Tyler, and Deirdre—we have had some of the greatest discussions about both life and science in that office, and I will miss it and all our memories dearly. Finally, thank you to all of the Arnold lab softball team members. These last two summers were so much fun with all of you—hopefully I will be able to catch one of your games next year.

Some of these friendships led to many adventures outside of the lab as well, including many days of climbing, hiking, camping, and skiing. Austin, Ella, Patrick, Nicholas, and Cynthia, we all have had some incredible trips together, and I hope there will be many more in our future. You have all fed my desire to be outdoors while surrounded by the greatest of friends—a great escape from the troubles of grad school. However, I also appreciated that you all liked to take time to relax as well, often

appearing equally as happy to rehash our weeks at Lucky Baldwins or over card games joined by Bruce, Ali, and Nick.

I was also lucky to have some great friendships outside of the lab. Charles and Dave, you were the best roommates I could have asked for. From furnishing our apartment together, to working out at the Caltech gym, to any of our various woodworking projects, to late night discussions, I always felt that I was learning so much from both of you. Thank you both for being wonderful dog uncles to Sierra too. Drew, although you never officially lived with us, I always thought of you as an extra roommate, and I was happy every time you showed up for horror movie night. Even though I didn't love them as you and Dave, your love of them was infectious and I have grown to think of them fondly.

I also want to thank my family, who have been so understanding of the craziness that is graduate school and being so supportive of my desire to pack up and move across the country to study bioengineering. I have loved my time at Caltech, but I have also cherished the holiday memories we made in between. From Christmas in Florida playing Euchre on the beach to driving across Wisconsin in the middle of a snowstorm to ice fishing in -11 °F weather to hiking in the Spokane rain on a family reunion trip, I have managed to make many happy memories with you all. Thank you to my grandparents, Gary, Betty, and Carolyn, who were always so excited for me to talk about what was happening in California. Thank you to my aunts and uncles, Cory, Maija, Reggie, Treva, and Vonda who always were happy to offer advice. Thank you to my cousins, John, Sam, Aaron, Max, Tyler, Erik, and Allie, who I could always count on to come up with something fun to do with the family. And finally, thank you to my parents Brian and Kelly and my brother Will, who loved me unconditionally and who have shaped me into the person I am today.

Last of all, thank you to my partner, Patrick. You push me to achieve more than I thought I was capable in whatever I am up against, and I am grateful to you for it. Without you, I never would have learned to snowboard, and I would have had only a fraction of the adventures. You are assured in what you believe in, but you also take the time to listen and adjust those beliefs. Your laidback demeanor sets people at ease, and I appreciate your ability to strike up conversation—especially when you are getting advice on how to find the perfect backcountry campsite. I had a blast building out your truck to enable last-minute winter ski trips, and I am excited for all of the places it is yet to take us.

ABSTRACT

The emergence of machine learning methods for expediting directed evolution via protein fitness prediction has recently shed light on the need for more, high quality sequence-fitness data from which to learn the mapping from sequence to fitness. Enzymes specifically are highly selective catalysts and engineered enzymes are becoming increasingly important for human applications such as pharmaceutical synthesis. This thesis thus focuses on the collection of enzymatic sequence-fitness data to enable both development and validation of emerging approaches. Chapter 1 describes the process of traditional directed evolution as well as ways that machine learning methods have been used to accelerate it. It also discusses the experimental considerations for applying machine learning to the various steps of protein engineering campaigns, as the experimental constraints are not always obvious to the machine learning community. One of the major constraints for the application of machine learning methods is the requirement to sequence all variants required for model training, a step that is often skipped by traditional, lab-only directed evolution due to it not being worth the time and cost. Chapter 2 introduces a solution to this problem with “every variant sequencing” (evSeq), which enables higher throughput collection of sequencing data for a similar time and cost as commonly used Sanger sequencing methods. This method not only enables implementation of ML methods such as machine learning-assisted directed evolution (MLDE) and focused training MLDE (ftMLDE) by sequencing variants during an evolution campaign, but also offers promise to fill existing protein sequence-fitness databases with protein engineering datasets. This type of data collection can enable the development of newer, more accurate ML methods, and was an inspiration for the work presented in Chapter 3, which details the collection of a combinatorially complete, epistatic sequence-fitness landscape in an enzyme active site. Oftentimes, the effects of mutations on protein fitness can be considered largely independent and laboratory recombination of them can find an optimal variant. This general principle breaks down when the effects of mutations are not independent, termed epistasis, and sequence-fitness landscapes with these interactions are difficult to traverse. Thus, collection of this dataset provides a challenging task for the development of both ML and physics-based models and pushes the boundary of predictive methods for protein engineering.

PUBLISHED CONTENT AND CONTRIBUTIONS

† denotes equal contribution
*denotes corresponding author

1. **Johnston, K. E.,** Watkins-Dulaney, E. J., Almhjell, P.J., Liu, G., Porter, N. J., Yang, J. & Arnold, F. H.* A combinatorially complete epistatic fitness landscape in an enzyme active site. *Manuscript in preparation.*

K.E.J. and E.J.W conceived the project. K.E.J. designed double- and quadruple-site landscapes and did all experimental data collection except crystallography. K.E.J., P.J.A., G.L., and J.Y. wrote sequence processing and data analysis software. K.E.J. wrote initial draft. K.E.J., E.J.W., P.J.A., and F.H.A edited and revised manuscript.

2. **Johnston, K. E.†,** Fannjiang, C.†, Wittmann, B. J.†, Hie, B. L.†, Yang, K. K.†, & Wu, Z.†* "Machine learning for protein engineering," a book chapter from "Machine Learning in Molecular Sciences" co-edited by Dr. Hanchao Liu and Chen Qu in the series "Challenges and Advances in Computational Chemistry and Physics." *Accepted for publication.*

K.E.J, Z.W., and K.K.Y. determined topics to be covered by the manuscript. All authors contributed to the writing of the manuscript. K.E.J., C.F., B.H., and Z.W. made figures for the manuscript. K.E.J. wrote sections pertaining to the background and experimental considerations of applying machine learning and directed evolution.

3. Wittmann, B. J., **Johnston, K. E.**, Almhjell, P. J., & Arnold, F. H.* evSeq: Cost-effective amplicon sequencing of every variant in a protein library. *ACS Synth. Biol.* **11**, 1313–1324 (2022). doi: 10.1021/acssynbio.1c00592.

B.J.W. conceived the project and performed initial design and execution of research and software development. B.J.W., K.E.J, and P.J.A. optimized the experimental workflow and software. K.E.J. and P.J.A. wrote software for data visualization and installation. B.J.W., K.E.J., and P.J.A. wrote the manuscript and prepared figures.

4. Wittmann, B. J., **Johnston, K. E.**, Wu, Z., & Arnold, F. H.* Advances in machine learning for directed evolution. *Curr. Opin. Struct. Biol.* **69**, 11–18 (2021). doi: 10.1021/acssynbio.1c00592.

B.J.W. and K.E.J. determined topics for the review together, wrote the manuscript, and constructed figures. B.J.W., K.E.J., Z.W., and F.H.A. revised and edited the manuscript.

PUBLISHED CONTENT NOT INCLUDED IN THESIS

† denotes equal contribution
*denotes corresponding author

5. Wu, Z., **Johnston, K. E.**, Arnold, F. H., & Yang, K. K.* Protein sequence design with deep generative models. *Curr. Opin. Chem. Biol.* **65**, 18–27 (2021). doi: 10.1016/j.cbpa.2021.04.004.

Z.W., K.K.Y., and K.E.J. generated ideas for and wrote the manuscript. Z.W. and K.K.Y. made figures for the manuscript. All authors edited the manuscript.

6. Dallago, C.†, Mou, J.†, **Johnston, K. E.**, Wittmann, B. J., Bhattacharya, N., Goldman, S., Madani, A., & Yang, K. K. FLIP: Benchmark tasks in fitness landscape inference for proteins. *bioRxiv.* (2021). doi: 10.1101/2021.11.09.467890.

K.E.J. assisted in conceptualization of the work, including selection of the datasets and splits to provide. K.E.J. made figures for the manuscript and assisted in writing and editing of the manuscript.

7. Yang, J., Ducharme, J., **Johnston, K.E.**, Li, F-Z, Yue, Y., & Arnold F. H.* DeCOIL: Optimization of degenerate codon libraries for machine learning-assisted protein engineering. *ACS Synth. Biol.* **12**, 2444–2454 (2023). doi: 10.1021/acssynbio.3c00301.

J.Y. conceptualized the work. K.E.J. assisted with methodology, experimental data collection, and writing and revision of the manuscript.

8. Almhjell, P. J., **Johnston, K. E.**, Porter, N. J., Kennemur, J. L., Bhethanabotla, V.C., Ducharme, J., & Arnold, F. H.* The β-subunit of tryptophan synthase is a latent tyrosine synthase. *Manuscript in review.*

P.J.A. conceptualized the work. K.E.J. assisted with methodology, experimental data collection, and writing and revision of the manuscript.

9. Sarai, N.S. †, Fulton, T.J. †, O’Meara, R.L. †, **Johnston, K. E.**, Brinkmann-Chen, S., Maar, R.R., Tecklenburg, R.E., Roberts, J.M., Reddel, J.C.T., Katsoulis, D.E.*, & Arnold, F.H.* Directed evolution of enzymatic silicon–carbon bond cleavage in siloxanes. *Manuscript in preparation*.

N.S.S. conceptualized the work. K.E.J. assisted with experimental data collection and revision of the manuscript.

TABLE OF CONTENTS

LIST OF FIGURES, TABLES, AND SCHEMES

Figure Number *Page*

ABBREVIATIONS

Å	Ångstrom
carb	carbenicillin
COMM domain	communication domain
epPCR	error-prone PCR
GC	gas chromatography
GC-MS	gas chromatography with mass spectrometry
GOI	gene of interest
h	hour(s)
HPLC	high performance liquid chromatography
HPLC-MS	high performance liquid chromatography with mass spectrometry
k	rate constant
K_{M}	Michaelis constant
KPi	potassium phosphate
LB	Lysogeny Broth (Luria-Bertani medium)
LC-MS	liquid chromatography with mass spectrometry
min	minute(s)
OD_{600}	optical density at 600 nm
PCR	polymerase chain reaction
PDB	Protein Data Bank
PLP	pyridoxal 5’-phosphate
RT	room temperature or retention time
Ser	L-serine
StEP PCR	staggered extension process PCR
SOB	Super Optimal Broth
SOC	Super Optimal broth with Catabolite repression
SSM	site saturation mutagenesis
TB	Terrific Broth
T_{50}	temperature at which a 1 h incubation irreversibly inactivates 50% of a protein
Tm	*Thermotoga maritima*
*Tm*TrpB	TrpB from *Thermotoga maritima*
Trp	L-tryptophan
TrpA	Tryptophan synthase α-subunit
TrpB	Tryptophan synthase β-subunit
TrpS	Tryptophan Synthase αββα dimeric complex
UV-vis	ultraviolet visible
WT	wild type

Chapter I

LEARNING FROM PROTEIN SEQUENCE-FITNESS DATA

Material from this chapter is reproduced with permission from Springer Nature and appears in: "**Johnston, K.E.**[†], Fannjiang, C.[†], Wittmann, B.J.[†], Hie, B.L.[†], Yang, K.K.[†], Wu, Z.[†]* "Machine learning for protein engineering," a book chapter from "Machine learning in molecular sciences" co-edited by Dr. Hanchao Liu and Chen Qu in the series "Challenges and advances in computational chemistry and physics." *Accepted for publication*."

[†]Equal authorship

*Corresponding author

K.E.J, Z.W., and K.K.Y. determined topics to be covered by the manuscript. All authors contributed to writing of the manuscript. K.E.J., C.F., B.H., and Z.W. made figures for the manuscript. K.E.J. wrote sections pertaining to the background and experimental considerations of applying machine learning and directed evolution.

Material from this chapter appears in: "Wittmann, B.J., **Johnston, K.E.**, Wu, Z., & Arnold, F.H.* Advances in machine learning for directed evolution. *Curr. Opin. Struct. Biol.* **69**, 11–18 (2021). doi: 10.1021/acssynbio.1c00592."

*Corresponding author

B.J.W. and K.E.J. determined topics for the review together, wrote the manuscript, and constructed figures. B.J.W., K.E.J., Z.W., and F.H.A. revised and edited the manuscript.

ABSTRACT

Directed evolution (DE) via iterative mutation and screening has proven itself time and again as an effective method for protein engineering. Through either single-step walks up a fitness landscape or recombination-based approaches, this laboratory-based method has proved a difficult baseline to beat. However, recent advances in machine learning (ML) approaches have shown that it is possible, with some ML-assisted DE implementations more efficiently reaching improved variants, especially on epistatic fitness landscapes. As the field of ML for DE has grown, new avenues have emerged, where ML might assist with identifying starting points or in designing libraries to test in the lab. Despite this, there has remained a disconnect between those doing experiments and those doing predictions, as laboratory constraints and practicalities are not always obvious. Furthermore, a specific protein engineering problem may have very constraints than another; there are a wide range of assay throughputs as well as assay-specific levels of noise that should be considered. This introduction seeks to provide experimental insight to those seeking to predict protein fitness, describing both situations where predictive models might help as well as experimental constraints new predictive approaches should consider. It also motivates the need for sequence-fitness datasets if we hope to develop even better predictive models.

1.1 Background

Enzymes provide solutions to life's most challenging chemical problems. The ability of enzymes to catalyze chemical reactions efficiently and selectively makes them useful not only to their host organisms, but also for myriad applications that humans have devised. As green, cheap, efficient catalysts, enzymes have been taken up by industries ranging from pharmaceuticals to consumer products, materials, food, and fuels, and their importance is expected to continue to grow[1–3].

Enzymes and many other proteins useful to humans often must function in non-native environments (non-aqueous solutions, high temperatures, in the presence of surfactants, etc.) that eliminate or reduce the activity of the natural protein. Additionally, although enzymes exhibit remarkable selectivity, they typically have a limited substrate scope, which often means that a new enzyme must be optimized for new target reactions or applications by engineering its amino acid sequence[4,5].

A protein's sequence encodes its function (the level of which is termed "fitness"), and the relationship between them is often conceptualized as a surface in high-dimensional space called the protein fitness landscape[6,7]. New proteins are developed by searching this landscape, but importantly, the number of possible protein sequences is immense, presenting a significant challenge to any protein engineering strategy. A typical protein is several hundred amino acids long, and at each position there are twenty canonical amino acids possible, resulting in 20^{300} (10^{390}) possible sequences. Interestingly, this does not mean that finding functional proteins or improving protein fitness is a hopeless endeavor. For evolution to occur, functional protein sequences must neighbor other functional sequences in protein sequence space, implying that functional proteins exist clustered together within a vast sea of non-functional ones[6]. Therefore, rather than throwing metaphorical darts at this astronomical space and hoping for a hit, protein engineers can begin with a small level of activity for their function of interest and leverage methods for local exploration to improve it. The process most commonly used is directed evolution[7].

Directed evolution proceeds by subjecting a protein having at least a small amount of the desired function to iterative rounds of mutagenesis and screening, using the best variant in each round as the starting point for the next until the functional goal is achieved (**Figure 1-1A**). First, a protein with a measurable amount of the property of interest must be identified. Many protein engineering projects fail at this stage, since biochemical ingenuity is required to identify a protein able to accomplish a new goal such as breaking down polyethylene terephthalate (in man-made plastics[8]) or selectively modifying DNA for targeted gene therapy.[9] Second, the protein sequence is randomized to generate a pool of variants, often called a library. In nature, this process typically occurs through random mutagenesis, where random mutations in DNA correspond to random changes in the protein sequence, or recombination of existing protein fragments. Third, the library is tested for the desired property. Some examples of approaches capable of generating the largest amounts of data (high-throughput assays), are based on protein fluorescence or binding, and reach hundreds of thousands of labels per month. However, other properties such as enzymatic activity for generating small molecule substrates are measured in much lower throughput. Thus, extensive laboratory characterization remains a bottleneck for the development of many engineered proteins.

To reduce the experimental burden of directed evolution, protein engineers are increasingly turning to in silico strategies for screening, particularly machine learning (ML). When applied to directed evolution, ML has thus far largely been cast as a supervised problem; that is, given a set of protein sequences with associated labels (e.g., catalytic activity, stability, etc.), the task is to learn a function that can predict the label of previously unseen sequences (**Figure 1-1B**). Using this function, large numbers of proteins can be evaluated computationally during each cycle of evolution, enabling much greater exploration of the protein fitness landscape than could be accomplished with laboratory screening alone.

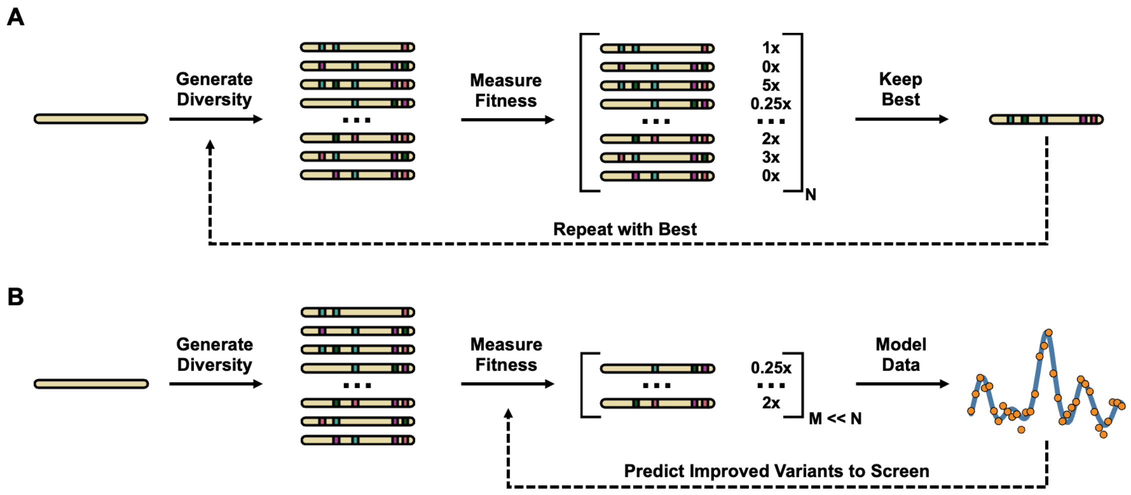


Figure 1-1. Example workflows of traditional and ML-assisted directed evolution. Both workflows begin by identifying a protein with activity for a target function. Once the starting point is identified, diversity is introduced by mutagenesis and resulting variants are screened for function. **A** In traditional directed evolution, many variants are screened and the best variant is then fixed as the parent for the next round of mutagenesis/screening. **B** When applying supervised machine learning to directed evolution, fewer variants are screened. Using the resulting sequence-function data, a function is fit that relates protein sequence to protein fitness (e.g., for f(x) = y, 'x' is the protein sequence and 'y' is the protein fitness). This function can be used to predict the fitness values of variants not experimentally evaluated or to propose a new set of variants to screen in the next round of evolution.

1.2 Protein fitness landscapes contain non-additive interactions that can constrain evolution

Due to the iterative nature of the optimization cycle that is directed evolution, it is commonly conceptualized as a greedy, uphill walk up the protein fitness landscape towards a fitness peak[7]. Each round of mutagenesis and screening searches through the local landscape, typically sampling only a few mutations away from the current position in sequence space. When a hit is identified, a step toward the fitness peak is taken and the local search is repeated, with the entire process continuing until the fitness is satisfactory or a peak is reached. Importantly, no downward steps into valleys of the fitness landscape are typically allowed in directed evolution.

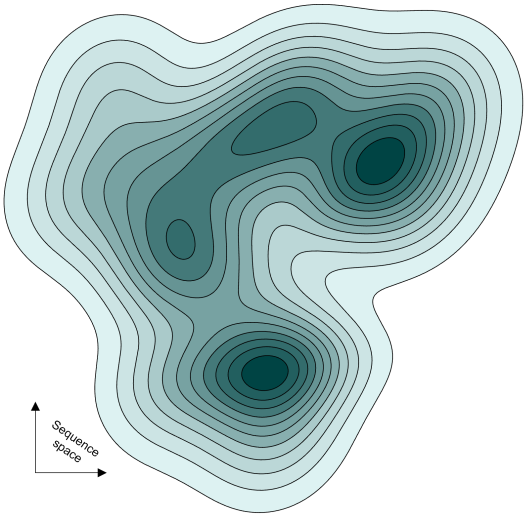


Figure 1-2. Two-dimensional representation of a protein fitness landscape. Sequence space is represented in the x and y axes despite being much more high-dimensional in practice. Fitness is represented by the contours of the map.

Fitness landscapes are often visualized as smooth, easy-to-navigate surfaces, but in reality, they are discrete, high-dimensional spaces, with many of the dimensions being quite rugged. This ruggedness is due to a phenomenon known in biology as epistasis, where mutational effects are dependent on higher order interactions rather than their individual contributions[10]. Epistasis arises most commonly from direct structural contacts, but interactions between residues can also be modulated by ligands, substrates, allostery, cofactors, or conformational dynamics. As a result, it is often reasonable to assume that distant mutations are mostly independent, but there are important cases where this assumption breaks down and epistasis must be considered. Otwinowski *et al.* explored the prevalence of global epistasis, where the mapping of genotype to phenotype showed inherent non-linearity not just due to pairwise interactions between residues[11]. This effect varied in magnitude depending on the protein being studied.

Intertwined with the idea of global epistasis, it is important to consider the pre-requisite protein properties that must be satisfied to take fitness measurements, such as expression, stability, and substrate binding. This means that fitness landscapes are a result of some combination of these factors, and changes in any of them can modulate fitness or cause epistasis. For example, Romero & Arnold outline how negative epistasis can arise from a protein stability threshold, where beneficial, but destabilizing mutations combine to completely ablate activity[7]. In fact, any mutation is more likely to be destabilizing than neutral or stabilizing, and therefore, most activity-improving mutations that improve activity are also most likely to be destabilizing[12]. Therefore, more stable starting proteins can be easier to evolve, as they allow for larger decreases in stability that occur as activating mutations are discovered[13].

1.3 The promise of machine learning for directed evolution

In traditional DE, the top k variants (often k = 1) are fed back into the diversity generation step for further improvement. For methods where DE is further enabled by ML, two other steps may follow the initial fitness determination step. Importantly, the advantages offered by ML currently depends on the protein being evolved and its fitness assay. Some protein engineering projects are already able to test millions of variants in a single round, minimizing the value of an ML approach. Other projects require a day or more to acquire a single data point and would not even be possible without further guidance from sources such as machine learning.

For ML-assisted approaches, the step following fitness determination is to fit ML models to the relationship between protein sequences and their fitness labels. A wide variety of approaches are available to the machine learning practitioner here, and there are multiple sources of prior knowledge that can be leveraged for proteins. One example is the rich historical record of protein sequences, which can be obtained from sequence databases such as UniRef.[14,15] From such a database, sets of evolutionarily related sequences (homologs) can be obtained and aligned in Multiple Sequence Alignments (MSAs), which can be used as priors on viable sequences. However, while this history represents sequences retained in nature, it does not necessarily represent the distribution of allowed sequences for a specific protein on a specific engineered task, especially for non-natural activities. For example, mutating the axial ligand of cytochrome P450$_{BM3}$ from a cysteine to a serine unlocked multiple non-natural activities, but an MSA would show high conservation of the cysteine residue and disfavor mutation at this position.[16]

The final step of an ML-assisted approach is to use a trained model to select optimal proteins for experimental validation. Again, a variety of approaches have been employed in this step such as gradient-based, reinforcement-learning-inspired, and active learning methods. The sampling strategy often depends on the modeling approach used in the previous step. Additionally, this step may be constrained by cost and availability of current molecular biology techniques. Such constraints have typically been enforced manually, but one can envision encoding them in the design process as well. From here, proposed sequences are tested for activity. At this point, a new diversity generation step could be pursued, more sequences could be proposed, or ML-assisted DE could be complete.

1.4 Identifying a starting variant for directed evolution

The first consideration in a directed evolution experiment is selecting the protein variant to evolve, which can be a nontrivial task. In the most extreme setting, there may be no known proteins that perform the desired function; a related setting is when the desired output of the directed evolution experiment needs to be substantially different from all known proteins due to considerations of scientific novelty or intellectual property.

1.4.1 Experimental considerations for finding starting variants

When screening for new activities, protein engineers typically begin by searching annotated databases of existing sequences and structures, looking for proteins which perform a similar function to the desired one. Such proteins are hoped to have at least a small level of "promiscuous" activity for the new function. For enzymes, this could mean looking for ones that have the desired mechanism but act on a different substrate. As another approach, plates of variants from previous evolution campaigns might be screened for a new but similar activity. These methods ultimately rely on the latent promiscuity of extant sequences, as the probability that a functional sequence is found through a random search is approximately zero.

1.4.2 Applying machine learning to finding starting variants

Alternatively, ML would be a useful way to identify a novel starting variant, typically with weak or suboptimal fitness, that can subsequently be given to a traditional or ML-assisted DE pipeline. Existing approaches fall somewhere between using ML to propose large and diverse collections of proteins to test for nonzero fitness values[17,18] and using ML to de-novo design the initial, functional protein.[19–21] As an example of the first approach, Shin *et al.* use an autoregressive language model, trained on approximately 1.2 million natural llama nanobody sequences, to generate a nanobody library that is screened to potentially identify novel binders to a target protein.[17] The generative model enables improved sequence diversity over previous synthetic libraries, enabling the authors to identify new proteins with high binding affinity using an efficient set of approximately 10^5 generated sequences, which is 1000-fold smaller than other libraries. The second approach, which promises robust starting points with desired properties without the need to physically screen lots of generated proteins, is an ideal yet unreached. However, progress towards designing new enzymes with new functions has continued, with the most notable example being luciferase enzymes designed in a completely novel scaffold with a combination of deep learning and rational design.[21]

These approaches target cases in which the starting variant is unknown or must be substantially different from existing proteins, but in many other cases a good, functional variant might already exist. DE could be used directly to evolve this variant, but it may be in a local optimum of the fitness landscape from which it is difficult to escape. Therefore, it may be desirable to find an alternate starting point or multiple starting points that could be more evolvable. Engineering a more evolvable starting variant, even at some fitness cost, is currently an open question,[22] as it is often unclear *a priori* if a given starting point is more evolvable. ML may be of some help here, as proteins with higher intrinsic stability are thought to be more evolvable[13], and many ML models have been developed that either directly (via supervision) or indirectly (as an emergent property of an unsupervised model) predict stability.[23,24]

1.5 Designing and building protein variant libraries

The data used to train an ML model determines what it learns and, by extension, in what situations it can be used to make effective predictions. For protein engineering, this means that the design of the library that will provide training data is critical to the eventual effectiveness of the trained model in finding improved sequences.

1.5.1 Experimental methods for designing and building protein variant libraries

After a starting point for evolution is identified, a variety of molecular biology methods are available for generating local sequence diversity. While these steps are physically separate from the downstream assay, which pairs a sequence to a fitness label, the assay throughput is a key factor in selecting the method of diversity generation. Several methods and their biases, an important consideration for applying ML, are introduced below, but notably, these methods are often combined in protein engineering campaigns.

Random mutagenesis is one of the most straightforward methods for creating initial sequence diversity. Errors are introduced throughout an initial DNA sequence by either randomly damaging DNA or by introducing errors during replication such as via error-prone PCR (epPCR), a process that introduces mutations randomly by increasing the error rate of the copying enzyme, the polymerase. However, it is important to note that errors introduced via "random" mutagenesis are not perfectly random in two major ways. First, mutations from one nucleotide to another do not occur at identical frequencies, so the original base can dictate what mutations are most likely at a given position.[25] Second, the genetic code is redundant, with the twenty canonical amino acids encoded by 61, three-nucleotide codons. Although it is possible for multiple nucleotides within a single codon to mutate simultaneously, this is rare, and generally only one nucleotide mutation occurs per codon during random mutagenesis, limiting the mutations available at the amino acid level. In comparison to other mutagenesis methods, the protein engineer has much less control over the generated library — only the rate of mutations can be changed.

Targeted mutagenesis, of which the focused mutagenesis discussed in the previous section is a subset, is an alternative to random mutagenesis that affords more control over the final library. Unlike random mutagenesis, targeted mutagenesis typically assumes that either (1) specific sites in the protein sequence are important to mutate or (2) it is important to be able to access all amino acids at a given position. Site selection usually requires structural knowledge or other biochemical insights into the protein system, and, unlike random mutagenesis, any amino acid can be accessed with equal probability. The pool of degenerate DNA oligos used for mutagenesis can also be modified to achieve a relatively even distribution across all twenty canonical amino acids[26] or to achieve a different distribution of interest.[27,28] Importantly, as the desired amino acid distribution becomes more complex and more sites are mutated simultaneously, both difficulty of laboratory implementation and the cost of oligos can become untenable, approaching direct gene synthesis costs). Therefore, ML methods relying on targeted mutagenesis for either training set design or evaluation of predicted designs must keep in mind these constraints.

Another strategy for library generation is recombination, which pieces together, or "recombines," initial diversity into different arrangements to create new diversity. The choice of recombination strategy typically relies on the type of initial diversity on hand. Such diversity could be comprised of a set of functional, homologous proteins, the top variants from a random mutagenesis library, or the top variants from a targeted mutagenesis library.

Another approach for recombination is the use of SCHEMA libraries,[29] where fragments of multiple parent proteins (selected by conserving contacts) are swapped, and which has been successfully engineered with ML methods.[30,31] Importantly, some recombination strategies are quite experimentally straightforward and a single round of recombination on top variants can yield much higher improvements in fitness than a single round of random or targeted mutagenesis. Overall, recombination is a very broad category of diversity generation, and due to the array of recombination strategies available, we have only briefly mentioned a few. Further strategies are well reviewed by Packer & Liu.[32]

Library design has typically been constrained by methods of library generation, but there is promise to disrupt these traditional paradigms with the advent of cheap gene synthesis technologies.[33] Rather than starting with an initial sequence or pool of sequences and building diversity with mutagenesis or recombination, a set of desired sequences can be synthesized directly for under $100 per protein — a cost which is currently dropping, particularly for proteins shorter than 100 residues. Ordering pools of sequences with targeted or random mutations is also possible, and emerging synthesis technologies can impart more control over the final distribution. As these DNA synthesis technologies improve, ML methods for protein engineering can begin to leverage and propose precisely defined libraries that were previously cost-prohibitive.

1.5.2 Applying machine learning to designing and building protein variant libraries

ML models tend to be more effective at interpolation than extrapolation and so will typically perform best when used to make predictions in the same domain as the data used to train them. In general, for a given design space of allowed proteins, this translates to collecting maximally diverse training data that best covers that space. For proteins, this means that training data with maximal sequence diversity will be most informative for modeling an underlying true fitness landscape: the more diverse the training sequences are, the more of the design space that is covered by the training data and the less a model must extrapolate to previously unseen regions of sequence space. For example, Romero *et al.*, Bedbrook *et al.*, and Greenhalgh *et al.* maximize the information entropy of the initial set of sequences when engineering P450s, channelrhodopsins, and acyl-ACP reductase, respectively.[30,34,35]

Randomly collecting sequences from a fixed design space (e.g., a combinatorial space defined by a given number of positions in a protein) can thus be a valuable strategy for training data collection, as this will on average result in the collection of highly diverse sequences. Random collection of training data is also an attractive approach based on available lab methods discussed previously, and this strategy has been combined with ML methods to engineer halohydrin dehalogenase,[36] fluorescent proteins,[37,38] and an adenovirus capsid protein.[39]

While building a perfect map of a fitness landscape would be ideal for model-guided engineering, it is not always feasible given our limited ability to collect experimental data. More complex fitness landscapes considering larger sections of sequence space require more data to model and a small amount of randomly selected training data may be spread too thinly across the design space to build a comprehensive map.[40] The goal of ML-assisted protein engineering is not to comprehensively map fitness landscapes, but to use ML to guide exploration of fitness landscapes to reach higher-fitness protein variants. As a result, if training data is expensive to collect, then it can be advantageous to build focused initial libraries that are biased toward protein variants believed *a priori* to be higher in fitness. It is more important to be able to identify the highest-fitness variants from the set of high-fitness variants than the lowest-fitness variants from the set of low-fitness variants, and so the idea of this strategy is to model (potentially) higher-fitness regions of the protein fitness landscape at higher resolution and lower-fitness regions of the protein fitness landscape at lower resolution.

Focused libraries can be particularly helpful when navigating protein fitness landscapes filled with many zero-fitness proteins, or “holes”.[41] As more mutations are made to a protein, the probability that it retains function decreases exponentially,[13] and so fitness landscapes consisting of combinations of mutations at multiple positions (combinatorial landscapes) tend to be dominated by such holes. These variants are conceptually distinct from fitness valleys mentioned previously, as they do not provide information about the *extent* to which a mutation impacts protein fitness, which is valuable information for training the regression models typically employed for ML-assisted protein engineering. Wittmann *et al.* demonstrated that by using so-called zero-shot predictors—models or strategies that can predict protein fitness prior to collection of new experimental data—focused training sets can be constructed that minimize inclusion of holes in training data.[41] Through simulation on a complex, hole-filled, combinatorial fitness landscape, they showed that models trained with these focused training sets tend to be far more effective at identifying the highest-fitness variants than models trained with data drawn randomly from the landscape.

The prior information needed to construct focused libraries can come from many sources. For instance, prediction of protein thermal stability,[41] use of meta-predictors of protein fitness,[42] or strategies based on evolutionary conservation can all be used to make zero-shot predictions of protein fitness.[43–47] The exact strategy that will be most effective, however, will vary depending on the fitness and protein being optimized. Take, for instance, zero-shot strategies that rely on sequence conservation. Such strategies assume that evolutionary fitness aligns with whatever fitness is being predicted; that is, they assume that mutant proteins more closely resembling known protein sequences (found in databases of protein sequence such as UniProt[48]) are more likely to be functional than others. Should this assumption not hold (for instance, the fitness of a protein being engineered for a new-to-nature activity may not correlate well with evolutionary fitness), or if there are simply not enough homologous protein sequences available to build an effective sequence-based zero-shot prediction model, then the zero-shot predictions are likely to be inaccurate. Inaccurate zero-shot predictions are unhelpful for focused library design: indeed, they may even be detrimental to effective learning by focusing training data collection on regions of the fitness landscape dominated by holes.

Ultimately, the decision between random library design and focused library design will depend on a number of factors. If the fitness landscape to be explored is expected to be minimally complex with few holes and large amounts of training data can be easily collected for it, then random library design is a reasonable approach, as the library itself will be simple to construct, and training data gathered from it will be sufficient to build a comprehensive map of the fitness landscape. If the fitness landscape to be explored is complex, full of holes, or it is challenging to gather training data for it, then focused libraries may be more viable, particularly if high-confidence zero-shot predictions can be made for the fitness landscape. Such libraries may be more challenging to construct in the laboratory, but they will likely result in more efficient ML-guided engineering. These are the applications where the dropping costs of gene synthesis will have the largest impact, but new methods are being developed to build focused libraries within the constraints of current molecular biology technology.[28]

1.6 Collecting protein sequence-fitness data

Once an initial variant has been selected and an initial library defined and built, is time to collect protein sequence-fitness data through screening (direct measurement of individual variants) or selection (assay where variants “compete” against each other, sometimes resulting in only the top variants remaining).

1.6.1 Assaying protein fitness creates labels for machine learning

With proteins performing such a wide variety of different functions, the protein engineering community has had to devise countless different assays to measure them all. As such, assays for protein function vary widely in both accuracy and throughput, with ranges from tens to millions of protein variants (**Figure 1-3**). This amount is typically dependent on the project definition of fitness. For instance, if the measurement of fitness can be directly coupled to a sequencing assay (as in deep mutational scanning, DMS), then large datasets (10^5–10^6) can be rapidly created. Many assays for fitness, however, are limited to comparatively low-throughput chromatographic methods (e.g., HPLC, LCMS, GCMS, etc.), which rely on physical separation of a mixture through a column, producing smaller datasets (10^1–10^4). The amount of data available for training a model will dictate how much of sequence space can be explored and how accurate the predictions of fitness for new sequences will be. At the same time, however, the goal of ML in protein engineering is to reduce the burden of experimental screening and expedite the process, so a balance must be found between the amount of data collected and the accuracy of predictions of models trained on that data.

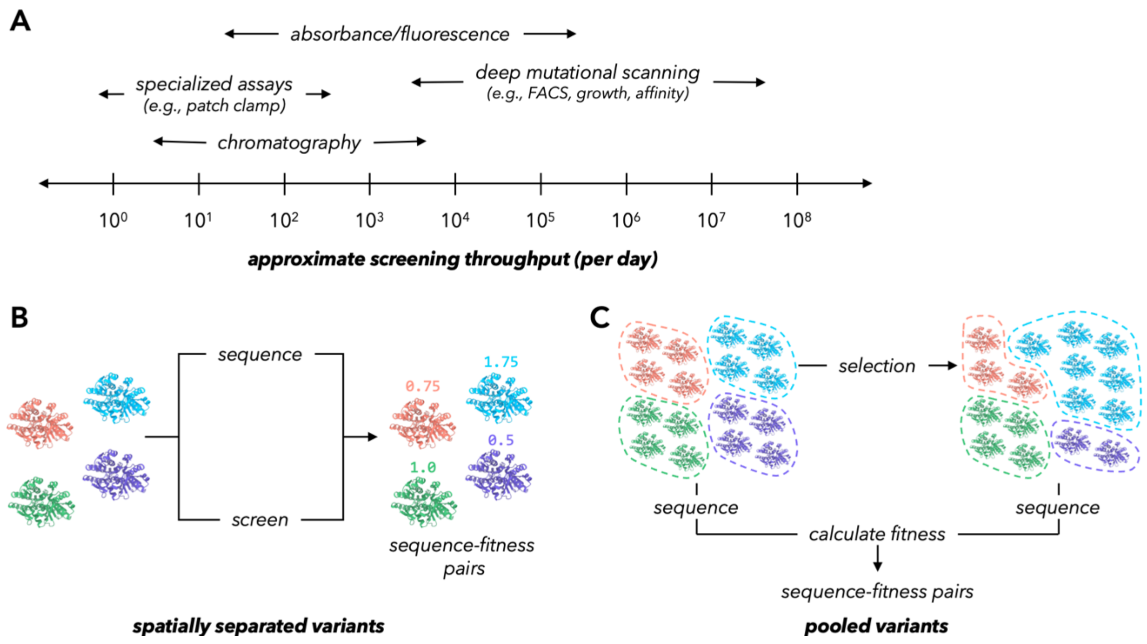


Figure 3. Many different screening throughputs exist for protein fitness. **A** Screening throughput can range from less than one variant per day for some specialized experiments to over 10^7 per day for some DMS-based screens. **B** Many protein screens require spatial separation of variants, including many chromatographic methods. This then requires separate sequencing and screening processes that can be later be combined into complete sequence-fitness datasets. **C** With deep-mutational scanning assays sequencing and fitness are linked. Sequencing is performed pre- and post-selection and changes in read frequency are used to calculate a fitness value for every variant

Additionally, one should consider where an ML method for protein engineering may have the biggest impact on efficiency, cost, and time. Protein functions that can be assayed in high-throughput provide more data for training downstream ML models; however, applying an ML-based engineering method to such a function may not impart as much benefit as it would being applied to engineering a function with a low-throughput assay. Therefore, many recent efforts for building ML methods for protein engineering have focused on the low-N regime, where few samples are used for training.[31,41,49–52]

Final considerations, for both the protein engineer choosing an assay and the ML scientist choosing a dataset, are the assay bias and noise. As interest has grown in applying ML to protein engineering, method developers have sought out sequence-fitness datasets with which to benchmark their approaches. This has typically resulted in the pursuit of large datasets built from multiplexed assays of variant effects (MAVEs) and using some fitness-related selection such as fluorescence-activated cell sorting (FACS) for a fluorescent protein.[53] Inherent biases in such protein fitness datasets are left under-discussed, making it difficult for the ML community to discern what datasets are appropriate for a given application.

Sequencing-based fitness measurements can be heavily impacted by the input library and selection methods, resulting in highly non-uniform error across a dataset.[54] For example, FACS-based assays have inherent bias due to fluorescence-based binning. Post-binning sequence processing attempts to smooth the categories into a continuous function, but systemic bias persists, leading Trippe *et al.* to propose a random gating strategy to reduce it.[55] Noise can be more difficult to address as there tends to be a balance between throughput and measurement accuracy for biological systems. However, examining and quantifying the noise within a dataset can be important for setting expectations for the success of downstream ML model. This could be done by comparing a subset of high-throughput measurements to more accurate, low-throughput measurements or using published statistical packages for data processing and analysis.[54,56] Note that even low-throughput assays can have their own noise and bias, so this should always be thoroughly considered when developing an ML method on any dataset.

1.6.2 Pairing sequences to fitness data through sequencing creates features for machine learning

Uncommon in the broader ML discipline, datasets that result from typical protein engineering campaigns are label-rich and feature-poor, with many assayed fitness values and relatively few variant sequences. Because only function is being optimized in directed evolution, this process does not technically require sequencing. Indeed, only the top few variants from each round are sequenced for validation in practice, and sequencing all the variants is considered an unnecessary, unjustifiable expense. Thus, the remainder of the unimproved variants are discarded without sequencing, resulting in fitness labels that are rendered useless. Notably, this is not true for MAVE libraries, where fitness is directly coupled to sequencing, which is a large part of the reason many ML approaches are currently developed on these types of datasets.[41,51,57–59]

Variant sequencing, especially for low-throughput assays, has traditionally been done via Sanger sequencing, but the cost of this method scales linearly with the number of variants, typically at a few dollars per sequence. Depending on the assay used for evaluating protein fitness, sequencing could easily become the most expensive part of a protein engineering campaign.

Fortunately, next generation sequencing (NGS) technologies[60] have begun disrupting this paradigm. The MAVE strategies mentioned in the previous section rely on sequencing to measure fitness, meaning variant sequencing and fitness assaying happen simultaneously.[61] New sequencing methods that incorporate NGS show promise to continue shifting the sequencing paradigm by spreading reads over many, multiplexed sequences, enabling large-scale sequencing for individually screened fitness values. For such methods, there is a trade-off between read length and cost per sequence. Both long- and short-read methods can return full-length sequences, but they typically require filling an entire NGS flow cell, which can cost upwards of $1000 per run and a few dollars per variant.[62,63] Alternatively, short amplicons from within an entire gene, which are restricted to only a few hundred amino acids, can be sequenced for cents per variant and are a cost-effective compromise when variation is confined to a smaller region of a gene.[64] Nonetheless, both sequencing technologies and sequencing methods for the protein engineering community have been growing rapidly in the past few years, showing promise to address these shortcomings in the near future.

1.6.3 Using alternative data sources to reduce the number of required experiments

Sequence-fitness data are not the only data that can be useful when applying ML to DE, and there are a variety of databases that can be leveraged. Of particular note are UniRef,[14,15] UniProt[48] and the Protein Data Bank (PDB).[65,66] From such a database like UniRef or UniProt, sets of evolutionarily related sequences (homologs) can be obtained and aligned in Multiple Sequence Alignments (MSAs), which can be used as priors on viable sequences. However, while this history represents sequences retained in nature, it does not necessarily represent the distribution of allowed sequences for a specific protein on an engineered task, especially for non-natural activities. For example, as mentioned earlier, mutating the axial ligand of cytochrome P450$_{BM3}$ from a conserved cysteine to a serine unlocked multiple non-natural activities. However, an MSA would show high conservation of the cysteine residue and disfavor mutation at this position.

UniProt also contains a large amount of information about proteins beyond their sequence, including cross-references to functional labels, disease-association and Protein family (Pfam) classifications at the per-residue level,[67] Gene Ontologies on the per-protein example,[68,69] and links to several other databases. UniProt also releases reference protein clusters which are currently almost ubiquitous as the training set for large protein language models.[70] This dataset will continue to grow as more metagenomes are sequenced[71] and more unique proteins are identified. Some approaches to modeling proteins are conditioned on protein functional labels[20] or other data available in UniProt,[57] but there is no clear optimal approach to incorporating annotations about all proteins for protein engineering campaigns, which are often focused on specific protein families.

The PDB is the primary source of experimental protein structure data, and it largely consists of static protein structures obtained through protein crystallography, although the number of structures obtained through cryo-EM and NMR are also increasing. Protein structures are invaluable to biologists in providing much-needed context for molecules that are otherwise difficult to probe. However, mutations may have effects that are not captured by static structures. For example, they may bias the protein's Boltzmann distribution toward different conformational states in the ensemble without perturbing the ground state crystal structure, or the crystal structure may simply have too low resolution to capture small changes. Nonetheless, structure can be a useful prior in directed evolution[41,72] and can also be directly applied to identify beneficial mutations.[73]

AlphaFold[74] has the potential to enable protein sequence-based protein engineering methods to incorporate structural information. AlphaFold incorporated several biophysical inductive biases in the CASP14 protein structure prediction contest. These include a variant of attention to account for the triangle inequality on distances, a lowered emphasis on the linear input sequence of a protein (which folds into a three-dimensional structure), and a variant of axial attention for the MSA. These methods have been successful for improving protein structure prediction, and it is likely that ML methods adapted from them (or even the predictions themselves) will improve each step of the protein engineering cycle.

This prior information can be leveraged for construction of focused libraries (as described in the previous section). Prediction of protein thermal stability,[41] use of meta-predictors of protein fitness,[42] or strategies based on evolutionary conservation can all be used to make zero-shot predictions of protein fitness.[43–47] The exact strategy that will be most effective, however, will vary depending on the fitness, predictor, and protein being optimized. Take, for instance, zero-shot strategies that rely on sequence conservation. Such strategies assume that evolutionary fitness aligns with whatever fitness is being predicted; that is, they assume that mutant proteins more closely resembling known protein sequences are more likely to be functional than others. Should this assumption not hold (for instance, the fitness of a protein being engineered for a new-to-nature activity may not correlate well with evolutionary fitness), or if there are simply not enough homologous protein sequences available to build an effective sequence-based zero-shot prediction model, then the zero-shot predictions are likely to be inaccurate. Inaccurate zero-shot predictions are unhelpful for focused library design: indeed, they may even be detrimental to effective learning by focusing training data collection on regions of the fitness landscape dominated by holes.

1.7 Training a protein sequence-fitness model

In traditional directed evolution, the top set of variants are fed back into the diversity generation step for further improvement. For methods where directed evolution is further enabled by supervised ML, two other steps may follow (Figure 2B). First, a model is trained on the sequence-fitness data collected. There are a large variety of models available to the ML practitioner and tools have emerged to make implementing a new model relatively straightforward. Once trained, the model can be used to predict a top set of variants to screen and another round of experimentation is performed.

1.7.1 Representing proteins for machine learning models

Proteins are variable-length sequences composed of twenty canonical amino acids that fold into three-dimensional structures to carry out their function. Due to both the variability in sequence length and the categorical nature of amino acids, an important and long-standing problem has been how to best represent a protein to facilitate ML. Roughly speaking, there are three general approaches for doing this, including (1) simple, one-hot encodings; (2) biophysical properties based on individual amino acids, structures, or full-protein simulations; and (3) using evolutionarily related sequences to learn likelihoods or embeddings to be used as representations. Any of these encoding methodologies are typically amenable to a variety of different downstream models.

One-hot encodings are the simplest way to encode protein input data for an ML model, and they are used across many different domains. At the most basic level, one-hot encodings are categorized inputs represented as computer-interpretable vectors, and for proteins the most common way to implement this encoding method is to designate each of the twenty canonical amino acids as its own category. Thus, each amino acid is described with a vector of length twenty where the position of a single “1” among nineteen “0” values indicates the identity of the amino acid. These encodings capture no information about similarities between amino acid properties (e.g., polarity, charge, size, etc.) and are thus a common baseline representation for more complex encoding strategies.

Biophysical properties are another way that proteins can be encoded, and such approaches can work at either a residue or whole-protein level. Per-residue methods can use curated sets of biophysical parameters unique to each amino acid such as the Georgiev parameters [75]. These sets of features as encodings begin to offer information on the similarities between amino acids, as those with more similar properties will have more similar encodings — downstream models may be able to pick up on these similarities and better share information between training data points. Building upon this approach, more complicated and compute-intensive encodings can be calculated with physical modeling from relatively fast force field calculations to molecular dynamics (MD) simulations to quantum mechanics/molecular modeling (QM/MM). With such approaches there is a trade-off between speed of calculation and accuracy, and the proper balance remains an open question for the field.

The final type of encoding comes from the information held within evolutionary related sequences as well as with protein sequences as a whole. Creating MSAs of homologous sequences and quantifying sequence conservation or covariation can provide scores used to augment one-hot encodings.[49] Alternatively, unsupervised large language models can be trained on these sequences or on the millions of protein sequences is sequence databases. They can then be repurposed to generate fully continuous vector representations of proteins known as embeddings. Protein embeddings from unsupervised models capture information learned during pretraining and define the relationships between proteins within the context of learned sequence constraints: similar sequences will be found closer together in embedding space and so can, for instance, be inferred to have similar properties by a downstream supervised model. In this way, learned protein embeddings allow information contained in unlabeled sequences to be passed to a downstream supervised task, in principle reducing the amount of labeled data needed compared to less informative encoding strategies.

1.7.2 Machine learning models for directed evolution

Once protein variants have been assayed to obtain fitness labels and represented to obtain features, a supervised model can be trained on these sequence-fitness pairs. These models can range from extremely simple models such as linear regression to more complicated models such as graph-neural networks, and model selection can depend on the data in question as well as the representation being used to encode a given protein variant. Because of these intricacies, there is currently no model type that dominates for protein fitness prediction, and thus model selection is based on rapidly evolving heuristics and comparisons.

1.8 The need for protein engineering datasets

The common theme for choosing a representation or model is that we require ways to validate each choice as well as compare new approaches to existing ones. This has resulted in a number of efforts to curate datasets for protein fitness prediction.[58,76,77]

1.8.1 The state-of-the-field in existing protein fitness landscapes

Discussed previously, DMS has been the predominant method for large-scale fitness landscape generation, where each residue of a protein sequence is independently altered to every other amino acid. Because of their prevalence, they have also been the main testing ground for ML models. However, these types of landscapes are not well-suited to the task of building ML approaches that work well for DE, as there is no possibility to test iterative approaches for improving sequences without generating more variants and measuring their fitness values. Therefore, there has been a focus upon landscapes that enable the testing of these approaches. For example, methodology for informed training set selection can be incorporated, or iterative "sample, screen, predict" strategies can be tested when more than single mutants are included.[12] Most importantly, they represent a complete validation set where every (or nearly every) sequence within the landscape has a measured fitness value that can be compared to predictions.

In protein engineering, mutational effects, especially for distant residues, are often presumed to be additive or at least cumulative; for such systems, simple linear regression models on easy-to-generate one-hot encodings work well, so there is no need to develop new approaches. Indeed, in these cases mutations can be directly recombined without the need for any ML. However, in the case of non-additive, epistatic, interactions the effects of combined mutations are more difficult to predict, and simple recombination or linear models can fail. Thus, ML-based enzyme engineering strategies must handle epistasis if they are to be applied to multiple mutations, making epistatic fitness landscapes some of the most interesting protein fitness landscapes. Availability of experimental, epistatic landscapes essential for the development of new ML methods for protein engineering.

1.8.2 A standout protein fitness landscape

One epistatic landscape has thus dominated method development in the space of ML-assisted DE: Wu *et al.* sampled all possible amino acids at each of four positions simultaneously (160,000 possible variants).[78] The authors built this fitness landscape on the B1 domain of protein G, an immunoglobulin-binding protein, targeting four sites known to have epistatic interactions that were found through complete single- and double-site saturation mutagenesis of GB1.[79] With 93.4% of all possible variants measured, nearly every sequence prediction can be "tested" *in silico*, thus making it an important benchmarking landscape for developing ML for protein engineering.

There are two major issues with this landscape, however. First, the function of the GB1 domain is binding immunoglobulin G, which involves optimizing a protein-protein interface to improve binding. This is vastly different than enzymatic catalysis, which requires first binding a small-molecule (and potentially also another small-molecule cofactor) into an enzyme active site, catalyzing product formation, and then releasing product to the environment. This difference in mechanism makes it uncertain if success at predicting on the GB1 landscape will translate to enzymes. Furthermore, GB1 has very few known homologous sequences, making its multiple-sequence alignment (MSA) extremely shallow.

As discussed above, MSAs have been instrumental in recent advances in protein structure prediction,[74] but GB1's shallow MSA has limited the effectiveness of these approaches and sparked the creation of methods to use deep mutagenesis data to attempt structural prediction instead.[80,81] With recent efforts to incorporate ML into protein engineering focusing on applying unsupervised learning to the creation of informative protein representations, the lack of homologous sequences makes the GB1 landscape a non-ideal benchmark to method developers since the representations are much less useful. Therefore, the desire for a comprehensive landscape for validation must currently be weighed against the desire for a landscape built on a protein from a large and diverse family.

Both DMS landscapes and multi-site saturation landscapes represent important facets of proteins that have been leveraged by ML models, but neither represents proteins perfectly. Thus, current ML method development is typically done across a panel of landscapes to assess performance in multiple tasks. Ideally, if we are to learn general rules about protein function, all landscapes would have depth, completeness, and epistasis, and differ most based on the protein background, not how the landscape was constructed.

1.9 Conclusion and outlook

By moving expensive experimental screens in silico, ML greatly expands our ability to explore protein sequence space. While ML has so far been cast mainly as a supervised problem when applied to directed evolution, there has been significant expansion in unsupervised ML strategies as well. These unsupervised approaches can be used to limit or eliminate required experimental characterization of proteins, assist with navigation of combinatorial sequence space, and generate new protein sequence diversity, all of which can improve the efficiency of directed evolution campaigns. Yet, ML for directed evolution is still a relatively young field with much room for continued advancement. In particular, continued decreases in the cost and time of gene synthesis and sequencing as well as increases in computational power will make the laboratory application of ML methods more feasible and enable expansion of both sequence and sequence-function databases. As data availability grows, continued and improved collaboration between ML scientists and protein engineers will prove critical to developing experimentally tractable ML strategies that advance the field and drive more widespread adoption of the technology.

Chapter I Bibliography

1. Blamey, J. M., Fischer, F., Meyer, H.-P., Sarmiento, F. & Zinn, M. Chapter 14 – Enzymatic biocatalysis in chemical transformations: A promising and emerging field in green chemistry practice. in *Biotechnology of microbial enzymes* (ed. Brahmachari, G.) 347–403 (Academic Press, 2017). doi:10.1016/B978-0-12-803725-6.00014-5.
2. Bornscheuer, U. T. *et al.* Engineering the third wave of biocatalysis. *Nature* **485**, 185–194 (2012).
3. Global enzymes market in industrial applications. https://www.bccresearch.com/market-research/biotechnology/global-markets-for-enzymes-in-industrial-applications.html (2018).
4. Rosenthal, K. & Lütz, S. Recent developments and challenges of biocatalytic processes in the pharmaceutical industry. *Curr. Opin. Green Sustain. Chem.* **11**, 58–64 (2018).
5. Devine, P. N. *et al.* Extending the application of biocatalysis to meet the challenges of drug development. *Nat. Rev. Chem.* **2**, 409–421 (2018).

6. Maynard Smith, J. Natural selection and the concept of a protein space. *Nature* **225**, 563–564 (1970).

7. Romero, P. A. & Arnold, F. H. Exploring protein fitness landscapes by directed evolution. *Nat. Rev. Mol. Cell Biol.* **10**, 866–876 (2009).

8. Austin, H. P. *et al.* Characterization and engineering of a plastic-degrading aromatic polyesterase. *Proc. Natl. Acad. Sci. U.S.A.* **115**, E4350–E4357 (2018).

9. Waehler, R., Russell, S. J. & Curiel, D. T. Engineering targeted viral vectors for gene therapy. *Nat. Rev. Genet.* **8**, 573–587 (2007).

10. Starr, T. N. & Thornton, J. W. Epistasis in protein evolution. *Protein Sci.* **25**, 1204–1218 (2016).

11. Miton, C. M., Buda, K. & Tokuriki, N. Epistasis and intramolecular networks in protein evolution. *Curr. Opin. Struct. Biol.* **69**, 160–168 (2021).

12. Otwinowski, J., McCandlish, D. M. & Plotkin, J. B. Inferring the shape of global epistasis. *Proc. Natl. Acad. Sci. U.S.A.* **115**, E7550–E7558 (2018).

13. Bloom, J. D., Labthavikul, S. T., Otey, C. R., Arnold, F. H. & Levitt, M. Protein stability promotes evolvability. *Proc. Natl. Acad. Sci U.S.A.* **103**, 5869–5874 (2006).

14. Suzek, B. E., Huang, H., McGarvey, P., Mazumder, R. & Wu, C. H. UniRef: Comprehensive and non-redundant UniProt reference clusters. *Bioinform.* **23**, 1282–1288 (2007).

15. Suzek, B. E. *et al.* UniRef clusters: A comprehensive and scalable alternative for improving sequence similarity searches. *Bioinform.* **31**, 926–932 (2015).

16. Hyster, T. K. & Arnold, F. H. P450$_{BM3}$-axial mutations: A gateway to non-natural reactivity. *Isr. J. Chem.* **55**, 14–20 (2015).

17. Shin, J.-E. *et al.* Protein design and variant prediction using autoregressive generative models. *Nat. Commun.* **12**, 2403 (2021).

18. Liu, G. *et al.* Antibody complementarity determining region design using high-capacity machine learning. *Bioinformatics* **36**, 2126–2133 (2020).

19. Huang, P.-S., Boyken, S. E. & Baker, D. The coming of age of de novo protein design. *Nature* **537**, 320–327 (2016).

20. Gligorijević, V. *et al.* Function-guided protein design by deep manifold sampling. 2021.12.22.473759. *bioRxiv* (2021). doi:10.1101/2021.12.22.473759.

21. Yeh, A. H.-W. *et al.* De novo design of luciferases using deep learning. *Nature* **614**, 774–780 (2023).

22. Peisajovich, S. G. & Tawfik, D. S. Protein engineers turned evolutionists. *Nat. Methods* **4**, 991–994 (2007).

23. Cao, H., Wang, J., He, L., Qi, Y. & Zhang, J. Z. DeepDDG: Predicting the stability change of protein point mutations using neural networks. *J. Chem. Inf. Model.* **59**, 1508–1514 (2019).

24. Li, B., Yang, Y. T., Capra, J. A. & Gerstein, M. B. Predicting changes in protein thermodynamic stability upon point mutation with deep 3D convolutional neural networks. *PLOS Comp. Biol.* **16**, e1008291 (2020).

25. Vanhercke, T., Ampe, C., Tirry, L. & Denolf, P. Reducing mutational bias in random protein libraries. *Anal. Biochem.* **339**, 9–14 (2005).

26. Kille, S. *et al.* Reducing codon redundancy and screening effort of combinatorial protein libraries created by saturation mutagenesis. *ACS Synth. Biol.* **15**, 83–92 (2012).

27. Weinstein, E. N. *et al.* Optimal Design of Stochastic DNA Synthesis Protocols based on Generative Sequence Models. in *Proc. 25th Int. Conf. Art. Int. and Stat., PLMR.* **151**, 7450–7482 (2022).

28. Yang, J. *et al.* DeCOIL: Optimization of degenerate codon libraries for machine learning-assisted protein engineering. *ACS Synth. Biol.* **12**, 2444–2454 (2023).

29. Voigt, C. A., Martinez, C., Wang, Z.-G., Mayo, S. L. & Arnold, F. H. Protein building blocks preserved by recombination. *Nat. Struct. Mol. Biol.* **9**, 553–558 (2002).

30. Romero, P. A., Krause, A. & Arnold, F. H. Navigating the protein fitness landscape with Gaussian processes. *Proc. Natl. Acad. Sci. U.S.A.* **110**, E193–E201 (2013).

31. Bedbrook, C. N. *et al.* Machine learning-guided channelrhodopsin engineering enables minimally invasive optogenetics. *Nat. Methods* **16**, 1176–1184 (2019).

32. Packer, M. S. & Liu, D. R. Methods for the directed evolution of proteins. *Nat. Rev. Genet.* **16**, 379–394 (2015).

33. Kosuri, S. & Church, G. M. Large-scale de novo DNA synthesis: Technologies and applications. *Nat. Methods* **11**, 499–507 (2014).

34. Bedbrook, C. N., Yang, K. K., Rice, A. J., Gradinaru, V. & Arnold, F. H. Machine learning to design integral membrane channelrhodopsins for efficient eukaryotic expression and plasma membrane localization. *PLOS Comp. Biol.* **13**, e1005786 (2017).

35. Greenhalgh, J. C., Fahlberg, S. A., Pfleger, B. F. & Romero, P. A. Machine learning-guided acyl-ACP reductase engineering for improved in vivo fatty alcohol production. *Nat. Commun.* **12**, 5825 (2021).

36. Fox, R. J. *et al.* Improving catalytic function by ProSAR-driven enzyme evolution. *Nat. Biotechnol.* **25**, 338–344 (2007).

37. Saito, Y. *et al.* Machine-learning-guided mutagenesis for directed evolution of fluorescent proteins. *ACS Synth. Biol.* **7**, 2014–2022 (2018).

38. Gonzalez Somermeyer, L. *et al.* Heterogeneity of the GFP fitness landscape and data-driven protein design. *eLife* **11**, e75842 (2022).

39. Bryant, D. H. *et al.* Deep diversification of an AAV capsid protein by machine learning. *Nat. Biotechnol.* **39**, 691–696 (2021).

40. Brookes, D. H., Aghazadeh, A. & Listgarten, J. On the sparsity of fitness functions and implications for learning. *Proc. Natl. Acad. Sci. U.S.A.* **119**, e2109649118 (2022).

41. Wittmann, B. J., Yue, Y. & Arnold, F. H. Informed training set design enables efficient machine learning-assisted directed protein evolution. *Cell Syst.* **12**, 1026-1045.e7 (2021).

42. Gray, V. E., Hause, R. J., Luebeck, J., Shendure, J. & Fowler, D. M. Quantitative missense variant effect prediction using large-scale mutagenesis data. *Cell Syst.* **6**, 116-124.e3 (2018).

43. Meier, J. *et al.* Language models enable zero-shot prediction of the effects of mutations on protein function. in *Adv. Neural Inf. Process.* **34**, 29287–29303 (2021).

44. Liao, J. *et al.* Engineering proteinase K using machine learning and synthetic genes. *BMC Biotechnol.* **7**, 16 (2007).

45. Musdal, Y., Govindarajan, S. & Mannervik, B. Exploring sequence-function space of a poplar glutathione transferase using designed information-rich gene variants. *PEDS* **30**, 543–549 (2017).

46. Hopf, T. A. *et al.* Mutation effects predicted from sequence co-variation. *Nat. Biotechnol.* **35**, 128–135 (2017).

47. Riesselman, A. J., Ingraham, J. B. & Marks, D. S. Deep generative models of genetic variation capture the effects of mutations. *Nat. Methods* **15**, 816–822 (2018).

48. Consortium, T. U. *et al.* UniProt: The universal protein knowledgebase in 2021. *Nucleic Acids Res.* **49**, D480–D489 (2021).

49. Hsu, C., Nisonoff, H., Fannjiang, C. & Listgarten, J. Learning protein fitness models from evolutionary and assay-labeled data. *Nat. Biotechnol.* **40**, 1114–1122 (2022).

50. Biswas, S., Khimulya, G., Alley, E. C., Esvelt, K. M. & Church, G. M. Low-N protein engineering with data-efficient deep learning. *Nat. Methods* **18**, 389–396 (2021).

51. Wu, Z., Kan, S. B. J., Lewis, R. D., Wittmann, B. J. & Arnold, F. H. Machine learning-assisted directed protein evolution with combinatorial libraries. *Proc. Natl. Acad. Sci.* **116**, 8852–8858 (2019).

52. Qiu, Y., Hu, J. & Wei, G.-W. Cluster learning-assisted directed evolution. *Nat. Comput. Sci.* **1**, 809–818 (2021).

53. Bonner, W. A., Hulett, H. R., Sweet, R. G. & Herzenberg, L. A. Fluorescence activated cell sorting. *Rev. Sci. Instrum.* **43**, 404–409 (2003).

54. Rubin, A. F. *et al.* A statistical framework for analyzing deep mutational scanning data. *Genome Biology* **18**, 150 (2017).

55. Trippe, B. L. *et al.* Randomized gates eliminate bias in sort-seq assays. *Protein Sci.* **31**, e4401 (2022).

56. Bloom, J. D. Software for the analysis and visualization of deep mutational scanning data. *BMC Bioinform.* **16**, 168 (2015).

57. Madani, A. *et al.* Large language models generate functional protein sequences across diverse families. *Nat. Biotechnol.* 1–8 (2023) doi:10.1038/s41587-022-01618-2.

58. Dallago, C. *et al.* FLIP: Benchmark tasks in fitness landscape inference for proteins. *bioRxiv* (2022). doi:10.1101/2021.11.09.467890.

59. Aghazadeh, A. *et al.* Epistatic Net allows the sparse spectral regularization of deep neural networks for inferring fitness functions. *Nat. Commun.* **12**, 5225 (2021).

60. Slatko, B. E., Gardner, A. F. & Ausubel, F. M. Overview of next-generation sequencing technologies. *Curr. Protoc. Mol. Biol.* **122**, e59 (2018).

61. Fowler, D. M. & Fields, S. Deep mutational scanning: a new style of protein science. *Nat. Methods* **11**, 801–807 (2014).

62. Currin, A. *et al.* Highly multiplexed, fast and accurate nanopore sequencing for verification of synthetic DNA constructs and sequence libraries. *Synth. Biol.* **4**, ysz025 (2019).

63. Appel, M. J. *et al.* uPIC–M: Efficient and scalable preparation of clonal single mutant libraries for high-throughput protein biochemistry. *ACS Omega* **6**, 30542–30554 (2021).

64. Wittmann, B. J., Johnston, K. E., Almhjell, P. J. & Arnold, F. H. evSeq: Cost-effective amplicon sequencing of every variant in a protein library. *ACS Synth. Biol.* **11**, 1313–1324 (2022).

65. Berman, H. M. *et al.* The Protein Data Bank. *Nucleic Acids Res.* **28**, 235–242 (2000).

66. Burley, S. K. *et al.* RCSB Protein Data Bank: Powerful new tools for exploring 3D structures of biological macromolecules for basic and applied research and education in fundamental biology, biomedicine, biotechnology, bioengineering and energy sciences. *Nucleic Acids Res.* **49**, D437–D451 (2021).

67. Mistry, J. *et al.* Pfam: The protein families database in 2021. *Nucleic Acids Res.* **49**, D412–D419 (2021).

68. Ashburner, M. *et al.* Gene Ontology: Tool for the unification of biology. *Nat. Genet.* **25**, 25–29 (2000).

69. The Gene Ontology Consortium. The Gene Ontology resource: Enriching a GOld mine. *Nucleic Acids Res.* **49**, D325–D334 (2021).

70. Rives, A. *et al.* Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci. U.S.A.* **118**, e2016239118 (2021).

71. Mitchell, A. L. *et al.* MGnify: The microbiome analysis resource in 2020. *Nucleic Acids Res.* **48**, D570–D578 (2020).

72. Bepler, T. & Berger, B. Learning protein sequence embeddings using information from structure. *arXiv* (2019). doi:10.48550/arXiv.1902.08661.

73. Shroff, R. *et al.* Discovery of novel gain-of-function mutations guided by structure-based deep learning. *ACS Synth. Biol.* **9**, 2927–2935 (2020).

74. Jumper, J. *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021).

75. Georgiev, A. G. Interpretable numerical descriptors of amino acid space. *J. Comp. Biol.* **16**, 703–723 (2009).

76. Rao, R. *et al.* Evaluating protein transfer learning with TAPE. *bioRxiv* (2019) doi:10.1101/676825.

77. Notin, P. *et al.* Tranception: Protein fitness prediction with autoregressive transformers and inference-time retrieval. in *Proc. 39th ICML, PLMR.* 16990–17017 (2022).

78. Wu, N. C., Dai, L., Olson, C. A., Lloyd-Smith, J. O. & Sun, R. Adaptation in protein fitness landscapes is facilitated by indirect paths. *eLife* **5**, e16965 (2016).

79. Olson, C. A., Wu, N. C. & Sun, R. A comprehensive biophysical description of pairwise epistasis throughout an entire protein domain. *Curr. Biol.* **24**, 2643–2651 (2014).

80. Rollins, N. J. *et al.* Inferring protein 3D structure from deep mutation scans. *Nat. Genet.* **51**, 1170–1176 (2019).

81. Schmiedel, J. M. & Lehner, B. Determining protein structures using deep mutagenesis. *Nat. Genet.* **51**, 1177–1186 (2019).

Chapter II

EVSEQ: COST-EFFECTIVE AMPLICON SEQUENCING OF EVERY VARIANT IN A PROTEIN LIBRARY

Material from this chapter appears in: "Wittmann, B. J., **Johnston, K. E.**, Almhjell, P. J. & Arnold, F.H. evSeq: Cost-effective amplicon sequencing of every variant in a protein library. *ACS Synth. Biol.* **11**, 1313–1324 (2022). doi: 10.1021/acssynbio.1c00592."

B.J.W. conceived the project and performed initial design and execution of research and software development. B.J.W., K.E.J, and P.J.A. optimized the experimental workflow and software. K.E.J. and P.J.A. wrote software for data visualization and installation. B.J.W., K.E.J., and P.J.A. wrote the manuscript and prepared figures.

ABSTRACT

Widespread availability of protein sequence-fitness data would revolutionize both our biochemical understanding of proteins and our ability to engineer them. Unfortunately, even though thousands of protein variants are generated and evaluated for fitness during a typical protein engineering campaign, most are never sequenced, leaving a wealth of potential sequence-fitness information untapped. Primarily, this is because sequencing is unnecessary for many protein engineering strategies; the added cost and effort of sequencing is thus unjustified. It also results from the fact that, even though many lower cost sequencing strategies have been developed, they often require at least some sequencing or computational resources, both of which can be barriers to access. Here, we present every variant sequencing (evSeq), a method and collection of tools/standardized components for sequencing a variable region within every variant gene produced during a protein engineering campaign at a cost of cents per variant. evSeq was designed to democratize low-cost sequencing for protein engineers and, indeed, anyone interested in engineering biological systems. Execution of its wet-lab component is simple, requires no sequencing experience to perform, relies only on resources and services typically available to biology labs, and slots neatly into existing protein engineering workflows. Analysis of evSeq data is likewise made simple by its accompanying software (found at github.com/fhalab/evSeq, documentation at fhalab.github.io/evSeq), which can be run on a personal laptop and was designed to be accessible to users with no computational experience. Low-cost and easy to use, evSeq makes collection of extensive protein variant sequence-fitness data practical.

2.1 Introduction

Engineered proteins are valuable tools across the biological and chemical sciences and have revolutionized industries ranging from food to fuels, pharmaceuticals, and textiles by providing green and efficient protein solutions to challenging chemical problems.[1] Over the course of a protein engineering campaign, hundreds to thousands or more protein variants will be constructed and have their fitnesses (level of, e.g., thermostability, catalytic activity, substrate binding, etc.) evaluated. Notably, sequence information is typically not gathered alongside the functional information, even though it could provide useful biochemical insight.[2–4] This is largely because many engineering strategies can be applied without sequencing. For example, during a typical directed evolution (DE) experiment, often only the best-performing variant or variants are sequenced in each round of mutagenesis and screening; sequencing every variant is viewed as an unnecessary expense. Given the massive amount of functional data gathered during a typical DE campaign, however, if sequencing *were* performed for the variants generated during these experiments, the resultant large datasets of sequence-fitness information could be revolutionary for biological, biochemical, and biocatalytic research. This is especially true for data-driven protein engineering strategies such as machine learning (ML), the development of which has benefitted tremendously from large sequence-fitness datasets made available by strategies like deep mutational scanning (DMS) and in databases like ProtaBank.[5–16]

Unfortunately, the standard sequencing strategy employed during DE—Sanger sequencing—is too expensive for sequencing all variants tested during a round of evolution.[17] Sanger sequencing is ubiquitous due to ease of sample preparation and ready availability of sequencing providers. However, the cost of Sanger sequencing scales linearly with the number of samples (Appendix A, **Figure A-1**). Thus, while the cost of sequencing just the top variants in a round of DE is minor, sequencing the hundreds or thousands of variants generated over the full engineering endeavor is not. Ideally, any new approach to sequencing during a protein engineering campaign would be comparable in cost, effort, and accessibility to that of sequencing just the top variants by Sanger sequencing. Here we present a collection of standardized and accessible protocols, components, and software that accomplishes this goal. This collection, which we call every variant sequencing (evSeq), democratizes barcode sequencing strategies and expands on services made available by multiplexed next-generation sequencing (NGS) providers to allow amplicon sequencing of a region of interest within every variant produced during a round of DE at a cost of cents per variant.[18,19] Sample preparation for evSeq is simple, and the method requires no experience with NGS to perform, relies only on resources and services typically available to biology labs, and slots neatly into existing protein engineering experimental workflows. The accompanying software for analysis of evSeq data (found at github.com/fhalab/evSeq, documentation at fhalab.github.io/evSeq) was designed to be accessible to users with no computational experience and can be run on a personal laptop.

In this paper, we detail the underlying strategies, protocol, and potential applications of evSeq. We begin by describing the strategies employed by evSeq to extend multiplexed NGS for sequencing protein variant libraries in a way that reduces both cost and effort. We then describe the wet-lab protocol of evSeq sample preparation, focusing on how it can be completed without disrupting an existing protein engineering workflow. Next, we discuss the features of the evSeq software before finally presenting two case studies that highlight potential applications of evSeq. In particular, we highlight how (1) the sequence-fitness data from evSeq can provide valuable information about the quality of variant libraries and the functional screen as well as how mutations modulate protein activity, and how (2) the data generated from evSeq can be used to implement ML for protein engineering. We designed evSeq for use as a routine procedure in many protein/enzyme assays (especially DE and protein engineering experiments leveraging mutagenesis strategies that target specific sites or a segment of the sequence). This tool brings cost-effective, easy-to-use sequencing to all protein engineers, regardless of experience with NGS and access to sequencing and computational resources. We believe that widespread adoption of evSeq—and the resultant datasets generated—will be invaluable for future ML-guided protein engineering and will help us better understand protein sequence-fitness relationships.

2.2 Results

2.2.1 evSeq uses inline barcoding to expand on commercially available multiplexed next-generation sequencing.

Unlike Sanger sequencing, which outputs a single chromatogram that represents the population of DNA in a sequenced sample, NGS outputs millions of individual DNA reads that represent a random draw from the population of DNA in the sequenced sample.[18] Confidence in NGS sequencing results is largely determined by the sequencing "coverage," which for the purposes of this paper is defined as the number of returned reads that map to a specific nucleotide on a reference sequence. Higher coverage enables more confident identification of mutations relative to a reference sequence as the increased redundancy allows distinguishing between true sequence mutations and errors that arise during library preparation, clustering, or sequencing.

A single NGS run is roughly three orders of magnitude more expensive than a Sanger sequencing run, but because the run outputs millions of reads, this cost can be spread over multiple samples using a technique known as "multiplexed NGS" (Appendix A, **Figure A-1**). In multiplexed NGS, each submitted sample is tagged with a "molecular barcode"—a unique piece of DNA that encodes the sample's original identity—before all samples are sequenced together in the same NGS run.[19–25] Post sequencing, the barcodes are used to assign individual reads to individual samples. For instance, barcodes can be used to distinguish reads coming from samples belonging to specific plates and wells.[26] Importantly, multiplexed NGS can be outsourced just like Sanger sequencing (making it accessible to all laboratories regardless of sequencing experience), and sequencing providers typically charge tens of dollars per sample in a multiplexed sequencing run, yielding on the order of 10^4–10^5 individual sequences per sample (assuming the run is performed on an Illumina MiSeq instrument).

The level of coverage granted by a set number of reads depends on the length of the DNA sample being sequenced, the length of the NGS read used to sequence it, and whether those reads are paired-end. NGS reads are short (300 bp or less on Illumina systems), and so reads must be spread across a longer sample to sequence it in full. The expected coverage (average coverage per nucleotide) obtained for a DNA sample thus depends both on its length and the read length used for sequencing. For example, with the ~10^5 reads returned by a commercial MiSeq multiplexed sequencing run, a 3 Mb genome could be sequenced with 150 bp paired-end reads to an expected coverage of ~10x, whereas a 20 kb plasmid could be sequenced to an expected coverage of ~1500x.

Because shorter samples can be sequenced at higher coverage for a given number of reads, it can be advantageous to sequence only the region of interest of a sample. This is exemplified by amplicon sequencing, a strategy in which a researcher sequences a PCR product (an amplicon) that targets a specific region of interest in the DNA.[27] For instance, continuing the example from above, with ~10^5 total 150 bp paired-end reads, a 300 bp PCR product could be sequenced to an expected coverage of ~100,000x.

Many mutagenesis methods employed in protein engineering (e.g., site-saturation[28] and tile-based mutagenesis[29] strategies) target mutations to a specific position or region in the sequence of a protein, and thus the variants produced can be sequenced with amplicon sequencing to high coverage.[20] Notably, however, even though increasing coverage yields more confident results, it comes with diminishing returns, and it is generally held that coverage in the tens is more than sufficient for effective reference-based identification of mutations (Appendix A, **Figure A-1**).[30] Indeed, clinical sequencing of human genomes targets 30x coverage or greater to minimize false base calls. Given this reference, it is clear that the ~100,000x coverage that would be returned from a multiplexed sequencing run for a 300 bp amplicon is far higher than necessary for effective identification of mutations—2,000 amplicons could be sequenced in the same run and still yield clinical-grade coverage.

evSeq achieves cost-effectiveness by relying on the facts that (1) at tens of dollars per sample, the cost of sending a single sample to an outsourced multiplexed NGS run is comparable to the total cost of Sanger sequencing the top variants in a round of DE, (2) amplicon sequencing can be used to identify mutations in protein variants from many protein engineering library types, and (3) enough reads are returned for a single sample in a commercial multiplexed

NGS run to sequence hundreds of amplicons. Specifically, the evSeq protocol (**Figure 2-1** and Appendix A, **Section A.2.3**, *evSeq Library Preparation/Data Analysis Protocol*) works by focusing all reads of a single multiplexed NGS sample to specific regions on hundreds of protein variants, achieving sequencing depths of 10^1–10^3 at the approximate cost and level of accessibility of using Sanger sequencing of just the top variants in a round of DE (Appendix A, **Figure A-1**).

The evSeq library preparation protocol begins with PCR amplification of the region of interest in each variant (i.e., the position/region where mutations were made) and appending inline DNA barcodes to the resultant amplicons that encode their original plate-well position (**Figure 2-1A**).[26,31,32] This is a one-pot, two-step, plate-based PCR procedure that uses two sets of primer pairs. Each primer in the first set of primers ("inner" primers) consists of a user-specified 3' "seed" region that binds to the regions flanking the region of interest as well as a 5' predefined universal adapter (Appendix A, **Section A.1.1**, *Inner Primer Design*). Each primer in the second set of primers ("outer" primers) consists of (1) a 3' region that matches the adapter of the inner primers, (2) a central 7-nucleotide barcode where each barcode pair between forward and reverse outer primers is unique to a plate-well position, and (3) a 5' sequence matching the Illumina Nextera transposase adapters (Appendix A, **Section A.1.2**, *Outer Primer Design,* Appendix A, **Section A.1.3**, *Barcode Design*, **Tables A-1** and **A-2**). We designed 96 unique forward and 96 unique reverse outer primers for evSeq which, because both forward and reverse outer primers contain a barcode, can be combined to encode up to 96^2 = 9,216 possible plate-well positions (Appendix A, **Section A.2.2**, *Preparation of evSeq Barcode Primer Mixes,* and Appendix A, **Tables A-3–10**. Note that we also provide a pre-filled IDT order form for the outer primers on the GitHub associated with this work—see Appendix A, **Section A.2.1**, *Ordering Barcode Primers from IDT* for details. While we recommend using these pre-tested barcodes, users can also design their own to, e.g., further expand the number of available combinations.). Importantly, this set of outer primers can be used to sequence any target region from any gene with evSeq, and so only needs to be ordered once, constituting a one-time initial setup cost in the range of a few hundred dollars (the exact cost will vary based on oligo provider and any institutional agreements set up with said provider). Once outer primers are ordered, only a new inner primer pair is needed for each new region of interest to be targeted by evSeq.

Once all barcoded amplicons have been produced, they are pooled and sent to a sequencing provider, who will then use the transposase adapters installed with the outer primers as a handle to perform a third and final PCR to barcode the *pool* of amplicons *once again* with a pair of sample-specific Illumina indices (**Figure 2-1B**). At this point each amplicon in the pool has one pair of sample-specific Illumina barcodes and one pair of plate-well-specific inline evSeq barcodes. This complete evSeq library is sequenced as a single sample in a multiplexed NGS run along with samples from other users (whether or not they are also evSeq samples). Post sequencing, the sequencing provider uses the sample-specific barcodes to identify those sequences belonging to the evSeq pool and returns them to the user (i.e., the provider "demultiplexes" the run, separating evSeq sequences from those of other users). The user then uses the evSeq software to analyze the returned sequences, assigning them to corresponding plate-well positions using the evSeq barcodes and identifying the mutations in the variants relative to a reference (**Figure 2-1B** and **2-1C**).

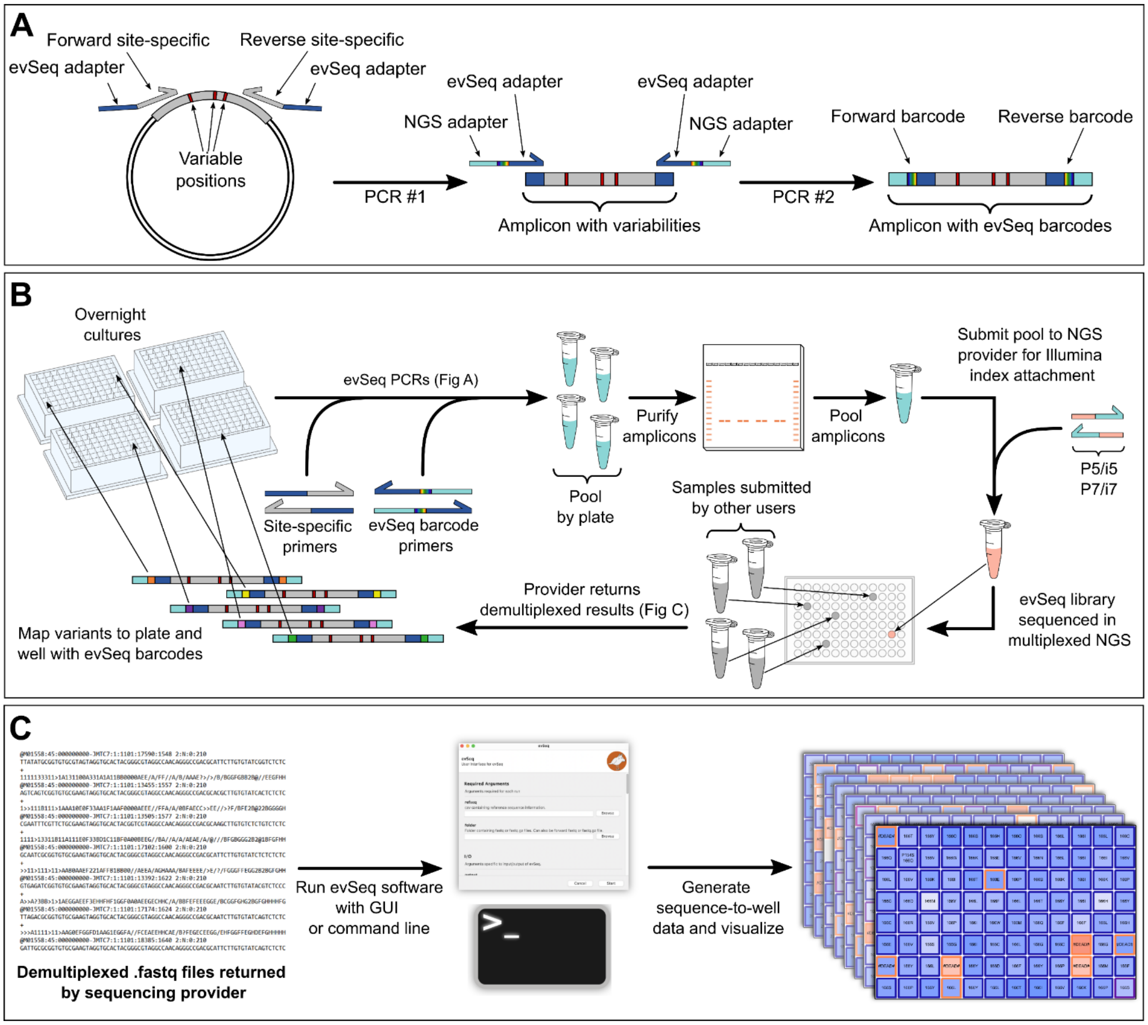

A
Forward site-specific
evSeq adapter
Reverse site-specific
evSeq adapter
Variable positions
evSeq adapter
evSeq adapter
NGS adapter
NGS adapter
PCR #1
Amplicon with variabilities
PCR #2
Forward barcode
Reverse barcode
Amplicon with evSeq barcodes
B
Overnight cultures
evSeq PCRs (Fig A)
Site-specific primers
evSeq barcode primers
Pool by plate
Purify amplicons
Pool amplicons
Submit pool to NGS provider for Illumina index attachment
P5/i5
P7/i7
Samples submitted by other users
evSeq library sequenced in multiplexed NGS
Provider returns demultiplexed results (Fig C)
Map variants to plate and well with evSeq barcodes
C
Demultiplexed .fastq files returned by sequencing provider
Run evSeq software with GUI or command line
Generate sequence-to-well data and visualize

Figure 2-1. Overview of evSeq library preparation and processing. A In the first stage of the PCR, a region of interest is amplified with primers that include a 3’ site-specific region (gray) with 5’ adapter sequences (dark blue). The second PCR stage adds molecular barcodes (rainbow) with primers that bind to the adapter regions and add adapters for downstream NGS processing (light blue). **B** To avoid costly DNA isolation steps, evSeq uses liquid cultures of cells harboring mutated DNA (e.g., an “overnight culture” of *E. coli*) as template during the one-pot two-step barcoding PCR described in **A**. Each plate is pooled individually and gel purified. Purified pools are then adjusted for concentration differences and pooled together before being sent to a sequencing provider, who then appends another set of barcodes as well as sequence elements necessary for Illumina NGS sequencing. This sample is now pooled with those of other users and a multiplexed sequencing run is performed. After sequencing, the sequencing provider uses the barcodes that they attached to separate (“demultiplex”) the evSeq reads from reads of other users; the provider returns evSeq reads in .fastq files. **c.** The .fastq files returned by the NGS provider are inputs to the evSeq software, which uses the evSeq forward/reverse barcode pair to map each read to a specific plate and well based on known barcode combinations. The software also processes the mapped reads (see Appendix A and evSeq documentation for more details) to, among other things, assign variant identities to each well and return interactive HTML visualizations.

2.2.2 evSeq library preparation fits into existing protein engineering and sequencing workflows and was designed to be resource efficient.

A typical procedure for evaluating protein variants involves (1) arraying colonies of an organism (e.g., *Escherichia coli*) that harbor a plasmid encoding a protein variant into the wells of a (usually 96-well) microplate, (2) growing the resulting cultures to stationary phase (colloquially, an "overnight culture"), (3) using the overnight culture to inoculate a fresh culture that will be used to express the protein variants, and (4) evaluating the fitnesses of expressed protein variants. The expression stage (step 3) typically involves downtime where the experimentalist must wait until the culture reaches sufficient density before inducing protein expression and then again as expression takes place. evSeq library preparation can be performed easily in either of these time windows. The evSeq library preparation protocol begins with the barcoding PCR described at the end of the previous section; this one-pot, two-step, plate-based PCR was designed to be compatible with outsourced sequencing workflows, minimize preparation time, and minimize laboratory resource usage (Appendix A, **Section A.2.3**, *evSeq Library Preparation/Data Analysis Protocol*). For instance, use of inline barcodes is a known, effective strategy for expanding the number of samples that can be multiplexed without having to modify the Illumina indices used during multiplexed sequencing.[31,32] Because evSeq library preparation uses inline barcodes, it grants the outsourced sequencing provider maximal flexibility in choice of Illumina indices. In other words, evSeq library preparation is decoupled from preparation of the Illumina library that will eventually be sequenced, allowing the evSeq library to be run just as any other sample would be that is submitted to a sequencing provider.

As mentioned in the previous section, use of a two-step PCR reduces the number of primers that must be ordered per new sequencing region of interest. Because evSeq relies on 96 unique forward barcodes and 96 unique reverse barcodes, a single-primer PCR would require ordering 192 new barcoding primers for each new target region evaluated in each library. In a two-primer protocol, however, the inclusion of a universal adapter on the inner primers allows the same 192 outer primers to be used regardless of target position in the variant—only two unique primers (forward and reverse inner) must be purchased for each new target region, and only if existing inner primers from previously targeted regions are not already compatible. Additionally, the evSeq PCR directly uses liquid from the overnight culture as a source of template DNA (**Figure 2-1B** and Appendix A, **Section A.2.3**, *evSeq Library Preparation/Data Analysis Protocol*); the template DNA is released from lysed cells during the initial heating step of the PCR, avoiding a costly and time-intensive DNA isolation/purification step and allowing researchers to use materials already prepared as part of the protein expression workflow.[32]

The remaining steps of evSeq library preparation were, like the PCR stage, also designed to be resource and time efficient. After completion of the PCR, the resulting barcoded amplicons are pooled by plate and purified via gel extraction. Pooling prior to purification goes against standard practice for multiplexed NGS library preparation, which is to purify samples individually, quantify their DNA concentration, then combine them in equimolar quantities to ensure more equal read distribution across samples after sequencing.[33] However, because individual plates in protein engineering libraries tend to contain variants from the same region of the same protein scaffold (e.g., as would be typical for variants from a comprehensive site-saturation library), it is assumed that the variation in PCR reaction yield will be minor within plates and that, as a result, the same plate can be pooled prior to quantification with only minor effects on read distribution. Using this "pooling first" strategy, only as many purifications as there are *plates* must be performed as opposed to as many as there are *variants*, thus enabling faster processing of evSeq amplicons while reducing resource usage. As will be shown in later sections, the distribution of reads returned using pooling first is perfectly acceptable for confidently identifying variant sequences.

Once all pooled plates have been purified, the concentrations of the individual purified pools are measured. The pools are then normalized by molarity and combined into a final evSeq library, which is in turn submitted as a single sample to a sequencing provider. As described in the previous section, the provider will perform a final PCR on the evSeq library to add sample-specific barcodes before sequencing it as a single sample in a multiplexed sequencing run. Outsourcing the sequencing stage has two main benefits: First, it allows evSeq to be performed by research groups with no prior sequencing experience and no direct access to sequencing equipment—groups need only be familiar with PCR, a ubiquitous technology in protein engineering laboratories. Second, to be cost effective, multiplexed sequencing should be run with tens of samples at least (Appendix A, **Figure A-1**). By outsourcing the sequencing stage, groups that do not frequently produce evSeq libraries need not wait until enough libraries have accumulated to run sequencing—a single outsourced submission, for instance, can be run along with those of other research groups with a variety of different sequencing needs.

The final stage of the evSeq workflow is data analysis using the evSeq software (github.com/fhalab/evSeq) (**Figure 2-1C**). Extensive documentation of the software and its capabilities is available as a website (fhalab.github.io/evSeq). The software was designed to be accessible to users with varying degrees of computational experience and can be run through either a graphical user interface (GUI), a command line application, or in a Python environment (e.g., a Jupyter notebook). Outputs from the software range from high-level overviews of data (e.g., an interactive "Platemap" graphic that displays sequencing coverage and identified mutations in each well of each plate; see **Figure 2-1C** for an example) to low-level details about the population of reads assigned to each well (e.g., in a well identified as polyclonal, the percentage of reads mapping to each of the identified variants). Functional data can also be easily associated with identified variants using the evSeq software outputs to produce sequence-fitness datasets, and we provide Jupyter notebooks and web pages that walk users through the process.

2.2.3 evSeq facilitates library construction, validation, and sequence-fitness pairing

To highlight the utility of evSeq for engineering and biochemical experiments, we first examined how it could be used to construct high-confidence and informative sequence-fitness data. Specifically, we constructed and screened eight single-site-saturation libraries of the enzyme Tm9D8*—an engineered β-subunit of tryptophan synthase from *Thermotoga maritima* (*Tm*TrpB)—for tryptophan-forming activity at 30 °C (**Figure 2-2**).[34] In two of the screened libraries, we targeted two positions distant from the active site (A118 and S292) that have been seen to play a role in allosteric regulation of *Tm*TrpB enzymes; in the other six libraries, we targeted active-site residues known to modulate the activity of TrpB (E105, L162, I166, F184, S228, and Y301) (**Figure 2-2A**).[35–37] As we show below, this type of sequence-fitness data can be used to assess the quality of a protein engineering library, identify improved variants during a round of directed evolution, and give insight into the significance of a given residue in catalysis.

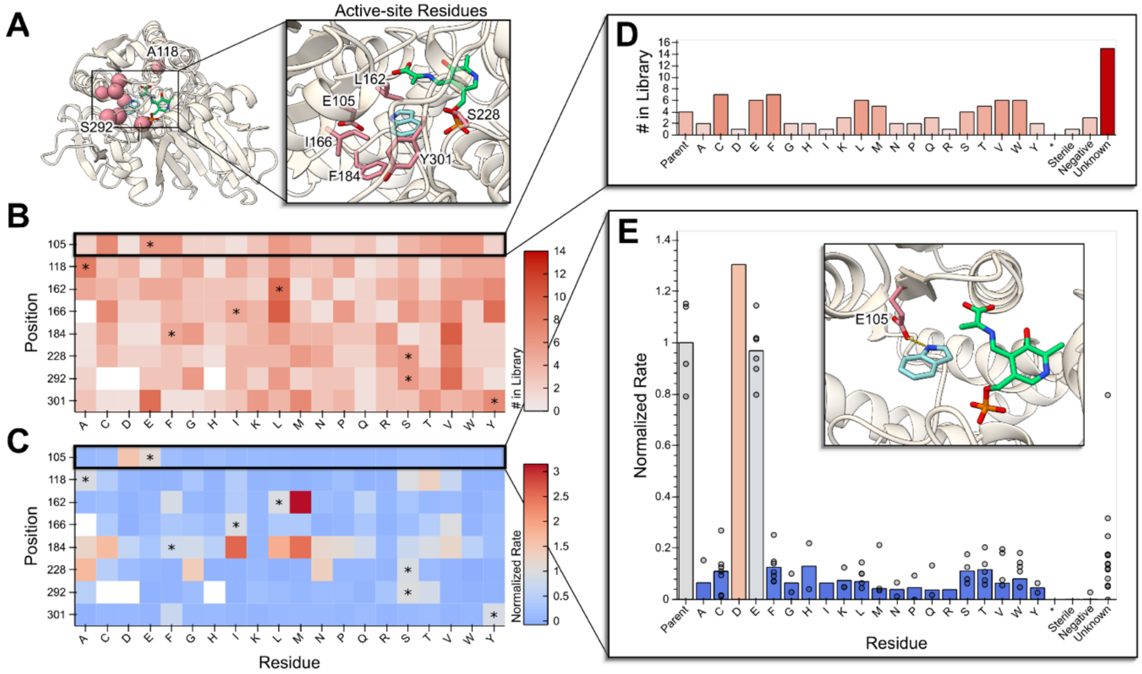


Figure 2-2. evSeq enables low-cost investigation of library quality and sequence-fitness pairing in site-saturation mutagenesis libraries. A Eight residues (red) known to modulate the activity of Tm9D8* were independently targeted with site-saturation mutagenesis: A118 and S292 (distal residues), E105, L162, I166, F184, S228, and Y301 (active-site residues). An active form of the pyridoxal 5'-phosphate cofactor is represented in green, and the substrate indole is shown in light blue. **B** Library quality can be investigated by plotting a heatmap of the number of times each variant/mutant was identified at each targeted position ("# in Library") from processed evSeq data. Parent amino acids are each marked with an asterisk. **C** Likewise, the effect of mutations and mutational "hotspots" can be identified by plotting a heatmap of the average activity measurements for each variant/mutation in each library, normalized to the average parent activity for that library ("Normalized Rate"), when fitness data is combined with evSeq data. **D** An example plot made possible by evSeq visualization functions shows the number of times each amino acid was found in a single TrpB library (position 105), also accounting for known controls and unidentified wells. **E** Another example output of the evSeq software shows activity for a single library (position 105), showing biological replicates. The inset displays the role of the mutated residue in this library, which is to coordinate the nitrogen of the indole substrate. Note that the circles in this plot correspond to individual measurements while the bar plot represents the mean of these measurements. If no circles are present for a bar (e.g., E105D), then this is because only a single instance of this mutation was observed. Circles are not shown in this case to allow distinguishing between a single replicate and a tight distribution of multiple replicates.

Many factors can introduce bias into a site-saturation mutagenesis experiment, such as annealing bias for the native nucleotides during the PCR for library construction or contamination with the template plasmid during transformation. Without sequencing all of the variants, it is impossible to know that the library is representative of the experimental design. Since evSeq reports exactly which variants are contained in a library, researchers can leverage this to implement important quality control practices as part of the standard protein screening workflow. For instance, of all 153 possible unique variants in our eight single-site-saturation libraries, we observed 149 of them (**Figures 2-2B** and **2-2C**); only I166A, S292C, S292D, and S292H could not be assigned with confidence, where we define >80% abundance in a well with >10 reads as our confidence threshold. Of the variants identified, many were found in replicate (**Figure 2-2D**) due to oversampling during colony picking, which ensures that all protein variants have a chance to be found and screened (All libraries were constructed with the 22-codon trick[38] and 88 individual colonies were screened for each library, so we expected a 98% probability of seeing all variants assuming perfect construction of libraries). Conveniently, this oversampling also allows us to evaluate the noise in our functional screen (**Figure 2-2E**) which further improves the confidence in the quality of data gathered.

Given just the fitness data gathered in this experiment, a protein engineer would identify 50 wells that are at least 1.2-fold improved over the parent enzyme Tm9D8*. However, with the sequence-fitness pairs constructed via evSeq, we know that these 50 *wells* correspond to only 16 unique *variants*. Depending on how conservative the engineer was as to what should be sequenced, a decision to sequence hits with Sanger sequencing could result in anywhere from 12 (2-fold improvement) to 50 (1.2-fold improvement) wells sent off for sequencing for a total cost of $36 to $150 (using an estimate of $3 per sequence). It would cost ~$2000 to sequence all eight plates via Sanger. Using evSeq, however, we obtained the sequences of *all* 625 wells of variants for only $100, corresponding to $0.13 per non-control well. In other words, using evSeq, we can produce far more sequence-fitness information than sequencing just the top hits using Sanger all for a similar cost. Importantly, although the evSeq defaults currently allow only eight plates to be sequenced at once, the number of variants included in this experiment could likely have been increased as the median number of reads per well was 86 (mean: 98), which is above what is needed for reliable sequencing. Assuming that doubling the number of plates would halve the number of reads seen for each well, doubling the number of plates sequenced would cause only 14 non-control well sequences to drop below the confidence threshold.

The per-variant cost of evSeq may be reduced even further using different services and sequencing platforms. For instance, in both this section and the next, the reported number of reads and ~$100 total cost are from outsourced MiSeq runs, which returned hundreds of thousands of total reads per evSeq library. We report these numbers because outsourced multiplexed MiSeq is a standard service available to all research groups. As an alternative to outsourcing, however, our institution provides multiplexed sequencing (via the Caltech Millard and Muriel Jacobs Genetics and Genomics Laboratory) on an Illumina NextSeq platform, returning an average of ~10x more reads than the outsourced MiSeq run for a total cost of ~$10. At 10x more reads and 10x less the total cost, the per-variant evSeq cost could decrease 100-fold to <$0.01. Indeed, we were able to re-sequence the TrpB libraries at a per-variant cost of ~\$0.01 with ~2.2 million total reads returned for an average of thousands of reads per variant, far higher than what is needed for reliable variant calling. It must be noted, however, that analysis of the millions of evSeq reads was no longer practical on a personal laptop, requiring a desktop workstation instead. Computational power beyond a laptop will be needed when processing more than hundreds of thousands of reads with the existing evSeq software.

Of final note, aside from providing valuable information for protein engineering experiments, evSeq can also facilitate investigation into the biochemical relevance of specific positions/mutations. Specifically, because all possible variants in a site-saturation library can be identified by evSeq, the sequence-fitness information generated can be used to explore the effects of mutations more fully than, for instance, an alanine scanning experiment.[39] Using an example from the TrpB data gathered here, an alanine scanning experiment would tell a biochemist that the mutation to the conserved catalytic residue E105A inactivates the enzyme, with no information about the effects of other amino acid changes at this position. Using site- saturation with evSeq, we instead find that all mutations to E105 except for E105D inactivate the enzyme. The fact that glutamate and aspartate are the only amino acids containing a carboxylic acid suggests that this functional group is critical for activity (**Figure 2-2E**, with inset).

2.2.4 evSeq facilitates library construction, validation, and sequence-fitness pairing

We next wanted to demonstrate the utility of evSeq for advancing and applying machine learning-assisted protein engineering (MLPE). In MLPE, models are trained to learn a function that relates protein sequence to protein fitness (i.e., they learn *f*(sequence) =

fitness).[5,6,9–11] These models are then used for rapid, low-cost *in silico* prediction of protein fitness, avoiding or greatly reducing the need for often-costly laboratory screening of variants (**Figure 2-3**).

Sequence-fitness data is critical for effective MLPE. Indeed, even though strategies exist that *can* predict protein fitness from sequence alone (e.g., those that use evolutionary data to predict protein fitness), their effectiveness is improved with the inclusion of sequence-fitness information.[7,14,15,40] As a result, the most effective MLPE workflows require that both sequence *and* fitness data be collected, unlike a DE workflow, which requires only fitness data.

The need to collect sequence data in addition to fitness data is an often-overlooked additional cost of MLPE strategies compared to standard DE. For instance, we recently developed an ML strategy known as machine learning-assisted directed evolution (MLDE) for efficient navigation of epistatic combinatorial protein variant libraries.[41,42] Previously, we used MLDE to evolve *Rhodothermus marinus* nitric oxide dioxygenase (*Rma*NOD) for greater enantioselectivity in a carbon–silicon bond-forming reaction.[41] Over the course of the engineering campaign, we collected six 96-well plates of sequence-fitness data for training ML models. In total, sequencing the variants in these plates by Sanger sequencing cost ~$1700. High additional sequencing costs like these can make MLPE methods far less attractive, even if they are more effective than traditional DE at finding high-fitness protein variants.[42] However, given that evSeq enables sequencing all variants for a cost similar to standard DE methods, it enables use of MLPE without added cost. In essence, evSeq eliminates the sequencing burden of MLPE.

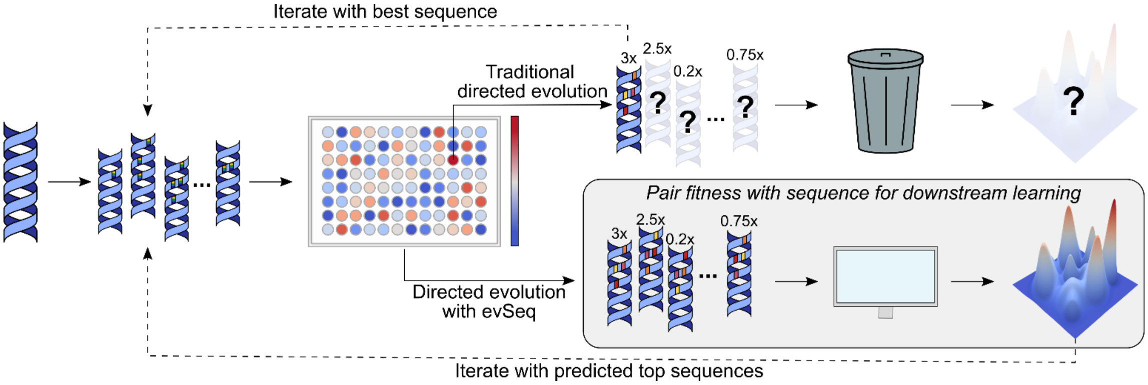

Iterate with best sequence
Traditional directed evolution
3x
2.5x
0.2x
0.75x
?
Pair fitness with sequence for downstream learning
Directed evolution with evSeq
Iterate with predicted top sequences

Figure 2-3. evSeq eliminates the sequencing burden of MLPE. Traditional DE only collects sequence information for top variants, essentially "throwing away" fitness data from inferior variants and learning nothing about the underlying fitness landscape. If, instead, evSeq is used to collect sequence information for all variants, MLPE methods, which require sequence-fitness pairs for supervised model training, can be implemented. Sampling from a fitness landscape, an ML model can be trained to predict the fitnesses of missing sequences and reconstruct the missing regions of this landscape.

To demonstrate the application of evSeq to MLPE, we used it to sequence five plates of *Rma*NOD variants from a four-site combinatorial library. Coupled with fitness data, the sequences resulting from this run could be used to drive a round of MLDE. Notably, sequencing these plates by Sanger sequencing would have cost ~$1400; in contrast, sequencing by evSeq using an outsourced multiplexed MiSeq run cost ~$100 for a per-variant cost of ~$0.21. The median read depth per variant in this run was 463 (mean: 506), much higher than is required for accurate sequencing, and so more plates—from either the same or a different library—could have reasonably been added to this evSeq run to decrease the per-variant sequencing cost even further (**Figure 2-3B**). Of course, as discussed in the previous section, in-house sequencing could have cut sequencing costs an additional ten-fold.

The cost of sequencing is most notably a barrier for MLPE strategies that focus on developing models for a single protein with a well-defined fitness (e.g., MLDE); however, the applicability of evSeq to MLPE is not limited solely to cost-reduction. For instance, ML strategies have been developed that, rather than focusing on a specific protein, train models on sequence-fitness information across multiple different protein scaffolds.[16,43] The goal is for these models to learn global determinants of protein fitness, then to use the models as general-purpose protein fitness predictors. By enabling the collection of sequence-fitness pairs across a wider array of proteins and fitness definitions, evSeq opens these approaches to new and more diverse data sources. Generally speaking, the more sequence-fitness data available to train and benchmark these strategies, the better we expect them to perform and the more rapidly we expect improvements to be developed.[16] It is no coincidence that large leaps forward in other ML disciplines have followed increased availability of large, diverse datasets, with the rapid advance in computer vision sparked by ImageNet being perhaps the most prominent example.[44] Widespread adoption of evSeq—and commitment to depositing sequence-fitness data in resources such as ProtaBank—would provide such a dataset for protein engineering.[8] This dataset would span the range of all engineered proteins and all target fitnesses, capture examples of sequences with both higher and lower/zero fitness relative to a parent (the latter of which is effectively never recorded with current DE sequencing practices), and overall enable rapid advancement in MLPE.

2.2.5 evSeq detects all variability in the sequenced amplicons

Although we focused here on demonstrating applications involving targeted mutagenesis strategies, evSeq is also applicable to other mutagenesis methods as the associated software can identify both user-specified and unspecified positions of variability (**Figure 2-4A**). This feature not only informs the user of potential unexpected mutations in the sequenced amplicon (Appendix A, **Table A-11**), but also allows it to work effectively with tile-based mutagenesis strategies and other semi-targeted mutagenesis strategies (e.g., error-prone PCR of specific regions or small genes). All that is required is that the amplicon length and read length be able to capture the full region containing mutations.

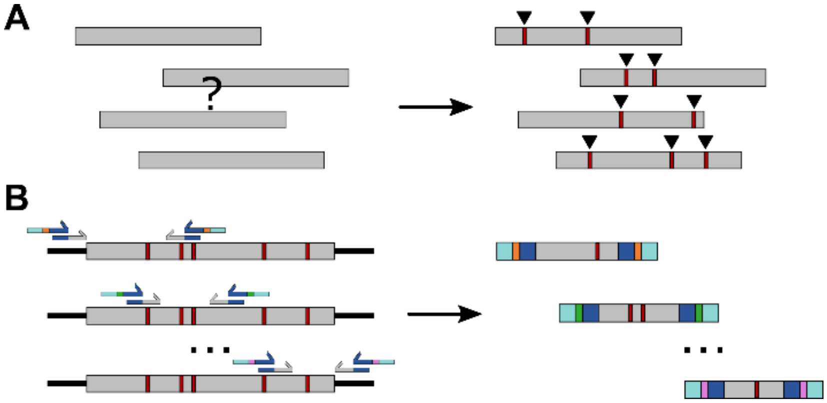


Figure 2-4. evSeq detects variability and can be expanded for random mutagenesis. A evSeq does not require that the user specify which position in the amplicon was targeted. Instead, the software can identify variable regions by comparing to a reference **B** evSeq can be used to sequence entire genes by designing a set of inner primer pairs which together capture the entire gene. Different evSeq barcodes can then be used for each region, and the user can reconstruct the entire sequence.

It should be noted that evSeq will not detect off-target mutations outside of the constructed amplicon as these regions are not sequenced, meaning that it is unable to identify other mutations in a larger DNA element that may be contributing to activity. Due to this fact, for exceedingly unexpected mutational effects that are not seen in replicate, we suggest sequencing the rest of the DNA element to confirm the presence or absence of any off-target mutations. However, this limitation is mitigated by the fact that off-target mutations are rare and, importantly, evSeq is agnostic to read length and will work with any length of paired-end sequencing.[45]

While the current software version is not yet suited for other, long-read sequencing technologies (e.g., PacBio or Oxford Nanopore), future versions could be updated and validated with these data formats and make full gene-length evSeq experiments more straightforward and cost effective. Given this, evSeq is currently best suited and most cost effective when all expected mutations exist in the sequenced amplicon, though sequencing of multiple overlapping amplicons can readily allow evSeq to be expanded to sequence entire genes of variants arrayed in microplates (**Figure 2-4B**). Care must be taken in such an application, however, to account for the fact that aggressive mutation rates could compromise the annealing efficiency of inner primers binding in the variable region, as could mutations to positions closer to the binding region of the 3' end of the inner primer. Such situations would lead to a higher proportion of wells failing sequencing.

2.3 Conclusion

Hundreds to thousands of protein variants (or more) are constructed and their fitnesses evaluated over the course of a standard protein engineering campaign. Without sequencing, these fitnesses are next to useless—the time, effort, and resources expended to produce them are largely wasted. Comparable in cost to existing protocols, accessible to scientists with no or minimal sequencing and computational experience, and easy to implement with existing technology, evSeq rescues these fitness data by making the collection of sequence data for every variant a practical and highly useful step of the protein engineering pipeline. Given the number of research groups working on DE and other protein engineering projects, widespread adoption of evSeq would lead to an explosion in the availability of sequence-fitness information. By sequencing every variant, no laboratory screening effort is wasted, and we open the door to advances in both our biochemical understanding of proteins and our ability to engineer them with data-driven methods.

2.4 Materials and Methods

2.4.1 Single-site-saturation library generation for TrpB.

Saturation mutagenesis libraries were prepared using a modification of the “22-codon trick” described by Kille *et al.*[38] We first designed primers using the templates given in Supplemental Table S12. For the forward primers, each sequence of “NNN” in these templates was replaced with “NDT,” “VHG,” and “TGG,” resulting in a total of three degenerate primers which could then be mixed at a ratio of 12:9:1, respectively. The reverse primers were used without changes.

We also designed primers that bind within the ampicillin resistance (AmpR) gene in pET22b(+) with sequences as given in Appendix A, **Table A-13**. These primers were designed such that, when used in combination with the site-specific primers to run a PCR, two medium-length fragments would be created with a break in the AmpR gene. For the forward site-saturation primers, a PCR was performed using the reverse AmpR primer, resulting in a fragment from ~1500–2000 bp long. For the reverse site-saturation primers, a PCR was performed using the forward AmpR primer, resulting in a fragment ~4500–5000 bp long.

Once PCRs finished, 1 μL of DpnI (NEB R0176S) was added to each of the reactions, which were then incubated at 37 °C for 1 h to digest the unmutated template plasmid. The presence of correctly sized fragments was confirmed via gel electrophoresis and each fragment was then excised from the gel and purified with the Zymoclean Gel DNA Recovery Kit (Zymo Research D4002).

Purified fragments were then assembled following the standard Gibson assembly method.[46] After 1 h at 50 °C, the reaction mixtures were desalted with a DNA Clean & Concentrator-5 kit (Zymo Research D4013) and used to transform electrocompetent E. cloni® cells (Lucigen 60051-1). Libraries were spread onto solid agar selection medium consisting of Luria Broth (RPI L24040-5000.0) supplemented with 100 µg/mL carbenicillin (LB_{carb}) and incubated at 37 °C until single colonies were observed. Individual colonies were then transferred into the wells of 96-well 2-mL deep-well plates containing 300 µL of LB_{carb} to isolate monoclonal enzyme variants, with 8 wells being reserved for control conditions, giving 4-fold oversampling of the 22-codon library. These cultures were grown overnight at 37 °C, 220 rpm, and 80% humidity in an Infors Multitron HT until they reached stationary phase, at which point 100 µL from each well were mixed with an equal volume of 50% glycerol and stored at –80 °C for future use.

For protein expression, 20 µL of the remaining culture were used to inoculate 630 µL of Terrific Broth with 100 µg/mL carbenicillin (TB_{carb}). These were then grown at 37 °C, 220 rpm, and 80% humidity for 3 hours in an Infors Multitron HT, at which point they were placed on ice for 30 minutes. Following this, 50 µL of a 14 mM solution of isopropyl-β-D-thiogalactoside (IPTG; GoldBio #I2481C100) in TB_{carb} were added to each well to induce protein expression at a final concentration of 1 mM IPTG. Expression proceeded in the same Infors Multitron HT shaker as before at 22 °C, 220 rpm for roughly 18 hours. Cells were harvested via centrifugation at 4500*g* for 10 minutes, the supernatant was removed, and the plates (now containing pelleted, expressed cells) were placed at –20 °C until needed.

Once cells had been harvested, cultures for evSeq were prepared. These cultures were started from the 96-well plate glycerol stocks prepared prior to moving into the cell expression protocol; the cultures were grown overnight (~18hrs) in an Infors Multitron HT (220 rpm, 37 °C) to saturation in 96-well deep-well plates in 300 μL of LB_{carb}. These cultures were then frozen and stored at –20 °C to be used for sequencing with evSeq.

A GenBank file detailing the plasmid and primers used in this section is available on the evSeq GitHub at https://github.com/fhalab/evSeq/tree/master/genbank_files/tm9d8s.gb.

2.4.2 Sequencing TrpB libraries with evSeq.

Frozen overnight cultures (preparation detailed in the previous section) were thawed at room temperature. Libraries were then sequenced with the process described in Appendix A, **Section A.2.3**, *evSeq Library Preparation/Data Analysis Protocol*; the evSeq software was run using all default parameters (`average_q_cutoff = 25, bp_q_cutoff = 30, length_cutoff = 0.9, match_score = 1, mismatch_penalty = 0, gap_open_penalty = 3, gap_extension_penalty = 1, variable_thresh = 0.2, variable_count = 10`) with the "`return_alignments`" flag thrown. The inner primers used for library preparation are in Appendix A, **Table A-14**. The barcode plates (Appendix A, **Tables A-3–10**) were paired to positions as given in Appendix A, **Table A-15**.

2.4.3 Measuring the rate of tryptophan formation.

Rate of tryptophan formation data was collected with the same procedure described in Rix *et al.* for non-heat-treated lysate preparation with a few modifications: lysis occurred in 300 μL KPi buffer with 100 μM pyridoxal 5'-phosphate (PLP) supplemented with 1 mg/mL lysozyme, 0.02 mg/mL bovine pancreas DNase I, and 0.1x BugBuster; lysis occurred at 37 °C for 1 h.[35]

2.4.4 Four-site-saturation library generation for *Rma*NOD.

Positions S28, M31, Q52, and L56 of a variant of *Rma*NOD (*Rma*NOD Y32G) were targeted for comprehensive site-saturation mutagenesis using a variant of the 22-codon trick originally described by Kille *et al*.[38] Due to the proximity of positions S28 and M31, it was easiest to use the same mutagenesis primers to target them; the same was done for positions Q52 and L56. Because the 22-codon trick requires three degenerate codons per position targeted, nine individual primers capturing all combinations (3 codons ^ 2 positions/per primer = 9 primers) of the degenerate codons had to be ordered for each of the two mutagenic primers. Sequences of these primers are given in Appendix A, **Table A-16**.

The primers from Appendix A, **Table A-16** were all ordered from IDT at 100 μM. Both a "forward" and a "reverse" primer mixture were prepared by combining individual forward and reverse primers in proportion to the number of individual codons they encoded. A 10 μM forward-reverse primer mixture was then prepared by adding 10 μL of both the forward and reverse primer mixtures to 80 μL ddH$_2$O. Once the forward-reverse primer mixture was prepared, it was used in a PCR to build a pool of DNA fragments containing the four-site combinatorial libraries. Two fragments that captured the remainder of the *Rma*NOD gene and host plasmid (pET22b(+)) were also produced by PCR. The primers used for these flanking fragments are given in Appendix A, **Table A-17**.

After PCR completed 1 μL DpnI (NEB R0176S) was added to each reaction. The reactions were then held at 37 °C in a thermalcycler for 1 h. The PCR fragments were then gel-extracted using a Zymoclean Gel DNA Recovery Kit (D4002).

Fragments were to eventually be assembled using Gibson assembly.[46] Because the efficiency of Gibson assembly increases with decreasing numbers of fragments, an assembly PCR was performed to combine flanking fragment 1 (see Appendix A, **Table A-17** for details) and the variant fragment. The resultant assembled fragment was then gel-extracted, again using a Zymoclean Gel DNA Recovery Kit (D4002).

To complete construction of the library of variant plasmids, a Gibson assembly was performed to combine the assembled PCR fragment and flanking fragment 0. After Gibson assembly, the Gibson reaction was cleaned using a Monarch PCR & DNA Cleanup Kit (NEB CAT T1030L). The cleaned Gibson product was next used to transform electrocompetent E. cloni® BL21 DE3. Transformed cells were spread onto solid agar selection medium consisting of Luria Broth (RPI L24040-5000.0) supplemented with 100 μg/mL ampicillin (LB_{amp}) and incubated at 37 °C until single colonies were observed.

To build the 96-well plates of *Rma*NOD variants used to demonstrate evSeq, 400 μL LB + 100 μg/mL ampicillin were first added to each well of 5x 96-well deep-well plates. Colonies from the agar plates grown overnight were then picked into the wells of the deep-well plates. The plates were placed in an Infors Multitron HT at 240 rpm, 37 °C for ~16 h. To glycerol stock the now-stationary-phase culture, 100 μL overnight culture were added to 100 μL 50% glycerol before being stored at –80 °C until its use in evSeq library preparation.

A GenBank file detailing the plasmid and primers used in this section is available on the evSeq GitHub at:

https://github.com/fhalab/evSeq/tree/master/genbank_files/rmanod_y32g.gb

2.4.5 Sequencing *Rma*NOD libraries with evSeq.

To begin preparation of culture for evSeq with the *Rma*NOD variants, cultures in 96-well deep-well plates (with 300 μL of LB_{carb}) were started from the 96-well plate glycerol stocks prepared in the previous section. The plates were placed in an Infors Multitron HT at 240 rpm; the cultures were grown overnight (~18hrs) before being frozen and stored at –20 °C.

To start the evSeq protocol, frozen overnight cultures were thawed in a room temperature water bath. Libraries were then sequenced with the process described in Appendix A, **Section A.2.3**, *evSeq Library Preparation/Data Analysis Protocol*; the evSeq software was run using the same parameters as for the TrpB data analysis (see **Section 2.4.2**, *Sequencing TrpB Libraries with evSeq*, above). The inner primers used for evSeq library preparation are given in Appendix A, **Table A-18**. The barcode plates (Appendix A, **Tables A-3–10**) were paired to positions as given in Appendix A, **Table A-19**.

Chapter II Bibliography

1. BCC Research Staff. Global markets for enzymes in industrial applications. *BCC Research LLC.* (2018) https://www.bccresearch.com/market-research/biotechnology/global-markets-for-enzymes-in-industrial-applications.html.
2. Podgornaia, A. I., and Laub, M. T. Pervasive degeneracy and epistasis in a protein-protein interface. *Science* **347**, 673–677 (2015).
3. Wu, N. C., Dai, L., Olson, C. A., Lloyd-Smith, J. O., and Sun, R. Adaptation in protein fitness landscapes is facilitated by indirect paths. *eLife* **5**, (2016).
4. Faure, A. J., Domingo, J., Schmiedel, J. M., Hidalgo-Carcedo, C., Diss, G., and Lehner, B. Global mapping of the energetic and allosteric landscapes of protein binding domains. *bioRxiv* (2021). doi: 10.1101/2021.09.14.460249.
5. Yang, K. K., Wu, Z., and Arnold, F. H. Machine-learning-guided directed evolution for protein engineering. *Nat. Methods* **16**, 687–694 (2019).
6. Li, G., Dong, Y., and Reetz, M. T. Can machine learning revolutionize directed evolution of selective enzymes? *Adv. Synth. Catal.* **361**, 2377–2386 (2019).
7. Hopf, T. A., Ingraham, J. B., Poelwijk, F. J., Schärfe, C. P. I., Springer, M., Sander, C., and Marks, D. S. Mutation effects predicted from sequence co-variation. *Nat. Biotechnol.* **35**, 128–135 (2017).
8. Wang, C. Y., Chang, P. M., Ary, M. L., Allen, B. D., Chica, R. A., Mayo, S. L., and Olafson, B. D. ProtaBank: A repository for protein design and engineering data. *Protein Sci.* **27**, 1113–1124 (2018).
9. Mazurenko, S., Prokop, Z., Machine learning in enzyme engineering. *ACS Catal.* **10**, 1210–1223 (2020).
10. Siedhoff, N. E., Schwaneberg, U., and Davari, M. D. Machine learning-assisted enzyme engineering, in *Methods in Enzymology* 1st ed., 281–315 (2020).
11. Wittmann, B. J., Johnston, K. E., Wu, Z., and Arnold, F. H. Advances in machine learning for directed evolution. *Curr. Opin. Struct. Biol.* **69**, 11–18 (2021).
12. Fowler, D. M., and Fields, S. Deep mutational scanning: A new style of protein science. *Nat. Methods* **11**, 801–807 (2014).

13. Livesey, B. J., and Marsh, J. A. Using deep mutational scanning to benchmark variant effect predictors and identify disease mutations. *Mol. Syst. Biol.* **16** (2020).

14. Riesselman, A. J., Ingraham, J. B., and Marks, D. S. Deep generative models of genetic variation capture the effects of mutations. *Nat. Methods* **15**, 816–822 (2018).

15. Meier, J., Rao, R., Verkuil, R., Liu, J., Sercu, T., and Rives, A. Language models enable zero-shot prediction of the effects of mutations on protein function. *bioRxiv* (2021). doi:10.1101/2021.07.09.450648.

16. Gray, V. E., Hause, R. J., Luebeck, J., Shendure, J., and Fowler, D. M. Quantitative missense variant effect prediction using large-scale mutagenesis data. *Cell Syst.* **6**, 116–124 (2018).

17. Sanger, F., and Coulson, A. R. A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase. *J. Mol. Biol.* **94**, 441–448 (1975).

18. Metzker, M. L. Sequencing technologies — the next generation. *Nat. Rev. Genet.* **11**, 31–46 (2010).

19. Smith, A. M., Heisler, L. E., St. Onge, R. P., Farias-Hesson, E., Wallace, I. M., Bodeau, J., Harris, A. N., Perry, K. M., Giaever, G., Pourmand, N., and Nislow, C. Highly-multiplexed barcode sequencing: an efficient method for parallel analysis of pooled samples. *Nucleic Acids Res.* **38** (2010).

20. Appel, M. J., Longwell, S. A., Morri, M., Neff, N., Herschlag, D., and Fordyce, P. M. uPIC–M: Efficient and scalable preparation of clonal single mutant libraries for high-throughput protein biochemistry. *ACS Omega* **6**, 30542–30554 (2021).

21. Srivathsan, A., Lee, L., Katoh, K., Hartop, E., Kutty, S. N., Wong, J., Yeo, D., and Meier, R. ONTbarcoder and MinION barcodes aid biodiversity discovery and identification by everyone, for everyone. *BMC Biol.* **19** (2021).

22. Glenn, T. C., *et al*. Adapterama I: Universal stubs and primers for 384 unique dual-indexed or 147,456 combinatorially-indexed Illumina libraries (iTru & iNext). *PeerJ* **7**, e7755 (2019).

23. Wierbowski, S. D. *et al.* A massively parallel barcoded sequencing pipeline enables generation of the first ORFeome and interactome map for rice. *Proc. Natl. Acad. Sci. U.S.A.* **117**, 11836–11842 (2020).

24. Chubiz, L. M., Lee, M.-C., Delaney, N. F., and Marx, C. J. FREQ-Seq: A rapid, cost-effective, sequencing-based method to determine allele frequencies directly from mixed populations. *PLoS One 7*, e47959 (2012).

25. Weile, J. *et al*. A framework for exhaustively mapping functional missense variants. *Mol. Syst. Biol.* **13**, 957 (2017).

26. Campbell, N. R., Harmon, S. A., and Narum, S. R. Genotyping-in-thousands by sequencing (GT-seq): A cost effective SNP genotyping method based on custom amplicon sequencing. *Mol. Ecol. Resour.* **15**, 855–867 (2015).

27. Wen, C., Wu, L., Qin, Y., Van Nostrand, J. D., Ning, D., Sun, B., Xue, K., Liu, F., Deng, Y., Liang, Y., and Zhou, J. Evaluation of the reproducibility of amplicon sequencing with Illumina MiSeq platform. *PLoS One* **12**, e0176716 (2017).

28. Siloto, R. M. P., and Weselake, R. J. Site saturation mutagenesis: Methods and applications in protein engineering. *Biocatal. Agric. Biotechnol.* **1**, 181–189 (2012).

29. Melnikov, A., Rogov, P., Wang, L., Gnirke, A., and Mikkelsen, T. S. Comprehensive mutational scanning of a kinase in vivo reveals substrate-dependent fitness landscapes. *Nucleic Acids Res.* **42**, e112 (2014).

30. Sims, D., Sudbery, I., Ilott, N. E., Heger, A., and Ponting, C. P. Sequencing depth and coverage: Key considerations in genomic analyses. *Nat. Rev. Genet.* **15**, 121–132 (2014).

31. de Muinck, E. J., Trosvik, P., Gilfillan, G. D., Hov, J. R., and Sundaram, A. Y. M. A novel ultra high-throughput 16S rRNA gene amplicon sequencing library preparation method for the Illumina HiSeq platform. *Microbiome* **5**, 68 (2017).

32. Tresnak, D. T., and Hackel, B. J. Mining and statistical modeling of natural and variant class IIa bacteriocins elucidate activity and selectivity profiles across species. *Appl. Environ. Microbiol.* **86**, e01646-20 (2020).

33. Illumina. Nextera XT DNA library prep reference guide. (2019) https://emea.support.illumina.com/content/dam/illumina-support/documents/documentation/chemistry_documentation/samplepreps_nextera/nextera-xt/nextera-xt-library-prep-reference-guide-15031942-05.pdf

34. Boville, C. E., Romney, D. K., Almhjell, P. J., Sieben, M., and Arnold, F. H.

Improved synthesis of 4-cyanotryptophan and other tryptophan analogues in aqueous solvent using variants of TrpB from *Thermotoga maritima*. *J. Org. Chem.* **83**, 7447–7452 (2018).

35. Rix, G., Watkins-Dulaney, E. J., Almhjell, P. J., Boville, C. E., Arnold, F. H., and Liu, C. C. Scalable continuous evolution for the generation of diverse enzyme variants encompassing promiscuous activities. *Nat. Commun.* **11**, 5644 (2020).

36. Romney, D. K., Murciano-Calles, J., Wehrmüller, J. E., and Arnold, F. H. Unlocking reactivity of TrpB: A general biocatalytic platform for synthesis of tryptophan analogues. *J. Am. Chem. Soc.* **139**, 10769–10776 (2017).

37. Buller, A. R., Brinkmann-Chen, S., Romney, D. K., Herger, M., Murciano-Calles, J., and Arnold, F. H. Directed evolution of the tryptophan synthase β-subunit for stand-alone function recapitulates allosteric activation. *Proc. Natl. Acad. Sci. U.S.A.* **112**, 14599–14604 (2015).

38. Kille, S., Acevedo-Rocha, C. G., Parra, L. P., Zhang, Z. G., Opperman, D. J., Reetz, M. T., and Acevedo, J. P. Reducing codon redundancy and screening effort of combinatorial protein libraries created by saturation mutagenesis. *ACS Synth. Biol.* **2**, 83–92 (2013).

39. Morrison, K. L., and Weiss, G. A. Combinatorial alanine-scanning. *Curr. Opin. Chem. Biol.* **5**, 302–307 (2001).

40. Hsu, C., Nisonoff, H., Fannjiang, C., and Listgarten, J. Learning protein fitness models from evolutionary and assay-labeled data. *Nat. Biotechnol.* (2022).

41. Wu, Z., Kan, S. B. J., Lewis, R. D., Wittmann, B. J., and Arnold, F. H. Machine learning-assisted directed protein evolution with combinatorial libraries. *Proc. Natl. Acad. Sci. U.S.A.* **116**, 8852–8858 (2019).

42. Wittmann, B. J., Yue, Y., and Arnold, F. H. Informed training set design enables efficient machine learning-assisted directed protein evolution. *Cell Syst.* **12**, 1026-1045.e7 (2021).

43. Alieva, A., Aceves, A., Song, J., Mayo, S., Yue, Y., and Chen, Y. Learning to make decisions via submodular regularization. *ICLR* (2021).

44. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet: Large scale visual recognition challenge. *Int. J. Comput. Vis.* **115**, 211–252 (2015).

45. McInerney, P., Adams, P., and Hadi, M. Z. Error rate comparison during polymerase chain reaction by DNA polymerase. *Mol. Biol. Int.* (2014).

46. Gibson, D. G., Young, L., Chuang, R. Y., Venter, J. C., Hutchison, C. A., and Smith, H. O. Enzymatic assembly of DNA molecules up to several hundred kilobases. *Nat. Methods* **6**, 343–345 (2009).

Chapter III

A COMBINATORIALLY COMPLETE EPISTATIC FITNESS LANDSCAPE IN AN ENZYME ACTIVE SITE

Material from this chapter appears in: "**Johnston, K. E.,** Watkins-Dulaney, E. J., Almhjell, P.J., Liu, G., Porter, N. J., Yang, J. & Arnold, F. H.* A combinatorially complete epistatic fitness landscape in an enzyme active site. *Manuscript in preparation.*"

K.E.J. and E.J.W conceived the project. K.E.J. designed double- and quadruple-site landscapes and did all experimental data collection except crystallography. K.E.J., P.J.A., G.L., and J.Y. wrote sequence processing and data analysis software. K.E.J. wrote initial draft. K.E.J., E.J.W., P.J.A., and F.H.A edited and revised manuscript.

ABSTRACT

Model-guided protein engineering approaches are quickly emerging as more efficient ways to navigate protein fitness landscapes. However, many of the existing datasets for testing and developing such approaches consist mainly of single substitutions across a protein sequence and are not compatible with the goals of protein engineering, which seeks to accumulate many activity-boosting substitutions. Furthermore, protein engineering often targets binding regions and active sites, which are commonly enriched in epistasis—non-additive interactions between substitutions which cannot be predicted from non-combinatorial datasets. Few existing datasets capture epistasis at large scale, and those that do often focus on binding, not catalysis. Here, we generate a combinatorially complete, 160,000-variant landscape across four residues in the active site of an enzyme. Assaying the native function of a thermostable β-subunit of tryptophan synthase (TrpB) in a non-native environment resulted in a landscape characterized by significant, difficult-to-navigate epistasis and many local optima. These effects prevent simulated directed evolution approaches from efficiently reaching the global optima in many cases. However, there is wide variability in effectiveness of the different approaches, which together provide experimental benchmarks for predictive workflows to beat. Within this landscape, the fittest variants all contained a substitution that is nearly absent in existing TrpB sequences—a result that conservation-based predictions would not capture. Thus, although fitness prediction using evolutionary data might work to classify inactive and active variants, these approaches may struggle to differentiate among the best ones, even for near-native functions. Overall, this work presents a new, large-scale testing ground for model-

guided enzyme engineering approaches and suggests that efficient navigation of epistatic fitness landscapes will require advances in both data-driven predictors and physical modeling.

3.1 Introduction

Engineered proteins are important pharmaceutical and industrial targets, but there are few ways to reliably engineer a protein for desired properties due to the complex, and largely unknown, relationship between a sequence and function (the level of which is its ‘fitness’). The effects of substitutions cannot be reliably predicted, especially in enzymes, which must guide substrates through intricate, often multi-step reaction pathways with high efficiency and selectivity. Currently, the most reliable way to optimize an enzyme for a desired function is by directed evolution, using multiple generations of semi-rational or random mutagenesis and screening to accumulate beneficial mutations.[1] However, this process is time- and resource-intensive, with directed evolution campaigns taking weeks to months. Predictive models that can shorten protein engineering timelines are valuable for quickly addressing emergent needs for better enzymes.[2] The development of such models requires high-quality datasets for testing and comparing new approaches.

Most datasets used for developing and testing predictive models have been generated by deep-mutational scanning (DMS). DMS measures the effects of every amino acid substitution across many or even all residues in a protein sequence,[3] providing information about all single substitutions in a specific protein background. However, DMS provides no information about the effects of substitutions in different backgrounds or how substitutions interact with other substitutions. In many instances, their effects are approximately independent (and therefore “additive”).[4] Under these circumstances, combining beneficial substitutions in a single sequence works well to identify improved variants with multiple substitutions, and this makes laboratory recombination a powerful evolutionary strategy.[1] Simple models can often predict the fitnesses of double and even triple mutants from single-site data alone.[5] However, additivity can break down due to epistasis, particularly for positions in close proximity or which simultaneously interact with cofactors or substrates; thus, binding sites and enzyme active sites are enriched in epistasis.[6] As these regions of proteins are vital for protein function, efficient navigation of epistasis is critical.

Despite a rapidly improving understanding of the mapping of protein sequence to structure and binding,[7–10] progress toward fitness prediction in epistatic regions of proteins has been more measured.[11,12] Multi-site saturation combinatorial landscapes have been a major testing ground for prediction and navigation of epistatic interactions,[13–15] but few landscapes exist that deeply examine these interactions, typically going no further than double-site saturation mutagenesis[16] or random sampling of multi-mutants.[17–20] Combinatorially complete landscapes, where every variant is characterized (within a constrained search space), are of particular interest because they enable exact calculation of epistasis within them and thorough testing of experimental or computational methodologies that aim to reach the most fit variants via simulation.[21,22] The few existing quadruple-site saturation landscapes measure only binding,[23,24] a simpler task than enzyme catalysis, which has proven to be much more difficult for protein design.[25,26] We sought to measure a combinatorial fitness landscape of an enzyme active site, furnishing a dataset that would enable examination of epistasis and evolutionary constraints in an enzymatic system, with a focus on how that epistasis might restrict enzyme engineering approaches such as directed evolution.

3.2 Results

3.2.1 Construction of multi-site saturation combinatorial landscapes using a growth-based assay

For this study, we chose the β-subunit of tryptophan synthase (TrpB), which synthesizes L-tryptophan (Trp) from indole and L-serine (Ser). TrpB is well suited for large-scale dataset generation because Trp is essential for proteome replication and cell growth, making TrpB amenable to a growth-based sequencing assay to obtain fitness and sequence data in high throughput. Furthermore, TrpB has been rigorously characterized and is essential in all kingdoms of life besides animals, providing a wealth of previous biochemical and structural data as well as sequence diversity. For the parent background, we chose a previously engineered TrpB variant *Tm*9D8*.[27] Derived from *Thermotoga maritima*, this variant is highly thermostable but can function at 37 °C, providing high activity at *Escherichia coli* growth temperatures. This feature is useful for decoupling loss of catalytic activity from loss of stability because fitness effects are less likely to be dominated by stability effects.[4] This TrpB variant was evolved to work as a stand-alone enzyme without its native partner,[28] the α-subunit of tryptophan synthase (TrpA), which allosterically activates TrpB and shuttles indole to the TrpB active site.

Previous work has shown that yeast harboring TrpB variants and provided exogenous indole can be growth-limited by Trp formation, enabling continuous evolution systems to evolve competent stand-alone TrpB variants.[29] A similar approach was implemented here using *E. coli* as the host organism. The *E. coli* strain auxotrophic for Trp was constructed through deletion of the *trpA* and *trpB* genes. Although deletion of *trpA* is not strictly necessary, it avoids potential confounding allosteric interactions between the native *E. coli* TrpA and the heterologous *Tm*TrpB.[30] The auxotroph strain harboring *Tm*9D8* exhibited both Trp- and TrpB-dependent growth (**Figure B-1**) in media lacking Trp (Trp-dropout media), and the concentrations of indole and gene-expression inducer were optimized via plate-based independent growth assays (Appendix B, **Figure B-2**, **Section B.1.6**).

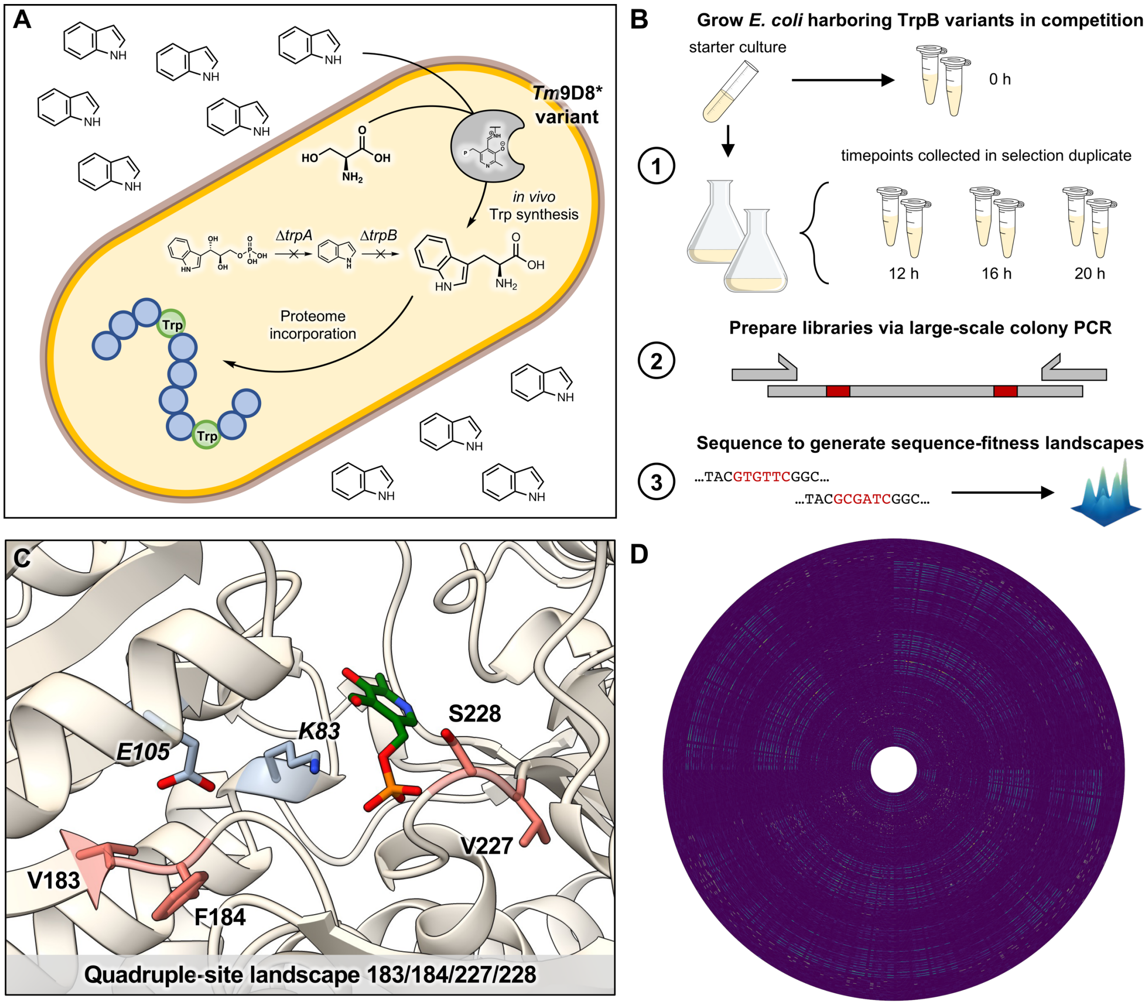

A
NH
Tm9D8*
variant
HO
OH
NH2
in vivo
Trp synthesis
ΔtrpA
ΔtrpB
HN
OH
Proteome
incorporation
Trp
Trp
B
Grow E. coli harboring TrpB variants in competition
starter culture
0 h
1
timepoints collected in selection duplicate
12 h
16 h
20 h
2
Prepare libraries via large-scale colony PCR
3
Sequence to generate sequence-fitness landscapes
...TACGTGTTCGGC...
...TACGCGATCGGC...
C
E105
K83
S228
V227
V183
F184
Quadruple-site landscape 183/184/227/228
D

Figure 3-1. Overview of TrpB-based combinatorial landscapes. A An *E. coli* strain with deletions of the trpA and trpB genes is transformed with plasmid harboring TrpB. When provided with exogenous indole, the harbored TrpB produces Trp according to its activity, enabling proteome and cellular replication. **B** For each landscape, the *E. coli* Trp auxotroph is transformed with a plasmid library and used as a starter culture to inoculate two replicate flasks as well as an initial timepoint, T_0, (with one or two replicates). Samples at different timepoints are collected in duplicate for up to 36 h and prepared for sequencing. **C** The quadruple-site saturation landscape

targeted two pairs of positions: 183/184 and 227/228 (pink). The pyridoxal 5'-phosphate (PLP) cofactor of TrpB is colored green and two important catalytic residues are light blue, showing the proximity of the selected sites to the catalytic core of TrpB. **D** Radial heatmap of the fitness values obtained for the quadruple-site landscape of TrpB. Missing values are white, demonstrating the high completeness of the landscape.

Single-site (20 possible variants) and double-site (400 possible variants) saturation libraries were constructed and assayed with plate-based independent growth rates and growth-based enrichment assays (**Figure 3-1A**). Sites were chosen to be near the active site or because they were previously seen to modulate TrpB activity in engineering campaigns. The plate-based independent growth rate assays monitored the cell density of *E. coli* Trp auxotrophs harboring TrpB variants via growth in Trp-dropout media (Appendix B, **Section B.1.6**, **Figure B-2**) while the growth-based enrichment assays were obtained by sequencing (**Figure 3-1B**, and Appendix B, **Section B.1.8**). We also compared to rates of Trp formation collected previously with *in vitro* lysate-based assays.[31] We observed a reasonable correlation across each of these activity measurements, indicating that the growth assays report on the enzyme-specific rate of Trp synthesis (more details on these assays and results can be found in Appendix B, **Section B.1.4**, **Figures B-3–5**).

We then moved on to designing triple-site saturation libraries and constructing these landscapes with the competitive growth-based assay (8,000 possible variants per landscape), targeting mainly residues in the active site known to impact activity, as well as their neighbors. In total, twenty different positions were targeted across nine landscapes with some overlapping positions between them (Appendix B, **Figures B-6–7**). These preliminary tests were designed to test scaling of methods to larger library sizes, identify epistatic residues, and identify residues which tolerated more than one different amino acid (Appendix B,

Figures B-8–10). Four of the nine landscapes had only a handful of variants with detectable activity, three had ~10–100 variants with detectable activity, and two had >100 variants with detectable activity. The positions sampled in these libraries varied in the breadth of substitutions tolerated. We also saw that this tolerance to substitution at each position depended on the sequence background in which they were sampled, indicating strong epistatic interactions. Inspecting these datasets, we chose four residues that we expected to display epistasis and provide a breadth of activities for scaling to a 160,000-variant, quadruple-site saturation landscape (**Figure 3-1C**). From this landscape 159,129 variants had an average number of input counts greater than ten, allowing us to quantify fitness for 99.45% of the total library (**Figure 3-1D**). We defined fitness based on work by Kowalsky *et al.*[32] (Appendix B, **Section B.1.11**, **Figures B-11–12**) and imputed the missing 871 fitness values for downstream analyses (Appendix B, **Figure B-13**).

3.2.2 Epistasis constrains navigability of fitness landscapes

The top variant of the quadruple-site landscape—the global optimum—contained substitutions at all four sites (V183A, F184I, V227K, and S228G with respect to parent) and is referred to hereafter as AIKG. Two of the substitutions, F184I and S228G, are reversions to wild-type residues in *Tm*TrpB and therefore not unexpected, since the assay was designed to capture the native function. A third substitution, V183A, incorporates the fourth-most-common residue at this position based on a multiple sequence alignment for *Tm*9D8* (referred to hereafter as VFVS), found in 10.41% of sequences (Appendix B, **Figure B-14**). The final V227K substitution, however, was surprising. V227K occurs at near-noise levels (0.01%) in natural sequences (Appendix B, **Figure B-14**) but is clearly beneficial under these assay conditions. Beyond AIKG, we observed a strong preference for 227K in the top variants in the landscape, occurring in all ten top variants and in nearly half of the top fifty (Appendix B, **Figure B-15**). Despite this strong preference, however, 227K is not uniformly beneficial. For example, in the parent background sequence of VFVS, it essentially ablates activity, requiring the S228G substitution before yielding an improved variant.

For further analysis, we determined an activity threshold as the upper bound of the 95% confidence interval of the fitness distribution of stop-codon-containing sequences (all of which are expected to be inactive due to proximity to the active site and occurrence in the middle of the coding sequence). We enforced the threshold over both replicates. Sequences with fitness values below this threshold were classified as "inactive," which left 9,783 "active" variants (6.11% of the library) whose activities could be reliably quantified (Appendix B, **Figure B-16**). This is a large fraction of inactive variants, but for residues so close to the active site being mutated simultaneously this fraction is not unexpected given the dynamic range of the assay. The previously reported activity threshold of 0.01 used for the GB1 binding landscape[16] classified 34,545 variants (21.59%) as active.

Using these thresholds, we quantified the prevalence of pairwise epistasis (Appendix B, **Section B.1.12**), including magnitude, reciprocal sign, and sign epistasis for all paths proceeding through active, quantifiable variants (**Figure 3-2A**). Magnitude epistasis, which occurs when the combined effects of two substitutions are in the same direction as expected but are of a smaller or larger magnitude than expected if perfectly additive, is navigable by step-wise or recombination DE approaches. Sign epistasis occurs when the effect of one of the substitutions changes direction in the background of the other, and therefore is only navigable by step-wise DE approaches if substitutions are made in the correct order, which is not known *a priori*. Finally, reciprocal sign epistasis occurs when the effects of both substitutions change direction when they are made together and is therefore not navigable by step-wise or recombination DE approaches that use only beneficial substitutions.

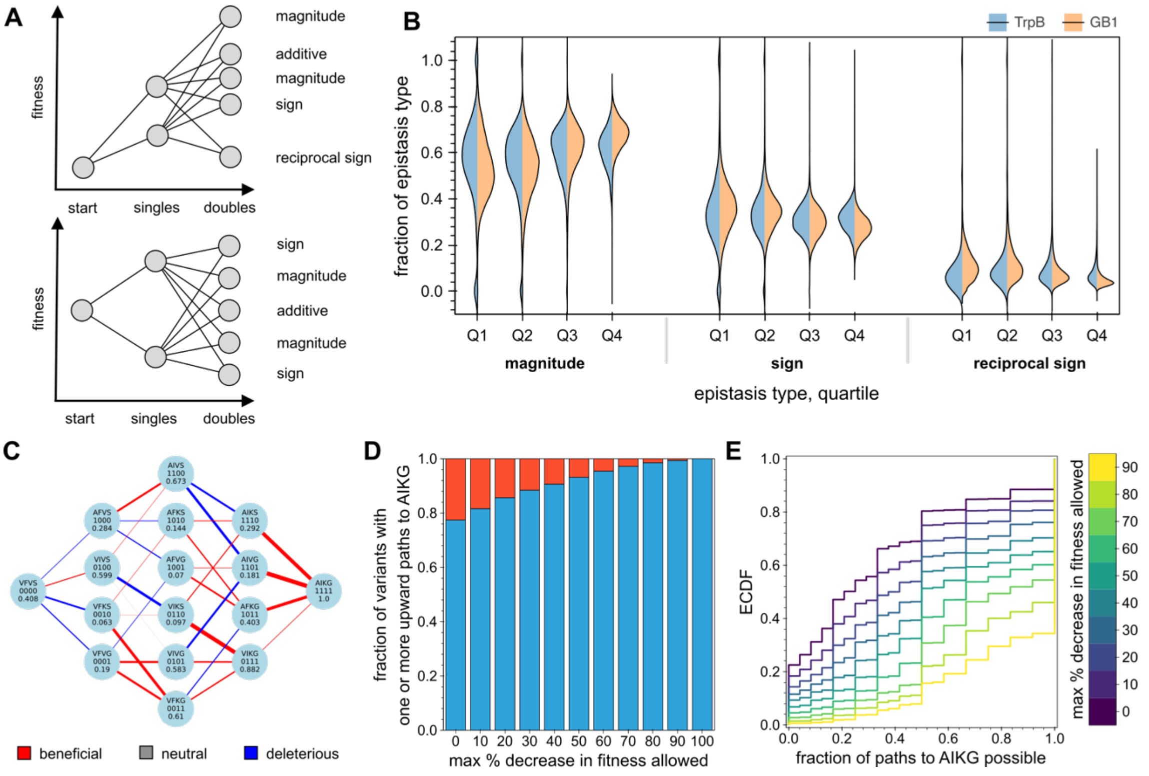


Figure 3-2. Examining how epistasis can constrain evolution. A The types of pairwise epistasis where both single substitutions are beneficial (top) or one single substitution is beneficial while the other is deleterious (bottom). **B** Distributions of the three types of pairwise epistasis within the TrpB and GB1 landscapes separated by quartile of the fitness (Q1, Q2, Q3, Q4) of the starting variant, differentiating epistasis prevalence from low-fitness variants (Q1) to high-fitness variants (Q4). **C** An example path map from the parent variant (*Tm*9D8*, VFVS) to the top variant in the landscape (AIKG). Nodes are labeled with the amino acids at positions 183, 184, 227, and 228, respectively, along with the fitness of that variant. Uphill paths are colored red, neutral paths (<10% change in fitness) are gray, and deleterious paths are blue. The width of the line indicates the magnitude of the increase or decrease. **D** A path map like that pictured in **C** can be built connecting every detectably active variant (imputed variants not considered as starting points but were used as intermediate variants in graphs) to the top variant for a total number of path maps equal to the total number of active variants. Considering each of these maps, the fraction of maps with at least one possible path to the top variant is colored in blue while the fraction with no possible paths is colored in red. When no downward steps are allowed, max fractional decrease in fitness allowed = 0 and

only strictly neutral or beneficial substitutions are allowed. This stringency is relaxed by accepting increasingly deleterious substitutions up to 100% (where all paths are accessible). **E** An empirical cumulative distribution function built from all possible starting points and displaying the fraction of paths reaching the top, given a specified neutral cutoff. The x-axis denotes the fraction of possible paths to the top variant, and the y-axis denotes the fraction of starting variants which have up to that fraction of paths possible to the top variant.

Overall, the fractions of epistasis (among the active variants) are similar in the TrpB and GB1 landscapes across all starting fitness quartiles of the starting variant (**Figure 3-2B**). Generally, additive effects and magnitude epistasis (non-sign epistasis) dominate across all fitness quartiles, followed by sign epistasis as the next most common type, and finally by reciprocal sign epistasis. Interestingly, however, we saw that the GB1 binding protein variants were more likely to experience non-sign epistasis as starting fitness increased across quartiles, decreasing the prevalence of sign and reciprocal sign epistasis. The dependence of non-sign epistasis on starting fitness in the enzymatic landscape stayed more consistent. This indicates that difficult-to-navigate epistasis persists into a higher fitness regime in the enzyme landscape while it attenuates at higher fitness in the binding landscape, suggesting that the binding landscape becomes smoother for higher fitness variants. The enzyme landscape remains more rugged for all levels of fitness. Further pairwise epistasis analyses can be found in Appendix B, **Figure B-17–22**.

Given this, we investigated how navigation of the landscape is constrained by epistasis. To do so, we first built directional graphs linking any active variant (Appendix B, **Section B.1.14**) to the best variant in the landscape, AIKG, via single substitutions (**Figure 3-2C**). For this analysis, only direct paths were considered with a maximum number of steps equal to the Hamming distance (HD) between the initial and final variants (i.e., we did not allow “side-steps” through other variants via conversion bypass or detour bypass). The number of possible paths is factorial of the HD from an initial variant to a final variant (e.g., for HD=4 there are 24 possible paths). Using these graphs, we determined the fraction of starting points which have at least one possible path to the top and found that if no deleterious steps are allowed, ~20% of the starting points cannot reach the global optimum, AIKG, via any single-step path, and so must navigate sign and/or reciprocal sign epistasis to do so (**Figure 3-2D**). Accounting for assay noise or a less restrictive evolutionary pressure, we looked at how changing the allowed magnitude of deleterious steps enabled access to more paths to the best variant. We allowed steps from a 0% decrease in fitness to a 100% decrease in fitness—at which point all steps are allowed, and thus all paths accessible—and observed that even allowing any step up to 50% worse still resulted in ~2% of starting variants having no possible pathway to the top. Such strongly deleterious substitutions are unlikely to be accumulated during natural or laboratory evolution under an explicit selective pressure, thereby constraining these starting points from reaching the global optimum.

We examined this in more detail using empirical cumulative distribution functions (ECDFs) that represent the fraction of variants that have at least a given fraction of paths accessible to the top variant (**Figure 3-2E**). The greater the number of variants that display a low fraction of accessible paths to the top (e.g., 1/24 paths), the more the ECDF is left-shifted. When no deleterious steps are allowed, we see that ~30% of the paths to the top variant are accessible from the median starting variant (the ECDF at $y = 0.5$), but by allowing steps up to a 50% reduction in fitness, the median starting variant now had all paths to the top variant accessible.

We next examined the local optima, defined as variants where no single substitution of an active variant yields a more fit variant. There are 520 optima (5.60% of the active variants), with one being the global optimum, AIKG, and 169 of which are greater than 10% of the maximum fitness. Of the remaining 519 local optima, 98.27% (510 variants) could be escaped via two simultaneous substitutions, while the remaining six required three simultaneous substitutions. Many fewer local optima were observed in the GB1 landscape, with only 30 total fitness peaks (0.07% of the active variants).[24] For further characterization of the local optima, we focused on the top twenty local optima to reduce the impact of noise from optima near the fitness threshold (Appendix B, **Section B.1.13**). Allowing no downward steps, we observed a tendency for the number of starting variants with at least one path to the local optimum to decrease as the fitness of the local optimum decreased (Appendix B, **Figures B-23–24**). Variant LPKG was the exception, potentially due to constraints imposed by incorporation of proline at position 184. Altogether, these results suggest that the TrpB landscape may be enriched in evolution-constraining epistasis compared to GB1, and experimental paths may more easily be trapped at local optima.

3.2.3 Performance of directed evolution and fitness predictors

We next examined how the effects of epistasis might change the results of different directed evolution approaches. We considered three different directed evolution approaches that can also serve as competitive benchmarks for predictive approaches: Method 1) site-saturation mutagenesis (SSM) at each of the four sites in parallel followed by recombination of the best variants at each site; Method 2) single-step sequential SSM, using the best variant at one site as the parent for the next until all four sites have been examined, starting from all sites; and Method 3) SSM at each site in parallel followed by direct synthesis of the top 96 additivity-predicted variants (**Figure 3-3A**, Appendix B, **Section B.1.15**). In all cases, because we enforced the sampling of every single substitution during the site-saturation mutagenesis steps, we would expect to obtain the max fitness every time with each of these approaches if the landscapes exhibited no epistasis. Only Method 2 can navigate sign and reciprocal sign epistasis, as it samples a new background after each round of SSM, and therefore can discover previously deleterious substitutions that have become beneficial.[33] However, it would require that the substitutions are made in the correct order, which is unknown *a priori*, for it to be efficient.

For both the TrpB data and the GB1 data, we saw the same pattern of performance: Method 3 performed the best, then Method 2, and then Method 1 (**Figures 3-3B** and **3-3C**) starting from one of the top 9,783 variants of either landscape. Importantly, these simulations were run by starting only with variants above the respective activity threshold of each landscape. This is the most realistic comparison to directed evolution since detectable starting activity is needed to begin an evolution campaign, and the comparison allows the use of the same number of starting variants for better comparison between TrpB and GB1. If simulations were allowed to start from any variant, the performance was much worse for both landscapes (Appendix B, **Figure B-27**). The performance drop was similar for the two landscapes for Methods 1 and 3, but surprisingly, Method 2 on GB1 saw a much less drastic drop in performance, working better than Method 3 on average. This may suggest that the single-step SSM greedy walk approach may be more robust than the SSM calculate and test top *N* approach and able to increase fitness more reliably even in a noisy, low-fitness region, potentially due to its ability to navigate sign and reciprocal sign epistasis.

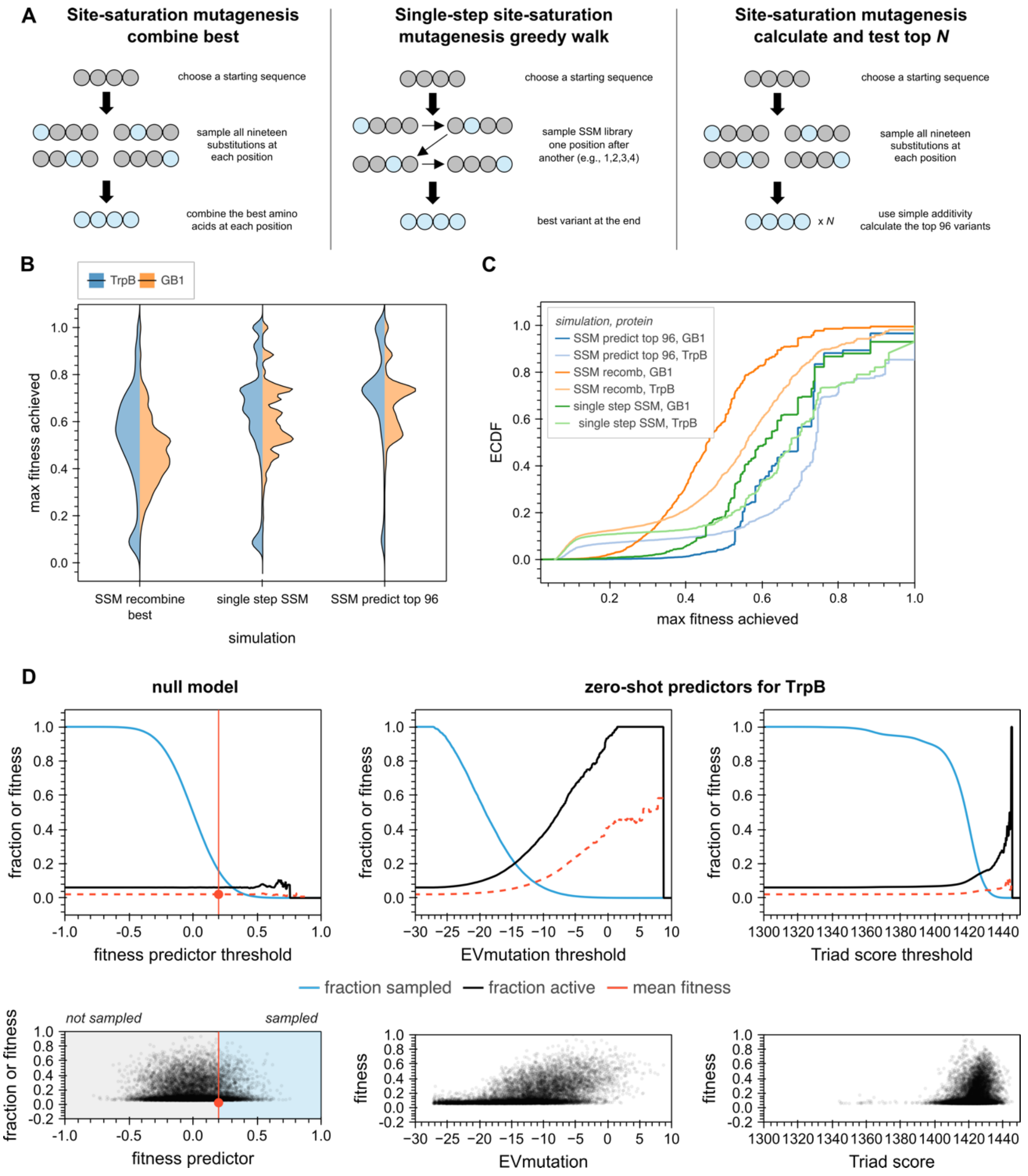


Figure 3-3. Evolutionary constraints in enzyme fitness landscapes. A Three different baselines of directed evolution methodologies. **B** The max fitness achieved from each starting point is plotted as a violin for each of the three directed evolution simulation methodologies. We show the results for both the quadruple-site saturation landscape on TrpB (blue) as well as on a binding landscape for GB1 (orange). **C** An empirical cumulative distribution function of the max fitness achieved from each active variant in the respective landscape for each directed evolution simulation method. Color indicates the simulation method, with TrpB results in the lighter shade and GB1 results in the darker. A left-shifted curve indicates fewer starting variants can achieve a high max fitness. **D** A hypothetical non-predictive model for comparing how using a fitness predictor score as a

threshold can determine the composition of the variants above that threshold. In the non-predictive model, fitness values (full distribution shown in blue, with active variants shown as black points) are normally distributed along the fitness predictor score. As this threshold is decreased, the fraction of the library sampled decreases (blue curve), while the fraction of active variants (black curve) and their mean activity (red curve) within that sample remains constant, since the fitness predictor does not enrich in active variants. **E** The same analysis as in D for the TrpB landscape using EVmutation (an evolutionary-based fitness predictor) and Triad score (a structure-based energy predictor) predictors.

Given the complexity of enzyme catalysis, we suspected that structure-based predictors that work well for a small binding protein[14,34] may not be as useful for an enzyme landscape. Instead, we recognized the abundance of TrpB sequences can be used to generate deep multiple sequence alignments (MSAs), making it potentially more amenable to evolutionary-scale predictors for the fitness effects of substitutions,[35,36] especially since we were assaying the native function of TrpB. We analyzed the enzyme data with both Triad protein design using a Rosetta energy function (Protabit, Pasadena, CA, USA: https://triad.protabit.com), which provides a score that aims to predict stability, and EVmutation,[35] which provides a score that aims to predict the fitness effect of a given set of substitutions based on conservation and evolutionary couplings. As a starting structure for the Triad calculations, we obtained a 2.15-Å resolution structure of *Tm*9D8* (Appendix B, **Sections B.1.20–21**, **Table B-12**). We found that, compared to the null model of zero predictivity (Figure 3-3D) and GB1 (Appendix B, **Figure B-28**), the evolutionary-scale predictor is moderately predictive for TrpB while the stability predictor performs much more poorly (**Figure 3-3E**). Both predictors work much better as classifiers of active and inactive variants and perform more poorly when asked to differentiate amongst the highest fitness variants.

3.2.4 Decoupling stability and activity with a thermostable parent sequence allows an evolutionarily unlikely residue to emerge

As noted above, sequences with K227 dominated the growth assay despite lysine at this site being nearly non-existent across known TrpB-like sequences (Appendix B, **Figure B-14**). This suggested to us that K227 may exert some deleterious effect that is not observed under the assay conditions but is subject to natural selection. For example, it may increase the K_M for indole such that it is not competitive under physiological conditions but works well when indole is added exogenously at 200 μM. Alternatively (or additionally), K227 may be highly destabilizing, but not enough to unfold the thermostable *Tm*TrpB variant at *E. coli* growth temperatures. Indeed, this possibility motivated the use of a thermostable parent sequence at the outset of this study. Therefore, we chose a set of variants to characterize in more depth, including all variants in the single possible path from *Tm*9D8* to the top variant (**Figure 3-2C**), as well the four other variants in the top five (CLKG, ALKG, CIKG, and VLKG), which also all contained K227.

To assess the stability of these variants, we determined the temperature at which a 1 h incubation causes an irreversible 50% reduction in activity as compared to a room temperature incubation (T_{50}) (**Table 3-1**, Appendix B, **Figures B-29, B-30 and B-33**, **Section B.1.18**). The only possible upward path between *Tm*9D8* (VFVS) and AIKG first requires F184I, which exerts no effect on T_{50}, followed by S228G, which imparts a >1 °C increase. From here, two larger decreases in stability come from the remaining two substitutions: a decrease of >7.2 °C from V227K and a decrease of 1.7 °C from V183A to 91.0 °C — a T_{50} about 8.0 °C lower than the starting variant. All five top variants (all of which contain K227) exhibited T_{50} values similar to AIKG (90.1–93.2 °C, or 5.8–8.9 °C below the starting variant). Drops in stability this large would likely lead to loss of function under native conditions, where proteins are typically only marginally more stable than the optimal growth temperature of their host,[37] suggesting why K227 might be absent from the evolutionary record but emerges as highly fit at *E. coli* growth temperatures.

Table 3-1. T_{50} values for selected variants. T_{50} is reported here as the temperature at which a 1 h incubation causes a fitness reduction of 50% as compared to a room temperature incubation.

variant	**T_{50} (°C)**
*Tm*9D8*	99.0 ± 0.6
VIVS	99.0 ± 0.8
VIVG	>100
VIKG	92.8 ± 0.3
AIKG	91.0 ± 0.3
CLKG	90.6 ± 0.3
ALKG	90.1 ± 0.4
CIKG	92.3 ± 0.3
VLKG	93.2 ± 0.4

Finally, we measured the kinetic parameters of a few of these: the starting variant, *Tm*9D8* (VFVS); the best variant (AIKG); and VIVG, the variant with two wild-type reversions and high stability in the middle of the path from *Tm*9D8* to AIKG. All three enzymes were expressed and purified (Appendix B, **Section B.1.19**) for characterization via Michaelis-Menten kinetics (**Table 3-2**, Appendix B, **Figures B-31–33**, **Section B.1.22**). As expected, based on preliminary comparisons of *in vitro* Trp formation and growth rate (Appendix B, **Figure B-3**), we observed that the k_{cat} values for the three enzymes roughly mirrored the fitness values we obtained for them in the high-throughput growth assay: 22.6 min^{-1} for *Tm*9D8*, 51 min^{-1} for VIVG, and 67 min^{-1} for AIKG. We also observed that the two wild-type reversions caused a significant decrease in K_M for indole in VIVG (4.2 μM), while the K_M of AIKG for indole (20 μM) was similar to *Tm*9D8* (23 μM).

Table 3-2. Kinetic parameters for selected variants. Monitoring absorbance at 290 nm over time enabled UV-Spectrometer collection of initial rate data.

	variant		
kinetic parameter	*Tm*9D8* (VFVS)	VIVG	AIKG
k_{cat} (min^{-1})	22.6 ± 0.3	51 ± 1.0	67 ± 1.1
$K_{M,serine}$ (mM)	0.30 ± 0.04	0.18 ± 0.01	0.17 ± 0.02
$K_{M,indole}$ (μM)	23 ± 1.4	4.2 ± 0.6	20 ± 1.5
$k_{cat}/K_{M,indole}$ (M^{-1}s^{-1})	1.44×10^4	2.03×10^5	5.53×10^4

Importantly, while AIKG displays a higher rate of Trp formation at the 200 μM indole concentration used during the growth assay, it reacts more slowly than VIVG at indole concentrations below ~50 μM, which better represents its native conditions. Both AIKG and VIVG had K_M values for Ser roughly half that of *Tm*9D8*, with values of 0.17 and 0.18 mM, respectively, compared to 0.30 mM (**Table 3-2**). These results, coupled with the decrease in stability, help explain how K227 can be nearly non-existent in native TrpB enzymes but optimal in an assay for its native reaction. The observation of effects like this was enabled by the choice of a highly thermostable parent enzyme that can decouple stability and activity.

3.3 Discussion

TrpB is conserved across all domains of life, acting in primary metabolism to perform the final step of Trp biosynthesis. Here we provide a combinatorially complete, 160,000-variant fitness landscape of substitutions at four active site residues of this ubiquitous enzyme, the first landscape of its kind that reports on enzymatic catalysis. The topography of this landscape reflects significant epistasis, which results in many indirect adaptive paths and local optima which can stymie traditional directed evolution methodologies. We expect this landscape to provide a useful testing ground for laboratory and predictive protein engineering approaches as we learn to navigate epistatic, enzymatic fitness landscapes.

The high-throughput fitness measurements are scalar quantities resulting from the aggregate influence of stability, substrate binding, catalytic rate, and environment (e.g., tunable assay conditions and intrinsic host cell conditions) on growth of the bacteria expressing TrpB. The emergence of the destabilizing but activating K227 substitution suggests that catalytic rate is a prominent factor contributing to the calculated fitness values. However, here we characterized only a few of the most active variants; loss of stability could have caused catastrophic loss of fitness for others that were not observed.[38] Combining these measurements with emerging high-throughput stability measurements[39] could disentangle the contributions of stability and activity within the fitness landscape. More drastic changes in K_{M} may also have significantly impacted some of the variants. For example, the top variant, AIKG, has a higher catalytic rate than the wild-type reversion variant, VIVG, at the 200 μM indole used for the assay, but a lower one below 50 μM indole. Assaying such a library again at different substrate concentrations would slightly alter the landscape, which could help deconvolute the specific fitness contributions of K_{M} and k_{cat}. Alternatively, these effects could be examined by characterizing a larger subset of the variants using, for example, high-throughput microfluidic methods.[40]

Navigation of epistatic landscapes is made more efficient by recognizing that the sources of epistasis are diverse, and these non-linear effects can arise due to changes in any one of the myriad factors contributing to fitness. For complex functions such as catalysis there can be more potential sources of epistasis. For example, beneficial substitutions that reduce stability may remain beneficial if they still meet minimal stability requirements, but when combined they could push the protein over the stability threshold and ablate activity altogether.[41–43] Alternatively, enzymatic epistasis can arise via a change in the rate-limiting step.[44] In these cases, predictors tailored to one particular attribute may fail when the fitness effects are due to effects on another attribute. A predictor for k_{cat}, a desirable parameter for enzyme engineering,[45] would fail if an observed deleterious fitness effect were due to stability changes. Likewise, a stability predictor would fail if fitness effects were due to a change in catalytic rate, and evolutionary-scale predictors struggle when an environment differs from the native one, as we show here. Even when each of these predictors can appropriately classify variant effects individually, they may break down in predicting fitness if other effects dominate.

Alone, methods that predict specific variant effects cannot be expected to accurately model a sequence-fitness landscape, especially of a complex enzymatic task, but we can envision a multi-modal approach where different models predict specific facets of enzyme fitness that can be aggregated into a final fitness prediction for a specific set of conditions. These methods might be composed of supervised and semi-supervised data-driven models,[12–14] physics-based approaches, or a combination. This work provides a complex, epistatic landscape for their testing and development. Alongside the existing binding dataset, as well as future datasets to come, we expect high-throughput fitness measurements to play a critical role in generalizable approaches for protein fitness prediction and engineering.

Chapter III Bibliography

1. Packer, M. S. & Liu, D. R. Methods for the directed evolution of proteins. *Nat. Rev. Genet.* **16**, 379–394 (2015).
2. Truppo, M. D. Biocatalysis in the pharmaceutical industry: The need for speed. *ACS Med. Chem. Lett.* **8**, 476–480 (2017).
3. Fowler, D. M. & Fields, S. Deep mutational scanning: A new style of protein science. *Nat. Methods* **11**, 801–807 (2014).
4. Starr, T. N. & Thornton, J. W. Epistasis in protein evolution. *Protein Sci.* **25**, 1204–1218 (2016).
5. Hsu, C., Nisonoff, H., Fannjiang, C. & Listgarten, J. Learning protein fitness models from evolutionary and assay-labeled data. *Nat. Biotechnol.* **40**, 1114–1122 (2022).
6. Miton, C. M., Buda, K. & Tokuriki, N. Epistasis and intramolecular networks in protein evolution. *Curr. Opin. Struct. Biol.* **69**, 160–168 (2021).
7. Jumper, J. *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021).
8. Mirdita, M. *et al.* ColabFold: Making protein folding accessible to all. *Nat. Methods* **19**, 679–682 (2022).
9. Baek, M. *et al.* Accurate prediction of protein structures and interactions using a three-track neural network. *Science* **373**, 871–876 (2021).
10. Lin, Z. *et al.* Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **379**, 1123–1130 (2023).
11. Yang, K. K., Wu, Z. & Arnold, F. H. Machine-learning-guided directed evolution for protein engineering. *Nat. Methods* **16**, 687–694 (2019).

12. Wittmann, B. J., Johnston, K. E., Wu, Z. & Arnold, F. H. Advances in machine learning for directed evolution. *Curr. Opin. Struct. Biol.* **69**, 11–18 (2021).
13. Wu, Z., Kan, S. B. J., Lewis, R. D., Wittmann, B. J. & Arnold, F. H. Machine learning-assisted directed protein evolution with combinatorial libraries. *Proc. Natl. Acad. Sci. U.S.A.* **116**, 8852–8858 (2019).
14. Wittmann, B. J., Yue, Y. & Arnold, F. H. Informed training set design enables efficient machine learning-assisted directed protein evolution. *Cell Syst.* (2021) doi:10.1016/J.CELS.2021.07.008.
15. Qiu, Y., Hu, J. & Wei, G.-W. Cluster learning-assisted directed evolution. *Nat Comput. Sci.* **1**, 809–818 (2021).
16. Olson, C. A., Wu, N. C. & Sun, R. A comprehensive biophysical description of pairwise epistasis throughout an entire protein domain. *Curr. Biol.* **24**, 2643–2651 (2014).
17. Sarkisyan, K. S. *et al.* Local fitness landscape of the green fluorescent protein. *Nature 2015 533:7603* **533**, 397–401 (2016).
18. Pokusaeva, V. O. *et al.* An experimental assay of the interactions of amino acids from orthologous sequences shaping a complex fitness landscape. *PLOS Genetics* **15**, e1008079 (2019).
19. Gonzalez Somermeyer, L. *et al.* Heterogeneity of the GFP fitness landscape and data-driven protein design. *eLife* **11**, e75842 (2022).
20. Dallago, C. *et al.* FLIP: Benchmark tasks in fitness landscape inference for proteins. *bioRxiv* (2022). doi:10.1101/2021.11.09.467890.

21. Weinreich, D. M., Lan, Y., Wylie, C. S. & Heckendorn, R. B. Should evolutionary geneticists worry about higher-order epistasis? *Curr. Opin. Genet. Dev.* **23**, 700–707 (2013).

22. Esteban, L. A. *et al.* HypercubeME: two hundred million combinatorially complete datasets from a single experiment. *Bioinform.* **36**, 1960–1962 (2020).

23. Podgornaia, A. I. & Laub, M. T. Pervasive degeneracy and epistasis in a protein-protein interface. *Science* **347**, 673–677 (2015).

24. Wu, N. C., Dai, L., Olson, C. A., Lloyd-Smith, J. O. & Sun, R. Adaptation in protein fitness landscapes is facilitated by indirect paths. *eLife* **5**, e16965 (2016).

25. Cao, L. *et al.* Design of protein-binding proteins from the target structure alone. *Nature* **605**, 551–560 (2022).

26. Kiss, G., Çelebi-Ölçüm, N., Moretti, R., Baker, D. & Houk, K. N. Computational enzyme design. *Angew. Chem. Int. Ed.* **52**, 5700–5725 (2013).

27. Boville, C. E., Romney, D. K., Almhjell, P. J., Sieben, M. & Arnold, F. H. Improved synthesis of 4-cyanotryptophan and other tryptophan analogues in aqueous solvent using variants of TrpB from *Thermotoga maritima*. *J. Org. Chem.* (2018) doi:10.1021/acs.joc.8b00517.

28. Murciano-Calles, J., Romney, D. K., Brinkmann-Chen, S., Buller, A. R. & Arnold, F. H. A panel of TrpB biocatalysts derived from tryptophan synthase through the transfer of mutations that mimic allosteric activation. *Angew. Chem. Int. Ed.* **55**, 11577–11581 (2016).

29. Rix, G. *et al.* Scalable continuous evolution for the generation of diverse enzyme variants encompassing promiscuous activities. *Nat. Commun.* **11**, (2020).

30. Buller, A. R. *et al.* Directed evolution of the tryptophan synthase β-subunit for stand-alone function recapitulates allosteric activation. *Proc. Natl. Acad. Sci. U.S.A.* **112**, 14599–14604 (2015).

31. Wittmann, B. J., Johnston, K. E., Almhjell, P. J. & Arnold, F. H. evSeq: Cost-effective amplicon sequencing of every variant in a protein library. *ACS Synth. Biol.* **11**, 1313–1324 (2022).

32. Kowalsky, C. A. *et al.* High-resolution sequence-function mapping of full-length proteins. *PLOS ONE* **10**, e0118193 (2015).

33. Park, Y., Metzger, B. P. H. & Thornton, J. W. Epistatic drift causes gradual decay of predictability in protein evolution. *Science* **376**, 823–830 (2022).

34. Nisthal, A., Wang, C. Y., Ary, M. L. & Mayo, S. L. Protein stability engineering insights revealed by domain-wide comprehensive mutagenesis. *Proc. Natl. Acad. Sci. U.S.A.* **116**, 16367–16377 (2019).

35. Hopf, T. A. *et al.* Mutation effects predicted from sequence co-variation. *Nat. Biotechnol.* **35**, 128–135 (2017).

36. Rives, A. *et al.* Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci. U.S.A.* **118**, e2016239118 (2021).

37. Taverna, D. M. & Goldstein, R. A. Why are proteins marginally stable? *Proteins* **46**, 105–109 (2002).

38. Atsavapranee, B., Stark, C. D., Sunden, F., Thompson, S. & Fordyce, P. M. Fundamentals to function: Quantitative and scalable approaches for measuring protein stability. *Cell Syst.* **12**, 547–560 (2021).

39. Tsuboyama, K. *et al.* Mega-scale experimental analysis of protein folding stability in biology and design. *Nature* **620**, 1–11 (2023).

40. Markin, C. J. *et al.* Revealing enzyme functional architecture via high-throughput microfluidic enzyme kinetics. *Science* **373**, eabf8761 (2021).

41. Bloom, J. D., Labthavikul, S. T., Otey, C. R. & Arnold, F. H. Protein stability promotes evolvability. *Proc. Natl. Acad. Sci. U.S.A.* **103**, 5869–5874 (2006).

42. Bershtein, S., Segal, M., Bekerman, R., Tokuriki, N. & Tawfik, D. S. Robustness–epistasis link shapes the fitness landscape of a randomly drifting protein. *Nature* **444**, 929–932 (2006).

43. Tokuriki, N., Stricher, F., Serrano, L. & Tawfik, D. S. How protein stability and new functions trade off. *PLOS Comput. Biol.* **4**, e1000002 (2008).

44. Fröhlich, C. *et al.* Epistasis Arises from Shifting the Rate-Limiting Step during Enzyme Evolution. *bioRxiv* (2023). Preprint at https://doi.org/10.1101/2023.06.29.547057.

45. Li, F. *et al.* Deep learning-based k_{cat} prediction enables improved enzyme-constrained model reconstruction. *Nat. Catal.* **5**, 662–672 (2022).

Appendix A
SUPPLEMENTARY INFORMATION FOR CHAPTER II

A.1 Oligo Design

A.1.1 Inner primer design

The inner primers of evSeq are specific to the region of interest. Each region of interest is captured by both a forward and reverse primer. These primers have the below general layout:

```
F: 5′ - CACCCAAGACCACTCTCCGGXXXXXXX… - 3′
R: 5′ - CGGTGTGCGAAGTAGGTGCXXXXXXXX… - 3′
```

The 5’ region is a universal adapter to which outer primers bind (see **Section A.2.2**, *Preparation of evSeq Barcode Primer Mixes*, below) while the 3’ region (denoted by “**X**” in the primers above) is specific to the region of interest. Note that the length of the variable 3’ region will vary depending on the target gene (this is indicated by the ellipses at the end of the poly-**X** region). Note that there is no need for the two primers in the pair to be equal length—we show them as such to highlight the fact that the forward universal adapter is one base longer than the reverse universal adapter. Detailed instructions for effective primer construction are provided on the evSeq wiki (https://fhalab.github.io/evSeq/1-lib_prep.html#inner-primer-design).

A.1.2 Outer primer design

The barcode (outer) primers used in evSeq all follow the below layout:

```
F: 5′ - TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGXXXXXXXCACCCAAGACCACTCTCCGG - 3′
R: 5′ - GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGXXXXXXXCGGTGTGCGAAGTAGGTGC - 3′
```

Each of these primers consists of (1) a 5' sequence matching the Illumina Nextera transposase adapters, (2) a central unique 7-nucleotide barcode (**Table A-1**), and (3) a 3' universal seed that matches the 5' adapter of the inner primers (see **Section A.1.1**, *Inner Primer Design*, above). Note that only Illumina indices compatible with the Nextera transposase adapters can be used with the provided outer primer designs; other indexing systems would require different adapters. The full set of outer primers used in this study can be found in **Table A-2**; they can be ordered from IDT by following the instructions provided in **Section A.2.1**, *Ordering Barcode Primers from IDT*, below.

A.1.3 Barcode design

evSeq uses 192 unique 7-nucleotide barcodes (**Table A-1**). The barcodes were designed to satisfy the below criteria:

1. All barcodes must have GC-content of 40–60%.
2. All barcodes must be at least 3 substitutions apart. This is to prevent misassignment of reads due to sequencing errors of the barcodes.
3. No barcode can have 3 of the same bases in a row. This is to reduce sequencing errors.
4. No barcode can be a sub-sequence of the Nextera transposase adapters or their reverse complements (see below). This is to avoid interference with downstream Illumina chemistry.
5. No barcode can be a sub-sequence of the Illumina p5 and p7 flow cell-binding sequences or their reverse complements (see below for sequences). Again, this is to avoid interference with downstream Illumina chemistry.

The Nextera transposase adapter sequences are below:

```
5′ - TCGTCGGCAGCGTCAGATGTGTATAAGAGACAG - 3'
```

```
5′ - GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAG - 3'
```

The p5 and p7 flow cell-binding sequences are below:

```
p5: 5′ - AATGATACGGCGACCACCGAGATCTACAC - 3'
p7: 5′ - CAAGCAGAAGACGGCATACGAGAT - 3′
```

A.2 Supplemental Protocols

A.2.1 Ordering barcode primers from IDT

We provide a pre-filled IDT order form for all evSeq primers on the evSeq GitHub repository (https://github.com/fhalab/evSeq/tree/master/lib_prep_tools/IdtOrderForm.xlsx). This order form can be used to order evSeq primers in the 96-well plate layout needed to prepare the evSeq barcode primer mixes (see **Section A.2.2**, *Preparation of evSeq Barcode Primer Mixes*, below). To order evSeq primers:

1. Navigate to the IDT DNA oligo ordering page: https://www.idtdna.com/pages/products/custom-dna-rna/dna-oligos/custom-dna-oligos.
2. Under “Ordering,” select “Plates.”
3. From the “Single-stranded DNA” table, select the amount (in nanomoles) of oligo you wish to order (denoted in the “Product” column) by clicking “Order” under the “96 Well” column. For the work described in this paper, 25 nmol oligos were ordered.
4. On the next page, click “UPLOAD PLATE(S).” Using the pop-up that results, upload the “IdtOrderForm.xls” provided on the evSeq GitHub repository. The pop-up should recognize two plates—one called “FBC” and the other called “RBC”—each consisting of 96 wells. Click “ADD PLATES” followed by “CLOSE THIS WINDOW” to close the window.

5. For the “FBC” plate, click “Plate Specifications.” Confirm that the below specifications are set as follows:
    a. Purification: Standard Desalting
    b. Plate Type: Deep Well
    c. **Ship Option: Wet**
    d. Buffer: IDTE 8.0 pH
    e. Normalization Type: Full Yield
    f. **Concentration: 100 µM**

Note that the bolded specifications are different from default. **While not strictly required, it is strongly recommended that primers be ordered *wet* at 100 µM; reconstituting plates of dry primers to 100 µM can be very time-consuming without robotic support.**

6. Once specifications are correctly set for the “FBC” plate, click “APPLY SETTINGS TO ALL PLATES” at the bottom of the specifications pop-up, followed by “YES” on the window that follows. Quickly check to make sure that the same settings as recommended in step 5 were applied to “RBC” by clicking on the “RBC” “Plate Specifications” option.
7. Add the primers to your order by clicking “ADD TO ORDER,” then follow standard IDT procedures for purchasing.

A.2.2 Preparation of evSeq barcode primer mixes

There are 96 unique forward and 96 unique reverse outer primers (**Table A-2**), corresponding to 96 unique forward and 96 unique reverse barcodes (**Table A-1**). The forward and reverse outer primers were ordered following the procedure given above in **Section A.2.1**, *Ordering Barcode Primers from IDT*.

Each well sequenced in evSeq is encoded by a different combination of forward and reverse barcode. Different primers from the forward and reverse outer primer plates can be mixed together to associate a barcode combination with a specific well in a specific plate. Because the same outer primers can be used regardless of inner primer, it is convenient to keep plates of barcode combinations on hand. Plates of outer primer combinations (hereafter also referred to as "barcode plates") can be stored for long periods of time.

Throughout this work, we used the same 8 barcode plates (consisting of 768 different combinations of forward and reverse outer primers) to encode plate and well locations. Barcode plates are named DI01–DI08, where "DI" stands for "dual-indexed." The exact barcode combinations used by evSeq are given in **Tables A-3–10**; these combinations can also be found in the "index_map.csv" file on the evSeq GitHub (https://github.com/fhalab/evSeq/tree/master/evSeq/util/index_map.csv). By default, the evSeq software assumes the barcode plates used for library preparation are laid out in the order given in the "index_map.csv" file. To build the barcode plates depicted in **Tables A-3–10**, we followed the below procedure:

1. 10-fold dilutions of each of the forward and reverse outer primer plates ordered from IDT were prepared by adding 10 µL of each primer stock to 90 µL ddH2O, keeping the well layout constant. Dilutions were performed in fully-skirted PCR plates (Bio-Rad HSP9601). The plates from IDT had a starting concentration of 100 µM, so the final concentration of these two diluted plates was 10 µM.

2. To 8 fully-skirted PCR plates, 80 µL ddH2O was added, followed by 10 µL diluted (10 µM) forward barcode plate. The well layout was kept constant for the forward barcode primers.

3. To the each of the 8 plates, 10 µL of diluted (10 µM) reverse barcode plate was added, shifting the well layout down by 1 row per plate. For instance, row A of the reverse plate went into row A of the first barcode plate, row B of the second barcode plate, row C of the third barcode plate, and so on; row H of the reverse plate went into row

H of the first barcode plate, row A of the second barcode plate, row B of the third barcode plate, and so on.

4. When not in use, the 10-fold dilutions prepared in step 1 were stored at –20 °C, while the barcode plates (each well of which had a combination of a specific forward and reverse primer at a final concentration of 1 μM) were stored at 4 °C. Both the 10 μM stock plates and 1 μM barcode plates can be stored for long periods of time—we have noticed no drop in effectiveness even after years of storage.

A.2.3 evSeq library preparation/data analysis protocol

The evSeq library preparation protocol was designed to be as cost-effective as possible. The quantities used in the below protocol were chosen to fit within the constraints of the resources available to our research group (these are the quantities used for all evSeq experiments performed in this paper). However, with automation support (e.g., liquid handling robots) and higher-capacity molecular biology equipment, the entire protocol could be scaled down to lower quantities, further improving cost-effectiveness.

The list of steps below can be followed to prepare an evSeq library for sequencing using the outer primers described in **Section A.2.2**, *Preparation of evSeq Barcode Primer Mixes*, above. Note that when first using a new set of inner primers, it is recommended to complete the below protocol for a few wells as a test before deploying them for plate-scale reactions.

The library preparation protocol can be completed with the below steps. Note that provided part numbers are for the materials/reagents we used while developing this protocol—the same components from other providers will almost certainly work as well. This protocol is also provided on the evSeq wiki (https://fhalab.github.io/evSeq/1-lib_prep.html#pcr-protocol).

1. Prepare a PCR master mix for the number of wells to be sequenced according to the below table. Note that we provide an excel calculator on the evSeq GitHub repository for easy calculation of master mix volumes based on the number of plates to be sequenced (https://github.com/fhalab/evSeq/tree/master/lib_prep_tools/MastermixCalculator.xlsx).

Component	Amount per 10 µL rxn (µL)
Thermopol Buffer (NEB B9004S)	1.00
10 mM dNTPs (NEB N0447)	0.20
Taq Polymerase (NEB M0267)	0.05
ddH_2O	5.33
Mol-Bio Grade DMSO (MP 194819)	0.40
Inner Primer Mix (10 µM)	0.02

   a. Note that the above table assumes that each evSeq PCR reaction will be 10 µL—if scaling down, adjust volumes accordingly.

   b. Note that the above table also assumes the same set of inner primers is used to prepare all plates. If this is not the case, a separate master mix will need to be prepared for each set of inner primers.

   c. The Inner Primer Mix (10 µM) is a combination of forward and reverse inner primers at a final concentration of 10 µM each in diH2O (this can be prepared, e.g., by adding 10 µL of 100 µM forward inner primer and 10 µL of 100 µM reverse inner primer to 80 µL diH2O).

2. Add 7 µL of master mix to each well of as many half-skirted PCR plates (USA Scientific 1402-9700) as will be sequenced. These are referred to as “PCR plates” in the remainder of this protocol.

3. Stamp 1 µL of overnight culture from each plate to be sequenced into the PCR plates.

   a. “Stamp” means “apply to all wells, keeping the plate layout consistent”. For example, 1 µL of culture from library 01 F02 is moved to PCR plate 01 F02, 1 µL of culture from library 02 C07 is moved to PCR plate 02 C07, etc.

   b. Note that both fresh culture and previously frozen culture (thawed before use as template) will work here. No modifications need to be made to the protocol.

4. Complete stage 1 PCR using the below thermal cycler conditions. This PCR amplifies the fragment of interest from the template DNA contained in the cell culture.

Step	Temperature (°C)	Time
1	95	5 min
2	95	20 s
3	TD 63-> 54	20 s
4	68	30 s
5	Return to 2, 9 x	
6	4	Hold

   a. "TD" above stands for "touchdown." A touchdown step decrements the temperature by 1 °C each cycle. The touchdown in the above PCR starts at 63 °C and drops to 54 °C by the end.
   b. Note that the extension step (step 4) is long enough to amplify a 500 bp fragment. Longer fragments will need a longer extension time. Note, however, that you may see reduced sequencing efficiency with fragments that are too large.
   c. While developing this protocol, we used the below thermal cycler models:
      i. Eppendorf Mastercycler ep Gradient S Thermal Cycler, Model 5345 with 96-well universal block
      ii. Eppendorf Mastercycler pro S vapo.protect
      iii. Eppendorf Mastercycler X50s 96-well silver block thermal cycler
5. Once PCR has completed, stamp 2 µL of 1 µM barcode primer mix from the barcode plates into the PCR plates (see **Section A.2.2**, *Preparation of evSeq Barcode Primer Mixes*, above, for details on preparation of barcode plates). **Record which barcode plate was stamped into which PCR plate.**
6. Perform the second step PCR using the below conditions:

Step	Temperature (°C)	Time
1	95	20s
2	68	50 s
3	Return to 1, 24 x	
4	68	5 min
5	4	Hold

   a. Again, longer fragments may need a longer extension time.

7. While the second PCR runs, prepare a 2% agarose gel with SYBR gold added (Thermo Fisher Scientific, S11494).
8. Once the second PCR has completed, for each plate, pool 5 µL of each reaction into 100 mM EDTA to a final concentration of 20 mM EDTA—this step quenches the reactions. Pooling will leave you with as many tubes as you have plates, each containing ~600 µL [96 rxns/plate × (5 µL per rxn + 1.25 µL 100mM EDTA per reaction)].
   a. Note: The most efficient way to do the pooling varies depending on the equipment available. Our group relies on 12-channel multichannel pipets for this step, and so will accomplish pooling by (1) adding 10 µL 100 mM EDTA to each well in a single row of a fresh PCR plate, (2) transferring 5 µL reaction from each row in the plate-to-be-pooled into the single row of EDTA, and (3) transferring 40 µL from each well in the single row of pooled reactions using a single-channel pipet (leaving 10 µL dead volume in each well) to a microcentrifuge tube. An alternate strategy might be, for instance, adding 120 µL 100 mM EDTA to a trough, then pipetting 5 µL of all reactions from a plate into this trough. **Whatever strategy is taken, what is important in pooling is that the ratios of the reactions in the pool remain equal—sacrificing some reaction as dead volume is perfectly acceptable to achieve equal mixing in this step.**
9. For each tube made in step 8, take 100 µL of pooled reaction and add it to 20 µL 6x loading dye (NEB B7025S) in a microcentrifuge tube. **It is critical that the loading dye does not contain SDS.** At this point, the remaining pooled reaction from step 8 can be stored at –20 °C for future use (i.e., if the later steps of this protocol ever need to be redone).
   a. Note that most of the pooled reaction is not moved into later steps with this protocol. Again, if relevant automation and molecular biology equipment is available, reactions can be scaled down below 10 µL, reducing wasted reaction. Current reaction sizes are set to minimize pipetting error.

10. Load the contents of each tube made in step 9 into the agarose gel prepared in step 7. The contents of each tube should be kept separate (i.e., loaded into different lanes in the gel). Load a ladder (we typically use 100 bp ladder from NEB, N3231S) in the flanking lanes.
11. Run the agarose gel at 130 V until the bands have sufficiently migrated. Often, you will see two bands: the lower band is usually primer dimer and the upper is the target. Reference the ladder to identify your product, remembering that the two-step PCR adds 120 bp of additional length (from the universal adapter, barcode, and transposase adapters) onto the gene fragment of interest.
12. Gel-extract the target bands from the agarose gel, again keeping bands from different plates separate. We typically use Zymoclean Gel DNA Recovery Kit (Zymo Research, D4001) for this step. Elution should be performed at a low volume—we typically elute in 10 µL of ddH2O.
13. After gel extraction, combine the gel-extracted pools from each plate in equimolar concentrations. We provide a calculator on the evSeq GitHub repository that can be used to normalize *equal-length* fragments to a pre-specified concentration (https://github.com/fhalab/evSeq/tree/master/lib_prep_tools/LibDilCalculator.xlsx).
    a. Note that the quantification here need not be extremely robust. For all results presented in this work, we performed this step using DNA concentrations output by a GE NanoVue Plus.
    b. Tip: It is generally not advised to pool amplicons drastically different in length. Shorter fragments are preferentially sequenced in NGS, and so the shorter amplicon will dominate the number of reads. Separate submissions should be made for libraries with very different lengths.
14. After the previous step, you should have a single tube of cleaned, normalized DNA consisting of all amplicons from all plates to be pooled. This DNA will be submitted to your sequencing provider for inclusion in a multiplexed sequencing run. You should work with your sequencing provider to ensure that all requirements are met to slot into their pipeline. For instance, this protocol assumes that the sequencing provider can add Nextera-compatible Illumina indices and flow-cell-binding

sequences via PCR—it should be confirmed that your sequencing provider can do this before submitting your sample.

   a. Note: Throughout this work, we used the "Customized PCR Amplicon Sequencing" services of Laragen Inc., http://www.laragen.com/laragen_nextgen.php.
   b. Also note that, depending on your sequencing provider, it may be possible (or even necessary) to add the Illumina indices yourself. Again, you should work with your provider to determine the best course of action for submitting evSeq libraries. Adding indices simply requires one final PCR on the pooled evSeq library.

15. Once sequencing is complete, your sequencing provider should return two fastq (or fastq.gz) files to you. One will contain the forward reads for your pooled samples and the other will contain the reverse reads—both files are needed by the evSeq software for processing.
16. Using the files returned in step 15, run the evSeq software to process results and assign variants to their original wells. Detailed instructions on how to use the evSeq software and interpret its outputs are provided on the evSeq Wiki https://fhalab.github.io/evSeq/4-usage.html

A.3 Supplemental Figures

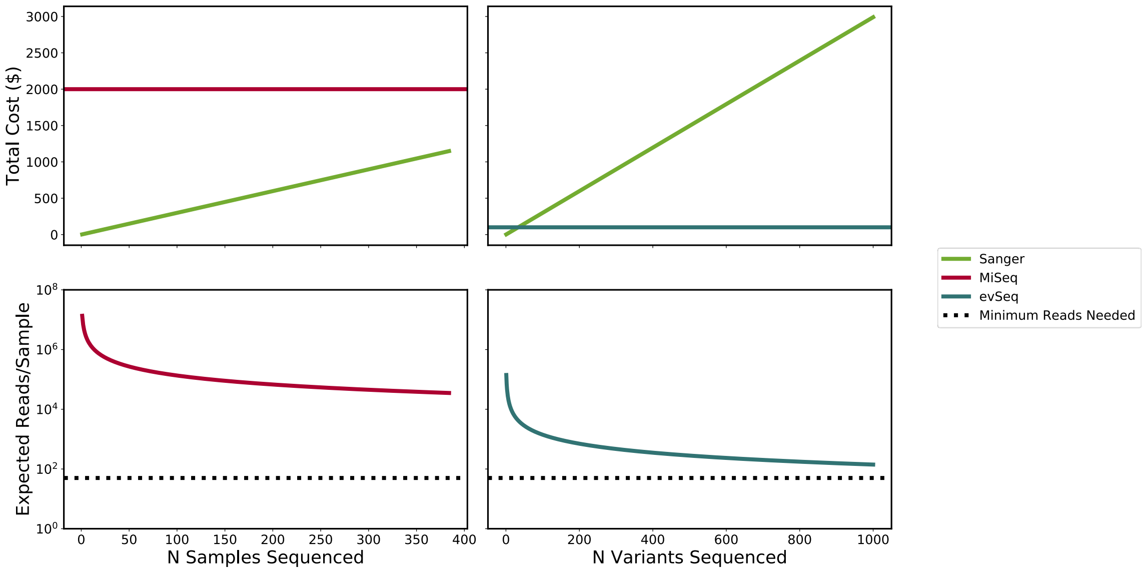


Figure A-1. Comparison of the tradeoff between sequencing depth and cost for Sanger sequencing (green), a multiplexed MiSeq run (red), and an evSeq library (blue). The top row gives the total cost for sequencing a given number of variants; the bottom row gives the expected number of reads per variant for sequencing a given number of variants. Note that the x-axes for the left and right columns are different. The limit on the x-axis for the left column is set to reflect what is typically the maximum level of multiplexed NGS available (384 samples) when outsourcing sequencing. To be consistent with the language used throughout the main text, the x-axis labels refer to elements run in a multiplexed NGS run as "samples" and elements contained in an evSeq library as "variants". We assume that the elements sequenced in these examples are derived from protein mutant libraries amenable to sequencing by evSeq (i.e., the sequenced elements are targeted amplicons). **Top Row:** We see that both multiplexed NGS on a commercial MiSeq run and evSeq have constant cost with an increasing number of elements sequenced; Sanger, in contrast, scales linearly with the number of elements sequenced. Many elements (669 with the cost estimates used to make this figure) need to be added to a multiplexed MiSeq run before it becomes more cost-effective than Sanger. Even though research groups may frequently meet or exceed 669 variants in a standard protein engineering experiment, the flat cost of $2000 is far too high to justify regular sequencing of every variant. Many fewer variants (34) need to be added to an evSeq run before it becomes cost-effective over Sanger. A flat cost of ~$100 is justifiable for regularly sequencing all variants. **Bottom Row:** NGS technologies trade off sequencing depth for cost effectiveness. Notably, the per-sample sequencing depth achieved by commercially available multiplexed runs is much higher than what is needed for reliable sequencing. evSeq, in contrast, more efficiently spreads reads, keeping the expected number of reads closer to, yet still above the minimum needed for effective sequencing. **Notes on Figure Generation:** Cost of a single MiSeq run ($2000) is based on an estimate provided by Laragen Inc. Cost of a single Sanger sequencing run ($2.99) is based on a quote from MCLAB for sequencing a single 96-well plate. The number of expected reads from a MiSeq run (13.5 million) is based on estimates provided by Illumina for a MiSeq Reagent Kit v2 (note that almost double the number of reads can be achieved using a v3 kit—we used v2 here to be conservative with our estimates for NGS/evSeq). The number of expected reads for a variant sampled with evSeq assumes the evSeq library was sequenced as 1 of 96 samples on a multiplexed sequencing run using a MiSeq Reagent Kit v2. The cost of a single evSeq run is based on an estimate provided by Laragen for a single sample in a multiplexed sequencing run using a PE150 kit.

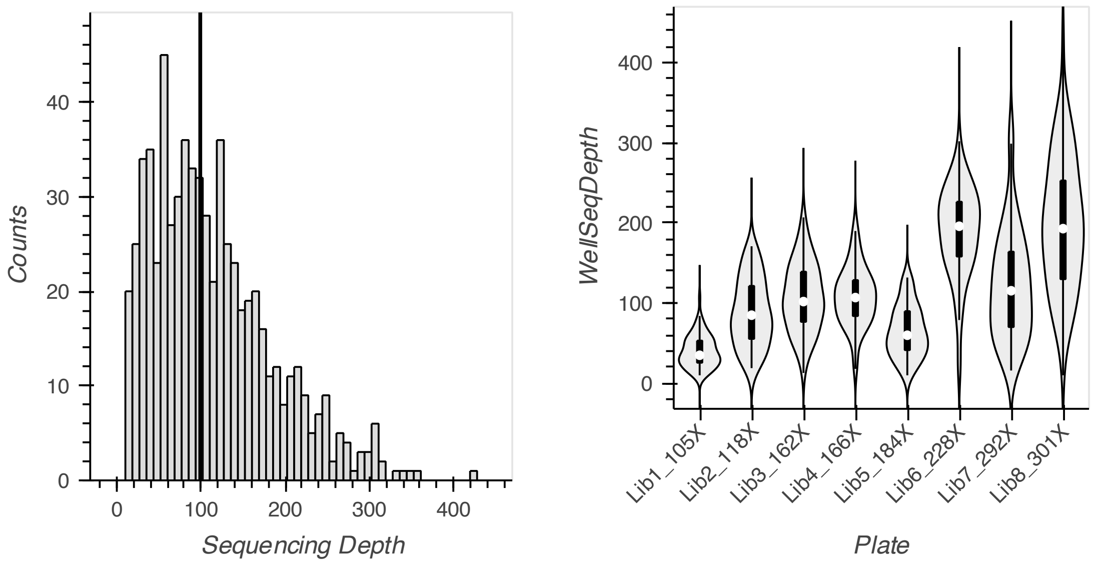


Figure A-2. Sequencing depths for the Tm9D8* evSeq libraries. Left: A histogram of sequencing depths for each Tm9D8* variant contained in the full evSeq library. The vertical black line gives the median. Right: Violin plots showing the distribution of read depths over the wells in each sequenced plate. Variability between plates likely indicates inaccurate quantification of pooled plates prior to final assembly of the evSeq library. Notable, libraries 1-5 use different evSeq primers than libraries 6-8.

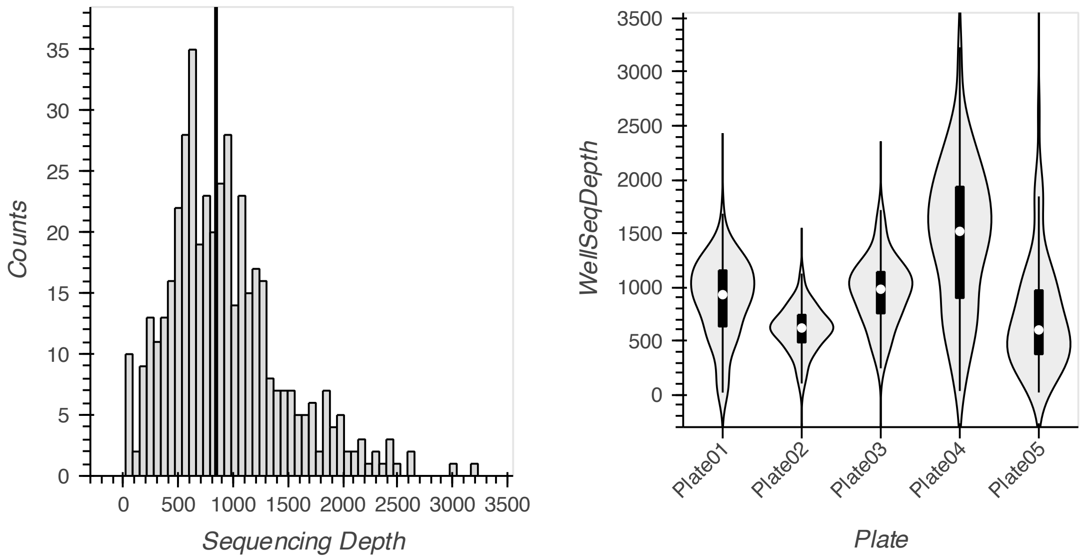


Figure A-3. Sequencing depths for the *Rma*NOD evSeq libraries. Left: A histogram of sequencing depths for each *Rma*NOD variant contained in the full evSeq library. The vertical black line gives the median. **Right:** Violin plots showing the distribution of read depths over the wells in each sequenced plate. Variability between plates likely indicates inaccurate quantification of pooled plates prior to final assembly of the evSeq library.

A.4 Barcode and Outer Primer Sequences

Table A-1. evSeq barcode sequences used in this work. The “Plate” and “Well” columns give the location of these sequences in the IDT order form provided on the evSeq GitHub repository (see **Section A.2.1**, *Ordering Barcode Primers from IDT* and **Section A.1.3**, *Barcode Design*, above). Note that barcode sequences can also be found in the “index_map.csv” file found on the evSeq GitHub repository (https://github.com/fhalab/evSeq/tree/master/evSeq/util/index_map.csv); this csv file also gives the combinations of barcodes used to define the dual indexing (DI) plates.

Plate	Well	Barcode
FBC	A01	GATCATG
FBC	A02	TACATGG
FBC	A03	AAGCACC
FBC	A04	TGGCTCA
FBC	A05	CTTGCTC
FBC	A06	GAAGCGT
FBC	A07	TCTCCAT
FBC	A08	TTGAAGG
FBC	A09	GAATGTC
FBC	A10	ATCTCCA
FBC	A11	GCGTTAT
FBC	A12	TGCACCA
FBC	B01	TGCCTAT
FBC	B02	AGGAATC
FBC	B03	TCCACTG
FBC	B04	TTGTACC
FBC	B05	TTCGAGT
FBC	B06	CTTCAGC
FBC	B07	CAGTGCA
FBC	B08	TGCTGTC
FBC	B09	CGCCATT
FBC	B10	GCCATGA
FBC	B11	CACAACG
FBC	B12	CTTCGCT
FBC	C01	TCGTGAA
FBC	C02	TTATCGG
FBC	C03	AGACCAT
FBC	C04	ACATGAG
FBC	C05	ACGTACT
FBC	C06	CACCTCA
FBC	C07	GTTGGAG
FBC	C08	TGTTCTG
FBC	C09	CTTACGT
FBC	C10	GAGGTTG

FBC	C11	ATGGACA
FBC	C12	ACACTGA
FBC	D01	ATCTGTG
FBC	D02	AATGTGC
FBC	D03	GAGTTGA
FBC	D04	TTCTCAC
FBC	D05	TGAAGCG
FBC	D06	GCTACAA
FBC	D07	AGAGAAC
FBC	D08	CAGAGTG
FBC	D09	TTCCGAA
FBC	D10	GTACGAC
FBC	D11	ACTCTTG
FBC	D12	CCAACCA
FBC	E01	CTCTAGA
FBC	E02	AATCGGA
FBC	E03	CGTCCTA
FBC	E04	GGAATGT
FBC	E05	TCCAAGC
FBC	E06	GCACCTA
FBC	E07	TTGCGTT
FBC	E08	CAGGATT
FBC	E09	CTGCATA
FBC	E10	CGTTGAG
FBC	E11	TGCTACT
FBC	E12	GTGATCC
FBC	F01	GCATGGT
FBC	F02	GTCGTTA
FBC	F03	CCTGACA
FBC	F04	AGTGTAG
FBC	F05	CGAGCAA
FBC	F06	CTACTCC
FBC	F07	GATGCCA
FBC	F08	GACCGAT
FBC	F09	ACGTTGG
FBC	F10	ATGAGCG
FBC	F11	TACTCCG
FBC	F12	GATTCAC
FBC	G01	ATGACGC
FBC	G02	GGTTGTT
FBC	G03	GTACTTG
FBC	G04	TAGCAAG

FBC	G05	CTGCCAT
FBC	G06	GAGAACA
FBC	G07	GTATAGC
FBC	G08	TGATGGA
FBC	G09	GGCAGTA
FBC	G10	GAAGAAG
FBC	G11	AGCGGTT
FBC	G12	TAAGGCC
FBC	H01	AACCTGT
FBC	H02	AGTACAC
FBC	H03	CTCGTAG
FBC	H04	CTAGGTG
FBC	H05	CGATACC
FBC	H06	TCGGCTA
FBC	H07	CGGTTGT
FBC	H08	ATTGCCT
FBC	H09	CATTCGA
FBC	H10	GCACAAT
FBC	H11	GCAGTAA
FBC	H12	CCTAATC
RBC	A01	GAACTGC
RBC	A02	ACCAGGT
RBC	A03	TCTAGAG
RBC	A04	CACACAA
RBC	A05	GTGGAAC
RBC	A06	ATATGCC
RBC	A07	GGTCTGA
RBC	A08	GTGAGAT
RBC	A09	TTGGCAG
RBC	A10	ATGCCTG
RBC	A11	TCCGAAG
RBC	A12	GGCTTAC
RBC	B01	AGTTGGC
RBC	B02	AACGATG
RBC	B03	ACTACCG
RBC	B04	GGTGTCT
RBC	B05	CCAGCTT
RBC	B06	TTAGACG
RBC	B07	ACCATAC
RBC	B08	GACGACT
RBC	B09	GTCACCT
RBC	B10	CGTGATG

RBC	B11	GCTTCCT
RBC	B12	TAGACGT
RBC	C01	CGGACTT
RBC	C02	ACCGGAA
RBC	C03	CCGAAGT
RBC	C04	TCACGCA
RBC	C05	ATCCTCG
RBC	C06	CGAATAG
RBC	C07	TATCCGG
RBC	C08	AGCAAGA
RBC	C09	TGTCGAC
RBC	C10	TTCCATG
RBC	C11	GCAATCG
RBC	C12	TGAGTGG
RBC	D01	TAGGAGA
RBC	D02	AGTCAGT
RBC	D03	GTGCTGT
RBC	D04	CAACAAC
RBC	D05	AATAGCC
RBC	D06	TCTGTGA
RBC	D07	TGTGGTA
RBC	D08	GCGTATG
RBC	D09	AGTTACG
RBC	D10	TTCCTGC
RBC	D11	TATGTCG
RBC	D12	GGAGAGA
RBC	E01	CCTTAGG
RBC	E02	TGTATCC
RBC	E03	CAACCTG
RBC	E04	CTGATGA
RBC	E05	AAGACAG
RBC	E06	AGCTCGT
RBC	E07	GATTGCG
RBC	E08	TCCTTCA
RBC	E09	TCACAGG
RBC	E10	AGAGCTG
RBC	E11	CCTCTGT
RBC	E12	CCTCGAA
RBC	F01	GTGTCTC
RBC	F02	ATTGAGG
RBC	F03	GACAATC
RBC	F04	CACTTGC

RBC	F05	TGAACGC
RBC	F06	CGTAGCA
RBC	F07	AGGTTCC
RBC	F08	GTACACA
RBC	F09	GATAGGT
RBC	F10	TAGCCTC
RBC	F11	TTCAGCC
RBC	F12	GGATTCA
RBC	G01	TGAGCCT
RBC	G02	AACGCGA
RBC	G03	TCATTGC
RBC	G04	AGCATCT
RBC	G05	TTGGTCT
RBC	G06	CAAGGAT
RBC	G07	AGACGTC
RBC	G08	AGGTCAA
RBC	G09	ATGCTAC
RBC	G10	CTCTGAT
RBC	G11	TCAAGTC
RBC	G12	TCGAGCT
RBC	H01	ACAGTCT
RBC	H02	CAGATAC
RBC	H03	TACGTTC
RBC	H04	ACGGTTC
RBC	H05	CATCGTC
RBC	H06	TACGCAT
RBC	H07	CTTAGAC
RBC	H08	AACTGAC
RBC	H09	ACTTGCA
RBC	H10	ACGCGAT
RBC	H11	TCGACAC
RBC	H12	ACTCAAC

Table A-2. Full-length evSeq barcode (outer) primer sequences used in this work. The "Plate" and "Well" columns give the location of these sequences in the IDT order form provided on the evSeq GitHub repository (see **Section A.2.1**, *Ordering Barcode Primers from IDT* and **Section A.2.2**, *Preparation of evSeq Barcode Primer Mixes*, above).

Plate	Well	Sequence
FBC	A01	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGGATCATGCACCCAAGACCACTCTCCGG
FBC	A02	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGTACATGGCACCCAAGACCACTCTCCGG
FBC	A03	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGAAGCACCCACCCAAGACCACTCTCCGG
FBC	A04	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGTGGCTCACACCCAAGACCACTCTCCGG
FBC	A05	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGCTTGCTCCACCCAAGACCACTCTCCGG
FBC	A06	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGGAAGCGTCACCCAAGACCACTCTCCGG
FBC	A07	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGTCTCCATCACCCAAGACCACTCTCCGG
FBC	A08	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGTTGAAGGCACCCAAGACCACTCTCCGG
FBC	A09	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGGAATGTCCACCCAAGACCACTCTCCGG
FBC	A10	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGATCTCCACACCCAAGACCACTCTCCGG
FBC	A11	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGGCGTTATCACCCAAGACCACTCTCCGG
FBC	A12	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGTGCACCACACCCAAGACCACTCTCCGG
FBC	B01	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGTGCCTATCACCCAAGACCACTCTCCGG
FBC	B02	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGAGGAATCCACCCAAGACCACTCTCCGG
FBC	B03	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGTCCACTGCACCCAAGACCACTCTCCGG
FBC	B04	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGTTGTACCCACCCAAGACCACTCTCCGG
FBC	B05	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGTTCGAGTCACCCAAGACCACTCTCCGG
FBC	B06	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGCTTCAGCCACCCAAGACCACTCTCCGG
FBC	B07	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGCAGTGCACACCCAAGACCACTCTCCGG
FBC	B08	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGTGCTGTCCACCCAAGACCACTCTCCGG
FBC	B09	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGCGCCATTCACCCAAGACCACTCTCCGG
FBC	B10	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGGCCATGACACCCAAGACCACTCTCCGG
FBC	B11	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGCACAACGCACCCAAGACCACTCTCCGG
FBC	B12	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGCTTCGCTCACCCAAGACCACTCTCCGG
FBC	C01	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGTCGTGAACACCCAAGACCACTCTCCGG
FBC	C02	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGTTATCGGCACCCAAGACCACTCTCCGG
FBC	C03	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGAGACCATCACCCAAGACCACTCTCCGG
FBC	C04	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGACATGAGCACCCAAGACCACTCTCCGG
FBC	C05	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGACGTACTCACCCAAGACCACTCTCCGG
FBC	C06	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGCACCTCACACCCAAGACCACTCTCCGG
FBC	C07	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGGTTGGAGCACCCAAGACCACTCTCCGG
FBC	C08	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGTGTTCTGCACCCAAGACCACTCTCCGG
FBC	C09	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGCTTACGTCACCCAAGACCACTCTCCGG
FBC	C10	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGGAGGTTGCACCCAAGACCACTCTCCGG
FBC	C11	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGATGGACACACCCAAGACCACTCTCCGG
FBC	C12	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGACACTGACACCCAAGACCACTCTCCGG
FBC	D01	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGATCTGTGCACCCAAGACCACTCTCCGG

FBC	D02	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGAATGTGCCACCCAAGACCACTCTCCGG
FBC	D03	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGGAGTTGACACCCAAGACCACTCTCCGG
FBC	D04	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGTTCTCACCACCCAAGACCACTCTCCGG
FBC	D05	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGTGAAGCGCACCCAAGACCACTCTCCGG
FBC	D06	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGGCTACAACACCCAAGACCACTCTCCGG
FBC	D07	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGAGAGAACCACCCAAGACCACTCTCCGG
FBC	D08	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGCAGAGTGCACCCAAGACCACTCTCCGG
FBC	D09	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGTTCCGAACACCCAAGACCACTCTCCGG
FBC	D10	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGGTACGACCACCCAAGACCACTCTCCGG
FBC	D11	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGACTCTTGCACCCAAGACCACTCTCCGG
FBC	D12	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGCCAACCACACCCAAGACCACTCTCCGG
FBC	E01	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGCTCTAGACACCCAAGACCACTCTCCGG
FBC	E02	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGAATCGGACACCCAAGACCACTCTCCGG
FBC	E03	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGCGTCCTACACCCAAGACCACTCTCCGG
FBC	E04	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGGGAATGTCACCCAAGACCACTCTCCGG
FBC	E05	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGTCCAAGCCACCCAAGACCACTCTCCGG
FBC	E06	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGGCACCTACACCCAAGACCACTCTCCGG
FBC	E07	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGTTGCGTTCACCCAAGACCACTCTCCGG
FBC	E08	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGCAGGATTCACCCAAGACCACTCTCCGG
FBC	E09	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGCTGCATACACCCAAGACCACTCTCCGG
FBC	E10	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGCGTTGAGCACCCAAGACCACTCTCCGG
FBC	E11	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGTGCTACTCACCCAAGACCACTCTCCGG
FBC	E12	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGGTGATCCCACCCAAGACCACTCTCCGG
FBC	F01	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGGCATGGTCACCCAAGACCACTCTCCGG
FBC	F02	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGGTCGTTACACCCAAGACCACTCTCCGG
FBC	F03	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGCCTGACACACCCAAGACCACTCTCCGG
FBC	F04	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGAGTGTAGCACCCAAGACCACTCTCCGG
FBC	F05	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGCGAGCAACACCCAAGACCACTCTCCGG
FBC	F06	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGCTACTCCCACCCAAGACCACTCTCCGG
FBC	F07	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGGATGCCACACCCAAGACCACTCTCCGG
FBC	F08	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGGACCGATCACCCAAGACCACTCTCCGG
FBC	F09	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGACGTTGGCACCCAAGACCACTCTCCGG
FBC	F10	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGATGAGCGCACCCAAGACCACTCTCCGG
FBC	F11	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGTACTCCGCACCCAAGACCACTCTCCGG
FBC	F12	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGGATTCACCACCCAAGACCACTCTCCGG
FBC	G01	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGATGACGCCACCCAAGACCACTCTCCGG
FBC	G02	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGGGTTGTTCACCCAAGACCACTCTCCGG
FBC	G03	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGGTACTTGCACCCAAGACCACTCTCCGG
FBC	G04	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGTAGCAAGCACCCAAGACCACTCTCCGG
FBC	G05	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGCTGCCATCACCCAAGACCACTCTCCGG
FBC	G06	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGGAGAACACACCCAAGACCACTCTCCGG
FBC	G07	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGGTATAGCCACCCAAGACCACTCTCCGG

FBC	G08	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGTGATGGACACCCAAGACCACTCTCCGG
FBC	G09	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGGGCAGTACACCCAAGACCACTCTCCGG
FBC	G10	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGGAAGAAGCACCCAAGACCACTCTCCGG
FBC	G11	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGAGCGGTTCACCCAAGACCACTCTCCGG
FBC	G12	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGTAAGGCCCACCCAAGACCACTCTCCGG
FBC	H01	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGAACCTGTCACCCAAGACCACTCTCCGG
FBC	H02	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGAGTACACCACCCAAGACCACTCTCCGG
FBC	H03	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGCTCGTAGCACCCAAGACCACTCTCCGG
FBC	H04	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGCTAGGTGCACCCAAGACCACTCTCCGG
FBC	H05	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGCGATACCCACCCAAGACCACTCTCCGG
FBC	H06	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGTCGGCTACACCCAAGACCACTCTCCGG
FBC	H07	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGCGGTTGTCACCCAAGACCACTCTCCGG
FBC	H08	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGATTGCCTCACCCAAGACCACTCTCCGG
FBC	H09	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGCATTCGACACCCAAGACCACTCTCCGG
FBC	H10	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGGCACAATCACCCAAGACCACTCTCCGG
FBC	H11	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGGCAGTAACACCCAAGACCACTCTCCGG
FBC	H12	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGCCTAATCCACCCAAGACCACTCTCCGG
RBC	A01	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGGAACTGCCGGTGTGCGAAGTAGGTGC
RBC	A02	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGACCAGGTCGGTGTGCGAAGTAGGTGC
RBC	A03	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTCTAGAGCGGTGTGCGAAGTAGGTGC
RBC	A04	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGCACACAACGGTGTGCGAAGTAGGTGC
RBC	A05	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGGTGGAACCGGTGTGCGAAGTAGGTGC
RBC	A06	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGATATGCCCGGTGTGCGAAGTAGGTGC
RBC	A07	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGGGTCTGACGGTGTGCGAAGTAGGTGC
RBC	A08	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGGTGAGATCGGTGTGCGAAGTAGGTGC
RBC	A09	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTTGGCAGCGGTGTGCGAAGTAGGTGC
RBC	A10	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGATGCCTGCGGTGTGCGAAGTAGGTGC
RBC	A11	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTCCGAAGCGGTGTGCGAAGTAGGTGC
RBC	A12	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGGGCTTACCGGTGTGCGAAGTAGGTGC
RBC	B01	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGAGTTGGCCGGTGTGCGAAGTAGGTGC
RBC	B02	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGAACGATGCGGTGTGCGAAGTAGGTGC
RBC	B03	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGACTACCGCGGTGTGCGAAGTAGGTGC
RBC	B04	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGGGTGTCTCGGTGTGCGAAGTAGGTGC
RBC	B05	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGCCAGCTTCGGTGTGCGAAGTAGGTGC
RBC	B06	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTTAGACGCGGTGTGCGAAGTAGGTGC
RBC	B07	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGACCATACCGGTGTGCGAAGTAGGTGC
RBC	B08	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGGACGACTCGGTGTGCGAAGTAGGTGC
RBC	B09	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGGTCACCTCGGTGTGCGAAGTAGGTGC
RBC	B10	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGCGTGATGCGGTGTGCGAAGTAGGTGC
RBC	B11	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGGCTTCCTCGGTGTGCGAAGTAGGTGC
RBC	B12	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTAGACGTCGGTGTGCGAAGTAGGTGC
RBC	C01	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGCGGACTTCGGTGTGCGAAGTAGGTGC

RBC	C02	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGACCGGAACGGTGTGCGAAGTAGGTGC
RBC	C03	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGCCGAAGTCGGTGTGCGAAGTAGGTGC
RBC	C04	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTCACGCACGGTGTGCGAAGTAGGTGC
RBC	C05	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGATCCTCGCGGTGTGCGAAGTAGGTGC
RBC	C06	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGCGAATAGCGGTGTGCGAAGTAGGTGC
RBC	C07	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTATCCGGCGGTGTGCGAAGTAGGTGC
RBC	C08	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGAGCAAGACGGTGTGCGAAGTAGGTGC
RBC	C09	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTGTCGACCGGTGTGCGAAGTAGGTGC
RBC	C10	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTTCCATGCGGTGTGCGAAGTAGGTGC
RBC	C11	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGGCAATCGCGGTGTGCGAAGTAGGTGC
RBC	C12	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTGAGTGGCGGTGTGCGAAGTAGGTGC
RBC	D01	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTAGGAGACGGTGTGCGAAGTAGGTGC
RBC	D02	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGAGTCAGTCGGTGTGCGAAGTAGGTGC
RBC	D03	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGGTGCTGTCGGTGTGCGAAGTAGGTGC
RBC	D04	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGCAACAACCGGTGTGCGAAGTAGGTGC
RBC	D05	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGAATAGCCCGGTGTGCGAAGTAGGTGC
RBC	D06	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTCTGTGACGGTGTGCGAAGTAGGTGC
RBC	D07	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTGTGGTACGGTGTGCGAAGTAGGTGC
RBC	D08	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGGCGTATGCGGTGTGCGAAGTAGGTGC
RBC	D09	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGAGTTACGCGGTGTGCGAAGTAGGTGC
RBC	D10	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTTCCTGCCGGTGTGCGAAGTAGGTGC
RBC	D11	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTATGTCGCGGTGTGCGAAGTAGGTGC
RBC	D12	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGGGAGAGACGGTGTGCGAAGTAGGTGC
RBC	E01	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGCCTTAGGCGGTGTGCGAAGTAGGTGC
RBC	E02	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTGTATCCCGGTGTGCGAAGTAGGTGC
RBC	E03	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGCAACCTGCGGTGTGCGAAGTAGGTGC
RBC	E04	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGCTGATGACGGTGTGCGAAGTAGGTGC
RBC	E05	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGAAGACAGCGGTGTGCGAAGTAGGTGC
RBC	E06	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGAGCTCGTCGGTGTGCGAAGTAGGTGC
RBC	E07	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGGATTGCGCGGTGTGCGAAGTAGGTGC
RBC	E08	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTCCTTCACGGTGTGCGAAGTAGGTGC
RBC	E09	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTCACAGGCGGTGTGCGAAGTAGGTGC
RBC	E10	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGAGAGCTGCGGTGTGCGAAGTAGGTGC
RBC	E11	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGCCTCTGTCGGTGTGCGAAGTAGGTGC
RBC	E12	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGCCTCGAACGGTGTGCGAAGTAGGTGC
RBC	F01	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGGTGTCTCCGGTGTGCGAAGTAGGTGC
RBC	F02	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGATTGAGGCGGTGTGCGAAGTAGGTGC
RBC	F03	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGGACAATCCGGTGTGCGAAGTAGGTGC
RBC	F04	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGCACTTGCCGGTGTGCGAAGTAGGTGC
RBC	F05	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTGAACGCCGGTGTGCGAAGTAGGTGC
RBC	F06	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGCGTAGCACGGTGTGCGAAGTAGGTGC
RBC	F07	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGAGGTTCCCGGTGTGCGAAGTAGGTGC

RBC	F08	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGGTACACACGGTGTGCGAAGTAGGTGC
RBC	F09	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGGATAGGTCGGTGTGCGAAGTAGGTGC
RBC	F10	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTAGCCTCCGGTGTGCGAAGTAGGTGC
RBC	F11	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTTCAGCCCGGTGTGCGAAGTAGGTGC
RBC	F12	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGGGATTCACGGTGTGCGAAGTAGGTGC
RBC	G01	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTGAGCCTCGGTGTGCGAAGTAGGTGC
RBC	G02	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGAACGCGACGGTGTGCGAAGTAGGTGC
RBC	G03	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTCATTGCCGGTGTGCGAAGTAGGTGC
RBC	G04	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGAGCATCTCGGTGTGCGAAGTAGGTGC
RBC	G05	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTTGGTCTCGGTGTGCGAAGTAGGTGC
RBC	G06	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGCAAGGATCGGTGTGCGAAGTAGGTGC
RBC	G07	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGAGACGTCCGGTGTGCGAAGTAGGTGC
RBC	G08	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGAGGTCAACGGTGTGCGAAGTAGGTGC
RBC	G09	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGATGCTACCGGTGTGCGAAGTAGGTGC
RBC	G10	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGCTCTGATCGGTGTGCGAAGTAGGTGC
RBC	G11	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTCAAGTCCGGTGTGCGAAGTAGGTGC
RBC	G12	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTCGAGCTCGGTGTGCGAAGTAGGTGC
RBC	H01	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGACAGTCTCGGTGTGCGAAGTAGGTGC
RBC	H02	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGCAGATACCGGTGTGCGAAGTAGGTGC
RBC	H03	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTACGTTCCGGTGTGCGAAGTAGGTGC
RBC	H04	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGACGGTTCCGGTGTGCGAAGTAGGTGC
RBC	H05	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGCATCGTCCGGTGTGCGAAGTAGGTGC
RBC	H06	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTACGCATCGGTGTGCGAAGTAGGTGC
RBC	H07	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGCTTAGACCGGTGTGCGAAGTAGGTGC
RBC	H08	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGAACTGACCGGTGTGCGAAGTAGGTGC
RBC	H09	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGACTTGCACGGTGTGCGAAGTAGGTGC
RBC	H10	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGACGCGATCGGTGTGCGAAGTAGGTGC
RBC	H11	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTCGACACCGGTGTGCGAAGTAGGTGC
RBC	H12	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGACTCAACCGGTGTGCGAAGTAGGTGC

A.5 Dual-Indexing Platemaps

This section contains all platemaps for the dual indexing plates (DI plates) used in this study. The tables that follow show how the primers from the forward and reverse barcode plates (**Table A-2**) were arrayed to produce the barcode plates. Each entry in the below platemaps follows the format “Well-Barcode Plate,” where the “-” delimits the plate and well. An “F” after the delimiter indicates that the well preceding the delimiter was from the forward barcode plate (“FBC” in **Table A-2**) and an “R” indicates that the well was from the reverse barcode plate (“RBC”). A detailed protocol for how the dual index plates were produced is given in **Section 2.2.2**, *Preparation of evSeq Barcode Primer Mixes*, above.

Table A-3. Platemap for DI01 used in this study.

DI01	01	02	03	04	05	06	07	08	09	10	11	12
A	A01-F, A01-R	A02-F, A02-R	A03-F, A03-R	A04-F, A04-R	A05-F, A05-R	A06-F, A06-R	A07-F, A07-R	A08-F, A08-R	A09-F, A09-R	A10-F, A10-R	A11-F, A11-R	A12-F, A12-R
B	B01-F, B01-R	B02-F, B02-R	B03-F, B03-R	B04-F, B04-R	B05-F, B05-R	B06-F, B06-R	B07-F, B07-R	B08-F, B08-R	B09-F, B09-R	B10-F, B10-R	B11-F, B11-R	B12-F, B12-R
C	C01-F, C01-R	C02-F, C02-R	C03-F, C03-R	C04-F, C04-R	C05-F, C05-R	C06-F, C06-R	C07-F, C07-R	C08-F, C08-R	C09-F, C09-R	C10-F, C10-R	C11-F, C11-R	C12-F, C12-R
D	D01-F, D01-R	D02-F, D02-R	D03-F, D03-R	D04-F, D04-R	D05-F, D05-R	D06-F, D06-R	D07-F, D07-R	D08-F, D08-R	D09-F, D09-R	D10-F, D10-R	D11-F, D11-R	D12-F, D12-R
E	E01-F, E01-R	E02-F, E02-R	E03-F, E03-R	E04-F, E04-R	E05-F, E05-R	E06-F, E06-R	E07-F, E07-R	E08-F, E08-R	E09-F, E09-R	E10-F, E10-R	E11-F, E11-R	E12-F, E12-R
F	F01-F, F01-R	F02-F, F02-R	F03-F, F03-R	F04-F, F04-R	F05-F, F05-R	F06-F, F06-R	F07-F, F07-R	F08-F, F08-R	F09-F, F09-R	F10-F, F10-R	F11-F, F11-R	F12-F, F12-R
G	G01-F, G01-R	G02-F, G02-R	G03-F, G03-R	G04-F, G04-R	G05-F, G05-R	G06-F, G06-R	G07-F, G07-R	G08-F, G08-R	G09-F, G09-R	G10-F, G10-R	G11-F, G11-R	G12-F, G12-R
H	H01-F, H01-R	H02-F, H02-R	H03-F, H03-R	H04-F, H04-R	H05-F, H05-R	H06-F, H06-R	H07-F, H07-R	H08-F, H08-R	H09-F, H09-R	H10-F, H10-R	H11-F, H11-R	H12-F, H12-R

Table A-4. Platemap for DI02 used in this study.

DI02	01	02	03	04	05	06	07	08	09	10	11	12
A	A01-F, H01-R	A02-F, H02-R	A03-F, H03-R	A04-F, H04-R	A05-F, H05-R	A06-F, H06-R	A07-F, H07-R	A08-F, H08-R	A09-F, H09-R	A10-F, H10-R	A11-F, H11-R	A12-F, H12-R
B	B01-F, A01-R	B02-F, A02-R	B03-F, A03-R	B04-F, A04-R	B05-F, A05-R	B06-F, A06-R	B07-F, A07-R	B08-F, A08-R	B09-F, A09-R	B10-F, A10-R	B11-F, A11-R	B12-F, A12-R
C	C01-F, B01-R	C02-F, B02-R	C03-F, B03-R	C04-F, B04-R	C05-F, B05-R	C06-F, B06-R	C07-F, B07-R	C08-F, B08-R	C09-F, B09-R	C10-F, B10-R	C11-F, B11-R	C12-F, B12-R
D	D01-F, C01-R	D02-F, C02-R	D03-F, C03-R	D04-F, C04-R	D05-F, C05-R	D06-F, C06-R	D07-F, C07-R	D08-F, C08-R	D09-F, C09-R	D10-F, C10-R	D11-F, C11-R	D12-F, C12-R
E	E01-F, D01-R	E02-F, D02-R	E03-F, D03-R	E04-F, D04-R	E05-F, D05-R	E06-F, D06-R	E07-F, D07-R	E08-F, D08-R	E09-F, D09-R	E10-F, D10-R	E11-F, D11-R	E12-F, D12-R
F	F01-F, E01-R	F02-F, E02-R	F03-F, E03-R	F04-F, E04-R	F05-F, E05-R	F06-F, E06-R	F07-F, E07-R	F08-F, E08-R	F09-F, E09-R	F10-F, E10-R	F11-F, E11-R	F12-F, E12-R
G	G01-F, F01-R	G02-F, F02-R	G03-F, F03-R	G04-F, F04-R	G05-F, F05-R	G06-F, F06-R	G07-F, F07-R	G08-F, F08-R	G09-F, F09-R	G10-F, F10-R	G11-F, F11-R	G12-F, F12-R
H	H01-F, G01-R	H02-F, G02-R	H03-F, G03-R	H04-F, G04-R	H05-F, G05-R	H06-F, G06-R	H07-F, G07-R	H08-F, G08-R	H09-F, G09-R	H10-F, G10-R	H11-F, G11-R	H12-F, G12-R

Table A-5. Platemap for DI03 used in this study.

DI03	01	02	03	04	05	06	07	08	09	10	11	12
A	A01-F, G01-R	A02-F, G02-R	A03-F, G03-R	A04-F, G04-R	A05-F, G05-R	A06-F, G06-R	A07-F, G07-R	A08-F, G08-R	A09-F, G09-R	A10-F, G10-R	A11-F, G11-R	A12-F, G12-R
B	B01-F, H01-R	B02-F, H02-R	B03-F, H03-R	B04-F, H04-R	B05-F, H05-R	B06-F, H06-R	B07-F, H07-R	B08-F, H08-R	B09-F, H09-R	B10-F, H10-R	B11-F, H11-R	B12-F, H12-R
C	C01-F, A01-R	C02-F, A02-R	C03-F, A03-R	C04-F, A04-R	C05-F, A05-R	C06-F, A06-R	C07-F, A07-R	C08-F, A08-R	C09-F, A09-R	C10-F, A10-R	C11-F, A11-R	C12-F, A12-R
D	D01-F, B01-R	D02-F, B02-R	D03-F, B03-R	D04-F, B04-R	D05-F, B05-R	D06-F, B06-R	D07-F, B07-R	D08-F, B08-R	D09-F, B09-R	D10-F, B10-R	D11-F, B11-R	D12-F, B12-R
E	E01-F, C01-R	E02-F, C02-R	E03-F, C03-R	E04-F, C04-R	E05-F, C05-R	E06-F, C06-R	E07-F, C07-R	E08-F, C08-R	E09-F, C09-R	E10-F, C10-R	E11-F, C11-R	E12-F, C12-R
F	F01-F, D01-R	F02-F, D02-R	F03-F, D03-R	F04-F, D04-R	F05-F, D05-R	F06-F, D06-R	F07-F, D07-R	F08-F, D08-R	F09-F, D09-R	F10-F, D10-R	F11-F, D11-R	F12-F, D12-R
G	G01-F, E01-R	G02-F, E02-R	G03-F, E03-R	G04-F, E04-R	G05-F, E05-R	G06-F, E06-R	G07-F, E07-R	G08-F, E08-R	G09-F, E09-R	G10-F, E10-R	G11-F, E11-R	G12-F, E12-R
H	H01-F, F01-R	H02-F, F02-R	H03-F, F03-R	H04-F, F04-R	H05-F, F05-R	H06-F, F06-R	H07-F, F07-R	H08-F, F08-R	H09-F, F09-R	H10-F, F10-R	H11-F, F11-R	H12-F, F12-R

Table A-6. Platemap for DI04 used in this study.

DI04	01	02	03	04	05	06	07	08	09	10	11	12
A	A01-F, F01-R	A02-F, F02-R	A03-F, F03-R	A04-F, F04-R	A05-F, F05-R	A06-F, F06-R	A07-F, F07-R	A08-F, F08-R	A09-F, F09-R	A10-F, F10-R	A11-F, F11-R	A12-F, F12-R
B	B01-F, G01-R	B02-F, G02-R	B03-F, G03-R	B04-F, G04-R	B05-F, G05-R	B06-F, G06-R	B07-F, G07-R	B08-F, G08-R	B09-F, G09-R	B10-F, G10-R	B11-F, G11-R	B12-F, G12-R
C	C01-F, H01-R	C02-F, H02-R	C03-F, H03-R	C04-F, H04-R	C05-F, H05-R	C06-F, H06-R	C07-F, H07-R	C08-F, H08-R	C09-F, H09-R	C10-F, H10-R	C11-F, H11-R	C12-F, H12-R
D	D01-F, A01-R	D02-F, A02-R	D03-F, A03-R	D04-F, A04-R	D05-F, A05-R	D06-F, A06-R	D07-F, A07-R	D08-F, A08-R	D09-F, A09-R	D10-F, A10-R	D11-F, A11-R	D12-F, A12-R
E	E01-F, B01-R	E02-F, B02-R	E03-F, B03-R	E04-F, B04-R	E05-F, B05-R	E06-F, B06-R	E07-F, B07-R	E08-F, B08-R	E09-F, B09-R	E10-F, B10-R	E11-F, B11-R	E12-F, B12-R
F	F01-F, C01-R	F02-F, C02-R	F03-F, C03-R	F04-F, C04-R	F05-F, C05-R	F06-F, C06-R	F07-F, C07-R	F08-F, C08-R	F09-F, C09-R	F10-F, C10-R	F11-F, C11-R	F12-F, C12-R
G	G01-F, D01-R	G02-F, D02-R	G03-F, D03-R	G04-F, D04-R	G05-F, D05-R	G06-F, D06-R	G07-F, D07-R	G08-F, D08-R	G09-F, D09-R	G10-F, D10-R	G11-F, D11-R	G12-F, D12-R
H	H01-F, E01-R	H02-F, E02-R	H03-F, E03-R	H04-F, E04-R	H05-F, E05-R	H06-F, E06-R	H07-F, E07-R	H08-F, E08-R	H09-F, E09-R	H10-F, E10-R	H11-F, E11-R	H12-F, E12-R

Table A-7. Platemap for DI05 used in this study.

DI05	01	02	03	04	05	06	07	08	09	10	11	12
A	A01-F, E01-R	A02-F, E02-R	A03-F, E03-R	A04-F, E04-R	A05-F, E05-R	A06-F, E06-R	A07-F, E07-R	A08-F, E08-R	A09-F, E09-R	A10-F, E10-R	A11-F, E11-R	A12-F, E12-R
B	B01-F, F01-R	B02-F, F02-R	B03-F, F03-R	B04-F, F04-R	B05-F, F05-R	B06-F, F06-R	B07-F, F07-R	B08-F, F08-R	B09-F, F09-R	B10-F, F10-R	B11-F, F11-R	B12-F, F12-R
C	C01-F, G01-R	C02-F, G02-R	C03-F, G03-R	C04-F, G04-R	C05-F, G05-R	C06-F, G06-R	C07-F, G07-R	C08-F, G08-R	C09-F, G09-R	C10-F, G10-R	C11-F, G11-R	C12-F, G12-R
D	D01-F, H01-R	D02-F, H02-R	D03-F, H03-R	D04-F, H04-R	D05-F, H05-R	D06-F, H06-R	D07-F, H07-R	D08-F, H08-R	D09-F, H09-R	D10-F, H10-R	D11-F, H11-R	D12-F, H12-R
E	E01-F, A01-R	E02-F, A02-R	E03-F, A03-R	E04-F, A04-R	E05-F, A05-R	E06-F, A06-R	E07-F, A07-R	E08-F, A08-R	E09-F, A09-R	E10-F, A10-R	E11-F, A11-R	E12-F, A12-R
F	F01-F, B01-R	F02-F, B02-R	F03-F, B03-R	F04-F, B04-R	F05-F, B05-R	F06-F, B06-R	F07-F, B07-R	F08-F, B08-R	F09-F, B09-R	F10-F, B10-R	F11-F, B11-R	F12-F, B12-R
G	G01-F, C01-R	G02-F, C02-R	G03-F, C03-R	G04-F, C04-R	G05-F, C05-R	G06-F, C06-R	G07-F, C07-R	G08-F, C08-R	G09-F, C09-R	G10-F, C10-R	G11-F, C11-R	G12-F, C12-R
H	H01-F, D01-R	H02-F, D02-R	H03-F, D03-R	H04-F, D04-R	H05-F, D05-R	H06-F, D06-R	H07-F, D07-R	H08-F, D08-R	H09-F, D09-R	H10-F, D10-R	H11-F, D11-R	H12-F, D12-R

Table A-8. Platemap for DI06 used in this study.

DI06	01	02	03	04	05	06	07	08	09	10	11	12
A	A01-F, D01-R	A02-F, D02-R	A03-F, D03-R	A04-F, D04-R	A05-F, D05-R	A06-F, D06-R	A07-F, D07-R	A08-F, D08-R	A09-F, D09-R	A10-F, D10-R	A11-F, D11-R	A12-F, D12-R
B	B01-F, E01-R	B02-F, E02-R	B03-F, E03-R	B04-F, E04-R	B05-F, E05-R	B06-F, E06-R	B07-F, E07-R	B08-F, E08-R	B09-F, E09-R	B10-F, E10-R	B11-F, E11-R	B12-F, E12-R
C	C01-F, F01-R	C02-F, F02-R	C03-F, F03-R	C04-F, F04-R	C05-F, F05-R	C06-F, F06-R	C07-F, F07-R	C08-F, F08-R	C09-F, F09-R	C10-F, F10-R	C11-F, F11-R	C12-F, F12-R
D	D01-F, G01-R	D02-F, G02-R	D03-F, G03-R	D04-F, G04-R	D05-F, G05-R	D06-F, G06-R	D07-F, G07-R	D08-F, G08-R	D09-F, G09-R	D10-F, G10-R	D11-F, G11-R	D12-F, G12-R
E	E01-F, H01-R	E02-F, H02-R	E03-F, H03-R	E04-F, H04-R	E05-F, H05-R	E06-F, H06-R	E07-F, H07-R	E08-F, H08-R	E09-F, H09-R	E10-F, H10-R	E11-F, H11-R	E12-F, H12-R
F	F01-F, A01-R	F02-F, A02-R	F03-F, A03-R	F04-F, A04-R	F05-F, A05-R	F06-F, A06-R	F07-F, A07-R	F08-F, A08-R	F09-F, A09-R	F10-F, A10-R	F11-F, A11-R	F12-F, A12-R
G	G01-F, B01-R	G02-F, B02-R	G03-F, B03-R	G04-F, B04-R	G05-F, B05-R	G06-F, B06-R	G07-F, B07-R	G08-F, B08-R	G09-F, B09-R	G10-F, B10-R	G11-F, B11-R	G12-F, B12-R
H	H01-F, C01-R	H02-F, C02-R	H03-F, C03-R	H04-F, C04-R	H05-F, C05-R	H06-F, C06-R	H07-F, C07-R	H08-F, C08-R	H09-F, C09-R	H10-F, C10-R	H11-F, C11-R	H12-F, C12-R

Table A-9. Platemap for DI07 used in this study.

DI07	01	02	03	04	05	06	07	08	09	10	11	12
A	A01-F, C01-R	A02-F, C02-R	A03-F, C03-R	A04-F, C04-R	A05-F, C05-R	A06-F, C06-R	A07-F, C07-R	A08-F, C08-R	A09-F, C09-R	A10-F, C10-R	A11-F, C11-R	A12-F, C12-R
B	B01-F, D01-R	B02-F, D02-R	B03-F, D03-R	B04-F, D04-R	B05-F, D05-R	B06-F, D06-R	B07-F, D07-R	B08-F, D08-R	B09-F, D09-R	B10-F, D10-R	B11-F, D11-R	B12-F, D12-R
C	C01-F, E01-R	C02-F, E02-R	C03-F, E03-R	C04-F, E04-R	C05-F, E05-R	C06-F, E06-R	C07-F, E07-R	C08-F, E08-R	C09-F, E09-R	C10-F, E10-R	C11-F, E11-R	C12-F, E12-R
D	D01-F, F01-R	D02-F, F02-R	D03-F, F03-R	D04-F, F04-R	D05-F, F05-R	D06-F, F06-R	D07-F, F07-R	D08-F, F08-R	D09-F, F09-R	D10-F, F10-R	D11-F, F11-R	D12-F, F12-R
E	E01-F, G01-R	E02-F, G02-R	E03-F, G03-R	E04-F, G04-R	E05-F, G05-R	E06-F, G06-R	E07-F, G07-R	E08-F, G08-R	E09-F, G09-R	E10-F, G10-R	E11-F, G11-R	E12-F, G12-R
F	F01-F, H01-R	F02-F, H02-R	F03-F, H03-R	F04-F, H04-R	F05-F, H05-R	F06-F, H06-R	F07-F, H07-R	F08-F, H08-R	F09-F, H09-R	F10-F, H10-R	F11-F, H11-R	F12-F, H12-R
G	G01-F, A01-R	G02-F, A02-R	G03-F, A03-R	G04-F, A04-R	G05-F, A05-R	G06-F, A06-R	G07-F, A07-R	G08-F, A08-R	G09-F, A09-R	G10-F, A10-R	G11-F, A11-R	G12-F, A12-R
H	H01-F, B01-R	H02-F, B02-R	H03-F, B03-R	H04-F, B04-R	H05-F, B05-R	H06-F, B06-R	H07-F, B07-R	H08-F, B08-R	H09-F, B09-R	H10-F, B10-R	H11-F, B11-R	H12-F, B12-R

Table A-10. Platemap for DI08 used in this study.

DI08	01	02	03	04	05	06	07	08	09	10	11	12
A	A01-F, B01-R	A02-F, B02-R	A03-F, B03-R	A04-F, B04-R	A05-F, B05-R	A06-F, B06-R	A07-F, B07-R	A08-F, B08-R	A09-F, B09-R	A10-F, B10-R	A11-F, B11-R	A12-F, B12-R
B	B01-F, C01-R	B02-F, C02-R	B03-F, C03-R	B04-F, C04-R	B05-F, C05-R	B06-F, C06-R	B07-F, C07-R	B08-F, C08-R	B09-F, C09-R	B10-F, C10-R	B11-F, C11-R	B12-F, C12-R
C	C01-F, D01-R	C02-F, D02-R	C03-F, D03-R	C04-F, D04-R	C05-F, D05-R	C06-F, D06-R	C07-F, D07-R	C08-F, D08-R	C09-F, D09-R	C10-F, D10-R	C11-F, D11-R	C12-F, D12-R
D	D01-F, E01-R	D02-F, E02-R	D03-F, E03-R	D04-F, E04-R	D05-F, E05-R	D06-F, E06-R	D07-F, E07-R	D08-F, E08-R	D09-F, E09-R	D10-F, E10-R	D11-F, E11-R	D12-F, E12-R
E	E01-F, F01-R	E02-F, F02-R	E03-F, F03-R	E04-F, F04-R	E05-F, F05-R	E06-F, F06-R	E07-F, F07-R	E08-F, F08-R	E09-F, F09-R	E10-F, F10-R	E11-F, F11-R	E12-F, F12-R
F	F01-F, G01-R	F02-F, G02-R	F03-F, G03-R	F04-F, G04-R	F05-F, G05-R	F06-F, G06-R	F07-F, G07-R	F08-F, G08-R	F09-F, G09-R	F10-F, G10-R	F11-F, G11-R	F12-F, G12-R
G	G01-F, H01-R	G02-F, H02-R	G03-F, H03-R	G04-F, H04-R	G05-F, H05-R	G06-F, H06-R	G07-F, H07-R	G08-F, H08-R	G09-F, H09-R	G10-F, H10-R	G11-F, H11-R	G12-F, H12-R
H	H01-F, A01-R	H02-F, A02-R	H03-F, A03-R	H04-F, A04-R	H05-F, A05-R	H06-F, A06-R	H07-F, A07-R	H08-F, A08-R	H09-F, A09-R	H10-F, A10-R	H11-F, A11-R	H12-F, A12-R

A.6 Supplemental Tables

Table A-11. evSeq captures off-target mutations. This table is derived from the "AminoAcids_Coupled_Max.csv" output file from evSeq for the TrpB run, and shows all confident (defined as ≥0.80 alignment frequency and ≥10 total reads) unexpected mutations captured by evSeq; some columns have been removed. Note in the "VariantCombo" column that the amino acid at the expected mutagenized position has a "?" as the original amino acid—this is because the evSeq run generating this data was told the variable positions with the "NNN" convention. For unexpected variable positions, both the original amino acid and the new amino acid are shown.

IndexPlate	Plate	Well	VariantCombo	AlignmentFrequency	WellSeqDepth
DI02	Lib2_118X	E03	?118V_D164G	0.964286	28
DI04	Lib4_166X	B02	P154S_?166Q	0.977011	87
DI08	Lib8_301X	H11	G250D_?301L	0.99537	216

Table A-12. Primer sequences for TrpB saturation mutagenesis library construction.

Site	Direction	Sequence
105	Forward	GGCAAAACCCGTATCATTGCTNNNACGGGTGCTGGTCAGCAC
105	Reverse	AGCAATGATACGGGTTTTGCCCATTAGTTTTGCCAGCAGAACCTGGC
118	Forward	GGCGTAGCAACTGCTACCNNNGCAGCGCTGTTCGGTATGGAATGTGTAATCTATATGG
118	Reverse	GGTAGCAGTTGCTACGCCGTGCTGACCAGC
162	Forward	GTAAAATCCGGTAGCCGTACCNNNAAAGACGCAATTGACGAAGCTCTG
162	Reverse	GGTACGGCTACCGGATTTTACCGGTACAACTTTAGCACCCAGCAG
166	Forward	CGTACCCTGAAAGACGCANNNGACGAAGCTCTGCGTGACTGGATTACCAACC
166	Reverse	TGCGTCTTTCAGGGTACGGCTACCGGATTTTACCGG
184	Forward	CTGCAGACCACCTATTACGTGNNNGGCTCTGTGGTTGGTCC
184	Reverse	CACGTAATAGGTGGTCTGCAGGTTGGTAATCCAGTCACGCAGAGCT
228	Forward	TACATCGTTGCGTGCGTGNNNGGTGGTTCTAACGCTGCC
228	Reverse	CACGCACGCAACGATGTAGTCCGGCAGACGGCCTTCT
292	Forward	GATGACTGGGGTCAAGTTCAGGTGNNNCACTCCGTCTCCGCTG
292	Reverse	CACCTGAACTTGACCCCAGTCATCCTGCAGAACGAACGTCTTAGAACCG
301	Forward	TCCGCTGGCCTGGACNNNTCCGGTGTCGGTCCGGA
301	Reverse	GTCCAGGCCAGCGGAGACGGAGTGGCTCACCTGAACT

Table A-13. Primers specific to the ampicillin resistance gene of pET22b(+) used in TrpB library construction.

Site	Direction	Sequence
AmpR	Forward	CCAACTTACTTCTGACAACGATCGGAGGACCGAAGGAGCTAACCGCTTTTTTGC
AmpR	Reverse	CGATCGTTGTCAGAAGTAAGTTGGCCGCAGTGTTATCACTCATGGTTATGGCAG

Table A-14. Inner primers used for evSeq library preparation from the TrpB site-saturation mutagenesis libraries.

Name	Direction	Sites	Sequence
evSeq_102_f	Forward	105, 118, 162, 166, 184	CACCCAAGACCACTCTCCGGGCAAAACTAATGGGCAAAACCCG
evSeq_184_r	Reverse	105, 118, 162, 166, 184	CGGTGTGCGAAGTAGGTGCGATGCGGACCAACCACAGAG
evSeq_226_f	Forward	228, 292, 301	CACCCAAGACCACTCTCCGGGCCGGACTACATCGTTGCG
evSeq_304_r	Reverse	228, 292, 301	CGGTGTGCGAAGTAGGTGCCAATAGGCGTGTTCCGGACC

Table A-15. The evSeq barcode plates used for sequencing each position of the TrpB site-saturation mutagenesis libraries.

Position targeted	Barcode plate
105	DI01
118	DI02
162	DI03
166	DI04
184	DI05
228	DI06
292	DI07
301	DI08

Table A-16. Mutagenic primers used for the construction of the *Rma*NOD four-site-saturation library. Note that the names of the primers are delimited by "-" and that the delimited sections reflect the mutagenized positions, the degenerate codons at those positions, and the direction of the primer on the template DNA ([Positions]-[Codon1]-[Codon2]-[Direction]).

Name	Sequence
S28M31-NDT-NDT-F	AAACACTCAGTCGCTATTNDTGCCACGNDTGGTCGGCTGCTTTTCG
S28M31-NDT-VHG-F	AAACACTCAGTCGCTATTNDTGCCACGVHGGGTCGGCTGCTTTTCG
S28M31-NDT-TGG-F	AAACACTCAGTCGCTATTNDTGCCACGTGGGGTCGGCTGCTTTTCG
S28M31-VHG-NDT-F	AAACACTCAGTCGCTATTVHGGCCACGNDTGGTCGGCTGCTTTTCG
S28M31-VHG-VHG-F	AAACACTCAGTCGCTATTVHGGCCACGVHGGGTCGGCTGCTTTTCG
S28M31-VHG-TGG-F	AAACACTCAGTCGCTATTVHGGCCACGTGGGGTCGGCTGCTTTTCG
S28M31-TGG-NDT-F	AAACACTCAGTCGCTATTTGGGCCACGNDTGGTCGGCTGCTTTTCG
S28M31-TGG-VHG-F	AAACACTCAGTCGCTATTTGGGCCACGVHGGGTCGGCTGCTTTTCG
S28M31-TGG-TGG-F	AAACACTCAGTCGCTATTTGGGCCACGTGGGGTCGGCTGCTTTTCG
Q52L56-AHN-AHN-R	GGCCAACAGGGCCGACGCAHNCTTGTGTATAHNTCTCTCAGGAAGTTCAAACAAG
Q52L56-AHN-CDB-R	GGCCAACAGGGCCGACGCAHNCTTGTGTATCDBTCTCTCAGGAAGTTCAAACAAG
Q52L56-AHN-CCA-R	GGCCAACAGGGCCGACGCAHNCTTGTGTATCCATCTCTCAGGAAGTTCAAACAAG
Q52L56-CDB-AHN-R	GGCCAACAGGGCCGACGCCDBCTTGTGTATAHNTCTCTCAGGAAGTTCAAACAAG
Q52L56-CDB-CDB-R	GGCCAACAGGGCCGACGCCDBCTTGTGTATCDBTCTCTCAGGAAGTTCAAACAAG
Q52L56-CDB-CCA-R	GGCCAACAGGGCCGACGCCDBCTTGTGTATCCATCTCTCAGGAAGTTCAAACAAG
Q52L56-CCA-AHN-R	GGCCAACAGGGCCGACGCCCACTTGTGTATAHNTCTCTCAGGAAGTTCAAACAAG
Q52L56-CCA-CDB-R	GGCCAACAGGGCCGACGCCCACTTGTGTATCDBTCTCTCAGGAAGTTCAAACAAG
Q52L56-CCA-CCA-R	GGCCAACAGGGCCGACGCCCACTTGTGTATCCATCTCTCAGGAAGTTCAAACAAG

Table A-17. Additional primers used to build flanking fragments during construction of the four-site-saturation *Rma*NOD library.

Flanking Fragment	Primer Type	Primer Name	Sequence
0	Forward	Universal-F	CCAACTTACTTCTGACAACGATCGGAGGACCGAAGGAGCTAACCGCTTTTTTGC
0	Reverse	S28M31_Const-R	AATAGCGACTGAGTGTTTCTGCAGTGCAGGCAC
1	Forward	L56_Const-F	GCGTCGGCCCTGTTGGCCTACGCCCGTAGTATCGACAACCC
1	Reverse	Universal-R	CGATCGTTGTCAGAAGTAAGTTGGCCGCAGTGTTATCACTCATGGTTATGGCAG

Table A-18. Inner primers used for evSeq library preparation from the *Rma*NOD four-site-saturation mutagenesis library.

Plates	Forward primer	Reverse Primer
All plates	CACCCAAGACCACTCTCCGGCACTGCAGAAACACTCAGTCG	CGGTGTGCGAAGTAGGTGCACTACGGGCGTAGGCCAAC

Table A-19. The evSeq barcode plates used for sequencing each position of the *Rma*NOD four-site-saturation mutagenesis library.

Position targeted	Barcode plate
Plate #1	DI01
Plate #2	DI02
Plate #3	DI03
Plate #4	DI04
Plate #5	DI05

Appendix B
SUPPLEMENTARY INFORMATION FOR CHAPTER III

B.1 General procedures

B.1.1 *Escherichia coli* Trp knockout strain construction

An initial Trp auxotroph strain used in preliminary assays was constructed from the NEB5-α starting strain via λ red-mediated gene replacement.[1] Both *trpA* and *trpB* were deleted and replaced with a Chloramphenicol resistance cassette using primers based on those reported for deletion of *trpA* and *trpB* in the Keio collection. The Chloramphenicol resistance cassette was amplified out of pKD3 with NEB5α_TrpAB::CamR_fwd and NEB5α_TrpAB::CamR_rev and used as the linear DNA for homologous recombination. Gene deletion was confirmed via colony PCR and Sanger sequencing with NEB5α_TrpAB_external_fwd and NEB5α_TrpAB_external_rev. All primers used can be found in Table S1. This strain was used for preliminary single- and double-site saturation for plate-based growth assays and pooled, sequencing-based growth assays.

Due to unusually slow growth, we observed with larger libraries in the NEB5-α Trp auxotroph, we decided to build a new Trp auxotroph strain with Kanamycin resistance to see if this could be remedied. Starting with a K-12 derivative and the parent strain for the Keio collection of single gene knockouts,[2] BW25113, we performed λ red-mediated gene replacement [1]. Both *trpA* and *trpB* were deleted and replaced with a Kanamycin resistance cassette using primers based on those reported for deletion of *trpA* and *trpB* in the Keio collection. The Kanamycin resistance cassette was amplified out of pKD13 with BW25113_TrpAB::KanR_fwd and BW25113_TrpAB::KanR_rev and used as the linear DNA for homologous recombination. Gene deletion was confirmed via colony PCR and

Sanger sequencing with BW25113_TrpAB_external_fwd, BW25113_TrpAB_external_rev, BW25113_TrpAB_internal_fwd, and BW25113_internal_rev. All primers used can be found in Table S1. This strain was used as the host organism for all reported triple- and quadruple-site saturation landscapes from pooled growth assays.

The protocol "Recombineering/Lambda red-mediated gene replacement" from OpenWetWare was used to construct both knockouts following the system from Datsenko & Wanner.[1]

B.1.2 *Tm*9D8* plasmid construction

In vitro assays and protein expression for purification were all performed with a pET22b(+) plasmid harboring *Tm*TrpB genes as previously reported for *Tm*9D8*.[3] For use in the growth assays, we constructed an arabinose-inducible TrpB expression vector with the pBAD24 backbone. pBAD24-sfGFPx1 was a gift from Sankar Adhya & Francisco Malagon (Addgene plasmid # 51558; http://n2t.net/addgene:51558; RRID: Addgene 51558). The gene for *Tm*9D8* was exchanged for sfGFPx1 by amplification of *Tm*9D8* with TrpB_pBAD24_insert_fwd and TrpB_pBAD24_insert_rev and backbone amplification of pBAD24 with TrpB_pBAD24_bb_fwd and TrpB_pBAD24_bb_rev (primers found in Table S3). These pieces were then assembled into a circular plasmid via a two-piece Gibson assembly.[4]

B.1.3 Deep-well plate protein expression

First, to prepare overnight culture plates, *E. coli* colonies harboring TrpB variants in pET22b(+) vectors are picked into separate wells of a 96-well deep-well plate containing 300-500 μL of Luria Broth containing 100 μg/mL carbenicillin (hereafter referred to as

LB_{carb}), covered with a microporous film, and grown overnight at 37 °C, 220 rpm, and 80% humidity for 16–20 h. For expression plates, new deep-well plates are prepared with 630 µL of Terrific Broth containing 100 µg/mL carbenicillin (hereafter referred to as TB_{carb}) and 20 µL of the overnight cultures is dispensed to each well. Expression plates are then incubated at 37 °C, 220 rpm, and 80% humidity for 6 h before addition of 50 µL of 14 mM IPTG in TB_{carb} (final concentration 1 mM). These expression plates were then incubated at 30 °C, 250 rpm, and ambient humidity for 22 h, spun down for 5–10 min at 4500 g (until pellets form and supernatant is clarified). Supernatant was decanted and expression plates were frozen at -20 °C for later use.

B.1.4 Preliminary *in vitro* rate of tryptophan formation assays

Single-site saturation preliminary data were obtained from Wittmann *et al.*[5] and double-site saturation preliminary data were obtained using the same procedure. Sequences were obtained using evSeq.[5]

B.1.5 Preparation of Trp auxotroph electrocompetent cells and electroporation

On day 1, the strain of interest was streaked onto an LB+agar plate containing the requisite antibiotic and grown overnight at 37 °C. On day 2, a single colony is picked into LB containing the requisite antibiotic and grown overnight at 37 °C and 220 rpm. On day 3, the overnight culture is diluted between 50- and 500-fold into Super Optimal Broth (SOB) medium[6] and grown at 18 °C and 220 rpm until OD ~0.4–0.6. Cultures were then plunged into ice-cold water for 10+ minutes until they reached 4 °C and then spun down at 5000*g* for 5 min at 4 °C. The supernatant was decanted, and cells were resuspended in cold, sterile water to 1/5–1/10 the original volume. Cultures were spun down resuspended a second time with the same procedure. One final spin was performed, and cells were resuspended in 1/100 the original volume (100X concentrated). Once prepared, cells were used fresh on the day they were prepared.

Electroporation was performed by combining 50 μL of cells with 1–2 μL of plasmid in a 1 or 2 mM in a chilled electroporation cuvette (USA Scientific, 9104-1050 or 9104-5050) and applying current (BioRad MicroPulser^{TM}, Catalog # 165-2100, Ec1 or Ec3, respectively). Cells were then rescued via resuspension in Super Optimal broth with Catabolite repression (SOB medium with 20 mM glucose: SOC) and incubated for 15–60 min before plating onto LB+agar or transferring to overnight cultures.

B.1.6 Preliminary plate-based independent growth assays

Assay media was composed of 1X M9 Salts (Sigma Aldrich, Catalog # M6030), 0.74 g/L dropout supplement -Trp (Takara Bio Inc., Catalog # 630413), 2 mM $MgSO_4$, 100 μM $CaCl_2$, and 0.4% glycerol (hereafter referred to as Trp DO Media). To prepare this media, M9 salts, dropout supplement -Trp and glycerol were sterile filtered and stored at 4 °C. $MgSO_4$ and $CaCl_2$ 1 M stock solutions were sterilized by autoclaving and added to the media directly before beginning an assay. Antibiotics to select for the Trp auxotroph strain (35 μg/mL kanamycin) and the TrpB-containing plasmid (100 μg/mL carbenicillin) were also added to the media. Using pBAD24-Tm9D8*, arabinose concentrations from 0.001% to 0.1% and indole concentrations from 10 μM to 1000 μM were tested before choosing final concentrations of 0.05% arabinose (stock concentration 20% in M9) and 200 μM indole (stock concentration 500 mM in DMSO). This final mix will be hereafter referred to as Trp-dropout media.

For the assay, electrocompetent Trp auxotroph cells were transformed with the relevant pBAD24-TrpB library and plated onto LB+agar with 35 µg/mL kanamycin and 100 µg/mL carbenicillin. Single colonies were picked into liquid LBcarb,kan culture and grown overnight at 37 °C, 220 rpm, and 80% humidity for 16–20 h. Cultures were diluted 1:20–1:200 into Trp-dropout media into UV-transparent microplates (Caplugs/Evergreen Catalog # 290-8120-0AF) to a total volume of 200 µL. Plates were then incubated at 37 °C and 240 rpm with a 2 mm amplitude in a Tecan® SPARK™. Absorbance at 600 nm was measured every 10 min for 12–48 h to monitor cell growth. Between readings the plate remained covered.

B.1.7 DNA library construction

Libraries were constructed via simultaneous site-saturation mutagenesis using either NNK degenerate primers or the 22c-trick.[7] Unless otherwise stated, the following default PCR mix was used for all reactions, which used the Phusion® High-Fidelity DNA Polymerase according to manufacturer recommendations (New England Biolabs, Catalog # M0530L).

Reagent	Volume (µL)
5x HF Buffer	10
DMSO (100%)	1.5
dNTPs (10 mM)	1
Template	variable (x)
Phusion	0.5
Forward primer (10 µM)	2.5
Reverse primer (10 µM)	2.5
PCR water	32 - x
Total	50

To build the single- and double-site saturation DNA libraries were built with the 22-codon trick[7] using primers in **Table B-2**. Extension time was varied based on fragment length.

Temperature (°C)	Time (s)	Cycles
98	00:30	1
98	00:10	1
55 Ramp speed: 1 °C / s	00:15	
72	variable	
98	00:10	29
63	00:15	
72	variable	
72	05:00	1
10	infinite	1

Each fragment was DpnI digested according to manufacturer directions (New England Biolabs, Catalog # R0176L) and run on a 1% agarose gel containing SYBR™ Gold Nucleic Acid Gel Stain (ThermoFisher Scientific, Catalog # S11494). The relevant bands were excised, and the DNA fragments were purified using a Zymoclean Gel DNA Recovery Kit (Zymo Research, Catalog # D4002). Products were assembled into circular plasmid using NEBuilder® HiFi DNA Assembly (New England Biolabs, Catalog # E2621X) following manufacturer instructions. The resultant product was cleaned and concentrated with a DNA Clean & Concentrator®-5 kit (Zymo Research, Catalog # D4004).

When scaling up to larger DNA libraries composed of more possible sequences, we wanted to construct a relatively uniform input library and reduce the bias for the parent sequence; therefore, we adopted a two-step PCR approach used for both the triple- and quadruple site saturation libraries.

For the triple-site libraries, to produce the first fragment, an inner PCR (termed “gap” PCR) was performed where none of the variable region was included, using the gap primer (F gap) for its respective library (**Table B-5**) along with AmpR_internal_rev (**Table B-2**). The template plasmid used for all libraries was pBAD24-*Tm*9D8* except libraries F and G, which used a 301X plasmid library as the template sequence (prepared following the same method as described for single- and double-site saturation libraries using SSM primers (**Table B-2**). The following thermal cycler protocol was used for all reactions:

Temperature (°C)	Time (s)	Cycles
98	00:30	1
98	00:10	5
55 → 59 (+1 °C / cycle) *ramp speed*: 1 °C / s	00:15	
72	00:45	
98	00:10	25
59	00:15	
72	00:45	
72	10:00	1
10	infinite	1

The resulting PCR product was DpnI digested according to manufacturer directions (New England Biolabs, Catalog # R0176L) and run on a 1% agarose gel containing SYBR^{TM} Gold Nucleic Acid Gel Stain (ThermoFisher Scientific, Catalog # S11494). The relevant bands were excised, and the DNA fragments were purified using a Zymoclean Gel DNA Recovery Kit (Zymo Research, Catalog # D4002). This fragment was then used as template in a second PCR using the same reaction mix and thermal cycler settings but exchanging the "F gap" primer for the "F library" primer. This product was also run on a 1% agarose gel containing SYBR^{TM} Gold Nucleic Acid Gel Stain, excised, and purified with a Zymoclean Gel DNA Recovery Kit.

The second fragment was constructed in a single step using the same reaction mix as used for the first fragment. For these reactions, the forward primer was the AmpR_internal_fwd (**Table B-2**) and the reverse primer was a library specific "R primer" based on **Table B-5**. The thermal cycler conditions used are stated below:

Temperature (°C)	Time (s)	Cycles
98	00:30	1
98	00:10	30
72	00:15	
72	02:15	
72	10:00	1
10	infinite	1

This product was also run on a 1% agarose gel containing SYBR^{TM} Gold Nucleic Acid Gel Stain, excised, and purified with a Zymoclean Gel DNA Recovery Kit.

Due to the design of the quadruple-site saturation library having two sets of two variable sites, the method for building it was slightly different than that used for the triple-site saturation libraries. The same two-step approached was used, where the inner region between the variable regions was first amplified without the regions to be diversified. This PCR used the default PCR mix, "F Gap" and "R Gap" primers from Table B-6, the pBAD24-*Tm*9D8* as template, and the following PCR protocol:

Temperature (°C)	Time (s)	Cycles
98	00:30	1
98	00:10	5
72 → 68 (-1 °C / cycle) *ramp speed*: 1 °C / s	00:15	
72	00:30	
98	00:10	25
68	00:15	
72	00:30	
72	10:00	1
10	infinite	1

The resulting PCR product was DpnI digested according to manufacturer directions (New England Biolabs, Catalog # R0176L) and run on a 1% agarose gel containing SYBR^{TM} Gold Nucleic Acid Gel Stain (ThermoFisher Scientific, Catalog # S11494). The relevant bands were excised, and the DNA fragments were purified using a Zymoclean Gel DNA Recovery Kit (Zymo Research, Catalog # D4002). This fragment was then used as template in a second

PCR using the same reaction mix and thermal cycler settings but using “F Library” and “R Library” from **Table B-6**. The following thermal cycler program was used for amplification:

Temperature (°C)	Time (s)	Cycles
98	00:30	1
98	00:10	5
67 → 63 (-1 °C / cycle) *ramp speed*: 1 °C / s	00:15	
72	00:30	
98	00:10	25
63	00:15	
72	00:30	
72	10:00	1
10	infinite	1

This product was also run on a 1% agarose gel containing SYBR$^{\text{TM}}$ Gold Nucleic Acid Gel Stain, excised, and purified with a Zymoclean Gel DNA Recovery Kit.

The backbone was prepared in two pieces using the AmpR cassette break strategy described previously. The first backbone piece was prepared with the default PCR reaction mix using pBAD24-*Tm*9D8* as template and primers AmpR_internal rev (**Table B-2**) and “F” from **Table B-6**. The following thermal cycler program was used for amplification:

Temperature (°C)	Time (s)	Cycles
98	00:30	1
98	00:10	5
72 → 68 (-1 °C / cycle) *ramp speed*: 1 °C / s	00:15	
72	00:45	
98	00:10	25
69	00:15	
72	00:45	
72	10:00	1
10	infinite	1

The second backbone piece was also prepared with the default PCR reaction mix using pBAD24-*Tm*9D8* as template. The primers used for amplification were AmpR_internal_fwd (**Table B-2**) and "R" from **Table B-6**. The following thermal cycler program was used for amplification:

Temperature (°C)	Time (s)	Cycles
98	00:30	1
98	00:10	5
72 → 68 (-1 °C / cycle) *ramp speed*: 1 °C / s	00:15	
72	02:00	
98	00:10	25
69	00:15	
72	02:00	
72	10:00	1
10	infinite	1

Both backbone products were run on a 1% agarose gel containing SYBR^{TM} Gold Nucleic Acid Gel Stain, excised, and purified with a Zymoclean Gel DNA Recovery Kit.

Once necessary fragments were prepared, DNA concentrations were measured with a GE Healthcare NanoVue^{TM} Plus Spectrophotometer and they were assembled into circular plasmid using NEBuilder® HiFi DNA Assembly (New England Biolabs, Catalog # E2621X) following manufacturer instructions. The resultant product was cleaned and concentrated with a DNA Clean & Concentrator®-5 kit (Zymo Research, Catalog # D4004). To achieve high transformation efficiency for the triple- and quadruple-site libraries, assembled plasmid libraries were first transformed into high-efficiency electrocompetent cells. We used NEB® 10-beta electrocompetent *E. coli* (New England Biolabs Inc., Catalog # C3020K) and the manufacturer recommended protocol. DNA concentration and electroporation settings were optimized to achieve high transformation efficiency. Following application of current, cells were rescued for only 15 minutes in the provided rescue media before being transferred to an overnight culture of LB$_{carb}$ and grown overnight at 37 °C and 220 rpm. Simultaneously, a dilution was plated on LB$_{carb}$+agar to be used to estimate the transformation efficiency and ensure sampling depth. The liquid cultures were miniprepped using the QIAprep Spin Miniprep Kit (Qiagen, Catalog # 27104) to prepare plasmid DNA libraries in high concentration and purity.

B.1.8 Growth-based enrichment assay

To perform the growth-based enrichment assay, electrocompetent Trp auxotroph cells were transformed with the relevant DNA library. After a 1 h rescue, cells were transferred to LB$_{kan,carb}$ and incubated overnight at 37°C and 220 rpm for 16–20 h. At this point, cells were either used directly or diluted 1:1 with sterile 50% glycerol, aliquoted, and frozen at -80°C for later use. Frozen aliquots were prepared by thawing on ice and transferring into $LB_{carb,kan}$ and incubated overnight at 37°C and 220 rpm for 16-20 h. From here, the LB culture was spun down at 5,000 g, the supernatant was decanted, and the pellet was resuspended in Trp-DO media (single- and double-site libraries) or 1X PBS, pH 7.4 (triple- and quadruple-site libraries) (Invitrogen, Catalog # AM9625). The resuspension was once again spun down at 5,000*g* and the supernatant was decanted to remove as much Trp in the solution as possible. The pellets were then resuspended in Trp-DO media to the OD_{600} reported in **Tables B-9–11**.

Cells were incubated in Trp-dropout media at 37 °C and 250 rpm in total volumes of 25 (single-, double-, and triple-site libraries) or 50 mL (quadruple-site library) and 1.5 mL samples were collected at each timepoint seen in **Tables B-9–11**. These samples were centrifuged at 5,000*g* for 5 minutes and stored at -20 °C until further use and sequencing preparation.

B.1.9 Sequencing library preparation

All libraries were prepared for sequencing with the same overall strategy using inner primers from **Table B-7**. Mapping of these primers to the libraries can be found in **Table B-8**. First, an initial two-cycle PCR amplification attached inner handles for the Illumina barcodes using the default PCR mix and the following thermocycler program:

Step	Temperature (°C)	Ramp rate	Time	Cycles
Initial denaturation	98	max	3 min	1
Denaturation	98	max	30 sec	2

Anneal start temp	64	max	1 sec	
Anneal slow ramp	58	0.2 C/s	90 sec	
Extension	72	max	90 sec	
Final extension	72	max	5 min	1
Hold	4	-	-	-

These samples were then digested with ExoCIP (New England Biolabs, Catalog # E1050L) using a 20 min incubation at 37 °C followed by a 15 min inactivation step at 80 °C. The resulting product was used as template for a second PCR using IDT® for Illumina® DNA/RNA UD Indexes Set A, Tagmentation (Illumina, Catalog # 20027213).

Reagent	Volume (µL)
PCR water	1.25
5X KAPA HiFi	5
10 mM dNTP	0.75
UDP Primer Mix	5
DNA eluate	12.5
KAPA HiFi Polymerase	0.5
Total	25

Samples were then DpnI digested according to manufacturer directions (New England Biolabs, Catalog # R0176L) and purified via magnetic bead cleanup using Agencourt AMPure XP (Beckman Coulter, Catalog # A63880) according to manufacturer recommendations. Sample concentrations were measured using Quant-iT™ PicoGreen™ (ThermoFisher Scientific, Invitrogen, Catalog # P7581) and pooled equimolarly for submission to high-throughput sequencing with an Illumina HiSeq2500.

B.1.10 Sequencing data pre-processing

Processing of the sequencing data was performed to provide filtered and aligned data for determining accurate, sequence-dependent codon/amino acid counts. Forward and reverse reads were filtered independently using fastq-filter (https://github.com/LUMC/fastq-filter) with an average read quality (option –q) of 25 (or an error rate of 0.00316). Corresponding forward and reverse reads were then matched, retaining only pairs of reads that passed both filters. The first 13 bases of each read were trimmed to remove sequence- and experiment-specific identifiers. Pairs of files were then aligned using minimap2 (https://github.com/lh3/minimap2) using the following process: `minimap2 –ax sr ref.fasta forward.fastq reverse.fastq -k 5 -w 3`, where ref.fasta is a fasta file containing the Tm9D8* (parent) reference sequence, forward.fastq is the filtered and trimmed forward fastq file, and reverse.fastq is the matching filtered and trimmed reverse fastq file. The option -w 3 was used due to the trimmed reads being short for the four-site library (38 bp) to provide a sufficiently small window for proper alignment. (The -k 5 option is based on the standard ratio of k/w ~1.5.) Aligned reads were then filtered based on the following criteria: both reads must align with no indels, starting at the expected position (the starting base of the trimmed forward read in the Tm9D8* sequence), with the expected total length for each aligned forward+reverse read. (Specific values for each library are reported in the processing scripts.) Codon identities for each position were then indexed from these filtered and aligned reads for determining fitness. Python scripts and documentation can be found on the associated GitHub.

B.1.11 Fitness score calculations

Fitness calculations were determined based on theory proposed by Kowalsky *et al.* to obtain specific growth rates for each variant.[8] For each time point captured, a specific growth rate, μ_i, was calculated for each unique amino acid sequence as follows:

$$\mu_i = \ln\left(\frac{x_{fi}}{x_{0i}}\right)\frac{1}{t}$$

where x_{0i} and x_{fi} represent the concentration of E. coli harboring the given amino acid sequence i in the initial population and the population at time t, respectively. These values were calculated based on the OD_{600} of the culture and the frequency of sequence in each population. We observed that sequences containing stop codons, which can be presumed non-functional, had slightly non-zero μ_i values. Therefore, we subtracted the average μ_{stop} from each μ_i. We then scaled everything to the maximum μ value for that timepoint by dividing by $\mu_{\max}$.

$$\text{fitness} = \frac{\mu_i - \mu_{stop}}{\mu_{max}}$$

Fitness values were then calculated for each timepoint in a landscape were then averaged for each replicate. The active/inactive threshold was imposed at this point by enforcing that the fitness for a variant in each replicate was greater than the 95% confidence interval of the stop codon-containing variant distribution. Finally, the fitness values for the two replicates were averaged together to obtain a final fitness metric for each sequence. Timepoints were omitted when sequencing showed poor agreement between replicates. Missing variants were imputed with the KNN imputer from sklearn using weights='distance' and n_neighbors=2 for the

TrpB data (**Figure B-13**) and the imputed scores reported previously by the authors were used for GB1.[9]

B.1.12 Pairwise epistasis calculations and analyses

Fraction of pairwise epistasis was calculated using python and functions that classify each type into one of the three categories: magnitude, sign, and reciprocal sign epistasis. Notably, for this analysis additive effects were grouped into magnitude. For each unique starting variant (**00**), all possible double substitutions (**11**) were tested such that all variants within the set of **00**, **01**, **10**, and **11** were active and an epistasis type was assigned. Doing this for all possible double substitutions, we computed the fraction of each type of epistasis for the starting variant. These results were sorted into quartiles based on the fitness values of the starting variant to create the distributions.

B.1.13 Determination of local optima

Local optima were determined by looking at all non-imputed variants classified as active. For each of these variants, all single substitutions were made *in silico*. If no single-substitution variant had a higher fitness than the original, that original variant was classified as a local optimum. To determine if two simultaneous substitutions could enable escape from the local optimum, all double substitutions were made *in silico*. If at least one double-substitution variant had a higher fitness than the original, that original variant was said to be able to escape the local optimum via double-site saturation mutagenesis. Code can be found in the associated GitHub repository.

B.1.14 Path analyses

For each active variant, networkx[10] was used to construct a directed graph from that variant to the best variant in the landscape, AIKG. For this analysis, any fitness below zero was set to zero. No imputed variants were used as starting points, but they were used as intermediate variants when necessary, so that no graphs had missing nodes. The number of direct paths possible to the path were counted allowing downward steps ranging from 0% to 90% decrease in fitness. The same analyses were run for each of the top twenty local optima.

B.1.15 Simulations of directed evolution

Starting from every active variant, directed evolution methodologies were simulated using Python functions that can be found in the associated GitHub repository.

Site-saturation mutagenesis combine best

For every variant classified as active, all nineteen substitutions are made *in silico* at each of the four positions independently in the background of the initial sequence. The best amino acid at each position is obtained and the sequence consisting of these amino acids is built. The best variant from among the initial sequence, all single-site mutagenesis variants, and the recombined variant is reported as the maximum fitness achieved.

Single-step site-saturation mutagenesis greedy walk

For every variant classified as active and each possible order of sampling positions ($M!$ where M = number of positions) site-saturation mutagenesis is performed iteratively *in silico* with N rounds. For the first position, all nineteen substitutions are tested for that position in the starting background of the other positions. Once the best residue for that position is determined, the next position targeted, the best residue is fixed at that position, and so on until all positions have been targeted once. The fitness of the final variant is reported as the max fitness achieved.

Site-saturation mutagenesis calculate and test top N

For every variant classified as active, all nineteen substitutions are made *in silico* at each of the four positions independently in the background of the initial sequence. Using these M x 19 + 1 (starting variant) datapoints, fitness scores for all 20^M possible combinations are calculated as the product of the fold-change for each single substitution over the initial sequence. These sequences are then ranked, and the best N are tested *in silico*. The max fitness achieved is reported as the maximum fitness of the initial sequence, any of the single substitutions, and the top N predicted sequences.

B.1.16 Generating zero-shot scores

Using the methods from Wittmann *et al.*[11] we obtained zero-shot scores using both Triad estimates of ΔΔG and EVmutation.[12] For Triad, the crystal structure obtained here was used as the starting point for the calculations and EVmutation used *Tm*9D8* as the starting sequence.

B.1.17 Site-directed mutagenesis to construct variants for in-depth biochemical characterization

Using a template plasmid of *Tm*9D8* in pET22b(+), primers were ordered to make exact mutations at positions 183, 184, 227, and 228. Full gene sequences are provided in **Table B-3**.

B.1.18 *T_{50}* measurements

Using the plasmids prepared via site-directed mutagenesis, variants were expressed as described in **Section B 1.3** "*Deep-well plate protein expression*" with six biological replicate wells for each variant. Frozen pellets were fully thawed at room temperature and then resuspended by light vortexing in lysis buffer composed of 1 mg/mL lysozyme, 0.1X Bug Buster®, 0.2 mg/mL DNase I, and 200 μM PLP in 50 mM KPi. Plates were incubated at 37 °C, 220 rpm, and 80% humidity for 1 h and then spun down at 4500*g* for 10 min. Clarified lysate for each variant was pooled into individual 15-mL conical centrifuge tubes and stored at 4 °C until needed.

For heat treatments, 40 μL of clarified lysate was aliquoted into full-skirted PCR plates and incubated at the reported temperature for 1 h using a gradient on a Mastercycler® X50s (Eppendorf, Catalog # 6311000010). Room temperature controls were incubated for 1 h on the benchtop in 200-μL PCR tubes. Room temperature controls were then added to the PCR plate and all samples were spun down for 8 min at 4000*g* to remove accumulated debris.

B.1.19 Enzyme purification

Since all enzymes displayed T_{50} measurements much greater than 75 °C, enzyme was purified as described in Boville *et al.*[3] apart from the addition of 200 μM PLP and 1 mg/mL lysozyme in the lysis buffer. Protein concentrations were obtained with the Pierce BCA Protein Assay Kit (ThermoFisher Scientific, Catalog # 23225) and purified protein was frozen in aliquots on dry ice or liquid nitrogen.

B.1.20 *Tm*9D8* crystallization

For the crystallization of tryptophan synthase variant *Tm*9D8*, protein was purified as described above. Due to the similarity between this enzyme and previously described tyrosine synthase variants, the same precipitant (1.2 M NaH_2PO_4/0.8 M K_2HPO_4, 0.1 M *N*-cyclohexyl-3-aminopropanesulfonic acid (CAPS), 0.2 M Li_2SO_4) was used to crystallize this variant. In a 24-well CrysChem M Plate (Hampton Research), 2 mg/mL protein was screened using 1–6 µL protein drops and 2–5 µL precipitant drops. While small salt crystals were observed in drops containing higher initial precipitant concentrations, drops with a higher protein concentration remained clear after 6 days. At this point, these drops (5–6 µL 2 mg/mL *Tm*9D8* + 2 µL precipitant) were streak seeded using a cat whisker, the generous gift of Crick Boville, and a seed stock derived from crystals of the related tyrosine synthase variant *Tm*TyrS1 (9 mutations) (PDB 8EGY). Within 2 days, small hexagonal prism crystals formed in these wells.

To prepare samples for x-ray diffraction experiments, a cryoprotectant solution was prepared by mixing 80 µL of equilibrated reservoir solution with 20 µL of ethylene glycol. This solution was then added to the crystal drop, sequentially adding and removing equivalent volumes until no schlieren was observed. Following cryoprotection, all crystals were mounted in nylon loop, cooled in liquid nitrogen, and stored prior to data collection.

B.1.21 *Tm*9D8* crystal structure determination

Diffraction data were collected at the Stanford Synchrotron Radiation Laboratory (SSRL) beamline 12-2. Data reduction and integration were carried out using XDS[13] and scaled using Aimless in the CCP4 suite of programs.[14] Molecular replacement (MR) was performed using the structure of a holo *Tm*TyrS1 (PDB 8EGY) as a search model in Phaser.[15] Model building and modification in the electron density was performed using Coot and structure refinement was performed using Phenix.[16,17] Other ligands, including free PLP, as well as water molecules and ethylene glycol were added during later stages of refinement. Occasionally, spurious electron density peaks were present in the active site, dimer interface, and COMM domain that could not be unambiguously modeled by alternative protein conformations, solvent, or other additives applied during the procedure, so these were left uninterpreted. The quality of the final models was evaluated with MolProbity and PROCHECK.[18,19] Data collection and refinement statistics are presented in **Table B-12**.

B.1.22 Determination of kinetic parameters

Enzyme parameters, including K_M and k_{cat}, were determined via Michaelis-Menten kinetics by collecting 290 nm absorbance continuously over 500 seconds with a UV-vis spectrophotometer (Shimadzu, Catalog # EW-83400-20) for reactions containing enzyme (62.5–250 nM), indole (1.5625–500 µM), Ser (0.05–20 mM), and DMSO (4%) in KPi. Indole K_M was collected at 20 mM Ser and Ser K_M was collected at 200 µM indole (the concentration used for the growth rate assay). Initial rates were obtained using linear or exponential fits of the data within a time frame not impacted by burst phase kinetics. These rates were fit with a Michaelis-Menten model to obtain estimates for K_M and k_{cat}.

B.2 Supplemental Tables

Table B-1. Primers for knockout strain building and verification

Name	Sequence (5’ → 3’)
NEB5α_TrpAB::CamR_fwd	TGCCGCCAGCGGAACTGGCGGCTGTGGGATTAACTGCGCGTCGCCGCTTTGTGTAGGCTGGAGCTGCTTC
NEB5α_TrpAB::CamR_rev	TTGGCCTCGGTTTTCCAGACGCTGCGCGCATATTAAGGAAAGGAACAATGATGGGAATTAGCCATGGTCC
NEB5α_TrpAB_external_fwd	TGCCGCCAGCGGAACTGGC
NEB5α_TrpAB_external_rev	TCAAAGACGCACGTCTTTTGGCCTCGG
BW25113_TrpAB::KanR_fwd	TGCCGCCAGCGGAACTGGCGGCTGTGGGATTAACTGCGCGTCGCCGCTTTTGTAGGCTGGAGCTGCTTCG
BW25113_TrpAB::KanR_rev	TTGGCCTCGGTTTTCCAGACGCTGCGCGCATATTAAGGAAAGGAACAATGATTCCGGGGATCCGTCGACC
BW25113_TrpAB_external_fwd	GGTAAGCGAAACGGTAAAAAGATAAATATTAAATGAATTTAGG
BW25113_TrpAB_external_rev	GCGCCGGACTTGATTTTAATTCTGC
BW25113_TrpAB_internal_fwd	GCCCAGTCATAGCCGAATAGCC
BW25113_TrpAB_internal_rev	GGCTATTCGGCTATGACTGGGC

Table B-2. Primers for single and double-site saturation mutagenesis for preliminary assays.

Name	Sequence (5’ → 3’)
Tm9D8*_184X_fwd	CTGCAGACCACCTATTACGTG**XXX**GGCTCTGTGGTTGGTCC
Tm9D8*_118X_fwd	GGCGTAGCAACTGCTACC**XXX**GCAGCGCTGTTCGGTATGGAATGTGTAATCTATATGG
Tm9D8*_184_rev	CACGTAATAGGTGGTCTGCAGGTTGGTAATCCAGTCACGCAGAGCT
Tm9D8*_118_rev	GGTAGCAGTTGCTACGCCGTGCTGACCAGC
Tm9D8*_301X_fwd	TCCGCTGGCCTGGAC**XXX**TCCGGTGTCGGTCCGGA
Tm9D8*_301_rev	GTCCAGGCCAGCGGAGACGGAGTGGCTCACCTGAACT
AmpR_internal_fwd	CCAACTTACTTCTGACAACGATCGGAGGACCGAAGGAGCTAACCGCTTTTTTGC
AmpR_internal_rev	CGATCGTTGTCAGAAGTAAGTTGGCCGCAGTGTTATCACTCATGGTTATGGCAG

XXX = mix of three primers with NDT, VHG, or TGG at that position mixed in a ratio of 12:9:1. This is based on methods presented by Kille *et al.*[7]

Table B-3. TrpB sequences

Name	Sequence
*Tm*9D8* in pBAD24: 5276	ATGAAAGGCTACTTCGGTCCGTACGGTGGCCAGTACGTGCCGGAAATCCTGATGGGAGCTCTGGAAGAACTGGAAGCTGCGTACGAAGGAATCATGAAAGATGAGTCTTTCTGGAAAGAATTCAATGACCTGCTGCGCGATTATGCGGGTCGTCCGACTCCGCTGTACTTCGCACGTCGTCTGTCCGAAAAATACGGTGCTCGCGTATATCTGAAACGTGAAGACCTGCTGCATACTGGTGCGCATAAAATCAATAACGCTATCGGCCAGGTTCTGCTGGCAAAACTAATGGGCAAAACCCGTATCATTGCTGAAACGGGTGCTGGTCAGCACGGCGTAGCAACTGCTACCGCAGCAGCGCTGTTCGGTATGGAATGTGTAATCTATATGGGCGAAGAAGACACGATCCGCCAGAAACTAAACGTTGAACGTATGAAACTGCTGGGTGCTAAAGTTGTACCGGTAAAATCCGGTAGCCGTACCCTGAAAGACGCAATTGACGAAGCTCTGCGTGACTGGATTACCAACCTGCAGACCACCTATTACGTGTTCGGCTCTGTGGTTGGTCCGCATCCATATCCGATTATCGTACGTAACTTCCAAAAGGTTATCGGCGAAGAGACCAAAAAACAGATTCCAGAAAAAGAAGGCCGTCTGCCGGACTACATCGTTGCGTGCGTGAGCGGTGGTTCTAACGCTGCCGGTATCTTCTATCCGTTTATCGATTCTGGTGTGAAGCTGATCGGCGTAGAAGCCGGTGGCGAAGGTCTGGAAACCGGTAAACATGCGGCTTCTCTGCTGAAAGGTAAAATCGGCTACCTGCACGGTTCTAAGACGTTCGTTCTGCAGGATGACTGGGGTCAAGTTCAGGTGAGCCACTCCGTCTCCGCTGGCCTGGACTACTCCGGTGTCGGTCCGGAACACGCCTATTGGCGTGAGACCGGTAAAGTGCTGTACGATGCTGTGACCGATGAAGAAGCTCTGGACGCATTCATCGAACTGTCTCGCCTGGAAGGCATCATCCCAGCCCTGGAGTCTTCTCACGCACTGGCTTATCTGAAGAAGATCAACATCAAGGGTAAAGTTGTGGTGGTTAATCTGTCTGGTCGTGGTGACAAGGATCTGGAATCTGTACTGAACCACCCGTATGTTCGCGAACGCATCCGCCTCGAGCACCACCACCACCACCACTGA
VIVS in pET22b(+): 5277	ATGAAAGGCTACTTCGGTCCGTACGGTGGCCAGTACGTGCCGGAAATCCTGATGGGAGCTCTGGAAGAACTGGAAGCTGCGTACGAAGGAATCATGAAAGATGAGTCTTTCTGGAAAGAATTCAATGACCTGCTGCGCGATTATGCGGGTCGTCCGACTCCGCTGTACTTCGCACGTCGTCTGTCCGAAAAATACGGTGCTCGCGTATATCTGAAACGTGAAGACCTGCTGCATACTGGTGCGCATAAAATCAATAACGCTATCGGCCAGGTTCTGCTGGCAAAACTAATGGGCAAAACCCGTATCATTGCTGAAACGGGTGCTGGTCAGCACGGCGTAGCAACTGCTACCGCAGCAGCGCTGTTCGGTATGGAATGTGTAATCTATATGGGCGAAGAAGACACGATCCGCCAGAAACTAAACGTTGAACGTATGAAACTGCTGGGTGCTAAAGTTGTACCGGTAAAATCCGGTAGCCGTACCCTGAAAGACGCAATTGACGAAGCTCTGCGTGACTGGATTACCAACCTGCAGACCACCTATTACGTGATTGGCTCTGTGGTTGGTCCGCATCCATATCCGATTATCGTACGTAACTTCCAAAAGGTTATCGGCGAAGAGACCAAAAAACAGATTCCAGAAAAAGAAGGCCGTCTGCCGGACTACATCGTTGCGTGCGTGAGCGGTGGTTCTAACGCTGCCGGTATCTTCTATCCGTTTATCGATTCTGGTGTGAAGCTGATCGGCGTAGAAGCCGGTGGCGAAGGTCTGGAAACCGGTAAACATGCGGCTTCTCTGCTGAAAGGTAAAATCGGCTACCTGCACGGTTCTAAGACGTTCGTTCTGCAGGATGACTGGGGTCAAGTTCAGGTGAGCCACTCCGTCTCCGCTGGCCTGGACTACTCCGGTGTCGGTCCGGAACACGCCTATTGGCGTGAGACCGGTAAAGTGCTGTACGATGCTGTGACCGATGAAGAAGCTCTGGACGCATTCATCGAACTGTCTCGCCTGGAAGGCATCATCCCAGCCCTGGAGTCTTCTCACGCACTGGCTTATCTGAAGAAGATCAACATCAAGGGTAAAGTTGTGGTGGTTAATCTGTCTGGTCGTGGTGACAAGGATCTGGAATCTGTACTGAACCACCCGTATGTTCGCGAACGCATCCGCCTCGAGCACCACCACCACCACCACTGA
VIVG in pET22b(+): 5278	ATGAAAGGCTACTTCGGTCCGTACGGTGGCCAGTACGTGCCGGAAATCCTGATGGGAGCTCTGGAAGAACTGGAAGCTGCGTACGAAGGAATCATGAAAGATGAGTCTTTCTGGAAAGAATTCAATGACCTGCTGCGCGATTATGCGGGTCGTCCGACTCCGCTGTACTTCGCACGTCGTCTGTCCGAAAAATACGGTGCTCGCGTATATCTGAAACGTGAAGACCTGCTGCATACTGGTGCGCATAAAATCAATAACGCTATCGGCCAGGTTCTGCTGGCAAAACTAATGGGCAAAACCCGTATCATTGCTGAAACGGGTGCTGGTCAGCACGGCGTAGCAACTGCTACCGCAGCAGCGCTGTTCGGTATGGAATGTGTAATCTATATGGGCGAAGAAGACACGATCCGCCAGAAACTAAACGTTGAACGTATGAAACTGCTGGGTGCTAAAGTTGTACCGGTAAAATCCGGTAGCCGTACCCTGAAAGACGCAATTGACGAAGCTCTGCGTGACTGGATTACCAACCTGCAGACCACCTATTACGTGATTGGCTCTGTGGTTGGTCCGCATCCATATCCGATTATCGTACGTAACTTCCAAAAGGTTATCGGCGAAGAGACCAAAAAACAGATTCCAGAAAAAGAAGGCCGTCTGCCGGACTACATCGTTGCGTGCGTGGGTGGTGGTTCTAACGCTGCCGGTATCTTCTATCCGTTTATCGATTCTGGTGTGAAGCTGATCGGCGTAGAAGCCGGTGGCGAAGGTCTGGAAACCGGTAAACATGCGGCTTCTCTGCTGAAAGGTAAAATCGGCTACCTGCACGGTTCTAAGACGTTCGTTCTGCAGGATGACTGGGGTCAAGTTCAGGTGAGCCACTCCGTCTCCGCTGGCCTGGACTACTCCGGTGTCGGTCCGGAACACGCCTATTGGCGTGAGACCGGTAAAGTGCTGTACGATGCTGTGACCGATGAAGAAGCTCTGGACGCATTCATCGAACTGTCTCGCCTGGAAGGCATCATCCCAGCCCTGGAGTCTTCTCACGCACTGGCTTATCTGAAGAAGATCAACATCAAGGGTAAAGTTGTGGTGGTTAATCTGTCTGGTCGTGGTGACAAGGATCTGGAATCTGTACTGAACCACCCGTATGTTCGCGAACGCATCCGCCTCGAGCACCACCACCACCACCACTGA

VIKG in pET22b(+): 5279	ATGAAAGGCTACTTCGGTCCGTACGGTGGCCAGTACGTGCCGGAAATCCTGATGGGAGCTCTGGAAGAACTGGAAGCTGCGTACGAAGGAATCATGAAAGATGAGTCTTTCTGGAAAGAATTCAATGACCTGCTGCGCGATTATGCGGGTCGTCCGACTCCGCTGTACTTCGCACGTCGTCTGTCCGAAAAATACGGTGCTCGCGTATATCTGAAACGTGAAGACCTGCTGCATACTGGTGCGCATAAAATCAATAACGCTATCGGCCAGGTTCTGCTGGCAAAACTAATGGGCAAAACCCGTATCATTGCTGAAACGGGTGCTGGTCAGCACGGCGTAGCAACTGCTACCGCAGCAGCGCTGTTCGGTATGGAATGTGTAATCTATATGGGCGAAGAAGACACGATCCGCCAGAAACTAAACGTTGAACGTATGAAACTGCTGGGTGCTAAAGTTGTACCGGTAAAATCCGGTAGCCGTACCCTGAAAGACGCAATTGACGAAGCTCTGCGTGACTGGATTACCAACCTGCAGACCACCTATTACGTGATTGGCTCTGTGGTTGGTCCGCATCCATATCCGATTATCGTACGTAACTTCCAAAAGGTTATCGGCGAAGAGACCAAAAAACAGATTCCAGAAAAAGAAGGCCGTCTGCCGGACTACATCGTTGCGTGCGTGAAGGGTGGTGGTTCTAACGCTGCCGGTATCTTCTATCCGTTTATCGATTCTGGTGTGAAGCTGATCGGCGTAGAAGCCGGTGGCGAAGGTCTGGAAACCGGTAAACATGCGGCTTCTCTGCTGAAAGGTAAAATCGGCTACCTGCACGGTTCTAAGACGTTCGTTCTGCAGGATGACTGGGGTCAAGTTCAGGTGAGCCACTCCGTCTCCGCTGGCCTGGACTACTCCGGTGTCGGTCCGGAACACGCCTATTGGCGTGAGACCGGTAAAGTGCTGTACGATGCTGTGACCGATGAAGAAGCTCTGGACGCATTCATCGAACTGTCTCGCCTGGAAGGCATCATCCCAGCCCTGGAGTCTTCTCACGCACTGGCTTATCTGAAGAAGATCAACATCAAGGGTAAAGTTGTGGTGGTTAATCTGTCTGGTCGTGGTGACAAGGATCTGGAATCTGTACTGAACCACCCGTATGTTCGCGAACGCATCCGCCTCGAGCACCACCACCACCACCACTGA
AIKG in pET22b(+): 5280	ATGAAAGGCTACTTCGGTCCGTACGGTGGCCAGTACGTGCCGGAAATCCTGATGGGAGCTCTGGAAGAACTGGAAGCTGCGTACGAAGGAATCATGAAAGATGAGTCTTTCTGGAAAGAATTCAATGACCTGCTGCGCGATTATGCGGGTCGTCCGACTCCGCTGTACTTCGCACGTCGTCTGTCCGAAAAATACGGTGCTCGCGTATATCTGAAACGTGAAGACCTGCTGCATACTGGTGCGCATAAAATCAATAACGCTATCGGCCAGGTTCTGCTGGCAAAACTAATGGGCAAAACCCGTATCATTGCTGAAACGGGTGCTGGTCAGCACGGCGTAGCAACTGCTACCGCAGCAGCGCTGTTCGGTATGGAATGTGTAATCTATATGGGCGAAGAAGACACGATCCGCCAGAAACTAAACGTTGAACGTATGAAACTGCTGGGTGCTAAAGTTGTACCGGTAAAATCCGGTAGCCGTACCCTGAAAGACGCAATTGACGAAGCTCTGCGTGACTGGATTACCAACCTGCAGACCACCTATTACGCGATTGGCTCTGTGGTTGGTCCGCATCCATATCCGATTATCGTACGTAACTTCCAAAAGGTTATCGGCGAAGAGACCAAAAAACAGATTCCAGAAAAAGAAGGCCGTCTGCCGGACTACATCGTTGCGTGCAAGGGTGGTGGTTCTAACGCTGCCGGTATCTTCTATCCGTTTATCGATTCTGGTGTGAAGCTGATCGGCGTAGAAGCCGGTGGCGAAGGTCTGGAAACCGGTAAACATGCGGCTTCTCTGCTGAAAGGTAAAATCGGCTACCTGCACGGTTCTAAGACGTTCGTTCTGCAGGATGACTGGGGTCAAGTTCAGGTGAGCCACTCCGTCTCCGCTGGCCTGGACTACTCCGGTGTCGGTCCGGAACACGCCTATTGGCGTGAGACCGGTAAAGTGCTGTACGATGCTGTGACCGATGAAGAAGCTCTGGACGCATTCATCGAACTGTCTCGCCTGGAAGGCATCATCCCAGCCCTGGAGTCTTCTCACGCACTGGCTTATCTGAAGAAGATCAACATCAAGGGTAAAGTTGTGGTGGTTAATCTGTCTGGTCGTGGTGACAAGGATCTGGAATCTGTACTGAACCACCCGTATGTTCGCGAACGCATCCGCCTCGAGCACCACCACCACCACCACTGA
CLKG in pET22b(+): 5281	ATGAAAGGCTACTTCGGTCCGTACGGTGGCCAGTACGTGCCGGAAATCCTGATGGGAGCTCTGGAAGAACTGGAAGCTGCGTACGAAGGAATCATGAAAGATGAGTCTTTCTGGAAAGAATTCAATGACCTGCTGCGCGATTATGCGGGTCGTCCGACTCCGCTGTACTTCGCACGTCGTCTGTCCGAAAAATACGGTGCTCGCGTATATCTGAAACGTGAAGACCTGCTGCATACTGGTGCGCATAAAATCAATAACGCTATCGGCCAGGTTCTGCTGGCAAAACTAATGGGCAAAACCCGTATCATTGCTGAAACGGGTGCTGGTCAGCACGGCGTAGCAACTGCTACCGCAGCAGCGCTGTTCGGTATGGAATGTGTAATCTATATGGGCGAAGAAGACACGATCCGCCAGAAACTAAACGTTGAACGTATGAAACTGCTGGGTGCTAAAGTTGTACCGGTAAAATCCGGTAGCCGTACCCTGAAAGACGCAATTGACGAAGCTCTGCGTGACTGGATTACCAACCTGCAGACCACCTATTACTGTCTGGGCTCTGTGGTTGGTCCGCATCCATATCCGATTATCGTACGTAACTTCCAAAAGGTTATCGGCGAAGAGACCAAAAAACAGATTCCAGAAAAAGAAGGCCGTCTGCCGGACTACATCGTTGCGTGCAAGGGTGGTGGTTCTAACGCTGCCGGTATCTTCTATCCGTTTATCGATTCTGGTGTGAAGCTGATCGGCGTAGAAGCCGGTGGCGAAGGTCTGGAAACCGGTAAACATGCGGCTTCTCTGCTGAAAGGTAAAATCGGCTACCTGCACGGTTCTAAGACGTTCGTTCTGCAGGATGACTGGGGTCAAGTTCAGGTGAGCCACTCCGTCTCCGCTGGCCTGGACTACTCCGGTGTCGGTCCGGAACACGCCTATTGGCGTGAGACCGGTAAAGTGCTGTACGATGCTGTGACCGATGAAGAAGCTCTGGACGCATTCATCGAACTGTCTCGCCTGGAAGGCATCATCCCAGCCCTGGAGTCTTCTCACGCACTGGCTTATCTGAAGAAGATCAACATCAAGGGTAAAGTTGTGGTGGTTAATCTGTCTGGTCGTGGTGACAAGGATCTGGAATCTGTACTGAACCACCCGTATGTTCGCGAACGCATCCGCCTCGAGCACCACCACCACCACCACTGA
ALKG	ATGAAAGGCTACTTCGGTCCGTACGGTGGCCAGTACGTGCCGGAAATCCTGATGGGAGCTCTGGAAGAACTGGAAGCTGCGTACGAAGGAATCATGAAAGATGAGTCTTTCTGGAAAGAATTCAATGACCTGCTGCGCGATTATGCGGGTCGTCCGACTCCGCTGTACTTCGCACGTCGTCTGTCCGAAAAATACGGTGCTCGCGTATATCTGAAACGTGAAGACCTGCTGCATACTGGTGCGCATAAAATCAATAACGCTATCGGCCAGGTTCTGCTGGCAAAACTAATGGGCAAAACCCGTATCATTGCTGAAACGGGTGCTGGTCAGCACGGCGTAGCAACTGCTACCGCAGCAGCGCTGTTCGGTATGGAATGTGTAATCTATATGGGCGAAGAAGACACGATCCGCCAGAAACTAAACGTTGAACGTATGAAACTGCTGGGTGCTAAAGTTGTACCGGTAAAATCCGGTAGCCGTACCCTGAAAGACGCAATTGACGAAGCTCTGCGTGACTGGATTACCAACCTGCAGACCACCTATTACTGTATTGGCTCT

in pET22b(+): 5282	GTGGTTGGTCCGCATCCATATCCGATTATCGTACGTAACTTCCAAAAGGTTATCGGCGAAGAGACCAAAAAACAGATTCCAGAAAAAGAAGGCCGTCTGCCGGACTACATCGTTGCGTGCAAGGGTGGTGGTTCTAACGCTGCCGGTATCTTCTATCCGTTTATCGATTCTGGTGTGAAGCTGATCGGCGTAGAAGCCGGTGGCGAAGGTCTGGAAACCGGTAAACATGCGGCTTCTCTGCTGAAAGGTAAAATCGGCTACCTGCACGGTTCTAAGACGTTCGTTCTGCAGGATGACTGGGGTCAAGTTCAGGTGAGCCACTCCGTCTCCGCTGGCCTGGACTACTCCGGTGTCGGTCCGGAACACGCCTATTGGCGTGAGACCGGTAAAGTGCTGTACGATGCTGTGACCGATGAAGAAGCTCTGGACGCATTCATCGAACTGTCTCGCCTGGAAGGCATCATCCCAGCCCTGGAGTCTTCTCACGCACTGGCTTATCTGAAGAAGATCAACATCAAGGGTAAAGTTGTGGTGGTTAATCTGTCTGGTCGTGGTGACAAGGATCTGGAATCTGTACTGAACCACCCGTATGTTCGCGAACGCATCCGCCTCGAGCACCACCACCACCACCACTGA
CIKG in pET22b(+): 5283	ATGAAAGGCTACTTCGGTCCGTACGGTGGCCAGTACGTGCCGGAAATCCTGATGGGAGCTCTGGAAGAACTGGAAGCTGCGTACGAAGGAATCATGAAAGATGAGTCTTTCTGGAAAGAATTCAATGACCTGCTGCGCGATTATGCGGGTCGTCCGACTCCGCTGTACTTCGCACGTCGTCTGTCCGAAAAATACGGTGCTCGCGTATATCTGAAACGTGAAGACCTGCTGCATACTGGTGCGCATAAAATCAATAACGCTATCGGCCAGGTTCTGCTGGCAAAACTAATGGGCAAAACCCGTATCATTGCTGAAACGGGTGCTGGTCAGCACGGCGTAGCAACTGCTACCGCAGCAGCGCTGTTCGGTATGGAATGTGTAATCTATATGGGCGAAGAAGACACGATCCGCCAGAAACTAAACGTTGAACGTATGAAACTGCTGGGTGCTAAAGTTGTACCGGTAAAATCCGGTAGCCGTACCCTGAAAGACGCAATTGACGAAGCTCTGCGTGACTGGATTACCAACCTGCAGACCACCTATTACGCGCTGGGCTCTGTGGTTGGTCCGCATCCATATCCGATTATCGTACGTAACTTCCAAAAGGTTATCGGCGAAGAGACCAAAAAACAGATTCCAGAAAAAGAAGGCCGTCTGCCGGACTACATCGTTGCGTGCAAGGGTGGTGGTTCTAACGCTGCCGGTATCTTCTATCCGTTTATCGATTCTGGTGTGAAGCTGATCGGCGTAGAAGCCGGTGGCGAAGGTCTGGAAACCGGTAAACATGCGGCTTCTCTGCTGAAAGGTAAAATCGGCTACCTGCACGGTTCTAAGACGTTCGTTCTGCAGGATGACTGGGGTCAAGTTCAGGTGAGCCACTCCGTCTCCGCTGGCCTGGACTACTCCGGTGTCGGTCCGGAACACGCCTATTGGCGTGAGACCGGTAAAGTGCTGTACGATGCTGTGACCGATGAAGAAGCTCTGGACGCATTCATCGAACTGTCTCGCCTGGAAGGCATCATCCCAGCCCTGGAGTCTTCTCACGCACTGGCTTATCTGAAGAAGATCAACATCAAGGGTAAAGTTGTGGTGGTTAATCTGTCTGGTCGTGGTGACAAGGATCTGGAATCTGTACTGAACCACCCGTATGTTCGCGAACGCATCCGCCTCGAGCACCACCACCACCACCACTGA
VLKG in pET22b(+): 5284	ATGAAAGGCTACTTCGGTCCGTACGGTGGCCAGTACGTGCCGGAAATCCTGATGGGAGCTCTGGAAGAACTGGAAGCTGCGTACGAAGGAATCATGAAAGATGAGTCTTTCTGGAAAGAATTCAATGACCTGCTGCGCGATTATGCGGGTCGTCCGACTCCGCTGTACTTCGCACGTCGTCTGTCCGAAAAATACGGTGCTCGCGTATATCTGAAACGTGAAGACCTGCTGCATACTGGTGCGCATAAAATCAATAACGCTATCGGCCAGGTTCTGCTGGCAAAACTAATGGGCAAAACCCGTATCATTGCTGAAACGGGTGCTGGTCAGCACGGCGTAGCAACTGCTACCGCAGCAGCGCTGTTCGGTATGGAATGTGTAATCTATATGGGCGAAGAAGACACGATCCGCCAGAAACTAAACGTTGAACGTATGAAACTGCTGGGTGCTAAAGTTGTACCGGTAAAATCCGGTAGCCGTACCCTGAAAGACGCAATTGACGAAGCTCTGCGTGACTGGATTACCAACCTGCAGACCACCTATTACGTGCTGGGCTCTGTGGTTGGTCCGCATCCATATCCGATTATCGTACGTAACTTCCAAAAGGTTATCGGCGAAGAGACCAAAAAACAGATTCCAGAAAAAGAAGGCCGTCTGCCGGACTACATCGTTGCGTGCAAGGGTGGTGGTTCTAACGCTGCCGGTATCTTCTATCCGTTTATCGATTCTGGTGTGAAGCTGATCGGCGTAGAAGCCGGTGGCGAAGGTCTGGAAACCGGTAAACATGCGGCTTCTCTGCTGAAAGGTAAAATCGGCTACCTGCACGGTTCTAAGACGTTCGTTCTGCAGGATGACTGGGGTCAAGTTCAGGTGAGCCACTCCGTCTCCGCTGGCCTGGACTACTCCGGTGTCGGTCCGGAACACGCCTATTGGCGTGAGACCGGTAAAGTGCTGTACGATGCTGTGACCGATGAAGAAGCTCTGGACGCATTCATCGAACTGTCTCGCCTGGAAGGCATCATCCCAGCCCTGGAGTCTTCTCACGCACTGGCTTATCTGAAGAAGATCAACATCAAGGGTAAAGTTGTGGTGGTTAATCTGTCTGGTCGTGGTGACAAGGATCTGGAATCTGTACTGAACCACCCGTATGTTCGCGAACGCATCCGCCTCGAGCACCACCACCACCACCACTGA

Table B-4. Primers for *Tm*9D8*-pBAD24 plasmid construction

Name	Sequence (5’ → 3’)
TrpB_pBAD24_insert_fwd	AGCAGGAGGAATTCGCCAATGAAAGGCTACTTCGGTCCGTACGG
TrpB_pBAD24_insert_rev	CCAAGCTTCCCGGGTCATCAGTGGTGGTGGTGGTGGTGC
TrpB_pBAD24_bb_fwd	TGACCCGGGAAGCTTGGCTGTTTTGGCGGATGAGAGAAGATTTTCAGC
TrpB_pBAD24_bb_rev	TGGCGAATTCCTCCTGCTAGCCCAAAAAAACGGGTATGGAGAAACAG
AmpR_internal_fwd	CCAACTTACTTCTGACAACGATCGGAGGACCGAAGGAGCTAACCGCTTTTTTGC
AmpR_internal_rev	CGATCGTTGTCAGAAGTAAGTTGGCCGCAGTGTTATCACTCATGGTTATGGCAG

Table B-5. Primers for construction of triple-site saturation libraries

Library	Primer	Primer name	Sequence (5’ → 3’)
A 104 105 106	F Gap	9D8s_106_gap_F	GGTGCTGGTCAGCACG
	F Library	9D8s_104-105-106_F	AATGGGCAAAACCCGTATCATTNNKNNKNNKGGTGCTGGTCAGCACG
	R	9D8s_104_R	AATGATACGGGTTTTGCCCATTAGTTTTGCCAGCAGAACCTGGC
B 105 106 107	F Gap	9D8s_107_gap_F	GCTGGTCAGCACGGC
	F Library	9D8s_105-106-107_F	AATGGGCAAAACCCGTATCATTGCTNNKNNKNNKGCTGGTCAGCACGGC
	R	9D8s_104_R	AATGATACGGGTTTTGCCCATTAGTTTTGCCAGCAGAACCTGGC
C 106 107 108	F Gap	9D8s_108_gap_F	GGTCAGCACGGCGTAG
	F Library	9D8s_106-107-108_F	AATGGGCAAAACCCGTATCATTGCTGAANNKNNKNNKGGTCAGCACGGCGTAG
	R	9D8s_104_R	AATGATACGGGTTTTGCCCATTAGTTTTGCCAGCAGAACCTGGC
D 117 118 119	F Gap	9D8s_119_gap_F	GCGCTGTTCGGTATGGAAT
	F Library	9D8s_117-118-119_F	ACGGCGTAGCAACTGCTNNKNNKNNKGCGCTGTTCGGTATGGAATGTGTAATC
	R	9D8s_117_R	AGCAGTTGCTACGCCGTGCTGACCAGCACCCGTTTCAG
E 184 185 186	F Gap	9D8s_186_gap_F	GTGGTTGGTCCGCATCC
	F Library	9D8s_184-185-186_F	CTGCAGACCACCTATTACGTGNNKNNKNNKGTGGTTGGTCCGCATCCATATCC
	R	9D8s_184_R	CACGTAATAGGTGGTCTGCAGGTTGGTAATCCAGTCACGCAGAGCT
F* 162 166 301	F Gap	9D8s_166_gap_F	GACGAAGCTCTGCGTGAC
	F Library	9D8s_162-166_F	GTAAAATCCGGTAGCCGTACCNNKAAAGACGCANNKGACGAAGCTCTG
	R	9D8s_162_R	GGTACGGCTACCGGATTTTACCGGTACAACTTTAGCACCCAGCAG
G* 227 228 301	F Gap	9D8s_228_gap_F	GGTGGTTCTAACGCTGCC
	F Library	9D8s_227-228_F	GGACTACATCGTTGCGTGCNNKNNKGGTGGTTCTAACGCTGCCGGTA
	R	9D8s_227_R	GCACGCAACGATGTAGTCCGGCAGACGGCCTTCTTTTTCTGG

H 228 230 231	F Gap	9D8s_231_gap_F	AACGCTGCCGGTATCTTCTAT
	F Library	9D8s_228-230-231_F	GGACTACATCGTTGCGTGCGTGNNKGGTNNKNNKAACGCTGCCGGTATCTTCTATCCG
	R	9D8s_227_R	GCACGCAACGATGTAGTCCGGCAGACGGCCTTCTTTTTCTGG
I 182 183 184	F Gap	9D8s_184_gap_F	GGCTCTGTGGTTGGTCC
	F Library	9D8s_182-183-184_F	CCAACCTGCAGACCACCTATNNKNNKNNKGGCTCTGTGGTTGGTCCGC
	R	9D8s_182_R	ATAGGTGGTCTGCAGGTTGGTAATCCAGTCACGCAGAGCTTCGT

*These libraries used a template of a 301X plasmid library created with primers in Table S2.

Table B-6. Primers for construction of quadruple-site saturation libraries

Library	Primer	Primer name	Sequence (5’ → 3’)
4-site 183 184 227 228	F	9D8s_foursite_183-184-227-228_bb_f	GGTGGTTCTAACGCTGCCGGTATCTTCTATCCGTTTATCG
	F Gap	9D8s_foursite_183-184-227-228_inner_f	GGCTCTGTGGTTGGTCCGCATCCATATCCG
	F Library	9D8s_foursite_183-184-227-228_outer_NNK_f	CCAACCTGCAGACCACCTATTACNNKNNKGGCTCTGTGGTTGGTCCGC
	R	9D8s_foursite_183-184-227-228_bb_r	GTAATAGGTGGTCTGCAGGTTGGTAATCCAGTCACGC
	R Gap	9D8s_foursite_183-184-227-228_inner_r	GCACGCAACGATGTAGTCCGGCAGACGGCCTTC
	R Library	9D8s_foursite_183-184-227-228_outer_NNK_r	GGCAGCGTTAGAACCACCMNNMNNGCACGCAACGATGTAGTCCG

Table B-7. Sequencing preparation primers for triple- and quadruple-site libraries

Primer name	Barcode	Seed (bp)	F/R	Sequence 5' → 3'
pr191_seq_TGT_266-285_F	TGT	266-285	F	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGNNNNNNNNNNTGTGCCAGGTTCTGCTGGCAAAA
pr192_seq_GTG_266-285_F	GTG	266-285	F	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGNNNNNNNNNNGTGGCCAGGTTCTGCTGGCAAAA
pr193_seq_ACA_266-285_F	ACA	266-285	F	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGNNNNNNNNNNACAGCCAGGTTCTGCTGGCAAAA
pr195_seq_TGT_415-394_R	TGT	415-394	R	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGNNNNNNNNNNTGTTCTGGCGGATCGTGTCTTCTTC
pr196_seq_GTG_415-394_R	GTG	415-394	R	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGNNNNNNNNNNGTGTCTGGCGGATCGTGTCTTCTTC
pr197_seq_ACA_415-394_R	ACA	415-394	R	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGNNNNNNNNNNACATCTGGCGGATCGTGTCTTCTTC
pr206_seq_TGT_448-473_F	TGT	448-473	F	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGNNNNNNNNNNTGTGCTAAAGTTGTACCGGTAAAATCCGG
pr207_seq_GTG_448-473_F	GTG	448-473	F	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGNNNNNNNNNNGTGGCTAAAGTTGTACCGGTAAAATCCGG
pr209_seq_TGT_589-565_R	TGT	589-565	R	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGNNNNNNNNNNTGTCGATAATCGGATATGGATGCGGACC
pr210_seq_GTG_589-565_R	GTG	589-565	R	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGNNNNNNNNNNGTGCGATAATCGGATATGGATGCGGACC
pr215_seq_TGT_659-678_F	TGT	659-678	F	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGNNNNNNNNNNTGTCGGACTACATCGTTGCGTGC
pr216_seq_GTG_659-678_F	GTG	659-678	F	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGNNNNNNNNNNGTGCGGACTACATCGTTGCGTGC
pr212_seq_TGT_929-911_R	TGT	929-911	R	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGNNNNNNNNNNTGTTAGGCGTGTTCCGGACCG
pr213_seq_GTG_929-911_R	GTG	929-911	R	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGNNNNNNNNNNGTGTAGGCGTGTTCCGGACCG
pr018_seq_TGT_518-540_F	TGT	518-540	F	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGNNNNNNNNNNTGTGGATTACCAACCTGCAGACCACC
pr019_seq_GTG_518-540_F	GTG	518-540	F	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGNNNNNNNNNNGTGGGATTACCAACCTGCAGACCACC
pr020_seq_ACA_518-540_F	ACA	518-540	F	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGNNNNNNNNNNACAGGATTACCAACCTGCAGACCACC

pr021_seq_TGT_709-688_R	TGT	709-688	R	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGNNNNNNNNNNTGTAGATACCGGCAGCGTTAGAACC
pr022_seq_GTG_709-688_R	GTG	709-688	R	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGNNNNNNNNNNGTGAGATACCGGCAGCGTTAGAACC
pr023_seq_ACA_709-688_R	ACA	709-688	R	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGNNNNNNNNNNACAAGATACCGGCAGCGTTAGAACC

Table B-8. Mapping sequencing primers to libraries

Primer name	Libraries
pr191_seq_TGT_266-285_F	A/T0 B/T0 C/T0 D/T0, D1/T1–T5
pr192_seq_GTG_266-285_F	A1/T1 B1/T1 C1/T1 D2/T1–T5
pr193_seq_ACA_266-285_F	A1/T4 B1/T4 C1/T4
pr195_seq_TGT_415-394_R	A/T0 B/T0 C/T0 D1/T1–T5
pr196_seq_GTG_415-394_R	A1/T1 B1/T1 C1/T1 D2/T1–T5
pr197_seq_ACA_415-394_R	A1/T4 B1/T4 C1/T4
pr206_seq_TGT_448-473_F	E/T0, E1/T1–T5 F/T0, F1/T1–T5 I/T0, I1/T1–T5
pr207_seq_GTG_448-473_F	E2/T1–T5 F2/T1–T5 I2/T1–T5
pr209_seq_TGT_589-565_R	E/T0 E1/T1–T5 I/T0, I1/T1–T5
pr210_seq_GTG_589-565_R	E2/T1–T5 I2/T1–T5
pr215_seq_TGT_659-678_F	G/T0, G1/T1–T5 H/T0, H1/T1–T5
pr216_seq_GTG_659-678_F	G2/T1–T5

	H2/T1–T5
pr212_seq_TGT_929-911_R	F/T0, F1/T1–T5 G/T0, G1/T1–T5 H/T0, H1/T1–T5
pr213_seq_GTG_929-911_R	F2/T1–T5 G2/T1–T5 H2/T1–T5
pr018_seq_TGT_518-540_F	4-site replicate #1/T0–T6
pr019_seq_GTG_518-540_F	4-site replicate #2/T0–T6
pr021_seq_TGT_709-688_R	4-site replicate #1/T0–T6
pr022_seq_GTG_709-688_R	4-site replicate #2/T0–T6

Table B-9. OD_{600} over time by library: libraries A, B, and C

Hours	A1	A2	B1	B2	C1	C2
0	0.1*	0.1	0.1*	0.1	0.1*	0.1
18	0.72*	0.75	0.75*	0.84	0.74*	0.76
20	0.78	0.83	0.83	0.98	0.78	0.84
24	0.94	1.01	1.09	1.50	0.86	0.92
44	2.55*	2.7	3.3*	3.85	1.95*	4.15

*These samples were sequenced. Based on preliminary results where most variants were inactive, the remaining timepoints were not sequenced.

Table B-10. OD_{600} over time by library: libraries D, E, F, G, H, and I

Time (h)	D1	D2	E1	E2	F1	F2	G1	G2	H1	H2	I1	I2
0	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
12	0.19	0.18	0.20	0.20	0.17	0.17	0.14	0.14	0.15	0.14	0.36	0.39
16	0.29	0.28	0.27	0.26	0.20	0.20	0.18	0.18	0.19	0.18	0.83	0.87
20	0.51	0.49	0.47	0.44	0.23	0.24	0.23	0.23	0.26	0.26	1.24	1.36
24	0.85	0.97	0.91	0.94	0.27	0.27	0.44	0.44	0.67	0.58	1.7	2.1
36	1.42	1.81	1.41	1.54	0.79	0.79	2.0	1.95	2.9	1.85	1.95	2.25

Table B-11. OD_{600} over time: quadruple-site library

Time (h)	4-site rep #1	4-site rep #2
0	0.025	0.025
12	0.19	0.19
16	0.51	0.52
20	1.26	1.34
24	1.50	1.625
28	1.675	1.75
36	1.75	1.875

Table B-12. Data collection and refinement statistics for the structure of *Tm*9D8*

Structure	*Tm*9D8*
Unit cell	
Space group	*I*4
a, b, c (Å)	165.7, 165.7, 83.06
α, β, γ (°)	90.0, 90.0, 90.0
Data collection	
Wavelength (Å)	0.97946
Resolution (Å)	45.99 – 2.15
Total/unique no. of reflections	825756/61153
R_{merge}[a,b]	0.17 (2.19)
$R_{p.i.m.}$[a,c]	0.05 (0.64)
$CC_{1/2}$[a,d]	0.99 (0.60)
$I/\sigma(I)$[a]	13.2 (1.7)
Redundancy[a]	13.5 (12.4)
Completeness[a] (%)	99.9 (99.4)
Refinement	
No. of reflections used in refinement/test set	61110/6021
R_{work}[a,e]	0.214 (0.303)
R_{free}[a,e]	0.237 (0.326)
No. of nonhydrogen atoms	
protein	5814
ligand	49
solvent	96
root-mean-square deviation from ideal geometry	
bonds (Å)	0.002
angles (°)	0.49
Ramachandran plot[f] (%)	
favored	97.24
allowed	2.37
disallowed	0.39
PDB accession code	N/A

[a] Values in parentheses refer to data in the highest shell.

[b] $R_{merge} = \sum_{hkl}\sum_i|I_{i,hkl} - \langle I\rangle_{hkl}|/\sum_{hkl}\sum_i I_{i,hkl}$, where $\langle I\rangle_{hkl}$ is the average intensity calculated for reflection *hkl* from replicate measurements.

[c] $R_{p.i.m.}= (\sum_{hkl}(1/(N-1))^{1/2}\sum_i|I_{i,hkl} - \langle I\rangle_{hkl}|)/\sum_{hkl}\sum_i I_{i,hkl}$, where $\langle I\rangle_{hkl}$ is the average intensity calculated for reflection *hkl* from replicate measurements and N is the number of reflections.

[d] Pearson correlation coefficient between random half-datasets.

[e] $R_{work} = \sum||F_o| - |F_c||/\sum|F_o|$ for reflections contained in the working set. $|F_o|$ and $|F_c|$ are the observed and calculated structure factor amplitudes, respectively. R_{free} is calculated using the same expression for reflections contained in the test set held aside during refinement.

[f] Calculated with PROCHECK.

B.3 Supplemental Figures

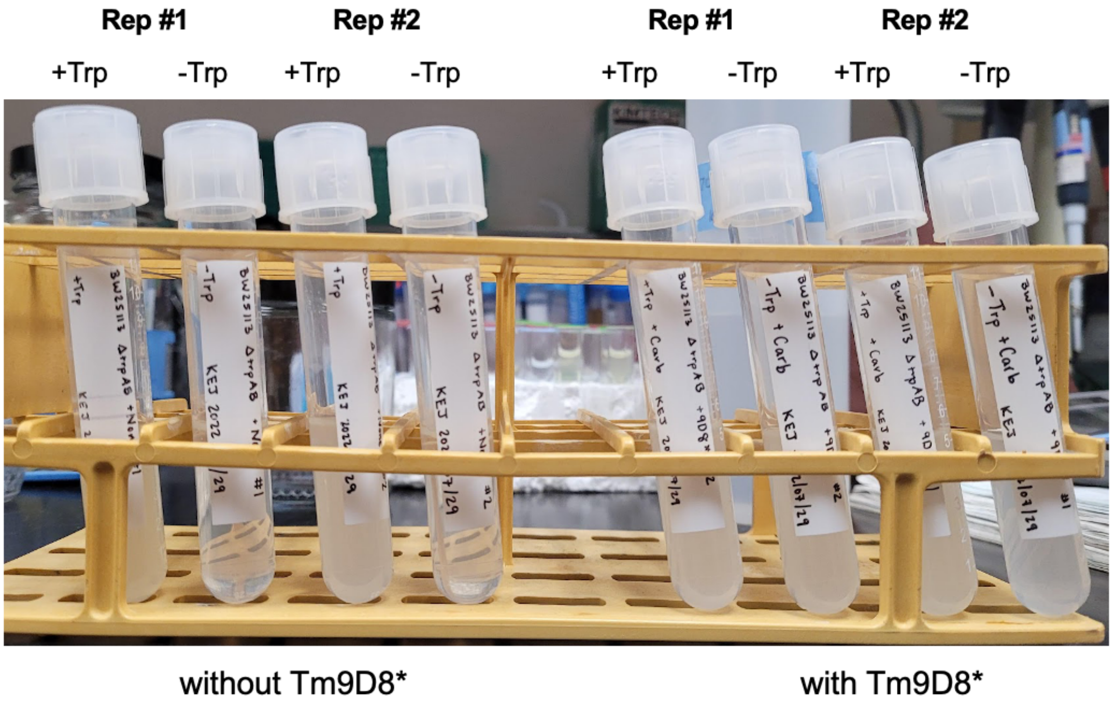


Figure B-1. TrpB- and Trp-dependent growth for the *E. coli* Trp auxotroph. The left set of tubes are cultures where the Trp auxotroph is given no TrpB variant while the right set does harbor an efficient TrpB (Tm9D8*). Within the sets, cultures were grown with and without exogenous Trp added to the Trp-DO media (all cultures contained indole and arabinose). Only cultures where exogenous Trp is added, or the cells harbor an active TrpB variant show detectable growth.

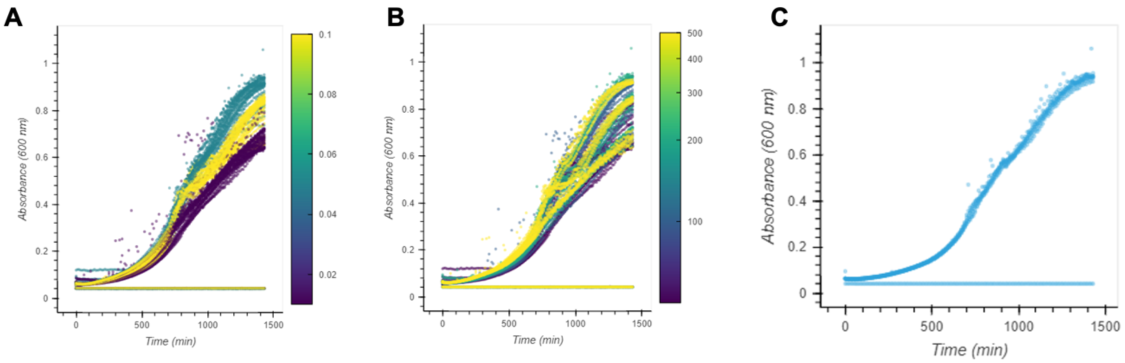


Figure B-2. Choosing growth assay conditions. *E. coli* Trp auxotroph cells harboring pBAD24-*Tm*9D8* and grown in Trp-dropout media supplemented with arabinose and indole. Sterile wells appear as flat lines in this assay. **A** Absorbance at 600 nm (OD_{600}) versus time for all wells colored by arabinose concentration (%). This showed that 0.05% arabinose was optimal. **B** Absorbance at 600 nm (OD_{600}) versus time for all wells colored by indole concentration. This showed that 200 μM indole was optimal. **C** Replicates of absorbance at 600 nm (OD_{600}) versus time at 0.05% arabinose and 200 μM indole.

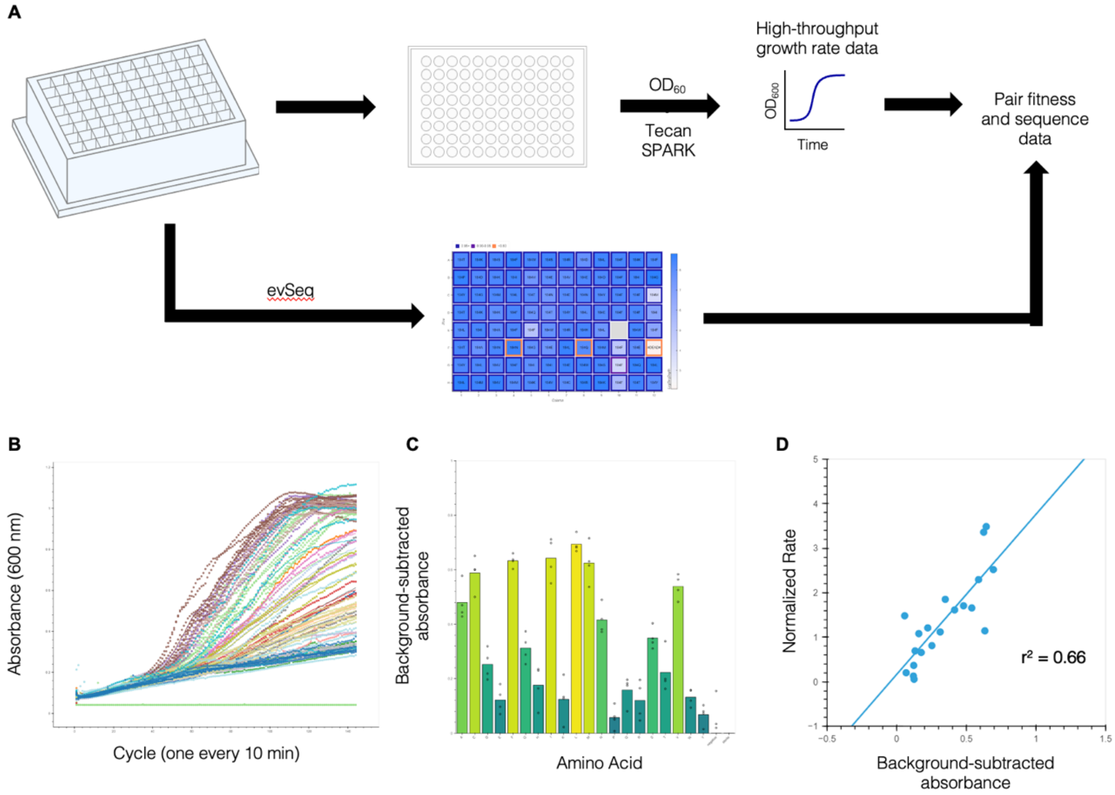


Figure B-3. Preliminary results for independent growth rates. **A** Independent growth rates for a 184X library were monitored by collecting the absorbance at 600 nm (OD_{600}) over time and pairing it with sequencing data obtained by evSeq. **B** Absorbance at 600 nm vs cycle number with the scatter plots colored by amino acid identity. We observed clear clustering between amino acids that indicated reproducibility. **C** Looking at a timepoint along the collected data, we subtracted the absorbance of the average negative control (empty pET22b(+) vector) well to obtain a background-subtracted absorbance for each well. Results were grouped by amino acid. **D** Plotting the background subtracted absorbance against the normalized rate of Trp formation (data obtained from Wittmann *et al.*,[5] we absorbed a reasonable correlation between the two that indicated the fitness we were measuring was similar.

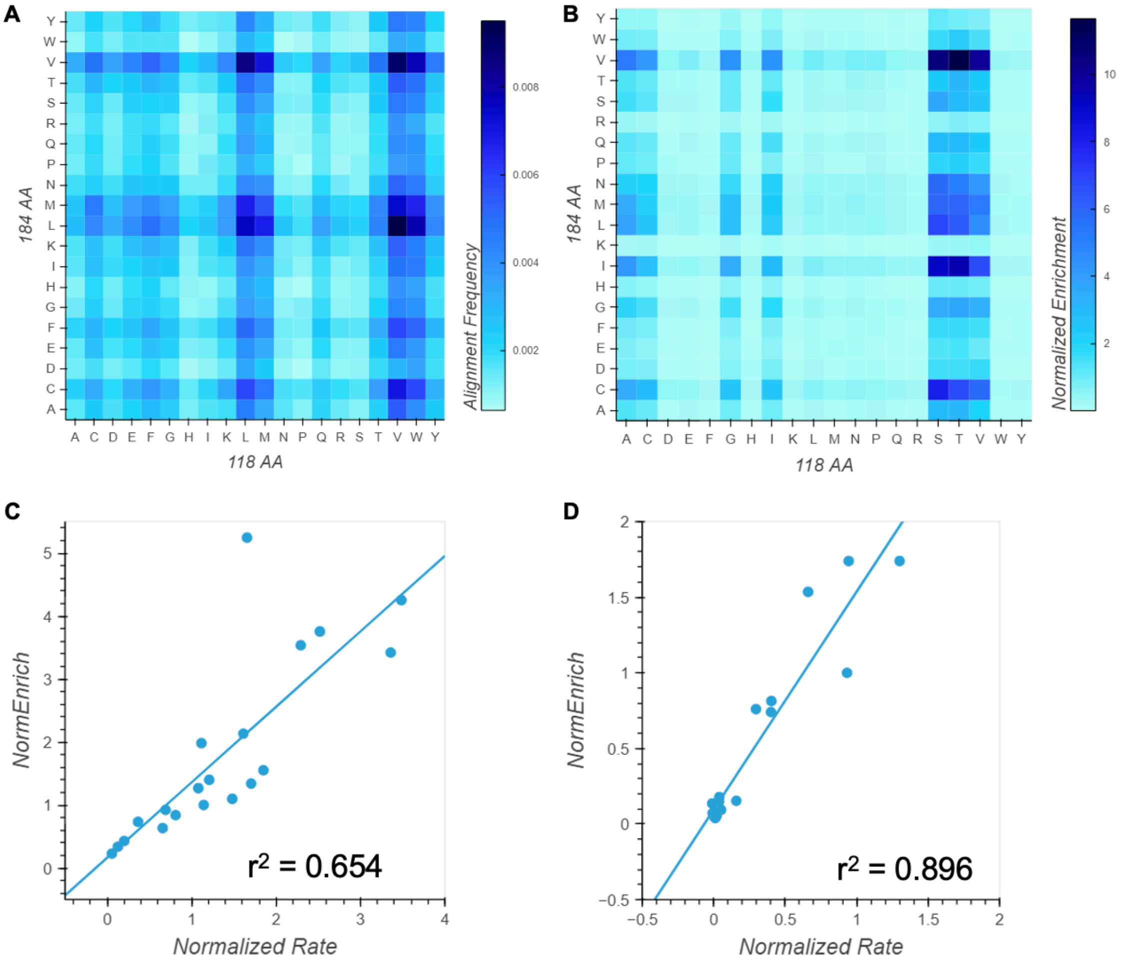


Figure B-4. 118X/184X growth-based enrichment assay. A Frequency of each 118/184 amino acid pair in the input DNA library. **B** Normalized enrichment was calculated as the output sequencing frequency divided by the input sequencing frequency and normalized to parent (A118, F184). **C** Correlation of normalized enrichment and normalized rate of Trp formation for all variants in the 118X/184X library with A118 (a proxy for a 184X library). **D** Correlation of normalized enrichment and normalized rate of Trp formation for all variants in the 118X/184X library with F184 (a proxy for a 118X library).

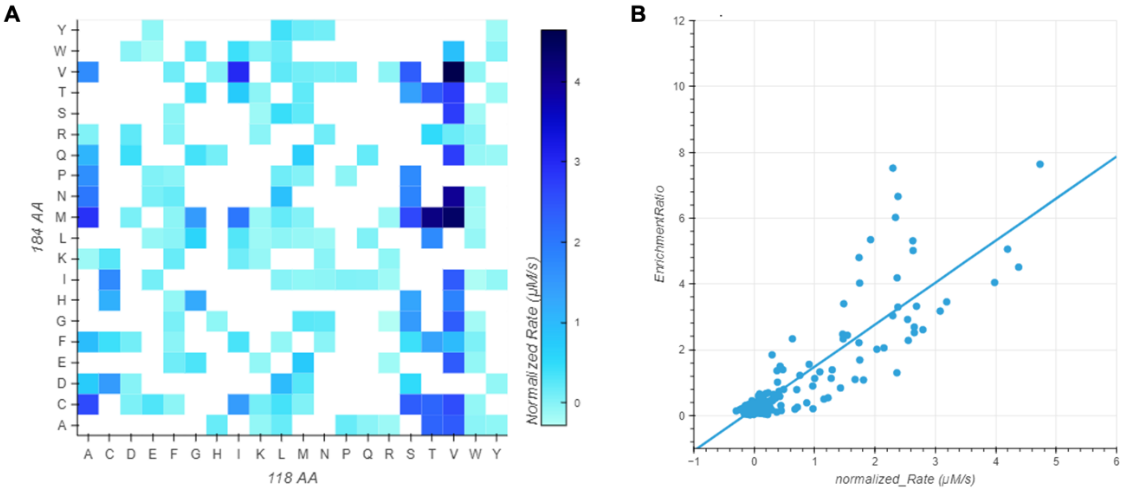


Figure B-5. Comparison of 118X/184X growth-based enrichment and *in vitro* rate of Trp formation. A Normalized rate of Trp formation for a subset of the 118X/184X library. **B** Correlation between the normalized rate of Trp formation and enrichment ratio (output frequency / input frequency) for the subset of the 118X/184X library.

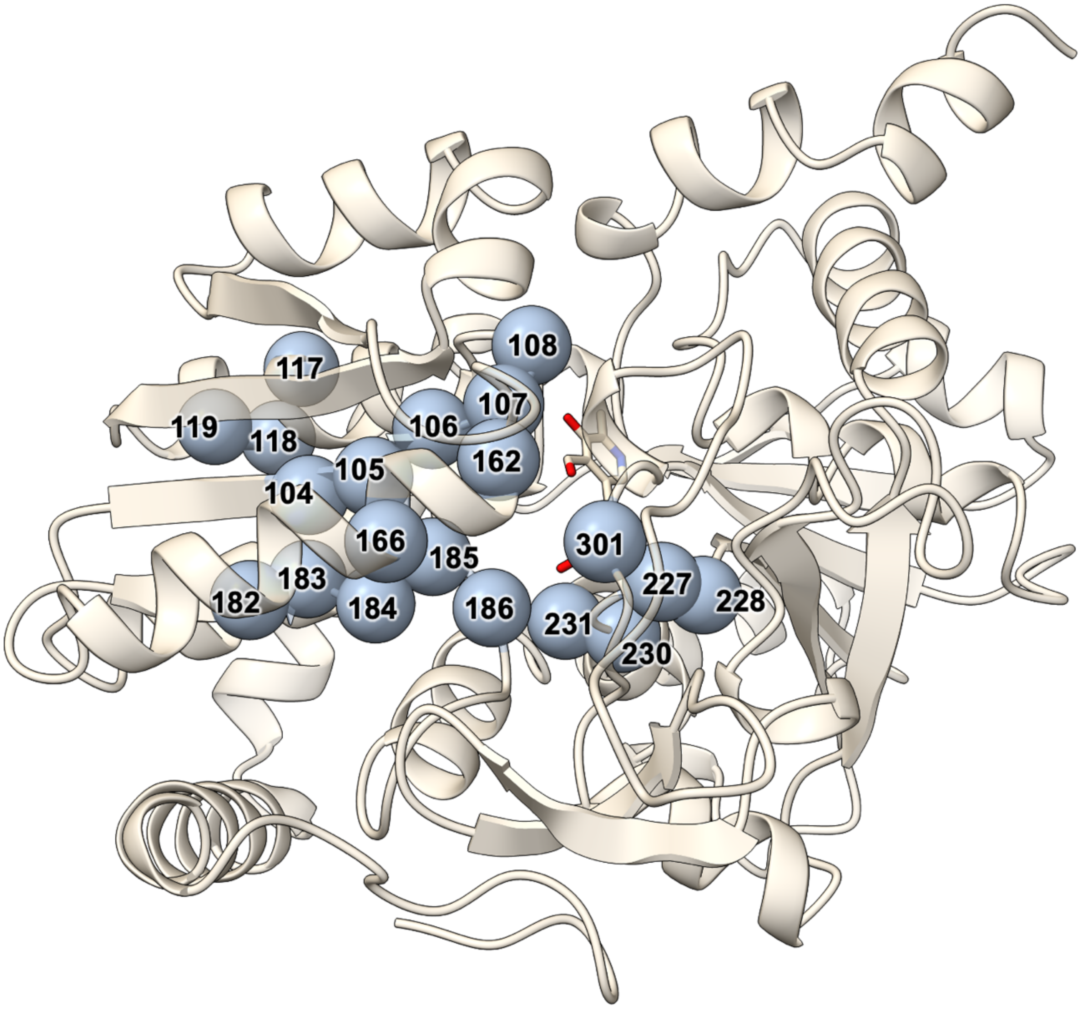


Figure B-6. All positions targeted within triple-site saturation libraries. A total of twenty different residues were targeted amongst nine triple-site saturation libraries. Residues were chosen to be near the active site or known to modulate the activity of TrpB.

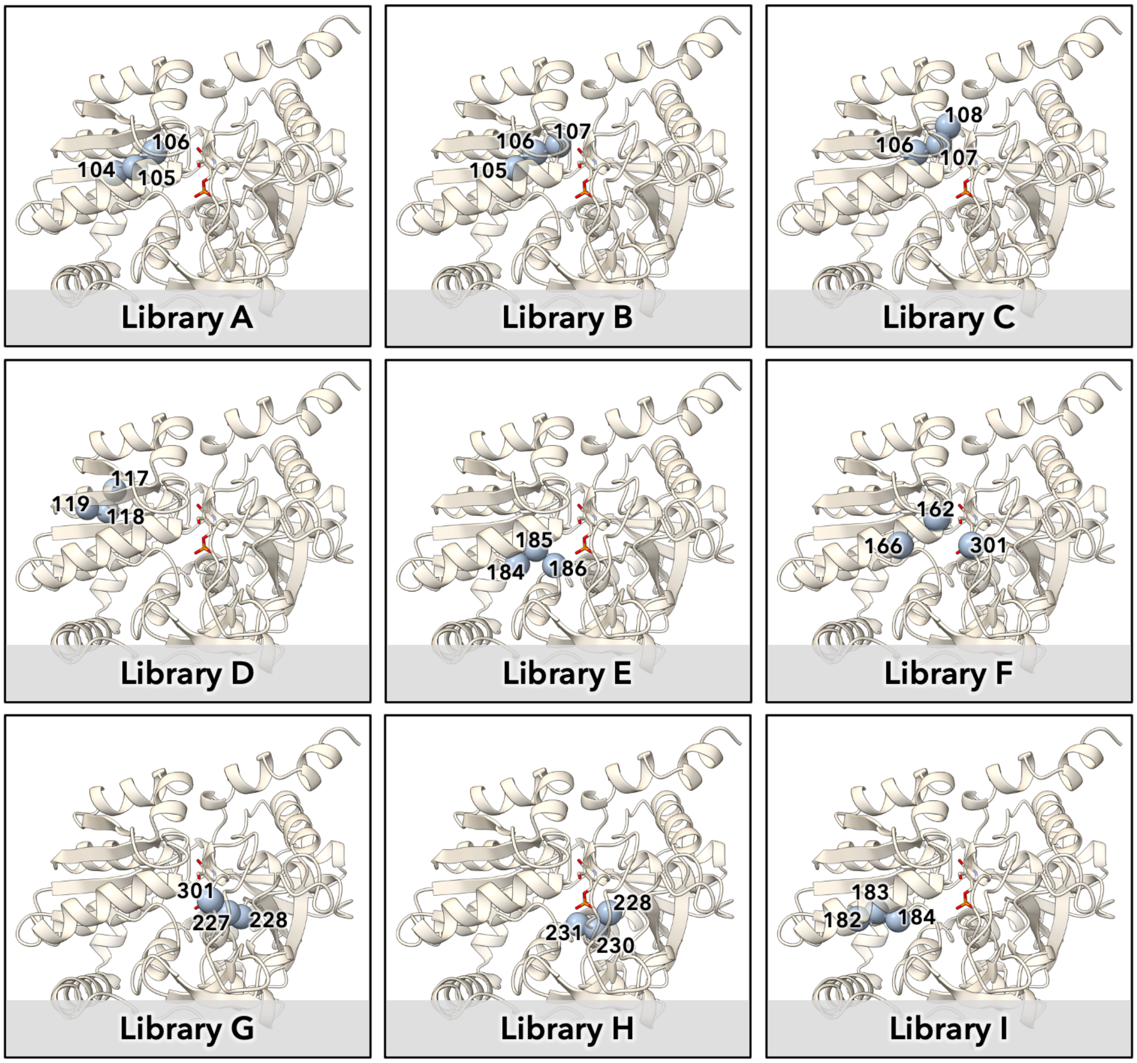


Figure B-7. Sets of residues chosen for the triple-site saturation libraries. Nine sets of three residues were targeted based on proximity to each other as well as each of construction with available molecular biology methods. Different numbers of replicates and timepoints were obtained for libraries A, B, and C vs libraries D, E, F, G, H, and I.

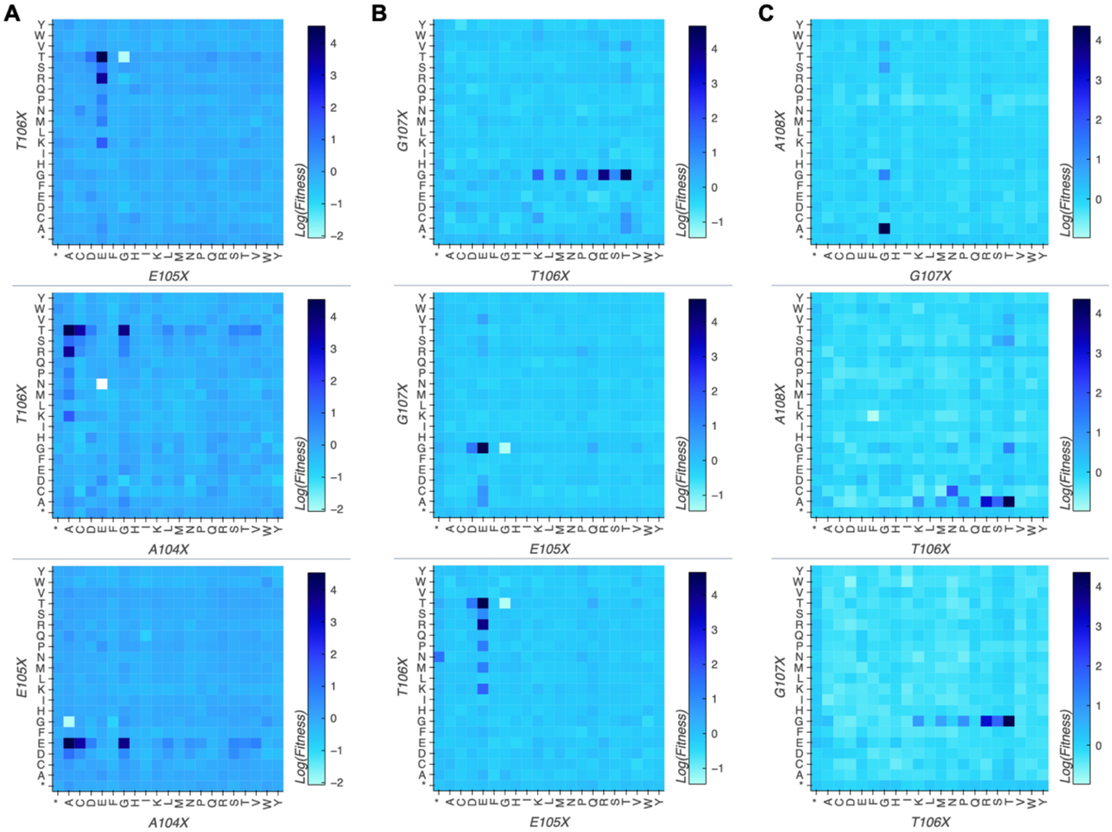


Figure B-8. Fitness for all pairs of residues for Libraries A, B, and C. Initial investigations into the utility of these sets of positions for larger libraries. Fitness was defined as the natural logarithm of the output frequency at T=44 over the input frequency. For each plot, the unplotted residue was held at the parent amino acid at that position, and for all three libraries very few amino acids are accepted at each position. **A** Library A separated into the three possible pairs of positions: 104 & 105, 105 & 106, and 104 & 106. **B** Library B separated into the three possible pairs of positions: 105 & 106, 106 & 107, and 105 & 107. **C** Library C separated into the three possible pairs of positions: 106 & 107, 107 & 108, and 106 & 108.

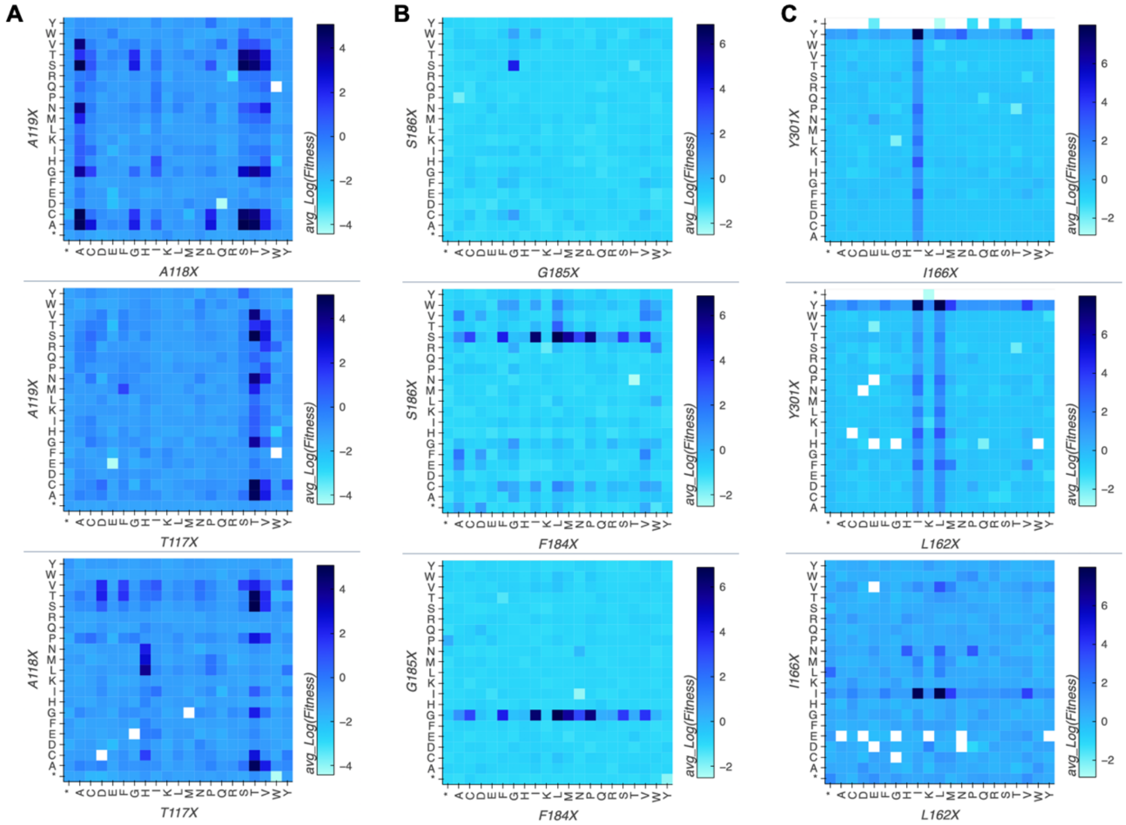


Figure B-9. Fitness for all pairs of residues for Libraries D, E, and F. Initial investigations into the utility of these sets of positions for larger libraries. Fitness was defined as the natural logarithm of the output frequency at T=36 over the input frequency and averaged between the two replicates. For each plot, the unplotted residue was held at the parent amino acid at that position. **A** Library D separated into the three possible pairs of positions: 117 & 118, 118 & 119, and 117 & 119. There is a relatively broad range of positions allowed at 118 and 119 while 117 allows many fewer. Interestingly, substitutions at 118 to Leu, Met, or Asp allow His to emerge as an option for 117. **B** Library E separated into the three possible pairs of positions: 184 & 185, 185 & 186, and 184 & 186. Position 184 shows a broad range of allowed amino acids while 185 and 186 are much more limited. **C** Library F separated into the three possible pairs of positions: 162 & 166, 166 & 301, and 162 & 301. This library allowed very few deviations from the parent sequence. Y301 appeared nearly mandatory as did I166 while either Leu or Iso were accepted at 162.

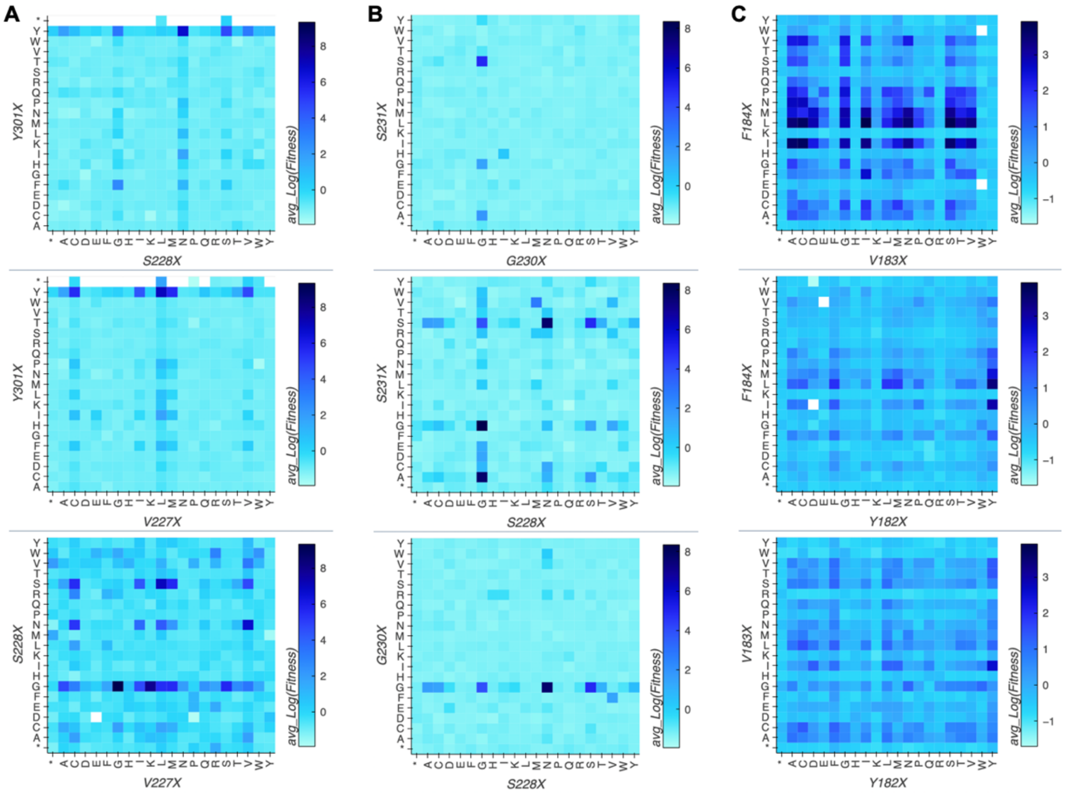


Figure B-10. Fitness for all pairs of residues for Libraries G, H, and I. Initial investigations into the utility of these sets of positions for larger libraries. Fitness was defined as the natural logarithm of the output frequency at T=36 over the input frequency and averaged between the two replicates. For each plot, the unplotted residue was held at the parent amino acid at that position. **A** Library G separated into all three possible pairs of residues: 227 & 228, 228 & 301, and 227 & 301. Once again Y301 appeared nearly mandatory. Positions 227 and 228 appeared to allow many more residues. One especially exciting observation was that G227 highly activating when paired with G228, but in the original S228 background it ablated activity. **B** Library H separated into all three possible pairs of residues: 228 & 230, 230 & 231, and 228 & 231. Very few amino acids were accepted at positions 230 and 231 while the range of those accepted at position 228 appeared similar to what was observed in library G. **C** Library I separated into all three possible pairs of residues: 183 & 184, 184 & 185, and 183 & 185. This library had many reasonably active variants, with some of the biggest improvements coming from the 183/184 pairing. Y182 appeared to be reasonably important for activity, but Phe, Leu, and Met were accepted at 182 to varying degrees with different residues at 183 and 184.

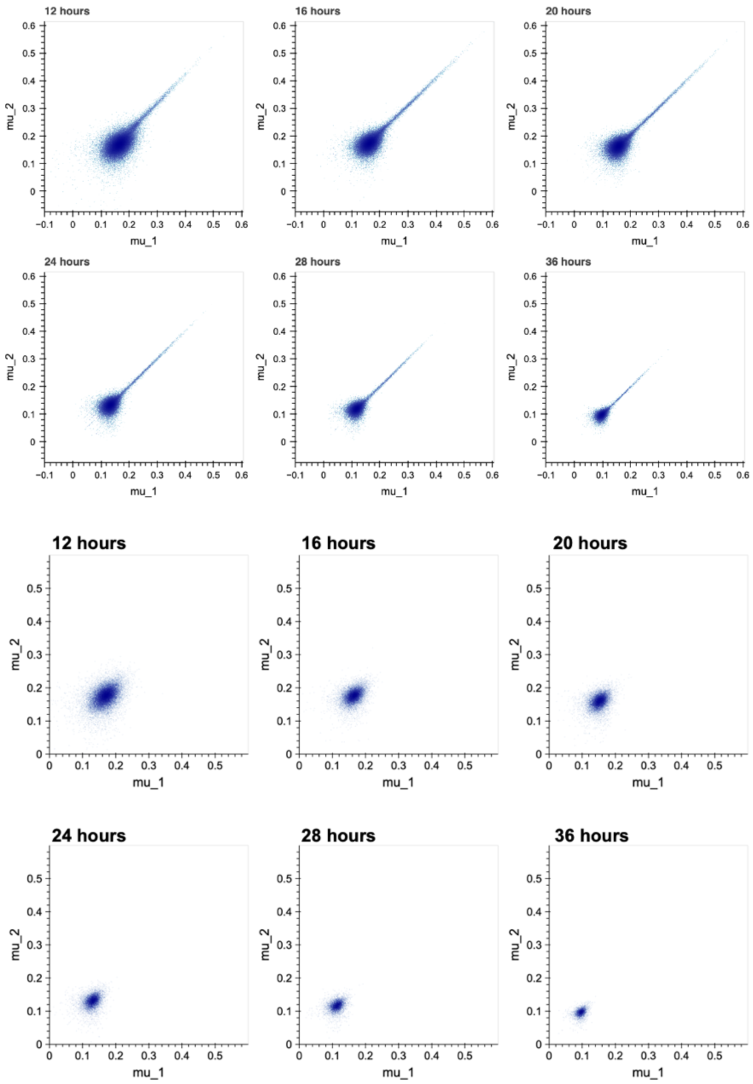


Figure B-11. Correlation of mu scores between replicates for all data and stop-codon sequences. Mu values were calculated as described by Kowalsky *et al*.[8] for all variants at all timepoints. In the top panel of six plots, mu for replicate 1 is plotted against mu for replicate 2 for all variants. In the bottom panel of six plots, mu for replicate 1 is plotted against mu for replicate 2 for just the stop codon-containing sequences. We observed that the distribution shrunk over time for both all variants as well as for stop codons, leading us to subtract the average mu for the stop codons within each replicate and timepoint and divide by the maximum mu value within that replicate and timepoint (mu_1-bg/max or mu_2-bg/max).

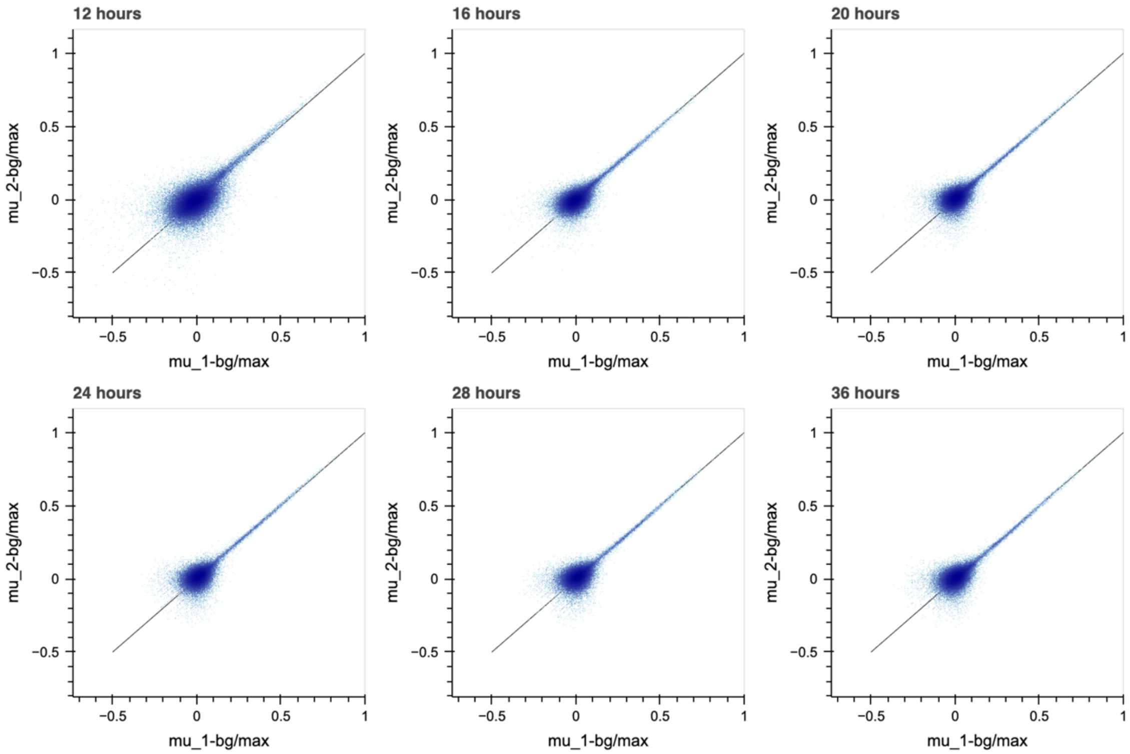


Figure B-12. Correlation of fitness scores between replicates for all timepoints. The scaled mu values for both replicates plotted against each other by timepoint overlaid on the identity line, y=x. There was a high degree of agreement between the replicates as well as between the timepoints.

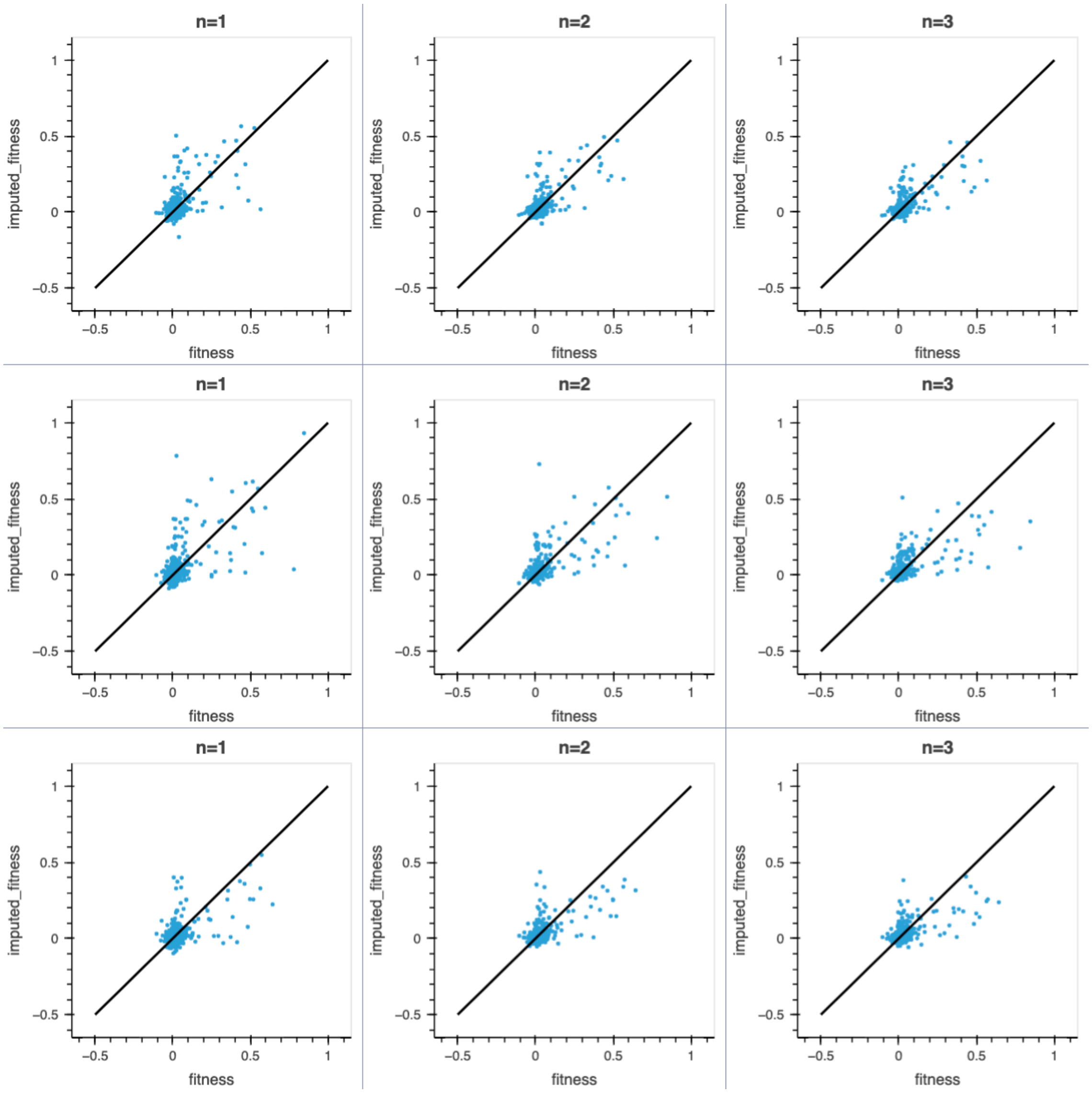
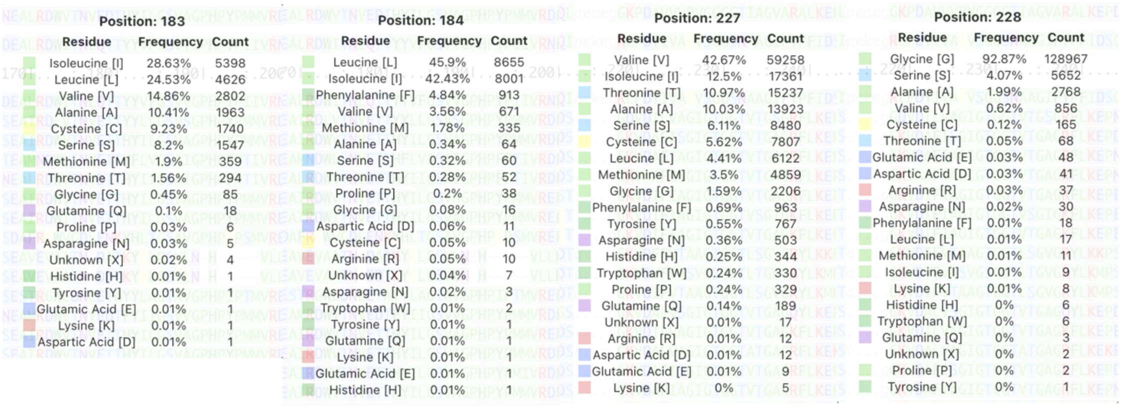


Figure B-13. Results of KNN imputation on 1,000 randomly ablated fitness scores. For three different sets of 1000 randomly ablated fitness scores, the KNN imputer was tested with weights=‘distance’ and n_neighbors = 1, 2, or 3. The imputed fitness values are plotted against the actual fitness values and overlaid with the line y=x. The ablated fitness values were restored and then the KNN imputer with n_neighbors = 2 was chosen to impute the 871 missing fitness values for the TrpB four-site landscape.



Position: 183

Residue	Frequency	Count
Isoleucine [I]	28.63%	5398
Leucine [L]	24.53%	4626
Valine [V]	14.86%	2802
Alanine [A]	10.41%	1963
Cysteine [C]	9.23%	1740
Serine [S]	8.2%	1547
Methionine [M]	1.9%	359
Threonine [T]	1.56%	294
Glycine [G]	0.45%	85
Glutamine [Q]	0.1%	18
Proline [P]	0.03%	6
Asparagine [N]	0.03%	5
Unknown [X]	0.02%	4
Histidine [H]	0.01%	1
Tyrosine [Y]	0.01%	1
Glutamic Acid [E]	0.01%	1
Lysine [K]	0.01%	1
Aspartic Acid [D]	0.01%	1

Position: 184

Residue	Frequency	Count
Leucine [L]	45.9%	8655
Isoleucine [I]	42.43%	8001
Phenylalanine [F]	4.84%	913
Valine [V]	3.56%	671
Methionine [M]	1.78%	335
Alanine [A]	0.34%	64
Serine [S]	0.32%	60
Threonine [T]	0.28%	52
Proline [P]	0.2%	38
Glycine [G]	0.08%	16
Aspartic Acid [D]	0.06%	11
Cysteine [C]	0.05%	10
Arginine [R]	0.05%	10
Unknown [X]	0.04%	7
Asparagine [N]	0.02%	3
Tryptophan [W]	0.01%	2
Tyrosine [Y]	0.01%	1
Glutamine [Q]	0.01%	1
Lysine [K]	0.01%	1
Glutamic Acid [E]	0.01%	1
Histidine [H]	0.01%	1

Position: 227

Residue	Frequency	Count
Valine [V]	42.67%	59258
Isoleucine [I]	12.5%	17361
Threonine [T]	10.97%	15237
Alanine [A]	10.03%	13923
Serine [S]	6.11%	8480
Cysteine [C]	5.62%	7807
Leucine [L]	4.41%	6122
Methionine [M]	3.5%	4859
Glycine [G]	1.59%	2206
Phenylalanine [F]	0.69%	963
Tyrosine [Y]	0.55%	767
Asparagine [N]	0.36%	503
Histidine [H]	0.25%	344
Tryptophan [W]	0.24%	330
Proline [P]	0.24%	329
Glutamine [Q]	0.14%	189
Unknown [X]	0.01%	17
Arginine [R]	0.01%	12
Aspartic Acid [D]	0.01%	12
Glutamic Acid [E]	0.01%	9
Lysine [K]	0%	5

Position: 228

Residue	Frequency	Count
Glycine [G]	92.87%	128967
Serine [S]	4.07%	5652
Alanine [A]	1.99%	2768
Valine [V]	0.62%	856
Cysteine [C]	0.12%	165
Threonine [T]	0.05%	68
Glutamic Acid [E]	0.03%	48
Aspartic Acid [D]	0.03%	41
Arginine [R]	0.03%	37
Asparagine [N]	0.02%	30
Phenylalanine [F]	0.01%	18
Leucine [L]	0.01%	17
Methionine [M]	0.01%	11
Isoleucine [I]	0.01%	9
Lysine [K]	0.01%	8
Histidine [H]	0%	6
Tryptophan [W]	0%	6
Glutamine [Q]	0%	3
Unknown [X]	0%	2
Proline [P]	0%	2
Tyrosine [Y]	0%	1

Figure B-14. Frequency of amino acids at each position of the quadruple-site landscape positions. EVCouplings was run with the webserver available at https://evcouplings.org/ to generate the multiple-sequence alignment used to determine the amino acid frequencies at each position.[20] These results are from the Tm9D8* sequence as an input and using a bitscore of 0.3.

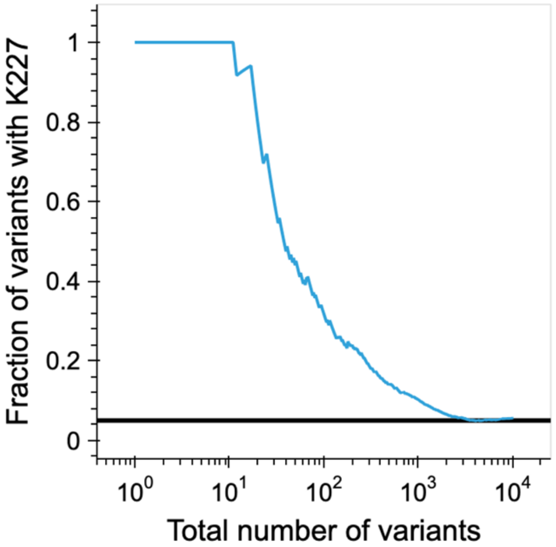


Figure B-15. Prevalence of K227 in the top sequences. The fraction of variants containing K227 with that ranking or better versus the ranking of the variant. All ten top sequences contained K227 and even among the top ~2,000 K227 remains overrepresented compared to the other nineteen possible amino acids. The black horizontal line is the expected fraction of sequences containing K227 (1/20) if sampling were random.

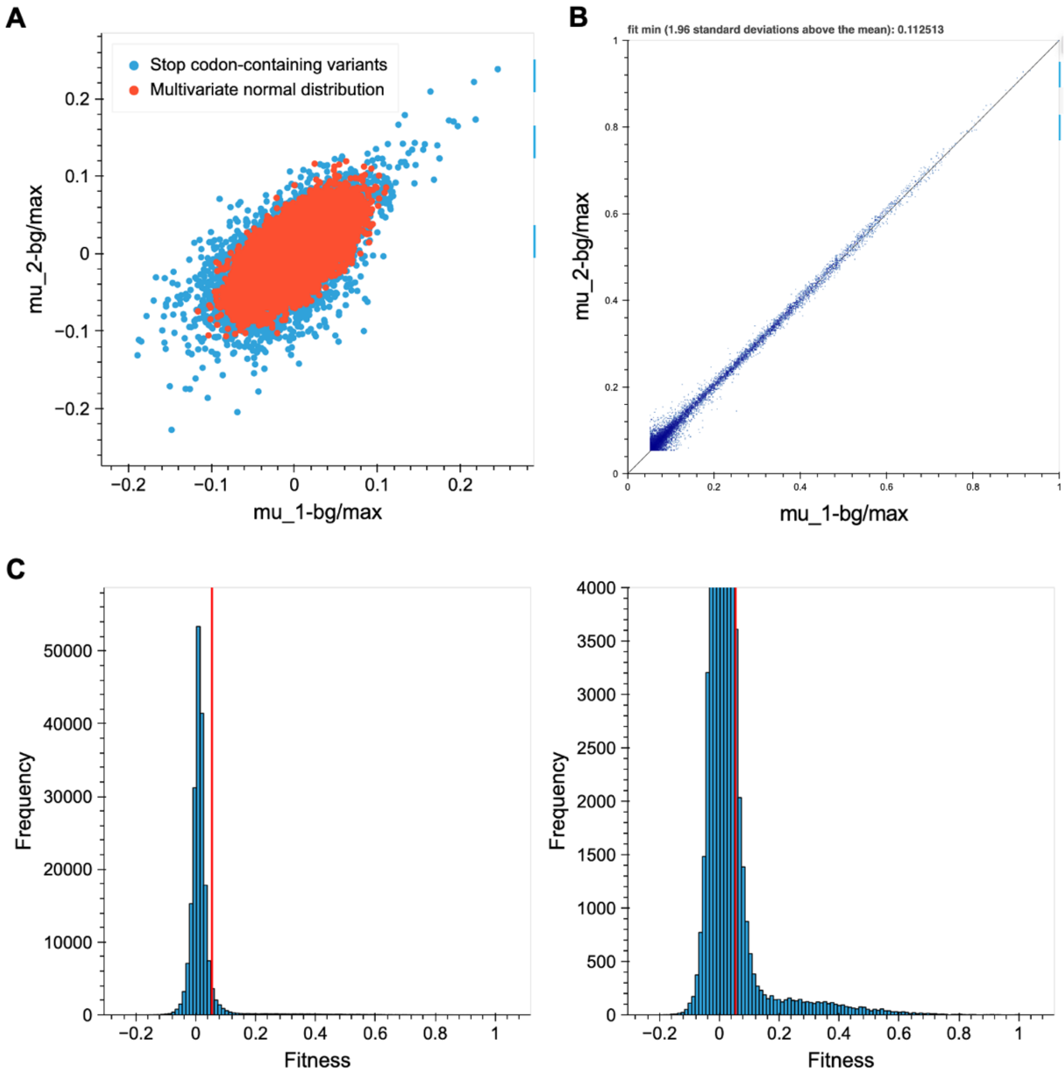


Figure B-16. Selecting an active/inactive threshold. A Correlation between fitness for the two replicates for stop-codon containing sequences (blue). This was overlaid with points sampled from a multivariate normal distribution based on the stop-codon distribution (red). This shows that the distribution is roughly normal. **B** Since it appeared that the fitness distribution was normal for each replicate, the active threshold was imposed such that the fitness of a variant was greater than the 95% confidence interval of stop-codon containing sequences for each replicate. **C** Histogram of the fitness values by averaging the two replications. The active/inactive threshold is displayed with a red, vertical line.

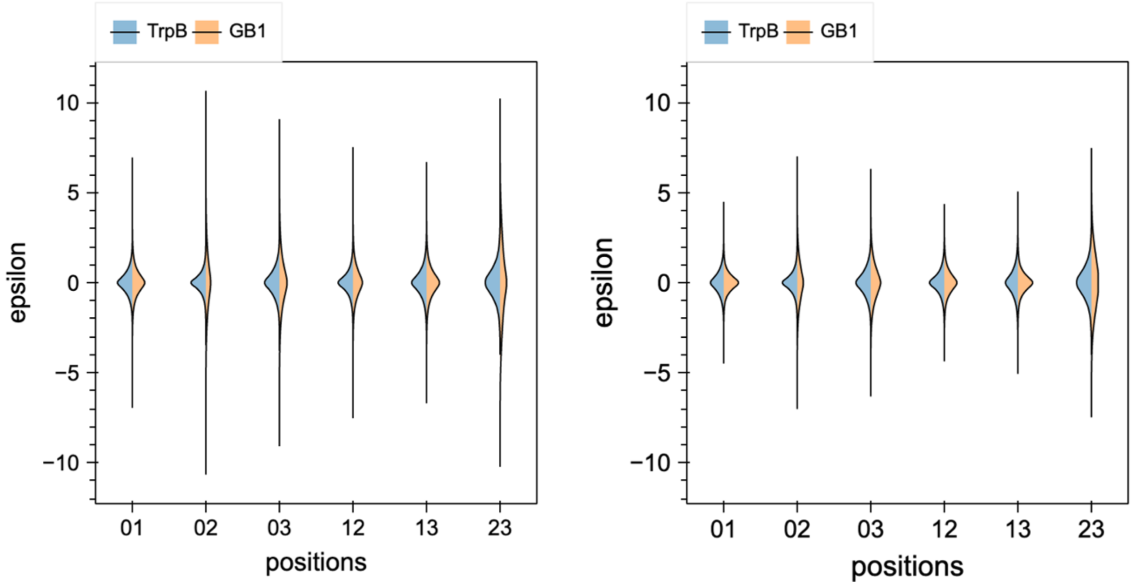


Figure B-17. Epsilon distribution by pair of positions for TrpB and GB1. Epsilon was calculated as described in Olson *et al.*[21] for each set of variants: $\epsilon = \ln\left(\frac{fit_{11}}{fit_{00}}\right) - \ln\left(\frac{fit_{01}}{fit_{00}}\right) - \ln\left(\frac{fit_{10}}{fit_{00}}\right)$ where 00 is the starting variant, 11 is the final variant with two substitutions, and 01 and 10 are the two single substitutions. Left: Epsilon distributions if all variants are required to be above the activity threshold. Right: Epsilon distributions if all variants are required to be in the top 9783 variants. Positions for TrpB and GB1 are listed in sequence order. TrpB: 0→183, 1→184, 2→227, 3→228. GB1: 0→39, 1→40, 2→41, 3→54. All following figures use the same nomenclature.

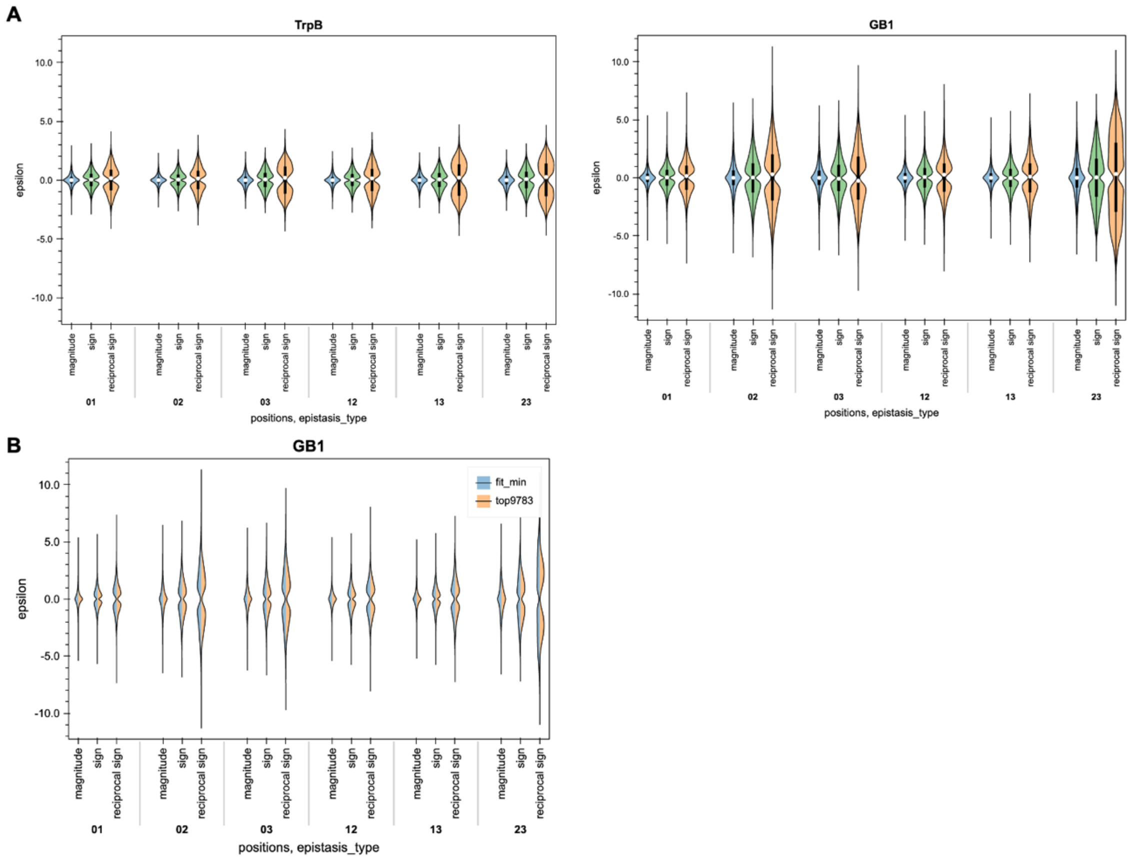


Figure B-18. Epsilon distributions by position pair and epistasis type. A Epsilon distributions for both TrpB and GB1 for sets of variants where the fitness of every variant is above the activity threshold. The distributions are symmetric because variants appear as both starting and final variants which results in an epsilon of equal magnitude and opposite direction. **B** Epsilon distributions for GB1 where each variant in the set is either above the activity threshold (blue) or in the top 9,783 variants (orange). For TrpB this does not change the distribution, so it is displayed only for GB1.

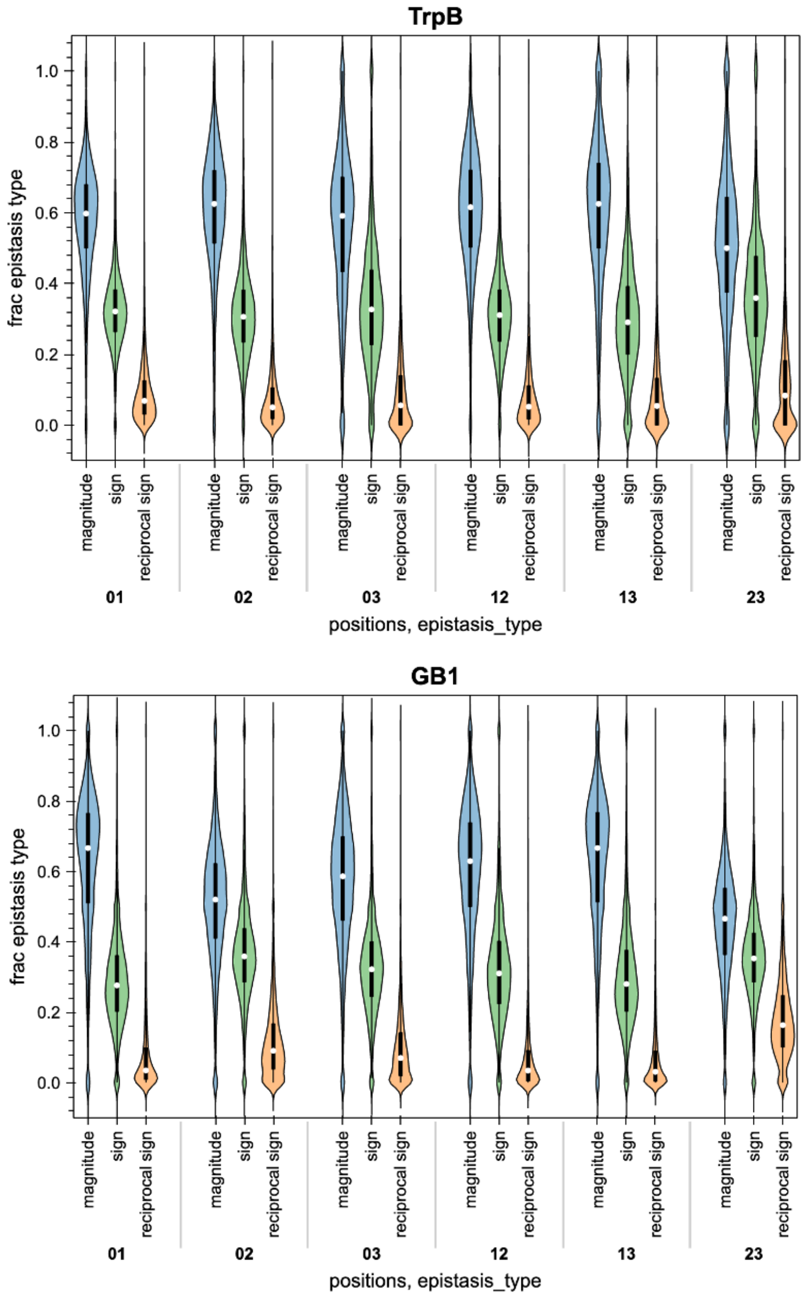


Figure B-19. Distribution of epistasis types by position pair. For all sequences above the respective activity thresholds, distributions of the fraction of each epistasis type grouped by pair of positions for both TrpB (upper) and GB1 (lower). All variants within a set were required to be above the activity threshold to determine the epistasis type. We see that the distributions are fairly similar across all position pairs.

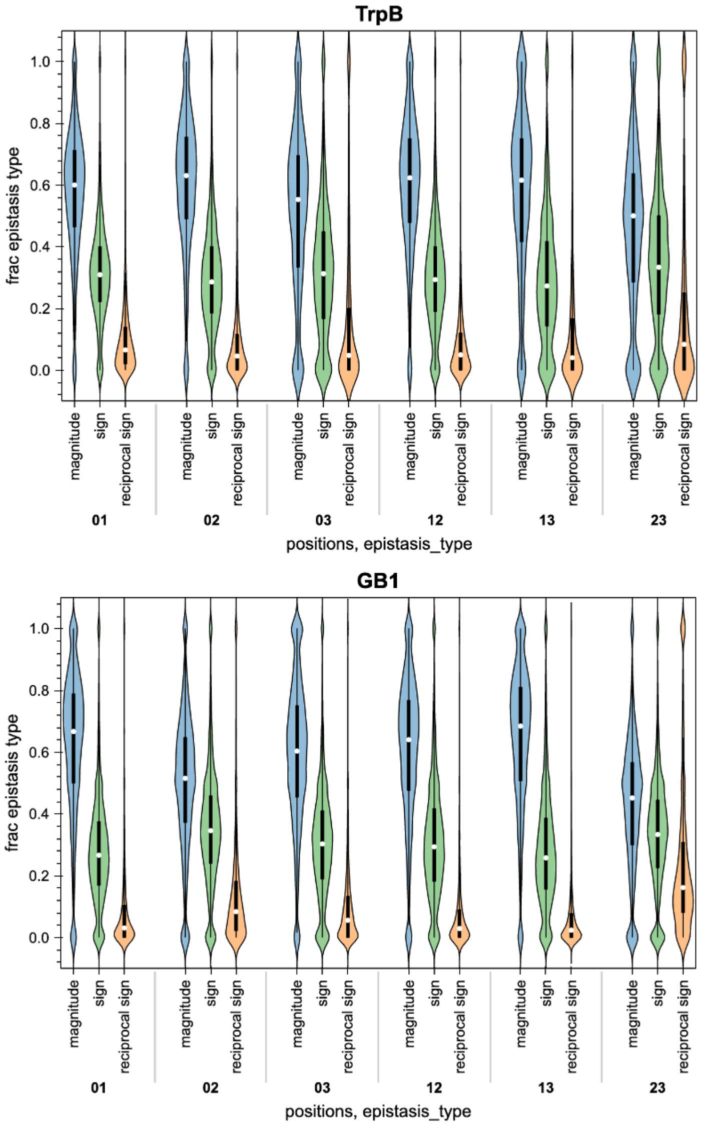


Figure B-20. Distribution of epistasis by position pair with final fitness > initial fitness. For all sequences above the respective activity thresholds, distributions of the fraction of each epistasis type grouped by pair of positions for both TrpB (upper) and GB1 (lower). All variants within a set were required to be above the activity threshold to determine the epistasis type. Additionally, it was enforced that the final fitness be greater than the initial fitness to investigate the distribution of epistasis for beneficial substitution pairs. Differences appeared to be relatively minor with a slight increase in the amount of magnitude epistasis across all positions pairs for both TrpB and GB1.

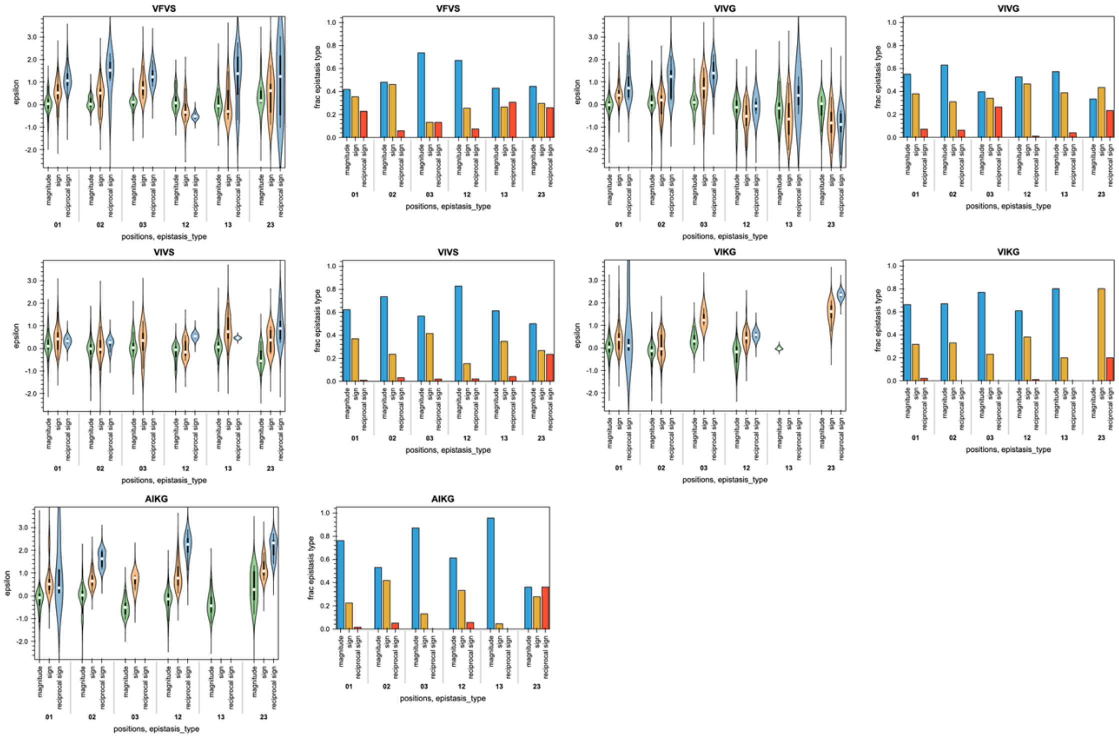


Figure B-21. Investigating the distribution of epsilon and epistasis type by variant. For the five variants involved in the path from parent (VFVS) to the best variant (AIKG), we plotted the distribution of epsilon separated by epistasis type and position pair as well as the fractions of epistasis types for each position pair. Although we had seen that the position pairs showed similar fractions of each epistasis type when examining all variants, when examined one background sequence at a time there is a lot of variation. VFVS and VIVG appear to have reciprocal sign epistasis for all position pairs, while VIKG has almost none. There is also much more variation in the fractions of epistasis between each of the pairs of positions for each background. For example, VIVS exhibits very little reciprocal sign epistasis except for between residue 227 (2) and 228 (3). For all sets, all variants were required to have fitness above the activity threshold, which is why some epsilon distributions are empty (no sets existed where all variants were above the threshold).

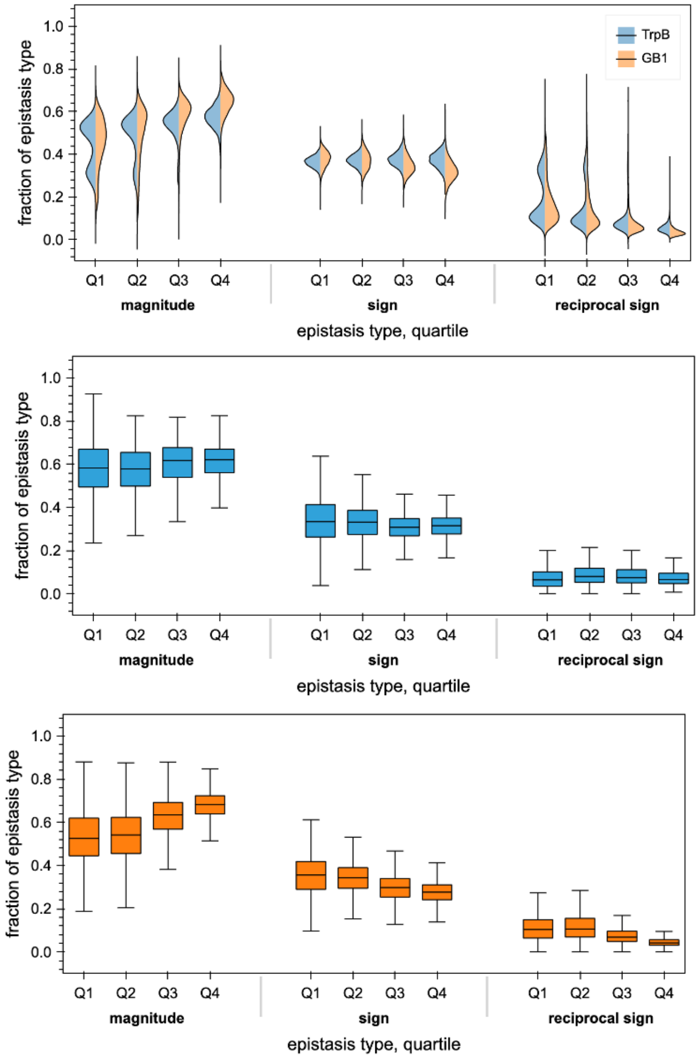


Figure B-22. Further information on the pairwise epistasis distributions across quartiles. Top: In the main text, when classifying types of epistasis, all variants were required to be above the activity threshold. If any variants can be used to classify epistasis (using only active variants as starting sequences), the distributions change, potentially due to noise impacting proper classification. More reciprocal sign epistasis is reported for TrpB (blue) while the GB1 (orange) distributions change less. Middle: Box and whisker plot of the data presented in **Figure 3-2B** for TrpB. Bottom: Box and whisker plot of the data presented in **Figure 3-2B** for GB1.

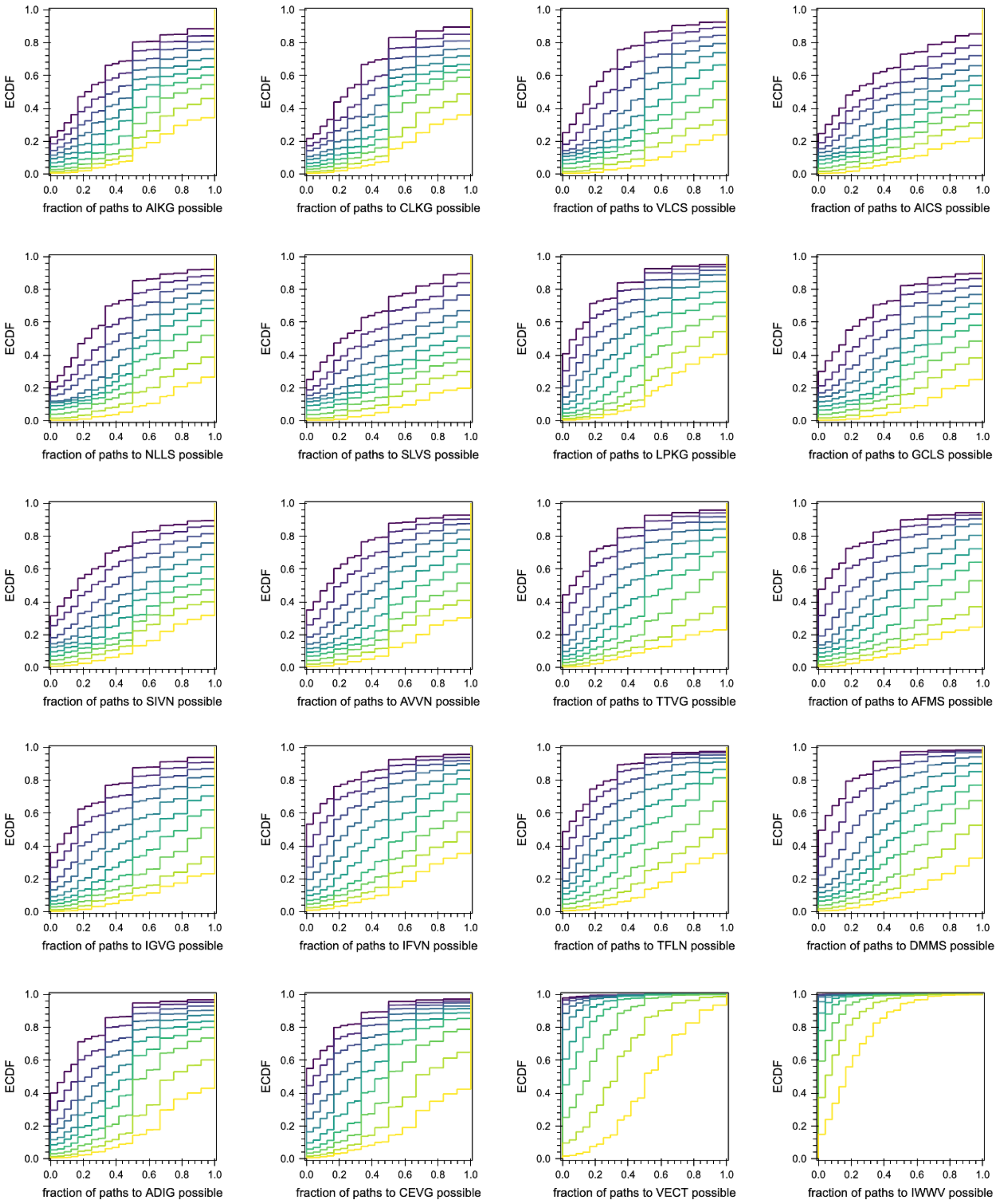


Figure B-23. Analysis of paths to each local optima. Fractions and ECDFs of active variants with at least one upward path to each of the top twenty local optima given the allowed fitness decreases. Most ECDFs appear roughly the same as that presented for AIKG with varying widths of distributions. However, the final two optima presented, VECT and IWWV, appear to be much more inaccessible than expected.

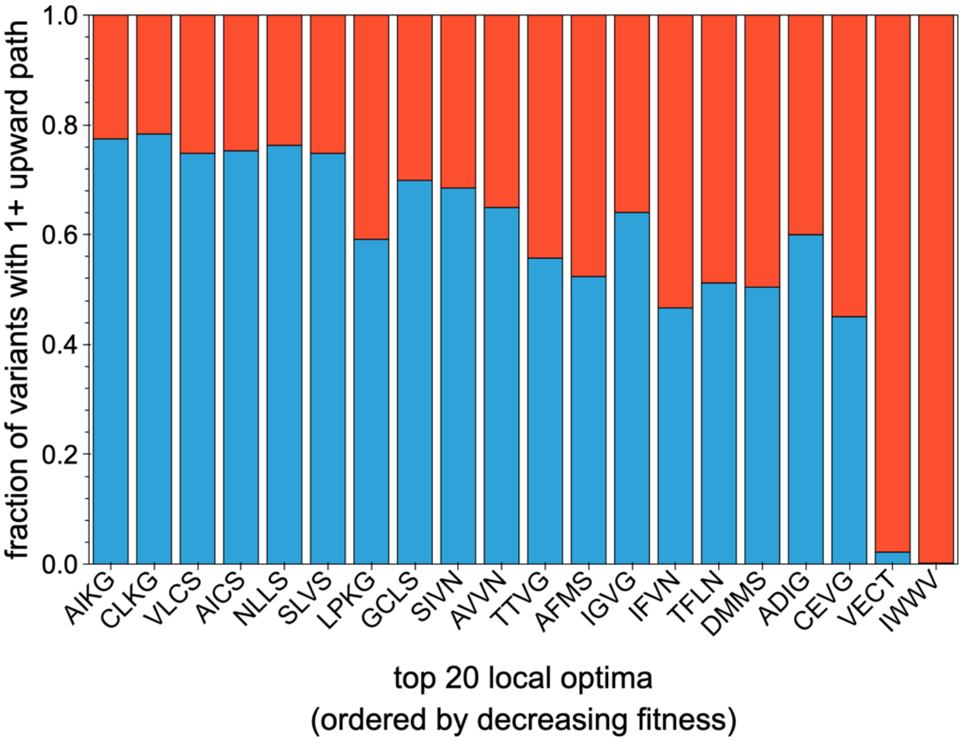


Figure B-24. Fraction of active variants with at least one upward path to each local optima. Allowing no decreases in fitness, this is the fraction of starting variants that have at least one upward path to each of the top 20 local optima (blue). The fraction of starting variants that do not have even one upward path to the local optima is in red. There is a general trend that as the fitness of the local optima decreases it becomes less accessible, likely because some of the paths require variants with higher fitness than the optima, which makes them inaccessible. However, many of these local optima are still similarly accessible as AIKG, meaning they could trap evolution campaigns.

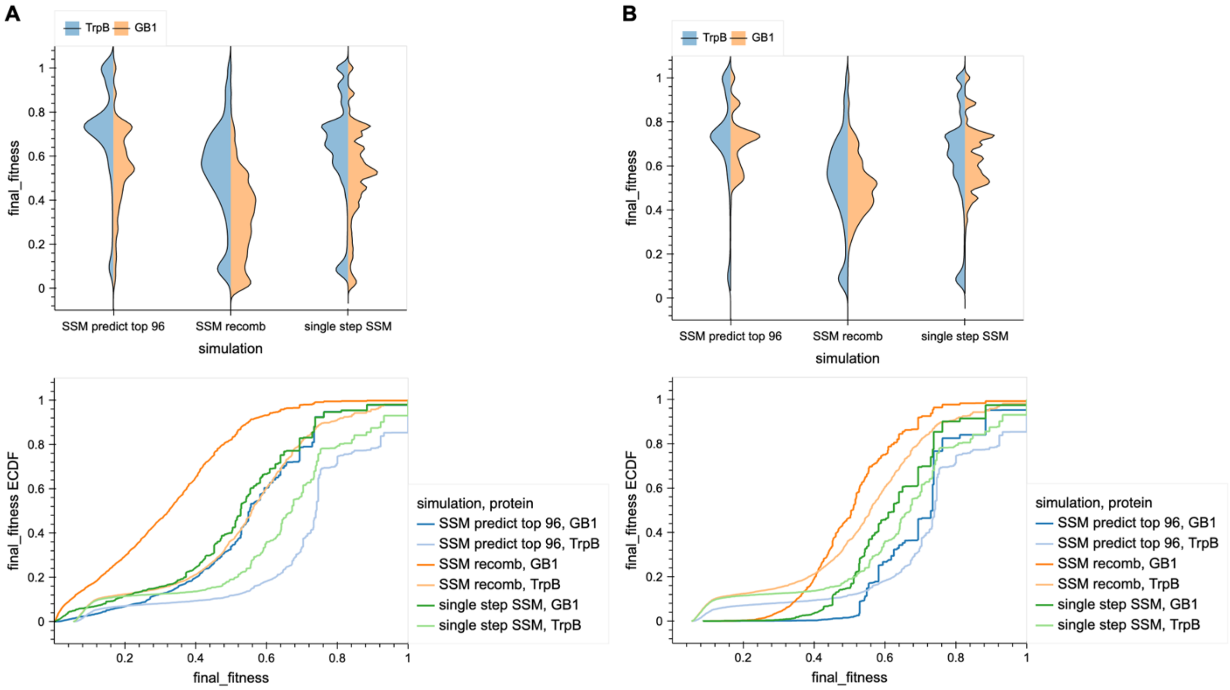


Figure B-25. Comparing simulation results with varying activity cutoffs for GB1 data. Presented in the main text, we show the simulation results starting from each of the top 9,783 non-imputed variants. We observed significant differences in the performance based on the starting cutoffs imposed for GB1 that made comparison between GB1 and TrpB difficult. **A** A comparison of the directed evolution simulation results starting from all variants above the respective activity thresholds for TrpB and GB1. In these results, the maximum fitness achieved is generally much higher for TrpB. However, the minimum fitness allowed for TrpB is ~1/20 the max while for GB1 it is ~1/1,000 the max. This means a given starting point for TrpB is fewer fold-improvements from the maximum to begin with. **B** Alternatively, the simulations can be run for GB1 using the same fraction of the maximum fitness as a cutoff. In this case, only 5587 variants from the GB1 landscape are within this cutoff. This makes the simulation results between TrpB and GB1 much more similar, with GB1 having fewer campaigns trapped at very low fitness (~0–0.3), but more trapped and medium fitness (~0.3–0.8).

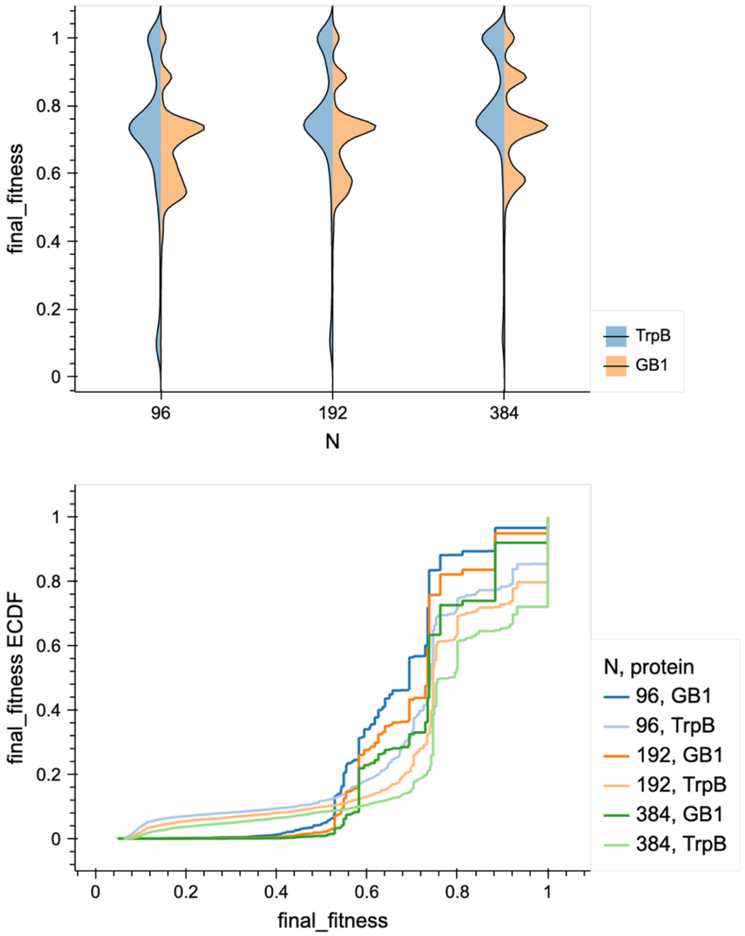


Figure B-26. Varying the number of exact variants tested in SSM predict top *N*. Since many machine learning-assisted directed evolution approaches require direct synthesis of *N* variants for a prediction round, we tested if increasing *N* improved the max fitness achieved via choosing top variants with SSM and recombination based on additivity. Testing more variants did shift the distributions slightly upward, but the return was quite small for a 2X or 3X increase in required synthesis costs.

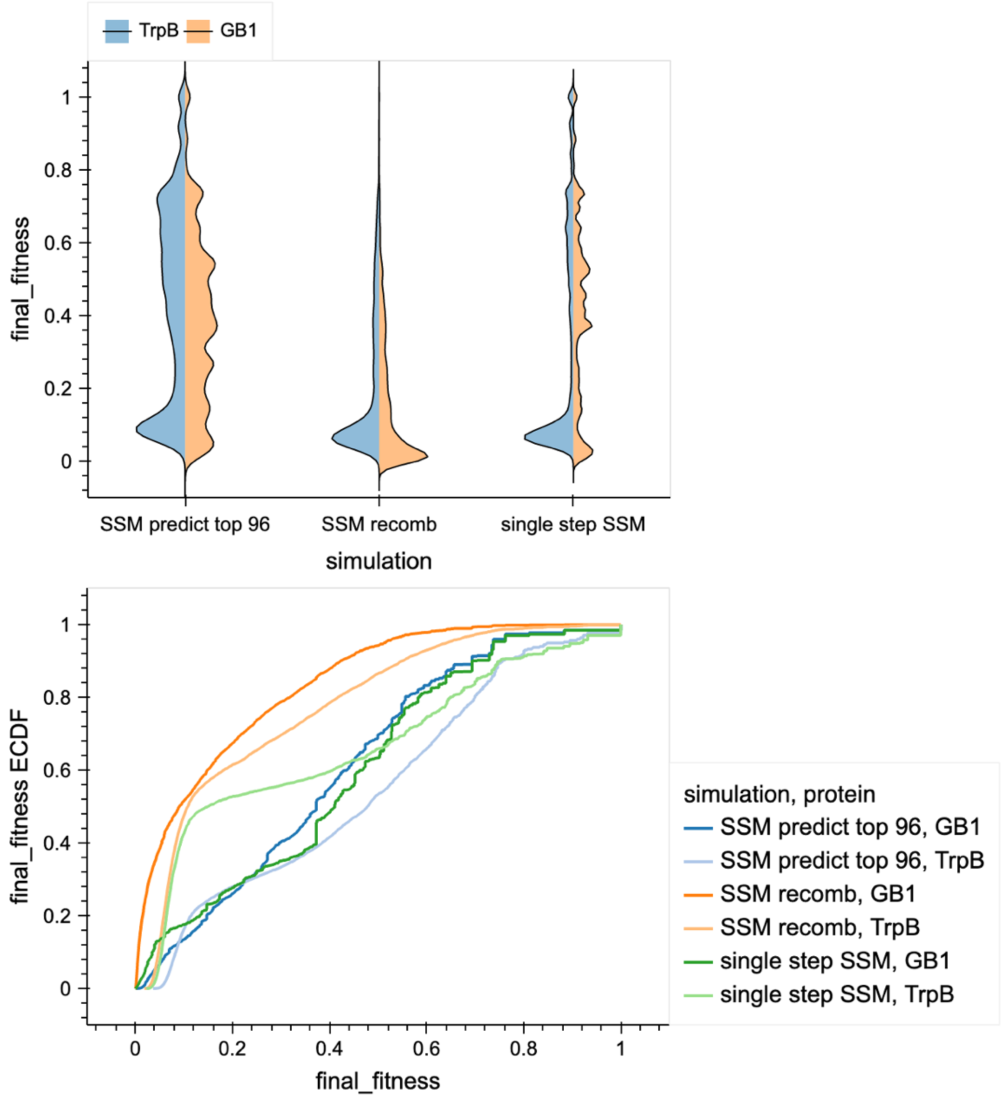


Figure B-27. Directed evolution simulations from any variant with fitness >0 in the landscape. As a final comparison, we tested how well simulations did starting from any random variant in the landscape of either TrpB or GB1. As expected, the performance was significantly worse, especially for site-saturation mutagenesis + recombine best (middle violin). Single-step site saturation and site-saturation predict top 96 performed similarly for GB1, but single-step SSM was by far the best for TrpB, likely because it allowed escape from the distribution of inactive variants. This suggests that a single-step walk may be a more robust approach, especially for early evolution campaigns when activity is near-noise levels even though SSM predict top 96 did somewhat better when starting from the detectably active variants.

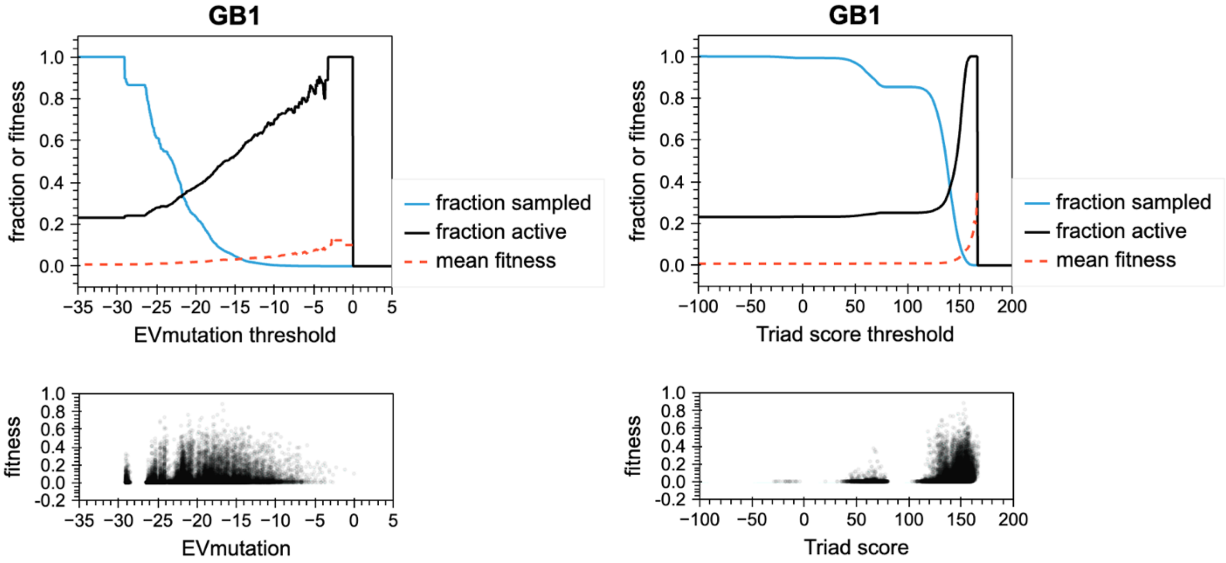


Figure B-28. Zero-shot fitness predictor performance for the GB1 binding protein. Left: EVmutation, Right: Triad with Rosetta energy function. EVmutation appears to be able to increase the fraction of active variants somewhat, but it barely increases the mean fitness achieved. Alternatively, Triad appears to be able to increase both the fraction of active variants sampled as well as the mean fitness.

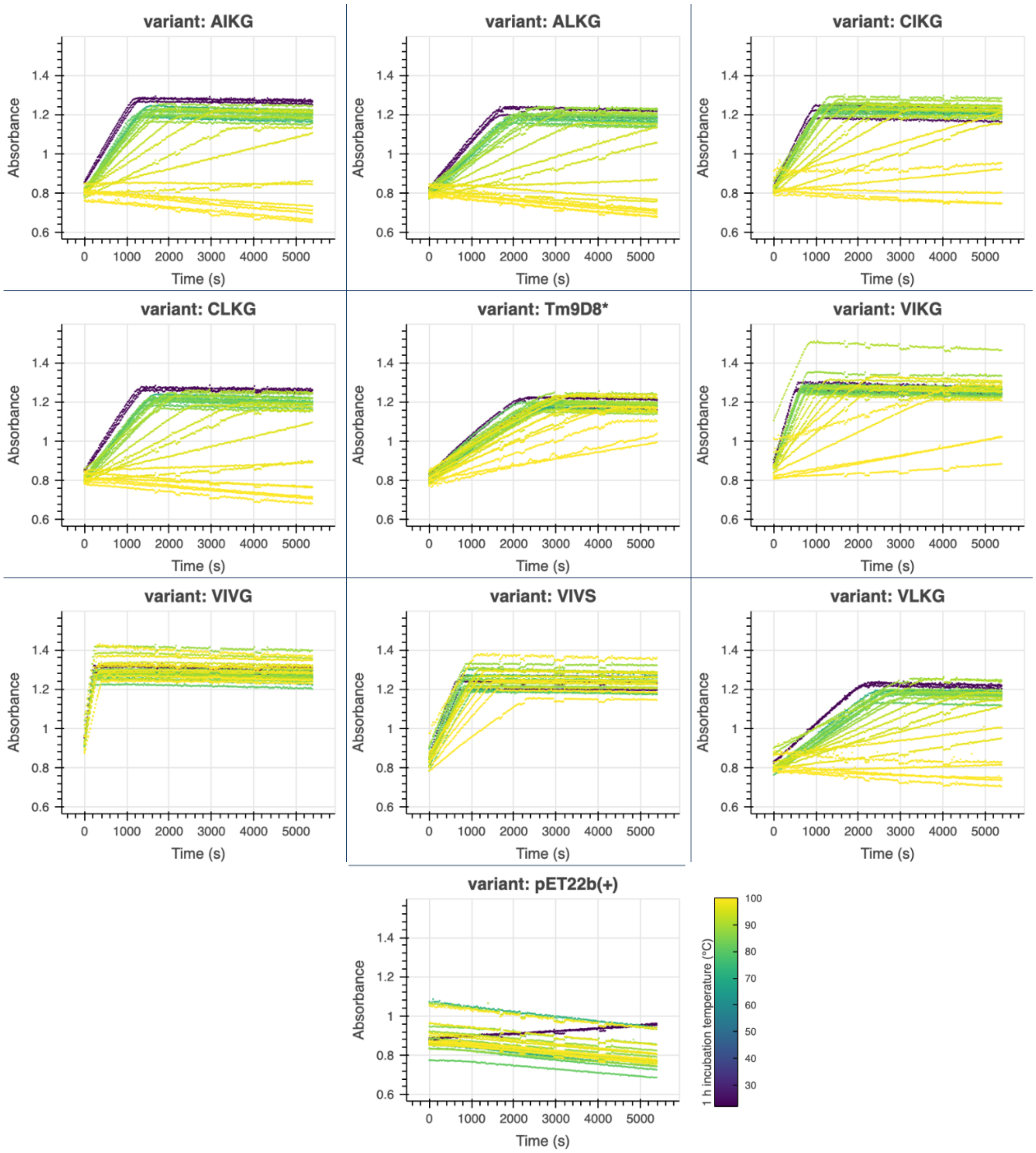


Figure B-29. *T_{50}* absorbance vs time (s) plots colored by temperature for each variant. Across three different collection plates, lysate harboring each variant was incubated for 1 h at a temperature between room temperature at 99 °C. Lysate was spun down and then added to a reaction mix containing indole and serine, and absorbance at 290 nm was collected over time. It is clear that some variants lose activity with temperature whiles others barely change (VIVG). The initial rates were captured with linear fits.

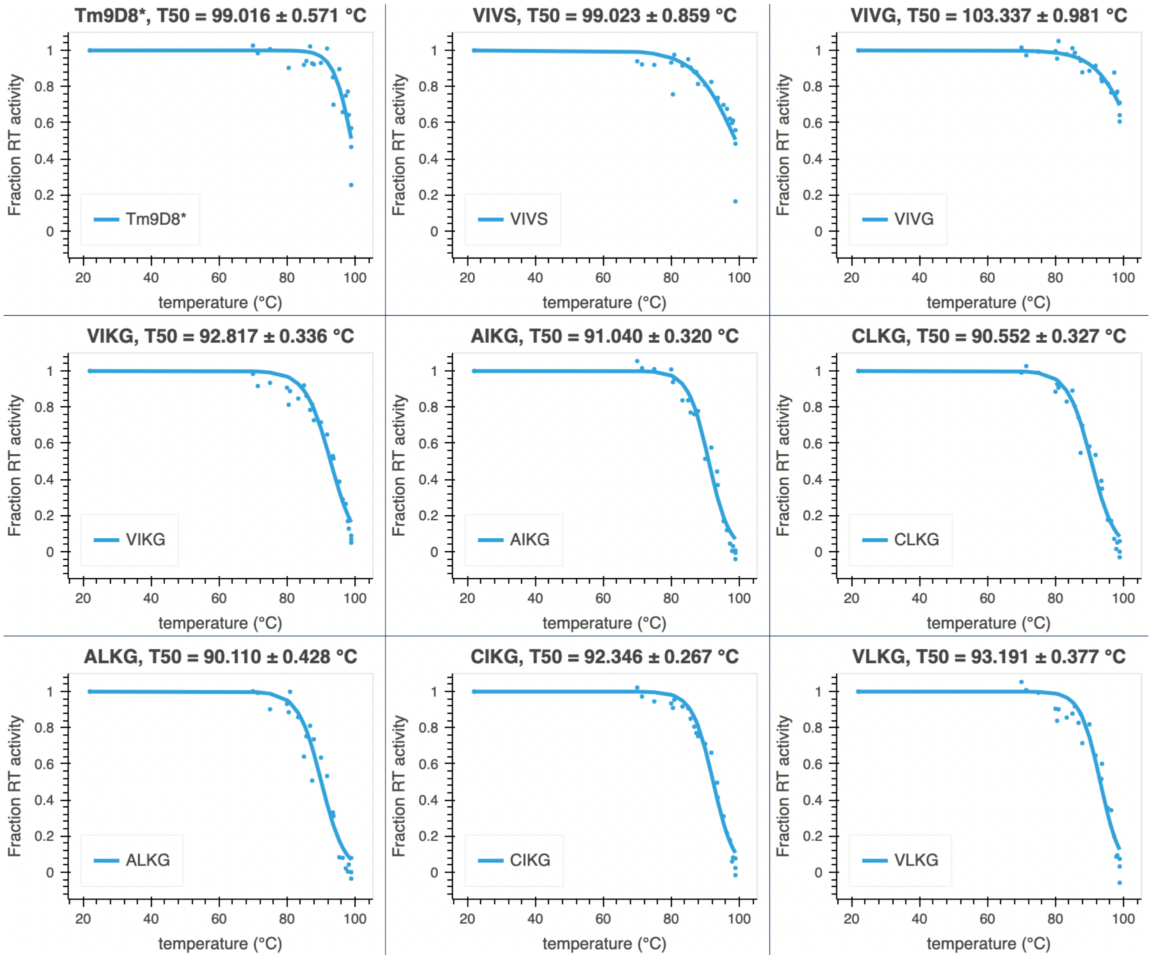


Figure B-30. Sigmoid fits for fraction of RT activity vs 1 h incubation temperature. Initial rates were divided by the initial rate of the room temperature-incubated sample and plotted versus temperature. These data were fit with a sigmoid that was used to estimate the T_{50} for each variant. No measurements were taken over an incubation temp of 99 °C.

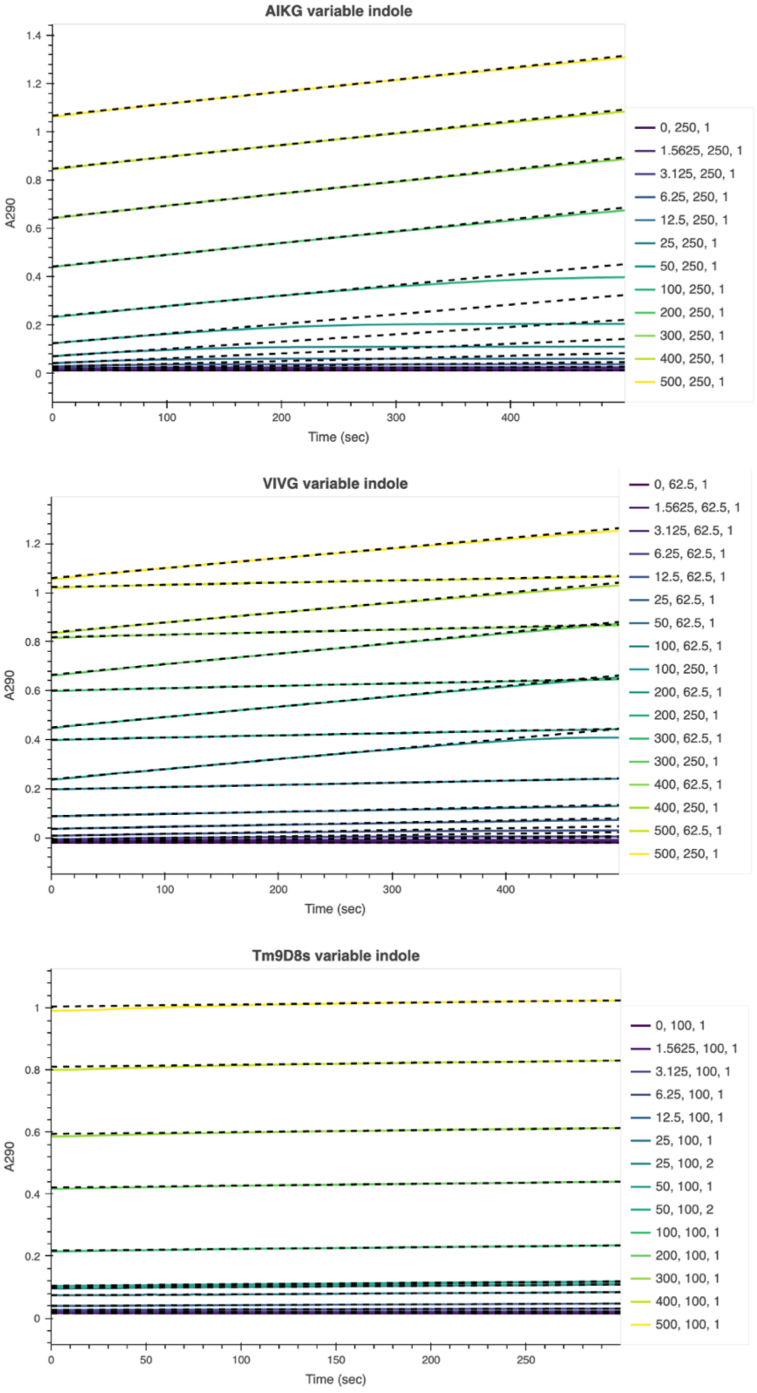


Figure B-31. Fits for variable indole Michaelis-Menten rate estimation data. Initial rates were estimated from linear or exponential rates. More details provided in the associated GitHub repository where data processing notebooks can be found.

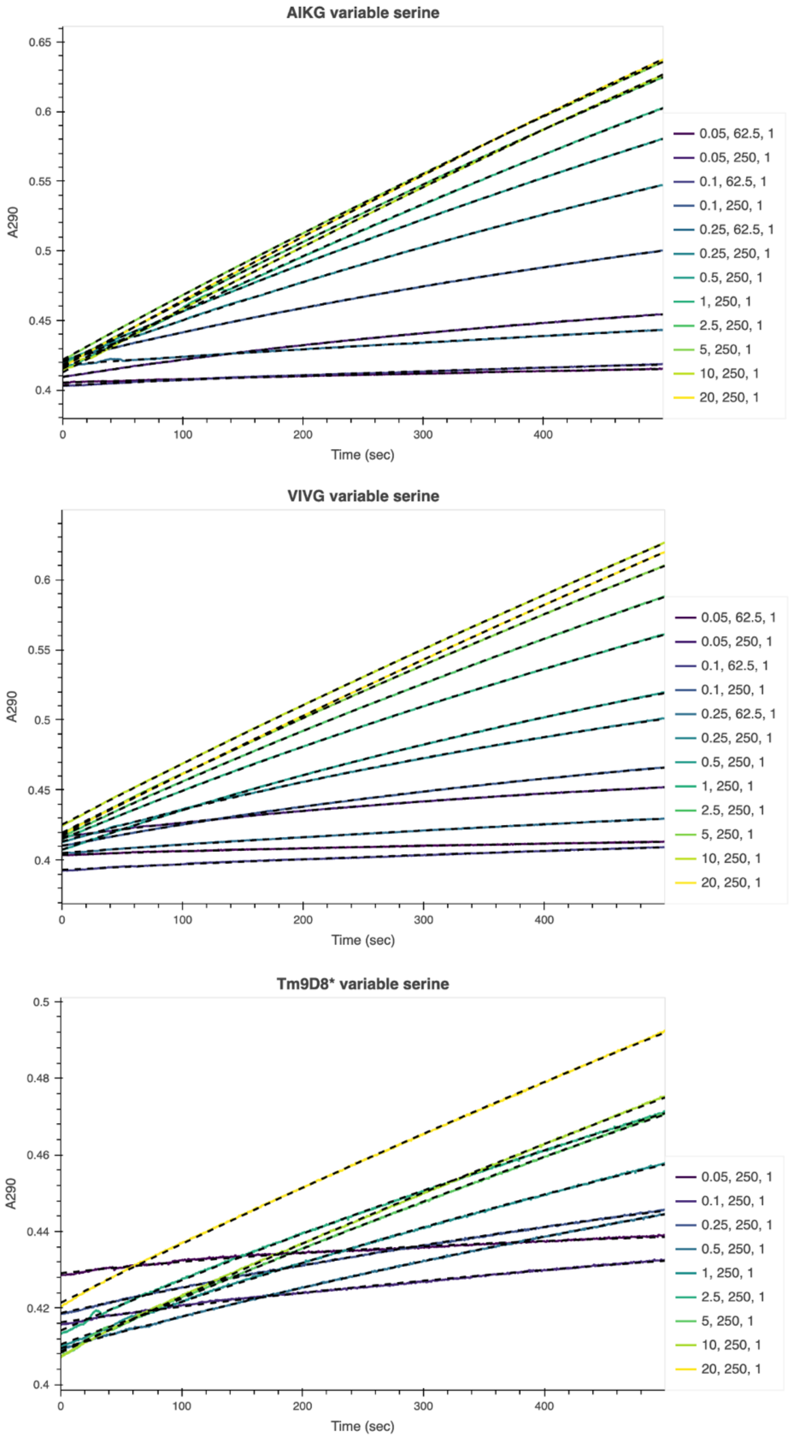


Figure B-32. **Fits for variable serine Michaelis-Menten rate estimation data.** Initial rates were estimated from linear or exponential rates. More details provided in the associated GitHub repository where data processing notebooks can be found.

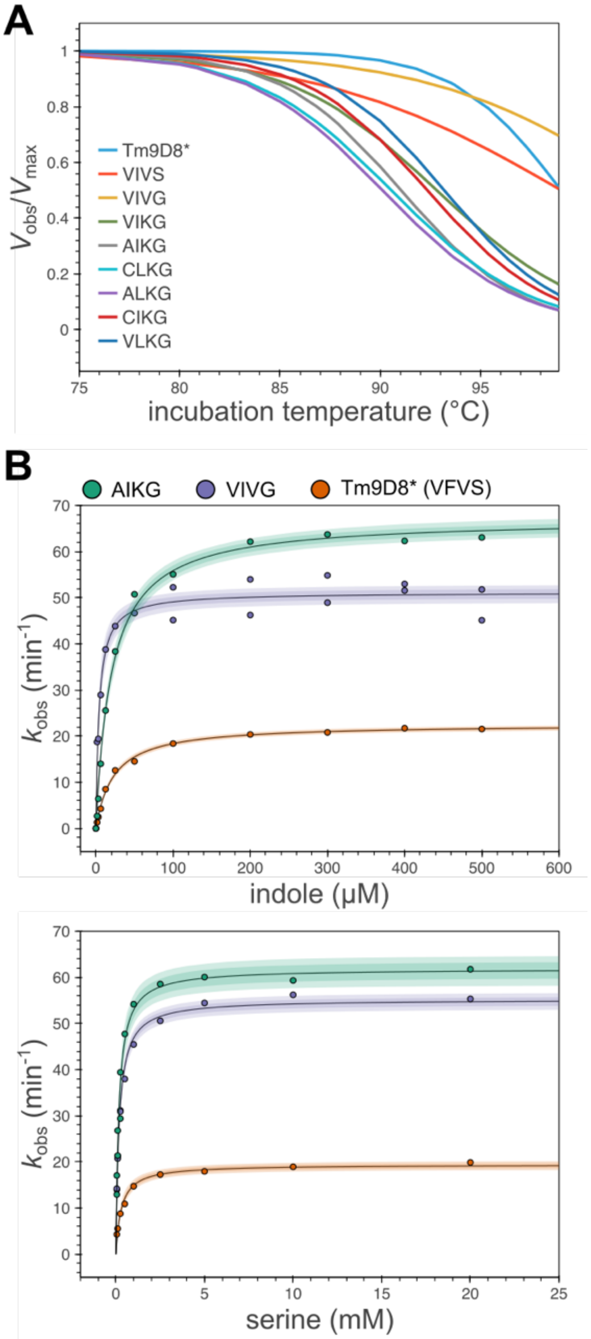


Figure B-33. **Biochemical investigation of the top variants.** Stability and activity measurements for select variants. **A** T_{50} curves for all variants in the single possible path from *Tm*9D8* to AIKG as well as the top five variants: AIKG, CLKG, ALKG, CIKG, and VLKG (in order of fitness). **B** Michaelis-Menten curves for *Tm*9D8*, VIVG, and AIKG with different indole concentrations at 20 mM Ser (left) and different Ser concentrations at 200 μM indole (right).

Appendix B Bibliography

1. Datsenko, K. A. & Wanner, B. L. One-step inactivation of chromosomal genes in Escherichia coli K-12 using PCR products. *Proc. Natl. Acad. Sci. U.S.A.* **97**, 6640–6645 (2000).
2. Baba, T. *et al.* Construction of Escherichia coli K-12 in-frame, single-gene knockout mutants: the Keio collection. *Mol. Syst. Biol.* (2006) doi:10.1038/msb4100050.
3. Boville, C. E., Romney, D. K., Almhjell, P. J., Sieben, M. & Arnold, F. H. Improved synthesis of 4-cyanotryptophan and other tryptophan analogues in aqueous solvent using variants of TrpB from *Thermotoga maritima*. *J. Org. Chem.* (2018) doi:10.1021/acs.joc.8b00517.
4. Gibson, D. G. *et al.* Enzymatic assembly of DNA molecules up to several hundred kilobases. *Nat. Methods* **6**, 343–345 (2009).
5. Wittmann, B. J., Johnston, K. E., Almhjell, P. J. & Arnold, F. H. evSeq: Cost-effective amplicon sequencing of every variant in a protein library. *ACS Synth. Biol.* **11**, 1313–1324 (2022).
6. SOB Medium. *Cold Spring Harb. Protoc.* **2018**, pdb.rec102723 (2018).
7. Kille, S. *et al.* Reducing codon redundancy and screening effort of combinatorial protein libraries created by saturation mutagenesis. *ACS Synth. Biol.* **2**, 83–92 (2013).
8. Kowalsky, C. A. *et al.* High-resolution sequence-function mapping of full-length proteins. *PLOS ONE* **10**, e0118193 (2015).
9. Wu, N. C., Dai, L., Olson, C. A., Lloyd-Smith, J. O. & Sun, R. Adaptation in protein fitness landscapes is facilitated by indirect paths. *eLife* **5**, e16965 (2016).
10. Hagberg, A., A., Schult, D. A. & Swart, P. J. Exploring Network Structure, Dynamics, and Function using NetworkX. in *Proceedings of the 7th Python in Science Conference (SciPy 2008)* 11–15 (Gäel Varoquaux, Travis Vaught, and Jarrod Millman, 2008).
11. Wittmann, B. J., Yue, Y. & Arnold, F. H. Informed training set design enables efficient machine learning-assisted directed protein evolution. *Cell Syst.* **12**, 1026-1045.e7 (2021).
12. Hopf, T. A. *et al.* Mutation effects predicted from sequence co-variation. *Nat. Biotechnol.* **35**, 128–135 (2017).
13. Kabsch, W. XDS. *Acta Cryst. D* **66**, 125–132 (2010).

14. Winn, M. D. *et al.* Overview of the CCP4 suite and current developments. *Acta Crystallogr. D Biol. Crystallogr.* **67**, 235–242 (2011).

15. McCoy, A. J. *et al.* Phaser crystallographic software. *J. Appl. Cryst.* **40**, 658–674 (2007).

16. Emsley, P., Lohkamp, B., Scott, W. G. & Cowtan, K. Features and development of Coot. *Acta Cryst. D Biol. Crystallogr.* **66**, 486–501 (2010).

17. Adams, P. D. *et al.* PHENIX: a comprehensive Python-based system for macromolecular structure solution. *Acta Cryst. D Biol. Crystallogr.* **66**, 213–221 (2010).

18. Chen, V. B. *et al.* MolProbity: all-atom structure validation for macromolecular crystallography. *Acta Cryst. D Biol. Crystallogr.* **66**, 12–21 (2010).

19. Laskowski, R. A., MacArthur, M. W., Moss, D. S. & Thornton, J. M. PROCHECK: a program to check the stereochemical quality of protein structures. *J. Appl. Cryst.* **26**, 283–291 (1993).

20. Hopf, T. A. *et al.* The EVcouplings Python framework for coevolutionary sequence analysis. **35**, 1582–1584 (2019).

21. Olson, C. A., Wu, N. C. & Sun, R. A comprehensive biophysical description of pairwise epistasis throughout an entire protein domain. *Curr. Biol.* **24**, 2643–2651 (2014).