Machine Learning Based Protein Engineering for Microbial Chemical Production

By

Jonathan C. Greenhalgh

A dissertation submitted in partial fulfillment of the requirements for the degree of:

Doctor of Philosophy

(Chemical and Biological Engineering)

UNIVERSITY OF WISCONSIN-MADISON

2022

Date of Final Oral Examination: 04/08/2022

This dissertation is approved by the following members of the Final Oral Committee:

Philip A. Romero, Assistant Professor, Biochemistry, Chemical and Biological Engineering

Brian F. Pfleger, Professor, Chemical and Biological Engineering

Eric V. Shusta, Professor, Chemical and Biological Engineering

Andrew R. Buller, Assistant Professor, Chemistry

Srivatsan Raman, Assistant Professor, Biochemistry, Chemical and Biological Engineering, Bacteriology

Acknowledgements

Thank you to the organizations that have provided the funding for the research that I performed in this dissertation: the Biotechnology Training Program, and the Great Lakes Bioenergy Research Center. Participating in these programs has helped me grow as a researcher and helped me deepen my professional and networking skills as well.

Thank you to my advisors, Phil Romero and Brian Pfleger, for your mentorship and support. I've learned so much working with each of you. I've really appreciated your flexibility throughout my PhD, especially in working with me as I strove to balance work and taking care of my family. I look up to both of you as scientists and as people, and I am grateful for the chance I had to work in your labs. Also thank you to my committee, Eric Shusta, Andrew Buller and Srivatsan Raman, for their feedback and guidance.

Thank you to all the people in the lab(s) who have made this dissertation possible. Working with both the Romero and Pfleger labs has been an incredible experience, and in the process, I have worked with many fantastic scientists and people, and I have learned so much from everyone in the labs. I'm especially grateful to those graduate students and postdocs who have acted as mentors and helped me refine my laboratory skills: Mark Politz, Jyun-Liang (Aaron) Lin, Job Grant, Nestór Hernández-Lozada and Chris Mehrer. I've also had the privilege to work with fantastic undergraduate students, Sarah Fahlberg, Haiyang (Ocean) Zheng, and Sam Gardner, who have been instrumental in collecting data and performing experiments included in this dissertation. I am deeply grateful for their help, and I have learned a lot by working with each of them. Additionally, I'm grateful to my other coauthors Apoorv Saraogee, and Pete Heinzelman, Ben Bremer,

and Jerry Duan for their roles in contributing to manuscripts and for teaching me new skills. Thanks to Sameer D'Costa for helping me improve my coding and computation al skills. Thank you as well to Hridindu Roychowdhury, Juan Diaz, and Leland Hyman, who have made up the core of the Romero lab since I joined, and to everyone else who has joined the lab since, for making the Romero lab the amazing place that it is to be. Thank you as well to everyone in the Pfleger lab for welcoming me as one of their own, even though I'm across the street most of the time. I'm especially grateful to Dylan Courtney, Paul Perkovich, Mike Jindra, Will Bothfeld, and Will Cordell, who have always been willing to answer my random questions about cell metabolism and metabolic pathways (as well as being willing to help me find *E. coli* strains in the freezer). Once again, thank you to everyone in the labs. I may not have mentioned everyone by name, but I really do feel that everyone in each lab has made an impact, and I'm grateful for the chance I've had to know you all.

I am also deeply grateful to my family. Thank you to my parents, Beverly and Cloyd, who gave me a love of learning and have always been rooting for me. You've always supported me in my desire to learn, and I am grateful for your support and your example.

Thank you to my wife Elizabeth, for always being there for me, for supporting me through all the ups and downs of graduate school, for helping read all my drafts, and for being all-around awesome. I truly couldn't have done this without you. Also thank you to my daughter Eleanor, who makes the world seem new, fresh, and exciting with each new day. I hope that somehow, perhaps in some small way, my work can help make the world a better place for you.

Table of contents

Table of figures

Table of tables

Abstract

The focus of this dissertation is on applications of machine learning-guided enzyme engineering for modulating metabolic pathways to produce chemicals. Enzyme engineering is especially useful for relieving bottlenecks in pathways, and machine learning strategies can accelerate enzyme engineering by efficiently leveraging data. We review machine learning-based protein engineering as a whole and demonstrate how we used machine learning-guided protein engineering to design acyl-CoA reductase enzymes that had increased activity for converting fatty acyl-ACPs to fatty alcohols. We also discuss machine learning strategies used to design a set of 1-Deoxy-D-Xylulose 5-Phosphate Synthase (DXS) enzymes (which are used for production of terpenoids) that had improved fitness *in vivo* and illustrate how machine learning was used to understand patterns in the fitness landscape. This dissertation demonstrates ways that machine learning tools can accelerate optimization of metabolic pathways by optimizing protein sequences, and the advances in this thesis can be used to help develop more efficient and sustainable routes to produce chemicals.

Chapter 1

Introduction

Author: Jonathan C. Greenhalgh

1.1 Overview of protein engineering for chemical production

Due to rising greenhouse gas emissions, there is a growing need to replace traditional chemical synthesis strategies with alternatives that are more sustainable. Traditionally, petroleum products have been used as chemical feedstocks and altered to make more complex commodity chemicals. Plant oils are also used frequently in chemical production, though their use raises concerns about deforestation and sustainability[1]. Producing chemical products in engineered microbial cells (such as bacteria or yeast) is a very promising alternative strategy to enable production of chemicals ranging from fuels to pharmaceuticals[2–5]. Microbes can produce chemicals through chemical synthesis from simple renewable feedstocks, like sugars from biomass. Even if the end-product is a fuel, using biomass as a source of carbon reduces the effect of carbon emissions, because the carbon came from a plant or photosynthetic organism capable of fixing carbon from the air. In addition to the environmental benefits, producing chemicals in microbes has many other advantages; traditional chemical synthesis frequently requires harsh chemicals or conditions, is limited in the scope of molecules that can be synthesized, and substantial work must be done to generate a selective process. On the other hand, cells can make products without harsh chemicals or conditions, including complex molecules and natural products that cannot be artificially synthesized, and cellular and enzymatic processes often have higher product selectivity than chemical methods.

Just like traditional chemical methods require optimization of the reagents and conditions, chemical production in microbes involves optimizing metabolic pathways, or metabolic engineering[2,5]. There are many ways that microbes have been and can be engineered to produce valuable chemicals. Common metabolic engineering strategies

include expressing heterologous pathways in a host organism (such as *E. coli*), overexpressing key enzymes, and protein or enzyme engineering. The combination of heterologous expression and overexpression allows individual proteins to be studied in high yield and has led to many key biological advances. Protein and enzyme engineering are also very useful tools, and can help optimize pathways further when other strategies such as overexpression fail[6]; protein engineering is often used to address bottlenecks in metabolic pathways, alter product specificity and introduce alternate metabolic routes to pathways[7]. Briefly, some examples of protein engineering to improve production of chemicals include engineering of thioesterase from *Cuphea palustris* to improve its catalytic activity on octanoyl-ACP[8], altering the substrate specificity of the *E. coli* thioesterase 'TesA[9] and thiolase enzymes from *Cupriavidus necator* and *zoogloea ramigera*[10], reducing product inhibition in deoxy-D-xylulose-phosphate synthase (DXS) from *Populus trichocharpa*[11], and developing a glycolyl-CoA carboxylase for capturing CO_2[12].

1.2 Protein engineering strategies

A protein's amino acid sequence determines its function, and protein engineering is the process by which a protein is optimized for a specific function by altering the sequence of its amino acids (i.e., introducing genetic mutations). Typically, protein engineering involves making or designing a library of protein sequences containing a variety of mutations. The library is then tested or screened for the desired function, and any sequences that result in a gain of function can be further characterized or used as a starting point for another round of engineering. Generally, there are two main strategies

for engineering proteins: 1) rational design and 2) directed evolution.

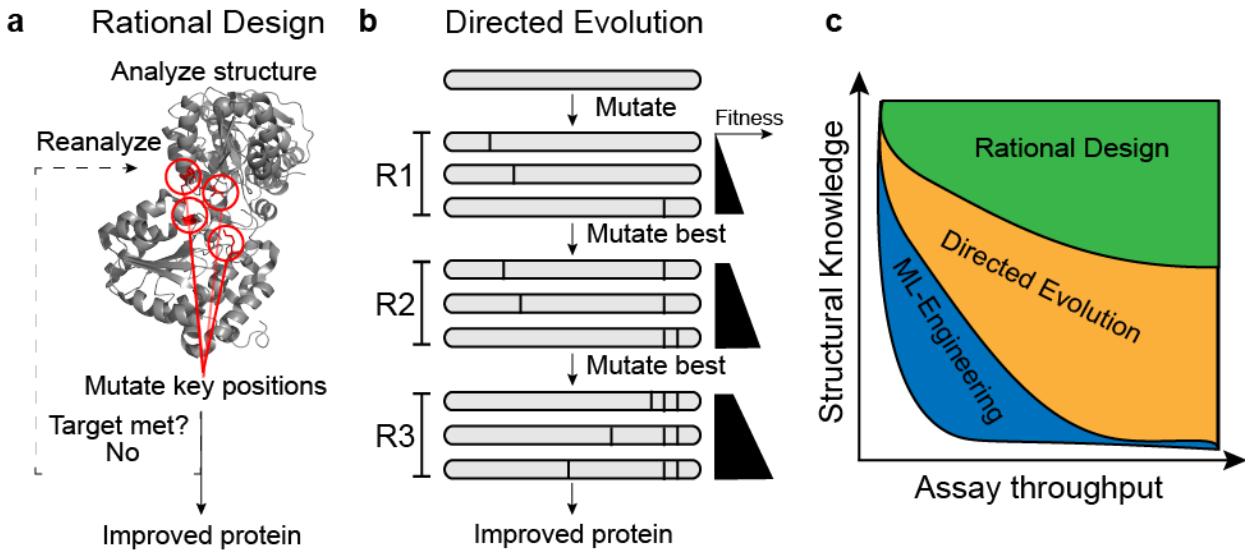

Figure 1.1: Traditional protein engineering workflows. a) Rational design involves detailed structural analysis followed by generating mutations at key residues. b) Directed evolution is an iterative process by which beneficial mutations are gradually accumulated over several evolutionary rounds. Directed evolution and rational design can sometimes be combined in protein engineering workflows. c) Tradeoffs associated with protein engineering strategies. Directed evolution usually requires access to a high throughput screen, whereas rational design typically requires high quality structural data. Machine learning can make new protein engineering targets accessible that can't be reasonably engineered by traditional methods.

1.2.1 Rational design

Rational design involves detailed study of protein structure and using prior knowledge of the structure and function to predict mutations that will enhance the desired trait. Because rational design tends to focus on mutating just a few key positions in a protein structure, libraries are not typically very large, and lower throughput or more expensive testing can be used to screen it. To be successful, rational design requires a very thorough understanding of the proteins structure, and therefore requires high resolution structural data (usually X-ray crystallography data).

Rational design has been used to engineer proteins for many purposes. In Grisewood & Hernández-Lozada et al.[9], the substrate chain length specificity of the thioesterase ‘TesA was altered with rational design in tandem with a program called IPRO (Iterative Protein Redesign and Optimization)[13]. Starting with crystal structures of ‘TesA and decanoyl-ACP, IPRO was used to identify specific mutations to ‘TesA that could improve the binding interaction between ‘TesA and various acyl-acyl-carrier proteins (acyl-ACPs). Mutations were then made *in vivo* and those that resulted in higher activity on shorter chain (C8-C12) fatty acyl-ACPs were kept for the next round. This process was repeated over four rounds, resulting in a ‘TesA variant that produced up to 50% C8 fatty acid (the wild type ‘TesA is specific for C14 fatty acid).

Another example of using rational design to alter enzyme specificity is work done by Bonk et al. to increase the ratio of C6 to C4 products in the reverse β-oxidation pathway (also reffered to as the 3-hydroxyacid pathway)[10], which is a pathway of interest for production of fatty alcohols and other related molecules in microbes[14–16]. In this work, structures of thiolases BktB and PhbA (from *C. necator* and *Z. ramigera* respectively) were computationally docked to butyryl-CoA. The binding energies of mutated thiolases were computed and compared to the wild-type and several specific mutations were selected for further screening. Resulting mutants had up to tenfold improvements in the ratio of C6 to C4 products, though most of this was driven by a reduction in activity for acetoacetyl-CoA (the substrate leading to the C4 product).

In the non-mevalonate pathway (or MEP pathway), which is an important microbial pathway for production of terpenoids[17], DXS is one of the key flux controlling enzymes[18]. It is also known that DXSs are feedback regulated by downstream prenyl-phosphates

(isopentenyl-diphosphate or IPP, and dimethylallyl diphosphate or DMAPP)[19]. Banerjee et al. sought to relieve this inhibitory effect in the DXS from *P. trichocarpa*, so they used rational design to identify a pair of key positions that when mutated from alanine to glycine reduced the amount of IPP based inhibition.

Carbon fixation is a grand challenge in renewable chemical production, but enzymes that are capable of fixing carbon from carbon dioxide in the air are notoriously inefficient. Scheffen et al.[12] used rational design coupled with microfluidics to enhance substrate promiscuity in a propionyl-CoA carboxylase to engineer a novel glycolyl-CoA carboxylase (or GCC, an enzyme that doesn't exist in nature) and enable a new cellular pathway from glycolate to glycerate. This pathway also results in fixation of carbon dioxide, and it is believed that replacing natural photorespiration with the GCC based pathway could improve the carbon efficiency of the cells by 150%[12].

1.2.2 Directed evolution

In contrast to rational design, directed evolution requires no structural information at all. Directed evolution mimics natural evolution in a laboratory; an artificial selective pressure is applied which favors enzymes with improved function[20,21]. Usually, a library of mutated proteins is subjected to either a screen or selection that links the proteins function to an observable trait. Improved variants can then be further mutated and subjected to additional rounds of evolution. This iterative process can result in proteins or enzymes with substantial improvements to their function. Directed evolution is most effective when library sizes are large, and a high throughput screen or selection is available, though it can be complimented with structural data in cases where high

throughput screening is not possible[22].Directed evolution has revolutionized biology and been widely applied across all subdisciplines of protein engineering. It is an especially powerful tool in advancing the introduction of chemical reactions that either weak or not found in nature to cellular processes, as has been done with recent work to design enzymes capable of creating carbon silicon bonds[23,24].

A few examples of how directed evolution has been used to engineer proteins for producing chemicals will be given here, but it is by no means comprehensive. Hernández-Lozada et al.[8] designed a high throughput selection designed to identify thioesterases with higher activity for converting octanoyl-ACP to octanoic acid. The selection works by using a ΔLipB strain of *E. coli* that is reliant on a source of either lipoic or octanoic acid for growth. A library was made by randomizing the low activity but C8-specific thioesterase from *C. palustris*. Thioesterase variants with higher C8 activity were capable of rescuing growth in the selection conditions, and the approach was used to make octanoic acid at very high levels (1.7 g/L) and very high purity (~90%)[8]. The thioesterase variant discovered in this workflow was also later used to make large amounts of the fatty alcohol 1-octanol[25].

Chen et al.[26] used three rounds of directed evolution to engineer the *Saccharomyces cerevisiae* mevalonate kinase enzyme to have improved activity in the mevalonate pathway. The study identified three mutations that when combined improved the ability of the enzyme to bind mevalonate and enabled a roughly 2.4-fold improvement in lycopene titers. Another study by Alvizo et al.[27] used directed evolution to develop a carbonic anhydrase enzyme with improved stability (in terms of thermal, pH and solvent tolerance). Carbonic anhydrases are enzymes capable of fixing carbon, and the

engineered carbon anhydrase from this study was tested at pilot plant scale and used to capture CO_2 from flue gas by converting CO_2 and water to bicarbonate.

1.2.3 Drawbacks of traditional protein engineering strategies

Despite their successes, there are drawbacks to traditional protein engineering strategies. For rational design, high resolution structural information is often unavailable (though recent advances in protein structure prediction will soon begin to help fill this knowledge gap). Additionally, rational design approaches fail to consider mutations that are far from known active sites or binding sites that can have often surprising effects on protein function. Directed evolution often involves making very large libraries with random mutations. Because random mutations tend to be deleterious to function, it can take extremely large libraries to find any functional variants, necessitating high throughput screens or selections. The requirement for high throughput significantly limits the kinds of enzymes that can be engineered using directed evolution.

1.2.4 Machine learning-guided protein engineering

To address the shortfalls of existing protein engineering strategies, machine learning is emerging as a valuable tool in protein and enzyme engineering complementing more traditional strategies. Machine learning is a process by which statistical models can be used to help computer algorithms learn from data and recognize patterns. The use of machine learning spans many disciplines and industries, and machine learning algorithms are increasingly common in everyday applications. Though machine learning-guided protein engineering is a very young discipline, it has already been used successfully to

accelerate protein engineering strategies and design enhanced proteins. Machine learning complements traditional protein engineering strategies and can be easily coupled with directed evolution or rational design workflows. A more detailed review of this topic appears in Chapter 2, but a few important studies related to this dissertation will be summarized here.

One of the first applications of machine learning to protein engineering was the development of the protein sequence activity relationships (ProSAR)[28,29] algorithm by Fox et al. ProSAR combines directed evolution with partial least squares regression models that maps protein sequences to their function. Sequences that are likely to be beneficial or neutral are identified and pooled for the next experimental round, and this process is repeated until the protein's activity is improved.

Another notable study was the use of Gaussian processes to iteratively sample a protein fitness landscape by Romero et al.[30] In this study, chimeric cytochrome P450 variants with substantial improvements to their thermostability were developed using an active learning style approach that optimizes the search through a fitness landscape. Briefly, gaussian process regression models were trained on chimeric protein thermostability data, and an optimization procedure that will be described in detail in Chapters 2 and 3 was used to design new chimeric proteins that were predicted to have high activity. Then, those proteins were built in the lab and tested, and the regression models were updated with the new data. At that point the cycle was repeated several more times. The result was a P450 variant with significantly higher thermostability than previously characterized variants. The optimization strategy used in this study has also been used to engineer difficult to engineer proteins called channelrhodopsins that cannot

be engineered by directed evolution alone[31,32].

There have also been substantial advances in how machine learning can complement structure based and rational design approaches. Recently, Alley et al.[33] developed a representation called UniRep. UniRep uses deep learning approaches trained on amino acid sequences to find a statistical representation of protein sequences that contains significant information about protein features and can be used for training downstream machine learning models[34].

Another major development in the use of machine learning to aid in protein engineering is the development of AlphaFold[35]. AlphaFold is a neural network-based model trained on protein sequences that is used to predict protein structures, and it recently garnered worldwide attention at the 14th Critical Assessment of protein Structure Prediction (CASP14), where it significantly outperformed competing methods. AlphaFold's high accuracy in predicting protein structures represents major progress towards solving the protein folding problem. AlphaFold has already been used to predict the structures of nearly every protein in the human proteome[36], and it promises to revolutionize protein structure prediction and fill in gaps in current protein structure databases and accelerate protein engineering efforts.

1.3 Overview of dissertation

The work in this dissertation focuses on machine learning-guided engineering of enzymes in two important pathways: acyl-ACP reductases for fatty alcohol production and DXS enzymes to improve performance of the MEP pathway. The chapters in this dissertation illustrate different ways of how machine learning and protein engineering can

be combined to improve production of valuable chemicals.

1.3.1 Chapter 2: Data-driven protein engineering

In Chapter 2, the field of machine learning-guided protein engineering will be introduced and summarized. The fundamental principles and terms of machine learning are reviewed in the context of protein engineering, and different methods for encoding protein data (sequence vs. structural) will be explained and compared. Important applications of different machine learning techniques from the literature will be described as well, beyond the examples described in section 1.2.4.

1.3.2 Chapter 3: Machine learning-guided acyl-ACP reductase engineering for improved in vivo fatty alcohol production

In Chapter 3, an active learning framework similar to the strategy used by Romero et al.[30] (described briefly in section 1.2.4) was used to engineer improved acyl-ACP reductase activity in chimeric protein sequences derived from three parental acyl-CoA reductases. We used machine learning models trained on data to help design new sequences that balanced the need to explore new sequence space and exploit knowledge of the sequence landscape, and then tested the recommended sequences *in vivo*. The fatty alcohol titers were used to update the models, and the design-build-test-learn cycle was repeated over ten rounds. The resulting enzyme had nearly threefold higher fatty alcohol titers from acyl-ACPs than the best wild-type parent, and the improvement in the titer correlated with an improvement in the k_{cat}.

1.3.3 Chapter 4: Engineering 1-Deoxy-D-Xylulose 5-Phosphate Synthase (DXS) for improved MEP pathway flux using a high-throughput growth selection

In Chapter 4 we build a large dataset of DXS activity by coupling a high-throughput growth selection with next generation sequencing. The dataset allows us to make a detailed mapping of the DXS fitness landscape. Then we use positive-unlabeled (PU) learning to design chimeric DXS sequences for improving flux through the MEP pathway. We then characterized the designed DXSs using *in vitro* coupled enzyme assays. We found that the highest activity DXSs from the selection had similar activity to the wild type DXSs *in vitro*, but interestingly, we also found that the designed DXSs were more strongly inhibited by IPP and DMAPP than the wild-type parents. This result suggests that the selection strategy we used likely favored enzymes that could be feedback regulated, potentially by allowing the flux to turn on and off depending on the levels of IPP.

1.3.4 Chapter 5: Conclusions and Future Directions

In Chapter 5 I'll summarize the work done in each of these chapters and examine how recent advances in machine learning and protein engineering can advance efforts to further optimize proteins and metabolic pathways. I'll briefly describe how some emerging techniques, such as neural networks and generative models, are being applied to protein engineering problems.

1.3.5 Chapter 6: Application of machine learning-based protein engineering to make an improved acyl-ACP reductase enzyme

Chapter 6 is part of a large project through the Wisconsin Initiative for Science Literacy (WISL) at UW-Madison and is directed towards the general public. Chapter 6 contains simple descriptions of ideas important for machine learning-guided protein engineering as well as a narrative summarizing Chapter 3 in simple, everyday terms.

References:

(1) Fargione, J.; Hill, J.; Tilman, D.; Polasky, S.; Hawthorn, P. Land Clearing and the Biofuel Carbon Debt. *Science (80-.).* **2008**, *319* (5867), 1235–1238. https://doi.org/10.1126/science.1245938.

(2) Keasling, J. D. Manufacturing Molecules through Metabolic Engineering. *Science (80-.).* **2010**, *330* (6009), 1355–1358. https://doi.org/10.1126/SCIENCE.1193990/ASSET/63A8B945-607A-40F7-AD37-03C9DB7C3A45/ASSETS/GRAPHIC/330_1355_F3.JPEG.

(3) Lennen, R. M.; Pfleger, B. F. Microbial Production of Fatty Acid-Derived Fuels and Chemicals. *Curr. Opin. Biotechnol.* **2013**, *24* (6), 1044–1053. https://doi.org/10.1016/j.copbio.2013.02.028.

(4) Yan, Q.; Pfleger, B. F. Revisiting Metabolic Engineering Strategies for Microbial Synthesis of Oleochemicals. *Metab. Eng.* **2020**, *58*, 35–46. https://doi.org/10.1016/J.YMBEN.2019.04.009.

(5) Nielsen, J.; Keasling, J. D. Engineering Cellular Metabolism. *Cell* **2016**, *164* (6), 1185–1197. https://doi.org/10.1016/j.cell.2016.02.004.

(6) Leonard, E.; Ajikumar, P. K.; Thayer, K.; Xiao, W.-H.; Mo, J. D.; Tidor, B.; Stephanopoulos, G.; Prather, K. L. J. Combining Metabolic and Protein Engineering of a Terpenoid Biosynthetic Pathway for Overproduction and Selectivity Control. *Proc. Natl. Acad. Sci.* **2010**, *107* (31), 13654–13659. https://doi.org/10.1073/pnas.1006138107.

(7) Li, C.; Zhang, R.; Wang, J.; Wilson, L. M.; Yan, Y. Protein Engineering for Improving and Diversifying Natural Product Biosynthesis. *Trends Biotechnol.*

2020, *38* (7), 729–744. https://doi.org/10.1016/J.TIBTECH.2019.12.008.

(8) Hernandez-Lozada, N. J.; Lai, R.-Y.; Simmons, T.; Thomas, K. A.; Chowdhury, R.; Maranas, C. D.; Pfleger, B. F. Highly Active C8-Acyl-ACP Thioesterase Variant Isolated by a Synthetic Selection Strategy. *ACS Synth. Biol.* **2018**, acssynbio.8b00215. https://doi.org/10.1021/acssynbio.8b00215.

(9) Grisewood, M. J.; Hernández-Lozada, N. J.; Thoden, J. B.; Gifford, N. P.; Mendez-Perez, D.; Schoenberger, H. A.; Allan, M. F.; Floy, M. E.; Lai, R. Y.; Holden, H. M.; et al. Computational Redesign of Acyl-ACP Thioesterase with Improved Selectivity toward Medium-Chain-Length Fatty Acids. *ACS Catal.* **2017**, *7* (6), 3837–3849. https://doi.org/10.1021/acscatal.7b00408.

(10) Bonk, B. M.; Tarasova, Y.; Hicks, M. A.; Tidor, B.; Prather, K. L. J. Rational Design of Thiolase Substrate Specificity for Metabolic Engineering Applications. *Biotechnol. Bioeng.* **2018**, *115* (9), 2167–2182. https://doi.org/10.1002/bit.26737.

(11) Banerjee, A.; Preiser, A. L.; Sharkey, T. D. Engineering of Recombinant Poplar Deoxy-D-Xylulose-5-Phosphate Synthase (PtDXS) by Site-Directed Mutagenesis Improves Its Activity. *PLoS One* **2016**, *11* (8). https://doi.org/10.1371/journal.pone.0161534.

(12) Scheffen, M.; Marchal, D. G.; Beneyton, T.; Schuller, S. K.; Klose, M.; Diehl, C.; Lehmann, J.; Pfister, P.; Carrillo, M.; He, H.; et al. A New-to-Nature Carboxylation Module to Improve Natural and Synthetic CO2 Fixation. *Nat. Catal.* **2021**, 1–11. https://doi.org/10.1038/s41929-020-00557-y.

(13) Saraf, M. C.; Moore, G. L.; Goodey, N. M.; Cao, V. Y.; Benkovic, S. J.; Maranas, C. D. IPRO: An Iterative Computational Protein Library Redesign and

Optimization Procedure. *Biophys. J.* **2006**, *90* (11), 4167–4180. https://doi.org/10.1529/biophysj.105.079277.

(14) Mehrer, C. R.; Incha, M. R.; Politz, M. C.; Pfleger, B. F. Anaerobic Production of Medium-Chain Fatty Alcohols via a β-Reduction Pathway. *Metab. Eng.* **2018**, *48* (April), 63–71. https://doi.org/10.1016/j.ymben.2018.05.011.

(15) Clomburg, J. M.; Vick, J. E.; Blankschien, M. D.; Rodríguez-Moyá, M.; Gonzalez, R. A Synthetic Biology Approach to Engineer a Functional Reversal of the Beta-Oxidation Cycle. *ACS Synth. Biol.* **2012**, *1* (11), 541–554. https://doi.org/10.1021/sb3000782.

(16) Dellomonaco, C.; Clomburg, J. M.; Miller, E. N.; Gonzalez, R. Engineered Reversal of the β-Oxidation Cycle for the Synthesis of Fuels and Chemicals. *Nature* **2011**, *476* (7360), 355–359. https://doi.org/10.1038/nature10333.

(17) Klein-Marcuschamer, D.; Ajikumar, P. K.; Stephanopoulos, G. Engineering Microbial Cell Factories for Biosynthesis of Isoprenoid Molecules: Beyond Lycopene. *Trends Biotechnol.* **2007**, *25* (9), 417–424. https://doi.org/10.1016/J.TIBTECH.2007.07.006.

(18) Volke, D. C.; Rohwer, J.; Fischer, R.; Jennewein, S. Investigation of the Methylerythritol 4-Phosphate Pathway for Microbial Terpenoid Production through Metabolic Control Analysis. *Microb. Cell Fact.* **2019**, *18* (1), 1–15. https://doi.org/10.1186/S12934-019-1235-5/FIGURES/8.

(19) Banerjee, A.; Wu, Y.; Banerjee, R.; Li, Y.; Yan, H.; Sharkey, T. D. Feedback Inhibition of Deoxy-d-Xylulose-5-Phosphate Synthase Regulates the Methylerythritol 4-Phosphate Pathway. *J. Biol. Chem.* **2013**, *288* (23), 16926–

16936. https://doi.org/10.1074/JBC.M113.464636.

(20) Arnold, F. H.; Volkov, A. A. Directed Evolution of Biocatalysts. *Curr. Opin. Chem. Biol.* **1999**, *3* (1), 54–59. https://doi.org/10.1016/S1367-5931(99)80010-6.

(21) Arnold, F. H. Combinatorial and Computational Challenges for Biocatalyst Design. *Nature* **2001**, *409* (6817), 253–257. https://doi.org/10.1038/35051731.

(22) Zhao, H. Directed Evolution of Novel Protein Functions. *Biotechnol. Bioeng.* **2007**, *98* (2), 313–317. https://doi.org/10.1002/bit.

(23) Kan, S. B. J.; Lewis, R. D.; Chen, K.; Arnold, F. H. Directed Evolution of Cytochrome c for Carbon-Silicon Bond Formation: Bringing Silicon to Life. *Science (80-.).* **2016**, *354* (6315), 1048–1051. https://doi.org/10.1126/SCIENCE.AAH6219/SUPPL_FILE/KAN.SM.PDF.

(24) Wu, Z.; Jennifer Kan, S. B.; 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.* **2019**, *116* (18), 8852–8858. https://doi.org/10.1073/pnas.1901979116.

(25) Hernández Lozada, N. J.; Simmons, T. R.; Xu, K.; Jindra, M. A.; Pfleger, B. F. Production of 1-Octanol in Escherichia Coli by a High Flux Thioesterase Route. *Metab. Eng.* **2020**, *61*, 352–359. https://doi.org/10.1016/j.ymben.2020.07.004.

(26) Chen, H.; Liu, C.; Li, M.; Zhang, H.; Xian, M.; Liu, H. Directed Evolution of Mevalonate Kinase in: Escherichia Coli by Random Mutagenesis for Improved Lycopene. *RSC Adv.* **2018**, *8* (27), 15021–15028. https://doi.org/10.1039/c8ra01783b.

(27) Alvizo, O.; Nguyen, L. J.; Savile, C. K.; Bresson, J. A.; Lakhapatri, S. L.; Solis, E.

O. P.; Fox, R. J.; Broering, J. M.; Benoit, M. R.; Zimmerman, S. A.; et al. Directed Evolution of an Ultrastable Carbonic Anhydrase for Highly Efficient Carbon Capture from Flue Gas. *PNAS* **2014**, *111* (46), 16436–16441. https://doi.org/10.1073/pnas.1411461111.

(28) Fox, R.; Roy, A.; Govindarajan, S.; Minshull, J.; Gustafsson, C.; Jones, J. T.; Emig, R. Optimizing the Search Algorithm for Protein Engineering by Directed Evolution. *Protein Eng. Des. Sel.* **2003**, *16* (8), 589–597. https://doi.org/10.1093/protein/gzg077.

(29) Fox, R. J.; Davis, S. C.; Mundorff, E. C.; Newman, L. M.; Gavrilovic, V.; Ma, S. K.; Chung, L. M.; Ching, C.; Tam, S.; Muley, S.; et al. Improving Catalytic Function by ProSAR-Driven Enzyme Evolution. *Nat. Biotechnol.* **2007**, *25* (3), 338–344. https://doi.org/10.1038/nbt1286.

(30) Romero, P. a; Krause, A.; Arnold, F. H. Navigating the Protein Fitness Landscape with Gaussian Processes. *Proc. Natl. Acad. Sci. U. S. A.* **2013**, *110* (3), E193-201. https://doi.org/10.1073/pnas.1215251110.

(31) 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 Comput. Biol.* **2017**, *13* (10), 1–21. https://doi.org/10.1371/journal.pcbi.1005786.

(32) Yang, K. K.; Elliott Robinson, J. Machine Learning-Guided Channelrhodopsin Engineering Enables Minimally Invasive Optogenetics. *Nat. Methods*. https://doi.org/10.1038/s41592-019-0583-8.

(33) Alley, E. C.; Khimulya, G.; Biswas, S.; AlQuraishi, M.; Church, G. M. Unified

Rational Protein Engineering with Sequence-Based Deep Representation Learning. *Nat. Methods* **2019**. https://doi.org/10.1038/s41592-019-0598-1.

(34) Biswas, S.; Khimulya, G.; Alley, E. C.; Esvelt, K. M.; Church, G. M. Low-N Protein Engineering with Data-Efficient Deep Learning. https://doi.org/10.1101/2020.01.23.917682.

(35) Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; et al. Highly Accurate Protein Structure Prediction with AlphaFold. *Nat. 2021 5967873* **2021**, *596* (7873), 583–589. https://doi.org/10.1038/s41586-021-03819-2.

(36) Tunyasuvunakool, K.; Adler, J.; Wu, Z.; Green, T.; Zielinski, M.; Žídek, A.; Bridgland, A.; Cowie, A.; Meyer, C.; Laydon, A.; et al. Highly Accurate Protein Structure Prediction for the Human Proteome. *590* | *Nat.* | **2021**, *596*. https://doi.org/10.1038/s41586-021-03828-1.

Chapter 2

Data-driven protein engineering

Authors and Contributors:

- Jonathan C. Greenhalgh is the primary author of sections 2.4 and 2.5, and also edited and reviewed the rest of the manuscript.
- Apoorv Saraogee is the primary author of sections 2.2 and 2.3, and also edited and reviewed the rest of the manuscript.
- Dr. Philip A. Romero is the primary author of sections 2.1 and 2.6 and provided guidance, edited and reviewed the manuscript.

This chapter is adapted from a book chapter of the same title in *Protein Engineering: Tools and Applications*, Wiley, **2021**

2.1 Introduction

A protein's sequence of amino acids encodes its function. This "function" could refer to a protein's natural biological function, or it could also be any other property including binding affinity toward a particular ligand, thermodynamic stability, or catalytic activity. A detailed understanding of how these functions are encoded would allow us to more accurately reconstruct the tree of life and possibly predict future evolutionary events, diagnose genetic diseases before they manifest symptoms, and design new proteins with useful properties. We know that a protein sequence folds into a three-dimensional structure, and this structure positions specific chemical groups to perform a function; however, we're missing the quantitative details of this sequence-structure-function mapping. This mapping is extraordinarily complex because it involves thousands of molecular interactions that are dynamically coupled across multiple length and time scales.

Computational methods can be used to model the mapping from sequence to structure to function. Tools such as molecular dynamics simulations or Rosetta use atomic representations of protein structures and physics-based energy functions to model structures and functions[1–3]. While these models are based on well-founded physical principles, they often fail to capture a protein's overall global behavior and properties. There are numerous challenges associated with physics-based models including consideration of conformational dynamics, the requirement to make energy function approximations for the sake of computational efficiency, and the fact that, for many complex properties such as enzyme catalysis, the molecular basis is simply unknown[4]. In systems composed of thousands of atoms, the propagation of small errors quickly

overwhelms any predictive accuracy. Despite tremendous breakthroughs and research progress over the last century, we still lack the key details to reliably predict, simulate, and design protein function.

Machine learning and artificial intelligence are transforming marketing, finance, healthcare, security, internet search, transportation, and nearly every aspect of our daily lives. These approaches leverage vast amounts of data to find patterns and quickly make optimal decisions. In this chapter, we present how these ideas are starting to impact the field of protein engineering. Instead of physically modeling the relationships between protein sequence, structure, and function, data-driven methods use ideas from statistics and machine learning to infer these complex relationships from data. This top-down modeling approach implicitly captures the numerous and possibly unknown factors that shape the mapping from sequence to function. These statistical models compliment physical models and can even be used to improve physics-based models. Statistical models have been used to understand the molecular basis of protein function and provide exceptional predictive accuracy for protein design. We present three key stages in data-driven protein engineering—(1) representation: how to encode protein sequence/structure/function data, (2) learning: automatic detection of patterns and relationships in data, and (3) prediction: applying the learned models to design new proteins.

2.2 The data revolution in biology

The volume of biological data has exploded over the last decade. This is being driven by advances in our ability to read and write DNA, which are progressing faster than

Moore's law[5]. Simultaneously, we have also gained unprecedented ability to characterize biological systems with advances in automation, miniaturization, multiplex assays, and genome engineering. It is now routine to perform experiments on thousands to millions of molecules, genes, proteins, and/or cells. The resulting data provides a unique opportunity to study biological systems in a comprehensive and less biased manner.

Protein sequence and structure databases have been growing exponentially for decades (Figure 2.1b, c). Currently, the UniProt database[6] contains over 100 million unique protein sequences and the Protein Data Bank[7] contains over 100,000 experimentally determined protein structures. While there is an abundance of protein sequence and structure data, there is still relatively little data mapping sequence to function. ProtaBank is a new effort to build a protein function database[8]. Function data is challenging to standardize because it is highly dependent on experimental conditions and even the particular researcher that performed the experiments. Therefore, statistical modeling approaches are most useful on data that is generated by an individual researcher/research group. This allows a consistent definition of "function" that is not influenced by uncontrolled experimental factors.

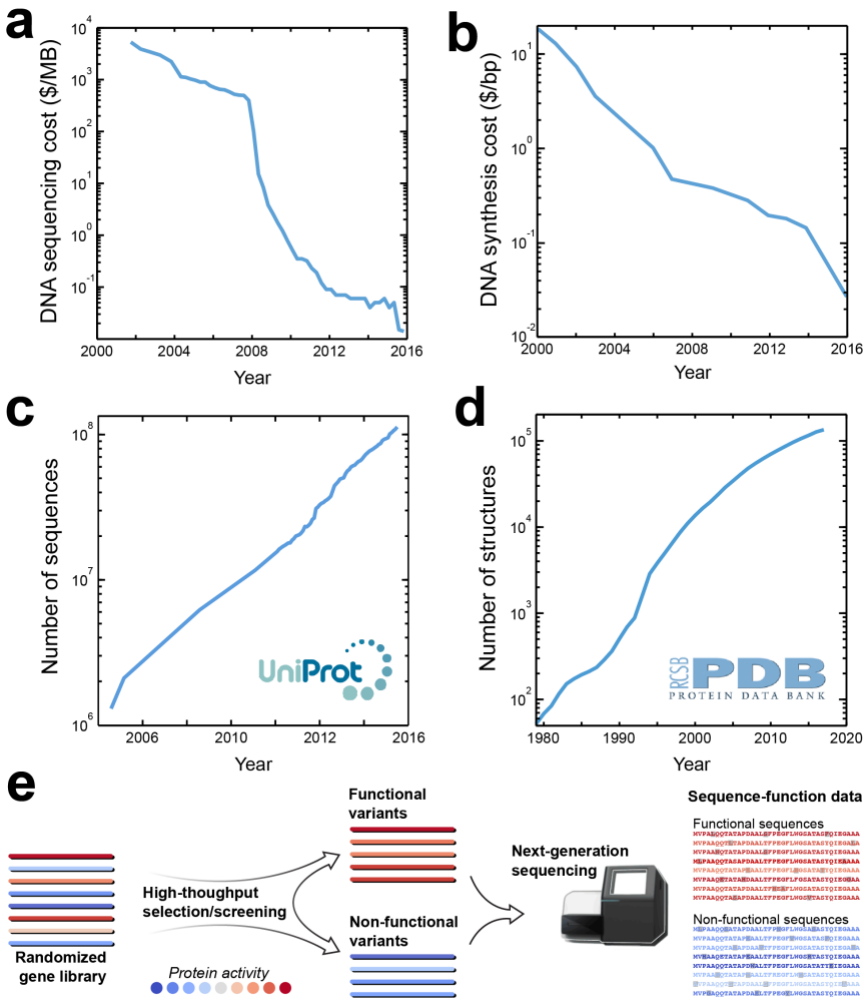

Figure 2.1: The growth of biological data. (a,b) DNA sequencing and synthesis technologies are advancing faster than Moore's law. As a result, costs have decreased exponentially over the last two decades. (c,d) Large-scale genomics, metagenomics, and structural genomics initiatives have resulted in exponential growth of protein sequence and structure databases. (e) Deep mutational scanning experiments combine high-throughput screens/selections with next-generation DNA sequencing to map sequence-function relationships for thousands to millions of protein variants.

Many sequence-function data sets are generated by protein engineering experiments that involve screening libraries of sequence variants for improved function. These variants may include natural homologs, random mutants, targeted mutants, chimeric proteins generated by homologous recombination, and computationally designed sequences. Each of these sequence diversification methods explores different

features of the sequence-function mapping and varies in their information content. Important factors include the sequence diversity of a library, the likelihood of functional vs nonfunctional sequences, and the difficulty/cost of building the desired gene sequences.

Recent advances in high-throughput experimentation have enabled researchers to map sequence-function relationships for thousands to millions of protein variants[9,10]. These “deep mutational scanning” experiments start with a large library of protein variants, and this library is passed through a high-throughput screen/selection to separate variants based on their functional properties (Figure 2.1e). The genes from these variant pools are then extracted and analyzed using next-generation DNA sequencing. Deep mutational scanning experiments generate data containing millions of sequences and how those sequences map to different functional classes (e.g. active/inactive, binds ligand 1/binds ligand and 2). The resulting data have been used to study the structure of the protein fitness landscape, discover new functional sites, improve molecular energy functions, and identify beneficial combinations of mutations for protein engineering[9,11–13].

2.3 Statistical representations of protein sequence, structure, and function

The growing trove of biological data can be mined to understand the relationships between protein sequence, structure, and function. This complex and heterogenous protein data needs to be represented in simple, machine-readable formats to leverage advanced tools in pattern recognition and machine learning. There are many possible ways of representing proteins mathematically including simple sequence-based representations or more advanced structure/ physics-based representations. In general,

a good representation is low dimensional but still captures the system's relevant degrees of freedom.

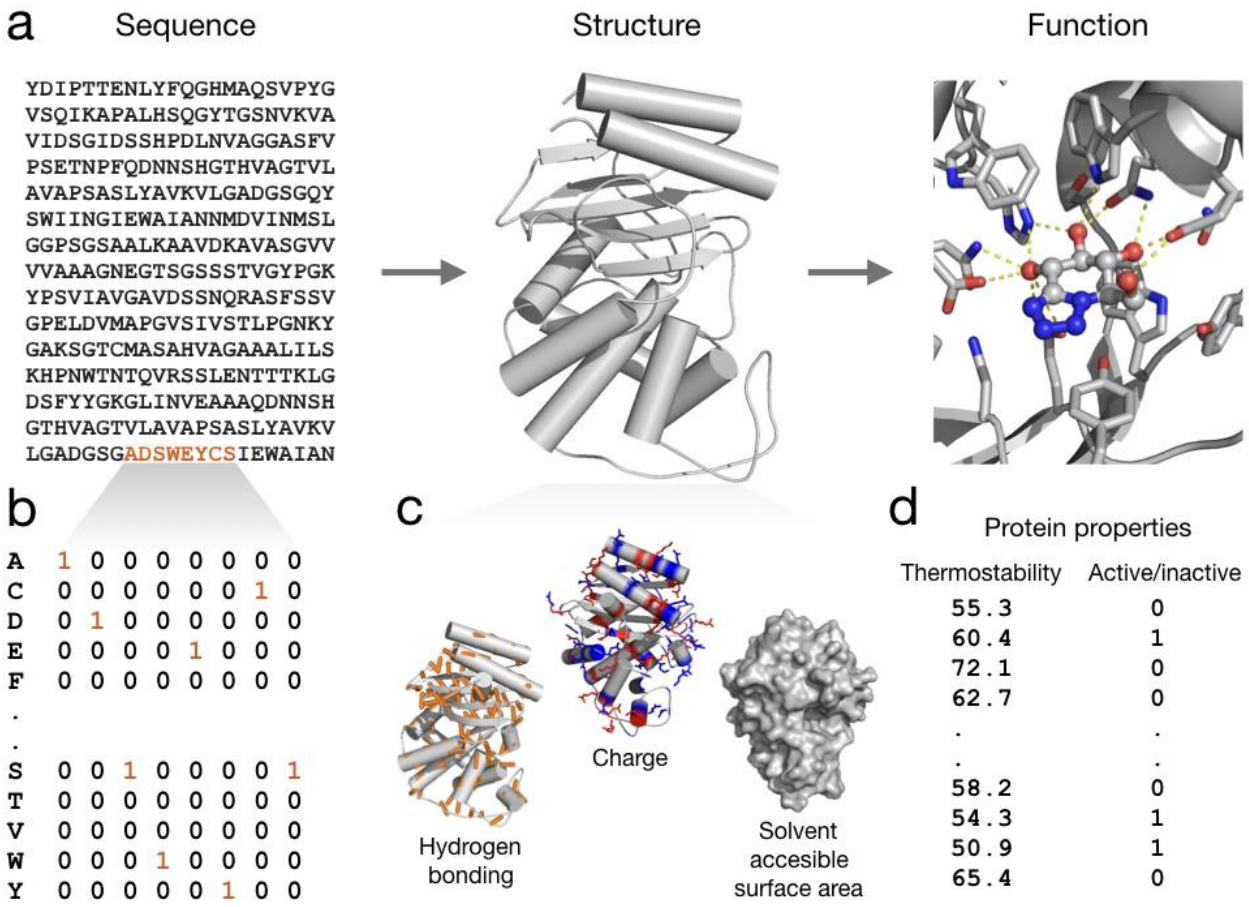

Figure 2.2: Sequence, structure, and function representations. (a) A protein's sequence folds into a three-dimensional structure, and this structure determines its function and properties. (b) Protein sequences can be represented using a one-hot encoding scheme that assigns 20 amino acid bits to each residue position. A bit is assigned a value of "1" if the protein has the corresponding amino acid at a particular residue position. (c) Structure-based representations use modeled protein structures to extract key physiochemical properties such as hydrogen bonds, total charge, or molecular surface areas. (d) Protein functions can be continuous properties such as thermostability or catalytic efficiency, or discrete properties such as active/inactive. Discrete properties can be represented using a binary (0 or 1) encoding.

2.3.1 Representing protein sequences

A protein's amino acid sequence contains all the information necessary to specify its structure and function. Each position in this sequence can be modeled as a categorical

variable that can take on one of twenty amino acid values. Categorical data can be represented using a one-hot encoding strategy that assigns one bit to each possible category. If a particular observation falls into one of these categories, it is assigned a “1” at that category’s bit, otherwise it is assigned a “0.” A protein sequence of length *l* can be represented with a vector of 20*l* bits; 20 bits for each sequence position (Figure 2.2). For example, assuming the amino acid bits are arranged in alphabetical order (A, C, D, E ... W, Y), if a protein has alanine (A) at the first position, the first bit would be 1 and the next 19 bits would be 0. If a protein has aspartic acid (D) at the first position, the first two bits would be 0, the third bit 1, and the next 17 bits 0. This encoding strategy can be applied to all amino acid positions in a protein and represent any sequence of length *l*. One-hot encoding sequence representations are widely used in machine learning because they are simple and flexible. However, they are also very high dimensional (20*l* ≈ thousands of variables for most proteins) and therefore require large quantities of data for learning.

Machine learning is widely used in the fields of text mining and natural language processing to understand sequences of characters and words. The tools word2vec and doc2vec use neural networks to learn vector representations that encode the linguistic context of words and documents[14,15]. These embeddings attempt to capture word/document “meaning” and are much lower dimensional than the original input space. Similar concepts have recently been applied to learn embedded representations of amino acid sequences[16]. This approach breaks amino acid sequences into all possible subsequences of length *k.* These subsequences are referred to as *k*-mers. As an example, the sequence PRFYLA contains the four 3-mers: PRF, RFY, FYL, and YLA. An amino acid sequence’s k-mers are treated as “words” and a neural network is used to

learn other words that are found before/after a given word (i.e. a word's context). Importantly, words that are found in similar contexts tend to have similar meanings. This concept can be used to build low-dimensional vector spaces that place similar words close together. For an amino acid sequence, this might mean that one amino acid triplet is comparable to another, and therefore, we only need one variable to represent both. This produces a low-dimensional representation or "protein embedding" that captures the entire protein sequence. These protein embeddings can then be used to model specific properties such as thermostability.

2.3.2 Representing protein structures

The properties of proteins depend on sequence through their structure, therefore structure-based representations provide a more direct link to function. Experimentally determining a protein's three-dimensional structure (via crystallography, NMR, CryoEM) is significantly more challenging and time consuming than determining sequence or function. Therefore, most sequence-function data sets do not contain experimentally determined protein structures. Instead, this missing structural information can be approximated by taking advantage of the extreme conservation of structures within a family. Homologous proteins with as low as 20% sequence identity still have practically identical three-dimensional structures[17].

A protein's overall fold can be represented by specifying which residues are "contacting" in the three-dimensional structure. These contacting residues could be defined as any pair of residues that has an atom within five angstroms. Other contact definitions could include different distance cutoffs, Cα-Cα distances, or Cβ-Cβ distances.

A protein's contact map specifies all pairs of contacting residues and provides a coarse-grained description of the protein's overall fold. Importantly, contact maps are highly conserved within a protein family, and therefore any two evolutionarily related proteins have practically identical contact maps. If we assume a fixed contact map for a protein family, structural information can be represented using a one-hot encoding scheme similar to sequence encoding described above. Each pair of contacting residues can take on one of 400 (20^2) possible amino acid combinations, which can be one-hot encoded using 400 bits. Therefore, the structure of a protein with *c* contacts can be represented with 400*c* bits. In contrast to sequence-based representations, this contact-based representation can capture pairwise interactions between residues. However, this increased flexibility comes at the cost of significantly higher dimensionality.

Three-dimensional protein structures can also be predicted using molecular modeling and simulation software. Most protein sequence-function data sets can take advantage of homology modeling approaches that start with a closely related template structure, mutate differing residues to the target sequence, and run minimization methods to relax the structure into a local energy minimum. State-of-the-art homology modeling methods can reliably predict protein structures with less than 2 angstrom atomic RMSD[18]. These predicted structures can be analyzed to extract key physiochemical properties such as surface areas, solvent exposure, and physical interactions (Figure 2.2). This approach was recently applied to model the kinetic properties of β-glucosidase point mutants[19]. The substrate was docked into β-glucosidase homology models, and this enzyme-substrate interaction was used to extract 59 physical features such as interface energy, number of intermolecular hydrogen bonds, and change in solvent accessible

surface area. A simple linear regression model could relate these physical features to β-glucosidase turnover number, Michaelis constant, and catalytic efficiency. Physics-based representations tend to be lower dimensional than the sequence and contact encodings described above. They may also have good generalization within a protein family or even across protein families because they are based on fundamental biophysical principles.

2.4 Learning the sequence-function mapping from data

Advanced pattern recognition and machine learning techniques can be used to automatically identify key relationships between protein sequence, structure, and function. These tools are used for two primary tasks: supervised learning and unsupervised learning. Supervised methods, such as regression and classification, attempt to learn the mapping between a set of input variables and output variables. The term "supervised learning" arises because the algorithms are given examples of input-output mapping to guide the learning process. In contrast, unsupervised methods are not given information about the output variable, but instead try to learn relationships between the various input variables. Similar concepts have been used extensively in quantitative structure-activity relationship (QSAR) models, which are typically used to predict the chemical and biological properties of small molecules[20]. QSAR models have also been applied to peptide and DNA sequences[21,22].

2.4.1 Supervised learning (regression/classification)

Regression is a supervised learning technique that is used to model and predict continuous properties. Continuous protein properties could include thermostability,

binding affinity, or catalytic efficiency. Regression methods span from simple linear models to advanced, nonlinear models such as neural networks.

Linear regression is the simplest regression technique and applies fixed weights to each input variable. A linear model is described by the following equation:

$$y = X\beta + \epsilon \tag{2.1}$$

where *y* is a vector of continuous output variables, *X* is a matrix of sequence/structure features (one protein variant per row), β is the weight vector, and ϵ is the model error. The model parameters (β) can be estimated by minimizing the sum of the squared error. This least-squares parameter estimate has an analytical solution:

$$\hat{\beta} = (X^T X)^{-1}(X^T y) \tag{2.2}$$

Here, $\hat{\beta}$ corresponds to an estimate of the true β. $\hat{\beta}$ can then be applied to new proteins to predict their properties:

$$\hat{y} = X_{new}\hat{\beta} \tag{2.3}$$

Linear regression provides a simple framework for relating sequence/structure to function, and predicting the properties of previously uncharacterized proteins.

Linear regression has been used to model chimeric cytochrome P450 thermostability[23]. A library of chimeric P450s was generated by shuffling sequence elements from three related bacterial P450s[24]. The thermostability of 184 randomly chosen chimeric P450s was determined, and a linear regression model was used to relate sequence to thermostability. Each chimeric protein's sequence was one-hot encoded by specifying which sequence elements were present. This encoding scheme is similar to the sequence-based one-hot encoding described above, but sequence "blocks" are used rather than individual amino acids. This simple regression model revealed a strong correlation between the predicted and observed thermostability (Figure 2.3). The model was applied to predict the thermostabilities of all 6,351 possible sequences in the chimeric P450 library, and the most stable predicted sequences were validated experimentally.

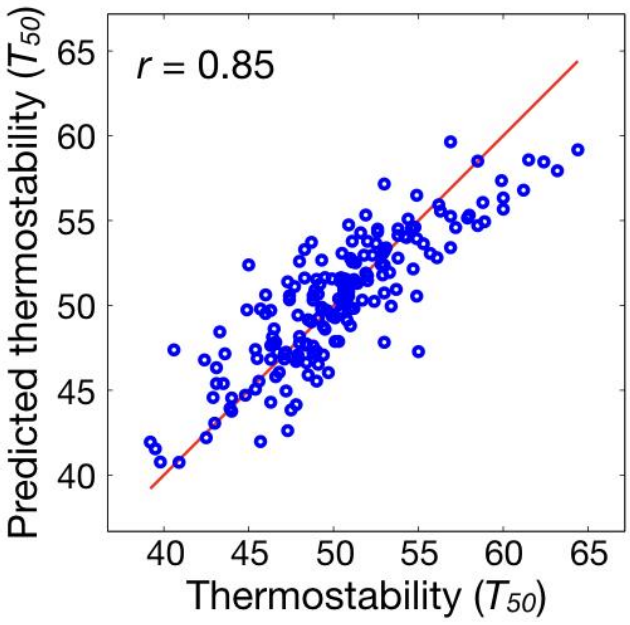

Figure 2.3: A linear regression model for cytochrome P450 thermostability. This model relates sequence blocks of chimeric P450s to their thermostability values. The plot shows the model's cross-validated predictions for 184 chimeric P450s.

Supervised learning methods, including linear regression, are highly susceptible to overfitting data. A linear model must have at least as many data points as model parameters to avoid overfitting. More complex nonlinear models require even more data. Overfitting occurs when there is not sufficient data and the model fits spurious correlations or noise, rather than the true underlying signal. An overfit model will display very small error on the training data, but large prediction error on new data points.

All statistical models must be evaluated for overfitting and their ability to generalize to new, unseen data points. One method for model validation involves training the model on some fraction of the data and using the remainder to evaluate the model's predictive ability. For example, one could train a model on 60% of the data and test the model on the remaining 40%. This holdout method is simple to implement, but also throws out valuable information because the model is not learning from the entire data set. Cross-validation is another method for model evaluation that more effectively utilizes the available data. Cross-validation is similar to the holdout method, but rotates through multiple training set-test set combinations. For example, ten-fold cross-validation breaks the data into ten subsets; a model is trained on nine of these subsets and used to predict the tenth subset. This process is repeated over all ten data folds (i.e. testing on all ten subsets) and the results are averaged. Cross-validation allows all data points to be used in model training and evaluation.

Overfitting can be reduced using regularization methods that favor simpler models. Regularized parameter estimation involves minimizing the model's squared error in addition to the magnitude of the model parameters. This can be achieved by including a penalty term on the norm of the parameter vector:

$$\min_{\beta}(X\beta - y)^2 + \lambda\|\beta\|_n \quad (2.4)$$

Here, the first term corresponds to the model's squared error, the second term is the magnitude of the model parameters, and λ tunes the relative influence of these two terms. *n* determines the type of vector norm and is typically equal to 0, 1, or 2. L0 regularization (n=0) penalizes the total number of non-zero parameters in the model, L1 regularization (n=1) penalizes the sum of the parameter absolute values, and L2 regularization (n=2) penalizes the sum of the squared parameters. This minimization problem can be solved analytically if n=2 or using convex optimization if n=1. The hyperparameter λ can be determined using cross-validation. Combinations of these penalties can also be used, such as elastic net regression, which utilizes both L1 and L2 norms.

While regression methods model continuous properties, classification methods are used to model discrete protein properties such as folded/unfolded or active/inactive. Classifiers are especially important for modeling data generated by high-throughput methods such as deep mutational scanning because these methods often bin proteins into broad functional classes. Classification methods try to relate input feature vectors to functional classes (e.g. active/inactive or folded/unfolded). Like the regression models discussed above, classification models can be evaluated using cross-validation, and regularization can be used to prevent overfitting.

Logistic regression is simple classification method that transforms a linear model through the logistic (sigmoid) function to produce binary outputs. The name "logistic regression" is a misnomer because it actually performs classification rather than

regression. Logistic regression parameters can be identified using iterative methods or convex optimization. Logistic regression was recently used to refine molecular energy functions for designing de novo miniproteins[25]. Thousands of miniproteins were designed using Rosetta protein design software, and these designs were screened for folding using a high-throughput yeast display assay. Each protein's structure was modeled and used to generate physical input features such as number of H-bonds, Lennard-Jones energies, and net charge. Logistic regression was then used to map these physical features to whether a design was successful or unsuccessful. The use of logistic regression led to the authors finding that a protein's buried nonpolar surface area was a dominant factor in determining design success. The logistic regression model was used to rank designs and drastically improved the rate of successful designs.

Kernel methods are another modeling approach that is widely used in machine learning and bioinformatics. In contrast to the parametric regression/classification methods described above, kernel methods do not require input feature vectors, but instead a user defined similarity function (or kernel function) is used to compute the "implicit features" by comparing pairs of data points. Kernel methods are more effective at dealing with high dimensional problems than parametric models because they do not have to store large parameter matrices. The similarity function could be as simple as an inner product between feature vectors, or it can represent more complex, potentially infinite dimensional, relationships between data points[26]. This flexibility allows them to learn from unstructured objects such as biological systems. Popular kernel methods include Support Vector Machines (SVMs) and Gaussian Process (GP) regression/classification.

Gaussian processes use kernel functions to define a prior probability distribution over a function space. This allows predictions of both the function mean and its confidence intervals. Gaussian processes have been used to model stability and activity of cytochrome P450s[27]. A structure-based kernel function was developed to define structural similarity between pairs of proteins. GP regression using this kernel function explained 30% more of the variation in P450 thermostability in comparison to linear regression and sequence-based kernels. The structure-based kernel was also used to model enzyme activity and binding affinity for several P450 substrates.

2.4.2 Unsupervised/semisupervised learning

Unlike supervised learning, where the data is labeled or categorized, in unsupervised learning there are no labels associated with each data point. Unsupervised learning can be used to find patterns such as clusters or correlations within data. The main drawback of unsupervised techniques is that the outputs are unknown, i.e. there is no mapping to protein function. However, these techniques still provide valuable information about proteins because of the massive amount of protein sequence data that is currently available. Examples of unsupervised methods include clustering, where data points are grouped based on similarity, and principal component analysis (PCA). PCA is a projection of data onto lower dimensional space in a way that maximizes the variance of the projection. This converts high dimensional input variables into a set of uncorrelated principle components that are ranked based on their variance. These principle components can be used to reduce the dimensionality of a problem and identify important relationships among variables[28].

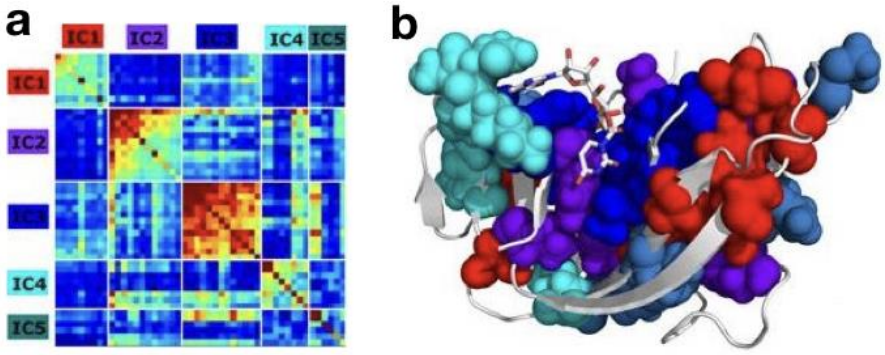

Figure 2.4: Unsupervised learning from protein sequences. (A) Statistical coupling analysis of the RNase superfamily reveals five independent components (ICs) that correspond to groups of coevolving residues (B) These five ICs form contiguous "sectors" in the three-dimensional protein structure. Figure was adapted from [31].

Unsupervised methods can be used to identify patterns in multiple sequence alignments (MSAs) of evolutionarily related proteins. Statistical coupling analysis (SCA) analyzes residue coevolution by performing principal component analysis on a protein family's MSA[29]. The dominant principle components consist of positions that coevolve and can reveal networks of spatially connected amino acids called protein sectors (Figure 2.4). Protein sectors have been demonstrated to play roles in protein dynamics and allostery and may represent functional modules[30,31]. EVmutation is another unsupervised method that models natural sequence variation and simultaneously considers epistasis (non-independence of mutational effects)[32]. Although EVmutation is only parameterized on an MSA (i.e. it is unsupervised), it is capable of predicting the functional effects of amino acid substitutions and residue interdependencies.

Semisupervised methods learn from data sets that contain both unlabeled and labeled data points. Semisupervised approaches can be used in protein engineering to transfer knowledge across protein families. A semisupervised approach was recently

developed that trained an unsupervised embedding model (doc2vec) on a large protein sequence database[16]. These embeddings were then used as the inputs for supervised Gaussian process regression. This approach was used to model channelrhodopsin membrane localization, P450 thermostability, and epoxide hydrolase enantioselectivity.

2.5 Applying statistical models to engineer proteins

Statistical modeling approaches provide unprecedented predictive accuracy for a wide variety of complex protein functions/properties. These models can be used to understand protein function and design new proteins. In addition, many classes of statistical models can provide confidence intervals for their predictions. These confidence intervals can be used to gauge whether a prediction is valid or if it contains too much uncertainty to be useful. We discuss several protein engineering strategies that leverage the predictive power of statistical models.

The most straightforward data-driven protein engineering approach involves training a model on a data set and then extrapolating that model to design best predicted sequences. This method was applied to engineer thermostable fungal cellobiohydrolase class II (CBHII) cellulases[33]. A panel of 33 chimeric CBHIIs was characterized for their thermal inactivation half-lives at elevated temperatures. This data was used to train a linear regression model that related sequence blocks to thermal tolerance. This model was then used to design 18 chimeras that were predicted to have enhanced stability relative to the parent enzymes. Most of these designed CBHII chimeras could hydrolyze cellulose at higher temperatures than most stable parent. A key feature of this extrapolation-based design approach is a relatively small training set (<1% of possible

chimeras) can be used to make predictions over a massive combinatorial sequence space. The CBHII regression model also pointed to a single sequence block that contributed over 8 °C of thermostability[34]. Further analysis revealed that a single amino acid substitution in that block (C313S) was responsible for the elevated thermostability. This example highlights how statistical models can be used to uncover molecular mechanisms contributing to protein function.

It is important to consider the space of sequences that a statistical model can make valid predictions on. This prediction domain is highly dependent on the model's sequence/structure representation. For example, consider a model that uses one-hot encoding to represent protein sequences. This model can only learn the effect of amino acids that are observed in the training set, and therefore can only make predictions about sequences composed of combinations of these observed amino acids. Representations that include information about amino acid properties and/or protein structure can broaden a model's prediction domain. Representations that use three-dimensional structural models to extract key physiochemical properties have potential to generalize well within a protein family and even across protein families.

Statistical models can be incorporated into an iterative directed evolution framework. ProSAR uses a statistical model to guide the search for beneficial mutations [35]. This model consists of a one-hot encoded sequence representation and a partial least squares linear regression model to relate sequence to function. A mutational library is screened, and the model classifies each amino acid substitution as deleterious, neutral, beneficial, or underdetermined (i.e. needing more information). Substitutions that are beneficial or underdetermined are combined with new substitutions in the next round, and

this screen-and-learn process is repeated over multiple rounds. The ProSAR method was used to engineer bacterial halohydrin dehalogenases (HHDH) to perform a cyanation reaction important for the synthesis of the cholesterol-lowering drug Lipitor[35]. 18 rounds of ProSAR yielded HHDH variants with over 35 mutations and increased the volumetric productivity of target reaction by ~4,000-fold. More recently, ProSAR-driven evolution was used to evolve ultra-stable carbonic anhydrase variants (107 ˚C thermostability at pH 10 in 4.2 M solvent) that enhanced the rate of CO_2 capture by 25-fold over the natural enzyme[36].

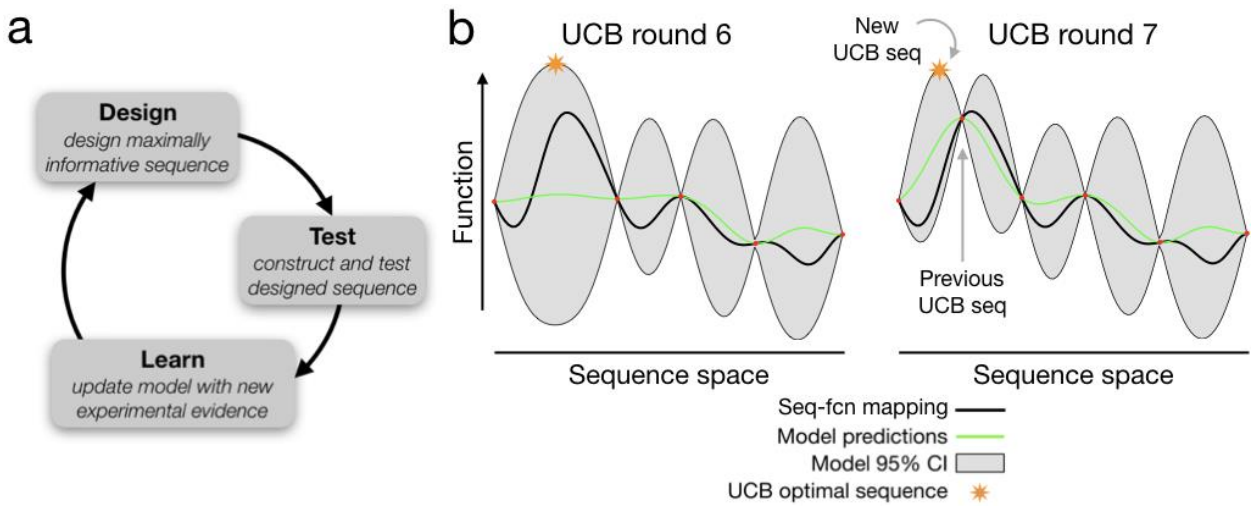

Figure 2.5: Active machine learning. (a) Active learning involves designing maximally informative sequences, experimentally characterizing these sequences, learning from the resulting data, and repeating this process over multiple iterations. (b) Upper-confidence bound (UCB) optimization involves iteratively selecting the sequence with the largest upper confidence bound (mean + confidence interval). The schematic illustrates sequence space in one dimension and the true mapping from sequence to function as a black line. Characterized sequences (small red dots) have accurate model predictions and small confidence intervals. The first panel shows five characterized sequences, which cause the model to propose one UCB optimal sequence (marked with a star). The second panel shows the results after this UCB optimal sequence is characterized—this causes a new UCB sequence to be proposed. This iterative process is guaranteed to efficiently converge to the optimal point.

Statistical models can also be used in an active learning setting that very efficiently explores protein sequence space. Active learning involves sequentially designing an informative experiment, performing that experiment, learning from the resulting data, and repeating the process over multiple cycles (Figure 2.5a. For protein engineering, the active learning algorithm must first learn the sequence-function mapping and then apply this knowledge to design optimized sequences. The primary challenge is how to allocate experimental resources toward understanding the sequence-function mapping versus designing optimized sequences. This trade-off is referred to as the “exploration-exploitation dilemma”, and the objective is to minimize the amount of exploration that is needed to predict optimized sequences. Upper confidence bound (UCB) algorithms

provide a principled framework for trading off between exploration and exploitation modes[37]. The UCB algorithm iteratively selects the point with the largest upper confidence bound (predicted mean plus confidence interval) and therefore encourages sampling of points that are simultaneously optimized and uncertain (Figure 2.5b). A UCB search algorithm was combined with a Gaussian process regression model to optimize cytochrome P450 thermostability[27]. Eight rounds of UCB optimization identified thermostable P450s that were more stable than variants made by rational design, recombination or directed evolution.

2.6 Conclusions and future outlook

The protein sequence-structure-function mapping involves thousands of interacting atoms, a practically infinite number of dynamic conformational states, and physical processes that span multiple length and time scales. This mapping is extremely difficult to model from a physical perspective. In contrast, statistical methods are able to learn complex interrelationships directly from experimental data. This top-down understanding of complex systems allows discovery of new functional mechanisms and provides exceptional predictive accuracy.

This chapter provides an overview of emerging data-driven approaches to model and engineer proteins. We have described statistical representations of proteins, how these representations can be used to learn from data, and practical protein engineering applications of these models. As a relatively new field, there is still significant room for improving these methods, especially in the area of sequence and structure representations. Ideal representations would be sparse, but still have a broad prediction

domain. These representations may integrate different sources of information (evolutionary, biochemical, and physical) into a single unified model. Advanced machine learning methods such as dictionary learning and deep learning attempt to learn new representations directly from data and could play an important role in protein modeling. Another key challenge for the field is data access and sharing. While there are many interesting sequence-structure-function data sets, they are often buried in a publication's supplemental information and very difficult to parse/organize. Efforts to share data on public repositories and databases such as ProtaBank will greatly accelerate progress in the field.

In addition to proteins, statistical approaches can be used to model genotype-phenotype relationships across all levels of biological organization. For example, linear regression was used to model product titers in a multi-enzyme biosynthetic pathway; this model was then used to optimize enzyme expression levels to maximize overall product production[38]. Another example used compressed sensing methods to model a protein's DNA-binding specificity[39]. Statistical methods have been widely used in genetics relate phenotypes to genetic loci using quantitative trait locus (QTL) mapping[40].

Data-driven approaches are transforming every field of science and engineering. This revolution has been triggered by the confluence of advances in data generation, data access, and data analysis/interpretation. Advanced experimental technologies are allowing us to analyze biological systems on an unprecedented scale and resolution. The resulting data is also becoming readily accessible through large, public biological databases and repositories. At the same time, there have been tremendous advances in artificial intelligence and pattern recognition. Widespread interest in machine learning has

also driven improvements in software packages such as the Scikit-learn and Keras deep learning Python libraries. Data-driven approaches leverage the continuously expanding sea of data and will play an increasingly important role in biological discovery and engineering.

References:

(1) Lazaridis, T.; Karplus, M. Effective Energy Functions for Protein Structure Prediction. *Curr. Opin. Struct. Biol.* **2000**, *10* (2), 139–145. https://doi.org/10.1016/S0959-440X(00)00063-4.

(2) Li, Z.; Yang, Y.; Zhan, J.; Dai, L.; Zhou, Y. Energy Functions in de Novo Protein Design: Current Challenges and Future Prospects. *Annu. Rev. Biophys.* **2013**, *42*, 315–335.

(3) Bradley, P.; Misura, K. M. S.; Baker, D. Toward High-Resolution de Novo Structure Prediction for Small Proteins. *Science (80-.).* **2005**, *309* (5742), 1868–1871.

(4) Baker, D. An Exciting but Challenging Road Ahead for Computational Enzyme Design. *Protein Sci.* **2010**, *19* (10), 1817–1819.

(5) Kosuri, S.; Church, G. M. Large-Scale de Novo DNA Synthesis: Technologies and Applications. *Nat. Methods* **2014**, *11* (5), 499–507. https://doi.org/10.1038/nmeth.2918.

(6) Apweiler, R.; Bairoch, A.; Wu, C. H.; Barker, W. C.; Boeckmann, B.; Ferro, S.; Gasteiger, E.; Huang, H.; Lopez, R.; Magrane, M.; et al. UniProt: The Universal Protein Knowledgebase. *Nucleic Acids Res.* **2017**, *45*, D158–D169. https://doi.org/10.1093/nar/gkw1099.

(7) Rose, P. W.; Prlic, A.; Altunkaya, A.; Bi, C.; Bradley, A. R.; Christie, C. H.; Di Costanzo, L.; Duarte, J. M.; Dutta, S.; Feng, Z.; et al. The RCSB Protein Data Bank: Integrative View of Protein, Gene and 3D Structural Information. *Nucleic Acids Res.* **2017**, *45*, D271–D281. https://doi.org/10.1093/nar/gkw1002.

(8) Wang, C. Y.; Chang, P. M.; Ary, M. L.; Allen, B. D.; Chica, R. A.; Mayo, S. L.; Olafson, B. D. ProtaBank: A Repository for Protein Design and Engineering Data. *Protein Sci.* **2018**. https://doi.org/10.1002/pro.3406.

(9) Hietpas, R. T.; Jensen, J. D.; Bolon, D. N. A. Experimental Illumination of a Fitness Landscape. *Proc. Natl. Acad. Sci. U. S. A.* **2011**, *108* (19), 7896–7901.

(10) Araya, C. L.; Fowler, D. M. Deep Mutational Scanning: Assessing Protein Function on a Massive Scale. *Trends Biotechnol.* **2011**, *29* (9), 435–442.

(11) Romero, P. A.; Tran, T. M.; Abate, A. R. Dissecting Enzyme Function with Microfluidic-Based Deep Mutational Scanning. *Proc. Natl. Acad. Sci.* **2015**, *112* (23), 201422285. https://doi.org/10.1073/pnas.1422285112.

(12) Whitehead, T. A.; Chevalier, A.; Song, Y.; Dreyfus, C.; Fleishman, S. J.; De Mattos, C.; Myers, C. A.; Kamisetty, H.; Blair, P.; Wilson, I. A.; et al. Optimization of Affinity, Specificity and Function of Designed Influenza Inhibitors Using Deep Sequencing. *Nat. Biotechnol.* **2012**, *30* (May), 1–9. https://doi.org/10.1038/nbt.2214.

(13) Bloom, J. D. An Experimentally Determined Evolutionary Model Dramatically Improves Phylogenetic Fit. *Mol. Biol. Evol.* **2014**, *31* (8), 1956–1978. https://doi.org/10.1093/molbev/msu173.

(14) Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *CoRR* **2013**, *abs/1301.3*.

(15) Le, Q. V; Mikolov, T. Distributed Representations of Sentences and Documents. *CoRR* **2014**, *abs/1405.4*.

(16) Zachary Wu, C. N. B. and F. H. A. K. K. Y. Learned Protein Embeddings for

Machine Learning. **2018**, No. April, 1–7. https://doi.org/10.1093/bioinformatics/xxxxx.

(17) Chothia, C.; Lesk, A. M. The Relation between the Divergence of Sequence and Structure in Proteins. *EMBO J.* **1986**. https://doi.org/060 fehlt.

(18) Misura, K. M. S.; Chivian, D.; Rohl, C. A.; Kim, D. E.; Baker, D. Physically Realistic Homology Models Built with ROSETTA Can Be More Accurate than Their Templates. *Proc. Natl. Acad. Sci.* **2006**. https://doi.org/10.1073/pnas.0509355103.

(19) Carlin, D. A.; Caster, R. W.; Wang, X.; Betzenderfer, S. A.; Chen, C. X.; Duong, V. M.; Ryklansky, C. V.; Alpekin, A.; Beaumont, N.; Kapoor, H.; et al. Kinetic Characterization of 100 Glycoside Hydrolase Mutants Enables the Discovery of Structural Features Correlated with Kinetic Constants. *PLoS One* **2016**, *11* (1), 1–14. https://doi.org/10.1371/journal.pone.0147596.

(20) Dudek, A.; Arodz, T.; Galvez, J. Computational Methods in Developing Quantitative Structure-Activity Relationships (QSAR): A Review. *Comb. Chem. High Throughput Screen.* **2006**, *9* (3), 213–228. https://doi.org/10.2174/138620706776055539.

(21) Hellberg, S.; Sjöström, M.; Skagerberg, B.; Wold, S. Peptide Quantitative Structure-Activity Relationships, a Multivariate Approach. *J. Med. Chem.* **1987**, *30* (7), 1126–1135. https://doi.org/10.1021/jm00390a003.

(22) Jonsson, J.; Norberg, T.; Carlsson, L.; Gustafsson, C.; Wold, S. Quantitative Sequence-Activity Models (QSAM) - Tools for Sequence Design. *Nucleic Acids Res.* **1993**, *21* (3), 733–739. https://doi.org/10.1093/nar/21.3.733.

(23) Li, Y.; Drummond, D. A.; Sawayama, A. M.; Snow, C. D.; Bloom, J. D.; Arnold, F. H. A Diverse Family of Thermostable Cytochrome P450s Created by Recombination of Stabilizing Fragments. *Nat. Biotechnol.* **2007**, *25* (9), 1051–1056. https://doi.org/10.1038/nbt1333.

(24) Otey, C. R.; Landwehr, M.; Endelman, J. B.; Hiraga, K.; Bloom, J. D.; Arnold, F. H. Structure-Guided Recombination Creates an Artificial Family of Cytochromes P450. *PLoS Biol.* **2006**, *4* (5), 789–798. https://doi.org/10.1371/journal.pbio.0040112.

(25) Rocklin, G. J.; Chidyausiku, T. M.; Goreshnik, I.; Ford, A.; Houliston, S.; Lemak, A.; Carter, L.; Ravichandran, R.; Mulligan, V. K.; Chevalier, A.; et al. Global Analysis of Protein Folding Using Massively Parallel Design, Synthesis, and Testing. *Science (80-.).* **2017**, *357* (6347), 168–175. https://doi.org/10.1126/science.aan0693.

(26) Rasmussen, C. E.; Williams, C. *Gaussian Processes for Machine Learning*; Adaptive Computation and Machine Learning; MIT Press: Cambridge, MA, 2006; Vol. 14.

(27) Romero, P. a; Krause, A.; Arnold, F. H. Navigating the Protein Fitness Landscape with Gaussian Processes. *Proc. Natl. Acad. Sci. U. S. A.* **2013**, *110* (3), E193-201. https://doi.org/10.1073/pnas.1215251110.

(28) Bishop, C. M. *Pattern Recognition and Machine Learning*; Springer Science+Business Media LLC, New York, NY, 2006.

(29) Lockless, S. W.; Ranganathan, R. Evolutionarily Conserved Pathways of Energetic Connectivity in Protein Families. *Science (80-.).* **1999**, *286* (5438),

295–299.

(30) Halabi, N.; Rivoire, O.; Leibler, S.; Ranganathan, R. Protein Sectors: Evolutionary Units of Three-Dimensional Structure. https://doi.org/10.1016/j.cell.2009.07.038.

(31) Narayanan, C.; Gagné, D.; Reynolds, K. A.; Doucet, N. Conserved Amino Acid Networks Modulate Discrete Functional Properties in an Enzyme Superfamily. *Sci. Rep.* **2017**, *7* (1), 1–9. https://doi.org/10.1038/s41598-017-03298-4.

(32) Hopf, T. A.; Ingraham, J. B.; Poelwijk, F. J.; Schärfe, C. P. I.; Springer, M.; Sander, C.; Marks, D. S. Mutation Effects Predicted from Sequence Co-Variation. *Nat. Biotechnol.* **2017**, *35* (2). https://doi.org/10.1038/nbt.3769.

(33) Heinzelman, P.; Snow, C. D.; Wu, I.; Nguyen, C.; Villalobos, A.; Govindarajan, S.; Minshull, J.; Arnold, F. H. A Family of Thermostable Fungal Cellulases Created by Structure-Guided Recombination. *Proc. Natl. Acad. Sci. U. S. A.* **2009**, *106* (14), 5610–5615.

(34) Heinzelman, P.; Snow, C. D.; Smith, M. A.; Yu, X.; Kannan, A.; Boulware, K.; Villalobos, A.; Govindarajan, S.; Minshull, J.; Arnold, F. H. SCHEMA Recombination of a Fungal Cellulase Uncovers a Single Mutation That Contributes Markedly to Stability. *J. Biol. Chem.* **2009**, *284* (39), 26229–26233.

(35) Fox, R. J.; Davis, S. C.; Mundorff, E. C.; Newman, L. M.; Gavrilovic, V.; Ma, S. K.; Chung, L. M.; Ching, C.; Tam, S.; Muley, S.; et al. Improving Catalytic Function by ProSAR-Driven Enzyme Evolution. *Nat. Biotechnol.* **2007**, *25* (3), 338–344. https://doi.org/10.1038/nbt1286.

(36) Alvizo, O.; Nguyen, L. J.; Savile, C. K.; Bresson, J. A.; Lakhapatri, S. L.; Solis, E. O. P.; Fox, R. J.; Broering, J. M.; Benoit, M. R.; Zimmerman, S. A.; et al. Directed

Evolution of an Ultrastable Carbonic Anhydrase for Highly Efficient Carbon Capture from Flue Gas. *Proc. Natl. Acad. Sci.* **2014**, *111* (46), 16436–16441. https://doi.org/10.1073/pnas.1411461111.

(37) Auer, P. Using Confidence Bounds for Exploitation-Exploration Trade-Offs. *J. Mach. Learn. Res.* **2002**, *3* (3), 397–422.

(38) Lee, M. E.; Aswani, A.; Han, A. S.; Tomlin, C. J.; Dueber, J. E. Expression-Level Optimization of a Multi-Enzyme Pathway in the Absence of a High-Throughput Assay. *Nucleic Acids Res.* **2013**, *41* (22), 10668–10678. https://doi.org/10.1093/nar/gkt809.

(39) AlQuraishi, M.; McAdams, H. H. Direct Inference of Protein-DNA Interactions Using Compressed Sensing Methods. *Proc. Natl. Acad. Sci.* **2011**, *108* (36), 14819–14824. https://doi.org/10.1073/pnas.1106460108.

(40) Doerge, R. W. Mapping and Analysis of Quantitative Trait Loci in Experimental Populations. *Nature Reviews Genetics*. 2002. https://doi.org/10.1038/nrg703.

Chapter 3

Machine learning-guided acyl-ACP reductase engineering for improved *in vivo* fatty alcohol production

Authors and Contributors:

- Jonathan C. Greenhalgh performed experiments, analyzed the data, made figures, and wrote the manuscript.
- Sarah A. Fahlberg assisted with experimental characterization of acyl-ACP reductases and protein purification and edited and reviewed the manuscript.
- Dr. Brian F. Pfleger provided guidance and edited and reviewed the manuscript.
- Dr. Philip A. Romero, provided guidance, made figures and edited and reviewed the manuscript.

This chapter is adapted from the following manuscript:

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*. Nature Communications. **2021**.

3.1 Abstract

Alcohol forming fatty acyl reductases (FARs) catalyze the reduction of thioesters to alcohols and are key enzymes for microbial production of fatty alcohols. Many metabolic engineering strategies utilize FARs to produce fatty alcohols from intracellular acyl-CoA and acyl-ACP pools; however, enzyme activity, especially on acyl-ACPs, remains a significant bottleneck to high-flux production. Here, we engineer FARs with enhanced activity on acyl-ACP substrates by implementing a machine learning (ML)-driven approach to iteratively search the protein fitness landscape. Over the course of ten design-test-learn rounds, we engineer enzymes that produce over twofold more fatty alcohols than the starting natural sequences. We characterize the top sequence and show that it has an enhanced catalytic rate on palmitoyl-ACP. Finally, we analyze the sequence-function data to identify features, like the net charge near the substrate-binding site, that correlate with *in vivo* activity. This work demonstrates the power of ML-models in improving function of traditionally difficult-to-engineer proteins.

3.2 Introduction

Fatty acyl reductases (FARs) are vital for the microbial synthesis of key primary and secondary metabolites such as fatty aldehydes, waxes, alkanes, and fatty alcohols. These enzymes often interface with fatty acid anabolic/catabolic pathways and catalyze the reduction of thioester bonds found in acyl-acyl carrier proteins[1] (acyl-ACPs) and acyl-coenzyme As (acyl-CoAs)[2]. These enzymes typically have a preference for either acyl-ACP or acyl-CoA substrates, but also display cross reactivity due to the common thioester bond in both substrates. Some FARs perform only one, two-electron, reduction step to produce aldehydes[3], while others can perform two sequential reduction steps (totaling four electrons) to produce alcohols directly[4–6].

The alcohol-forming FAR enzymes capable of complete reduction of thioesters to alcohols have been widely used in metabolic engineering for producing fatty alcohols[7–11]. The enzymes Maqu 2220 and MA-ACR from *Marinobacter aquaeloei* display high activity on acyl-CoA substrates and produce the corresponding fatty alcohols[2,4,11]. These enzymes can be incorporated to feed off of the reverse beta oxidation pathway to yield high levels of alcohols[8]. Another common metabolic engineering strategy involves terminating the host organism's fatty acid elongation cycle with a thioesterase to produce a fatty acid that can then be converted to an acyl-CoA by an ATP dependent ligase, and then finally converted to an alcohol by a FAR[7,9,12]. This approach was recently applied using an engineered C8-specific thioesterase to produce octanol at a titer of 1.3 g/L[9]. While these titers are impressive, alcohol production could be more efficient with enzymes that bypass the thioesterase-ligase route, and instead directly convert acyl-ACPs to alcohols[13].

Alcohol-forming FARs that prefer acyl-ACP substrates are less well characterized, and often display low to moderate activity relative to enzymes that prefer acyl-CoA substrates. Engineering alcohol-forming FARs such as MA-ACR to have higher activity on acyl-ACP substrates would open up new highly efficient pathways to making fatty alcohols *in vivo*. However, these enzymes are challenging to engineer using traditional protein engineering methods. MA-ACR and its close homologs lack high-resolution crystal structures needed for most computational and rational engineering approaches. Directed evolution strategies are also difficult because fatty alcohol production cannot be assayed in high-throughput. Machine learning (ML)-based protein engineering has recently emerged as an efficient strategy for engineering proteins with limited structural and functional information[14–20]. Machine learning algorithms can infer the protein sequence-function mapping given a limited experimental sampling of the landscape[14]. The resulting sequence-function models can be used to computationally explore sequence space and predict optimized sequences.

In this work, we apply an ML-based protein engineering framework to engineer acyl-ACP reductases to produce fatty alcohols *in vivo*. We start by characterizing the ability of MA-ACR and related enzymes to produce fatty alcohols from intracellular acyl-ACP pools. We then design a large library of chimeric enzymes and develop an ML-based protein optimization strategy to rapidly identify highly active sequences. Our approach consists of generating diverse initial sequence sampling to get a preliminary view of the landscape, followed by ten iterative design-test-learn cycles to efficiently search the landscape and discover optimized sequences. We show that the algorithm converges on highly active acyl-ACP reductases that produce 4.9-fold more fatty alcohols than MA-

ACR. We evaluate the performance of the engineered enzymes *in vitro* and find the improved alcohol titers are the result of engineered enzymes with increased catalytic efficiency. Finally, we perform a statistical analysis of the landscape and identify key sequence elements that contribute to enzyme activity. Many of these elements are located near the enzyme's putative substrate entry channel and may be involved with modulating the preference between acyl-CoA and acyl-ACP substrates. These results open future directions to engineer enzymes for efficient microbial production of fatty alcohols.

3.3 Results

3.3.1 In vivo fatty alcohol production by natural and chimeric acyl-ACP reductases

We focused our protein engineering efforts on MA-ACR from *Marinobacter aquaeloei* because it displays high *in vivo* activity on acyl-CoA substrates[7–9] and it was also suspected to accept acyl-ACP substrates. MA-ACR consists of two domains that sequentially reduce thioesters to alcohols (Figure 3.1a). The C-terminal acyl-thioester reductase (ATR) domain reduces thioesters from ACP or CoA substrates to aldehydes, and the N-terminal aldehyde reductase domain (AHR) reduces aldehydes to alcohols[4]. We also identified two related enzymes from *Marinobacter BSs20148* and *Methylibium Sp. T29* that have 60-81% sequence identity with MA-ACR (Figure 3.1b) and were previously shown to produce alcohols from acyl-CoAs[8,9]. Throughout the remainder of this chapter, we refer to the FAR enzymes from *Marinobacter aquaeloei, Marinobacter BSs20148*, and *Methylibium Sp. T29* as MA-ACR, MB-ACR, and MT-ACR, respectively.

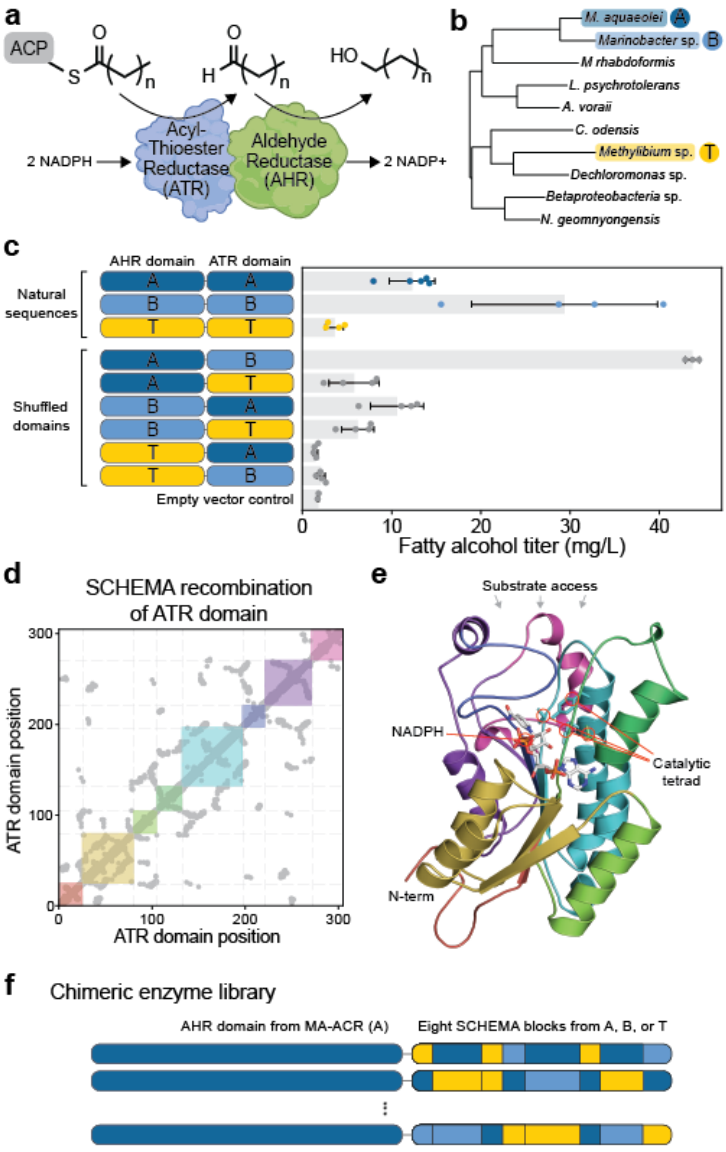

Figure 3.1: Acyl-ACP reductase activity of natural and chimeric enzymes**.** (a) Alcohol-forming acyl-ACP reductases consist of two domains that sequentially reduce acyl-ACP substrates to aldehydes, and then aldehydes to alcohols. (b) We focused our studies on three diverse sequences from *M. aquaeloei* (dark blue)*, Marinobacter BSs20148* (light blue), and *Methylibium Sp. T29* (yellow), which we refer to as A, B, and T, respectively. (c) Total fatty alcohol production by the three natural sequences and the six chimeric enzymes generated by shuffling their AHR and ATR domains. The error bars represent one standard deviation centered at the mean of four replicates (n=4) from cultures derived from individual colonies, except for MA-ACR (where n=5), parent B (fusion A-B, where n=3) and the empty vector (n=2). (d) ATR domain residue-residue contact map used for SCHEMA recombination. The colored squares depict the eight sequence blocks from the SCHEMA design that minimizes structural disruption. (e) The SCHEMA blocks mapped onto the ATR domain's three-dimensional structure (the colors correspond to the squares in panel d). (f) Our chimeric ATR library was fused to the AHR domain from MA-ACR.

We characterized the ability of these three natural enzymes to produce fatty alcohols from intracellular acyl-ACP pools by introducing them into *E. coli* RL08ara[21], a strain that lacks the *fadD* gene, which encodes an acyl-CoA ligase. Deletion of *fadD* decreases the formation of acyl-CoAs and thus presents the enzymes with substrates that are predominantly acyl-ACPs from fatty acid biosynthesis[10,13]. We grew each strain under aerobic conditions, extracted the fatty alcohols and measured the fatty alcohol (C6-C16) titers using gas chromatography. We found the enzyme MB-ACR from *Marinobacter BSs20148* displayed more than double the total fatty alcohol titer of MA-

ACR (Figure 3.1c). These results suggest that MB-ACR may have a preference for acyl-ACP substrates because it was previously shown to have lower activity than MA-ACR on acyl-CoA substrates[8].

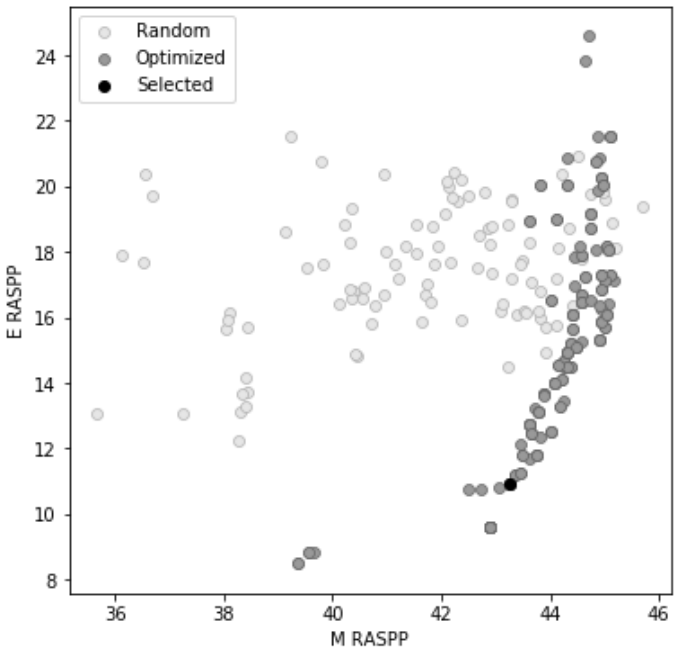

Figure 3.2: RASPP chimeric enzyme library design. The SCHEMA-RASPP algorithm was used to identify sets of breakpoints that simultaneously maximize the average mutation level (M) of the library and minimize the SCHEMA energy (E) of the library. Each point in the graph represents a library of chimeras, and the library that was selected is shown in black. Libraries with randomized breakpoints are shown in light gray for comparison.

We next characterized the fatty alcohol production from chimeric enzymes generated by swapping AHR and ATR domains between the three natural sequences. Of the six possible chimeric enzymes, we found the chimera with an AHR domain from MA-ACR and the ATR domain from MB-ACR displayed the highest fatty alcohol titers (Figure 3.1c). This chimeric enzyme produced ~50% more fatty alcohol than MB-ACR and roughly three-fold more fatty alcohol than MA-ACR. The ATR domain from MT-ACR also displayed increased activity (~1.5x) when fused to the AHR domain from MA-ACR. These results suggest that MA-ACR's AHR domain is more efficient than the AHR domains from the two other natural enzymes.

To further explore how gene shuffling can enhance fatty alcohol production, we designed a large library of ATR domains using SCHEMA[22–24] structure-guided recombination (Figure 3.1d). Our design used a homology model of MA-ACR's ATR domain to define the family's contact map and identified seven breakpoints within the domain that balance structural disruption with library diversity (Figure 3.2).

These seven breakpoints define eight sequence blocks that span the ATR domain's structure (Figure 3.1e). Notably, the structure's substrate access channel is composed of blocks 4, 5, 6, 7, and 8, and diversity at these positions may result in changes in the enzyme's substrate preference. Each of the eight sequence blocks can be inherited from one of the three natural enzymes to define a combinatorial sequence space of 3^8 sequences. However, block 6 from MA-ACR and MB-ACR happened to be perfectly conserved, and therefore the total library diversity is $2*3^7 = 4{,}374$ sequences. We fused our chimeric ATR domains with the highly active AHR domain from MA-ACR (Figure 3.1f).

For the remainder of the chapter, we refer to chimeras by a block sequence (e.g. A-ABTABTAB) that specifies which of the three enzymes each sequence fragment was inherited from. Here, A, B, and T correspond to MA-ACR, MB-ACR, and MT-ACR, respectively; the first position specifies the AHR domain and the remaining positions specify the ATR domain's eight SCHEMA blocks. We also refer to the three sequences that have all eight ATR blocks from a single natural enzyme as 'parental' enzymes. Here 'parent A' has the block sequence A-AAAAAAAA, 'parent B' is A-BBBBBBBB, and 'parent T' is A-TTTTTTTT.

3.3.2 Increasing fatty alcohol production with ML-driven enzyme engineering

We aimed to identify the most highly active enzymes from our chimeric ATR domain library. However, the chimera space consists of thousands of unique sequences and is much too large to fully characterize using our low-throughput gas chromatography assay. Instead, we developed an ML-based sequence optimization method to rapidly identify highly active sequences with minimal experimentation (Figure 3.3a). Our approach consists of generating diverse initial sequence sampling to get a preliminary view of the landscape, followed by iterative design-test-learn cycles to efficiently search the landscape and discover optimized sequences.

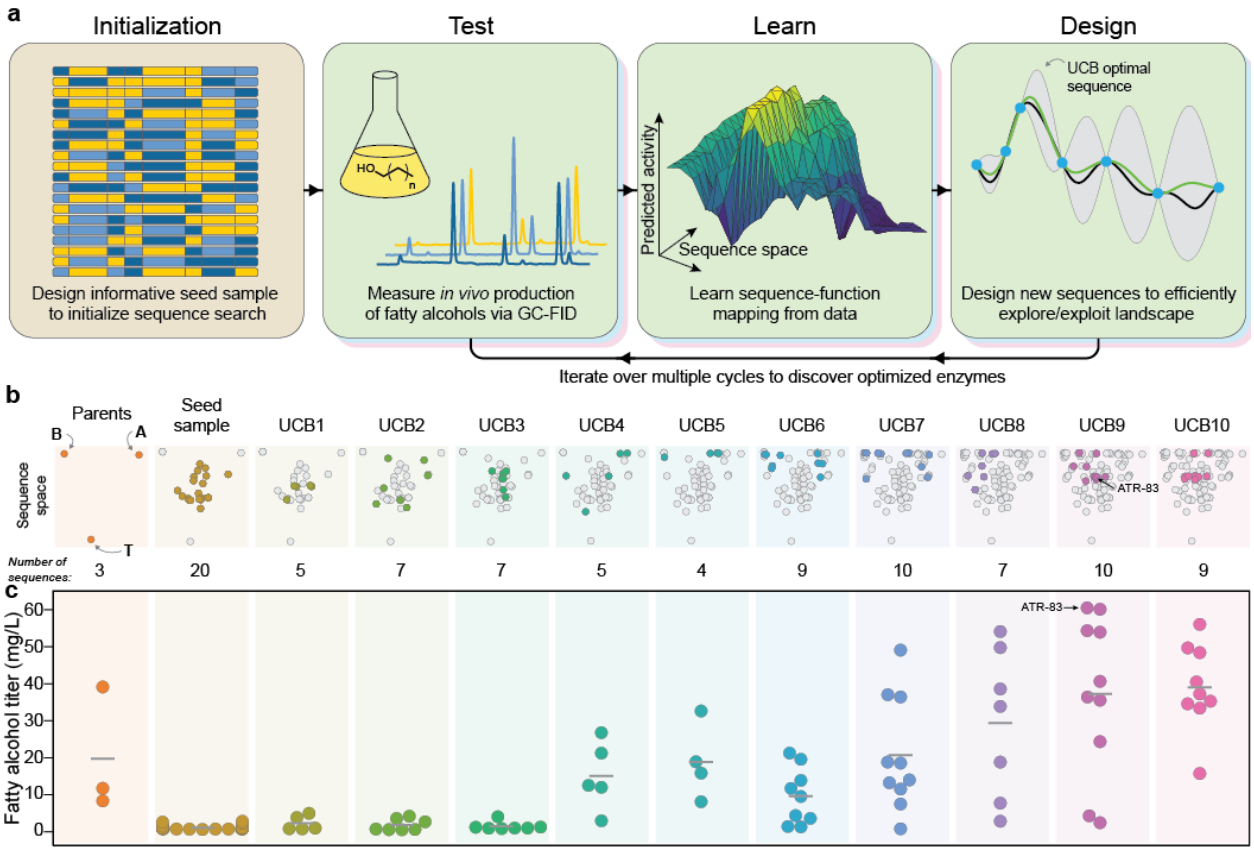

Figure 3.3: ML-accelerated protein sequence optimization**.** (a) An overview of our sequence space search strategy. We first initialize the search by designing a diverse set of sequences that broadly sample the landscape. We then iterate through multiple design-test-learn cycles to efficiently understand and optimize *in vivo* fatty alcohol production. (b) Sequence space visualization over ten rounds of UCB optimization (each round is shown as a different color). The three parent enzymes are found at the vertices of this chimeric sequence space and all chimeras fall within the parents' envelope. The UCB optimization started by broadly sampling the landscape, but quickly converged on highly active regions. (c) The *in vivo* fatty alcohol titers over the course of the sequence optimization. Each point depicts an individual sequence's mean fatty alcohol production in the sequence optimization phase and the horizontal grey bars represent the average titer during that round of sequence optimization. The mean, standard deviation and number of replicates, where *n* is equal to the number of cultures analyzed (each one from an individual colony), are shown in Supplementary Table 3.5.

We generated a diverse initial sampling of sequence space using a greedy algorithm to identify the set of 20 sequences that maximized the Gaussian mutual information with the full chimera space consisting of 4,374 sequences. We then constructed these sequences and experimentally measured their fatty alcohol titers in three *E. coli* strains (Figure 3.4). We evaluated the chimeras' titers in RL08ara under aerobic conditions to assess activity on acyl-ACP substrates. Seventeen of the twenty sequences displayed no measurable alcohol production in RL08ara and the remaining three produced low titers that were below the least productive parent (T). We also tested their activity in the CM24 strain[8] that was engineered to produce high concentrations of acyl-CoA substrates. In the CM24 strain under anaerobic conditions, we found two of the twenty chimeras produced alcohol titers comparable to least productive parent (B). Finally, we also evaluated alcohol titers in BL21(DE3) under aerobic conditions and found eight of the twenty chimeras produced measurable alcohols. Notably, the panel of twenty chimeras displayed differential activity across strains, which could be the result of varying substrate pools within each strain and different substrate preferences between the chimeric enzymes.

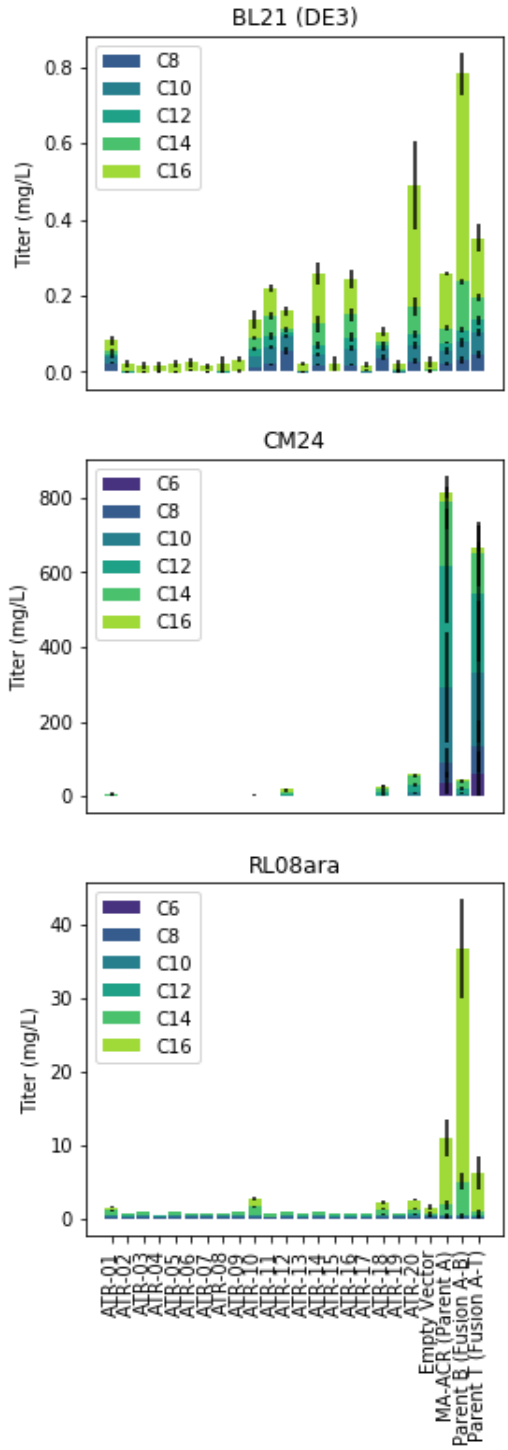

Figure 3.4: Total fatty alcohol titer data in BL21(DE3) (aerobic), CM24 (anaerobic) and RL08ara (aerobic) for the initial chimera seed sample. Different carbon chain lengths (i.e. C6, C8 etc.) are shown as different colors. The total titers from BL21 (DE3) and CM24 were used to train the models used for the first round of UCB optimization, and the RL08ara data was used for all subsequent rounds. Data are presented as the mean ± standard deviation (SD) for each fatty alcohol chain length.

The fatty alcohol titer data from these 20 initial sequences was used to train Gaussian process (GP) sequence-function models that can make predictions across the entire chimera space. Importantly, GPs also provide estimates of the model's uncertainty (confidence intervals) that can be used to gauge the reliability of predictions and highlight gaps in its understanding of the landscape[14,15].

With the initialized GP sequence-function model, we then iterated through multiple design-test-learn cycles with the goal of identifying the optimal sequence with minimal experimental samples. The sequences for the next round of experimentation were designed using an upper-confidence bound (UCB) criterion that simultaneously explores uncertain regions of the landscape and samples sequences that are predicted to be optimized. UCB optimization provides strong theoretical guarantees for efficiently balancing exploration and exploitation[25,26], and should rapidly converge on the optimal sequences. During each iteration, we designed 10-12 sequences using a batch mode UCB criterion (see section 3.5), assembled the corresponding genes, transformed them into *E. coli*, and measured each strain's fatty alcohol titer using gas chromatography. The new data was then used to update the sequence-function model and the process was repeated. We performed a total of ten rounds of UCB sequence optimization and saw gradual improvements in fatty alcohol titers (Figure 3.3). The details of each round of UCB optimization can be found in Supplementary Table 3.4.

The UCB sequence optimization converged on multiple highly active acyl-ACP reductases. The enzyme with the highest titer had a block sequence of A-ATBBAAAB and we refer to this top sequence as ATR-83. Additional *in vivo* characterization showed that ATR-83 produces a total titer of 54 $\pm$ 11 mg/L fatty alcohols (Supplementary Table

3.5), which is nearly five-fold greater than the titer of MA-ACR and about two-fold greater than the best natural sequence (MB-ACR). The alcohols produced by ATR-83 and the other top chimeras consisted of primarily hexadecanol (C16) and some tetradecanol (C14). This product distribution is expected since long chain acyl-ACPs are the primary precursors for the lipids that make up the cell membranes in *E. coli*[27,28].

3.3.3 Improved fatty alcohol production occurs via an enhanced catalytic rate on acyl-ACP substrates

Our engineered acyl-ACP reductase chimeras produce several-fold more fatty alcohols than the initial natural sequences. Increased flux through the metabolic pathway can be the result of improved protein stability and/or expression, enzyme kinetic properties, or possibly interactions with other components of the pathway. We performed further biochemical analysis of the engineered enzymes to better understand how they increase alcohol production.

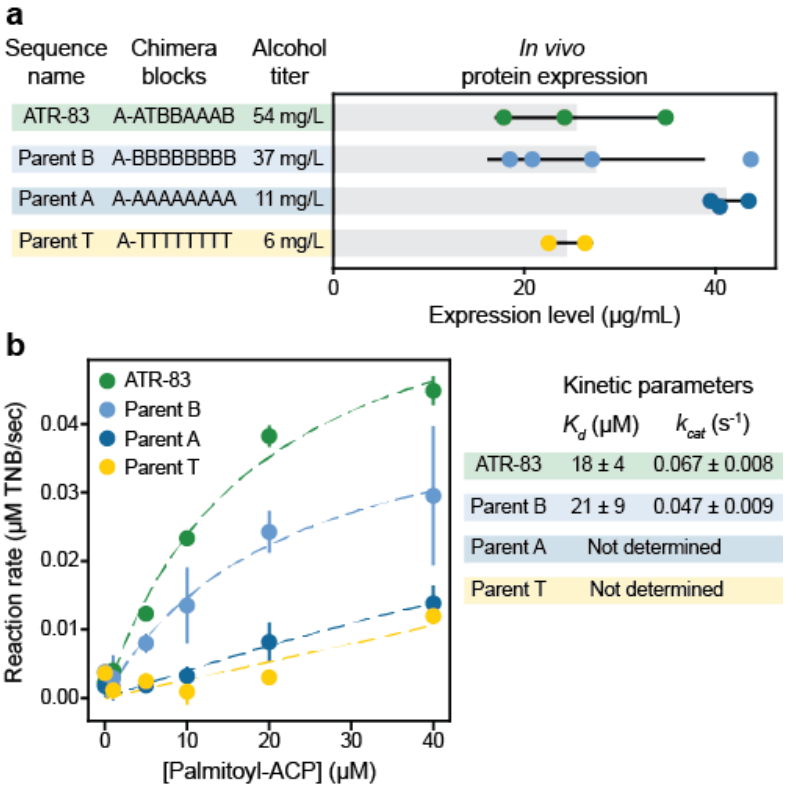

Figure 3.5: Expression levels and kinetic activity of selected ATRs. (a) We measured the expression levels of the ATR-83 chimera and the three parental enzymes. These four enzymes displayed no significant differences in expression despite the large differences in their alcohol titer. The error bars represent one standard deviation centered at the mean (n=3, 4, 3, and 2 for ATR-83, parent B, parent A and parent T respectively, where n is the number of cultures analyzed, each from individual colonies). (b) We characterized the kinetics of selected enzymes on palmitoyl-ACP. The error bars represent one standard deviation centered at the mean (n=4 technical replicates). ATR-83 displayed a higher turnover number (k_{cat}) relative to parent B, and higher activity overall compared to the other parents. The kinetic parameters for parents A and T could not be precisely determined due to their low overall activity and the resulting poor fit to the Michaelis-Menten model.

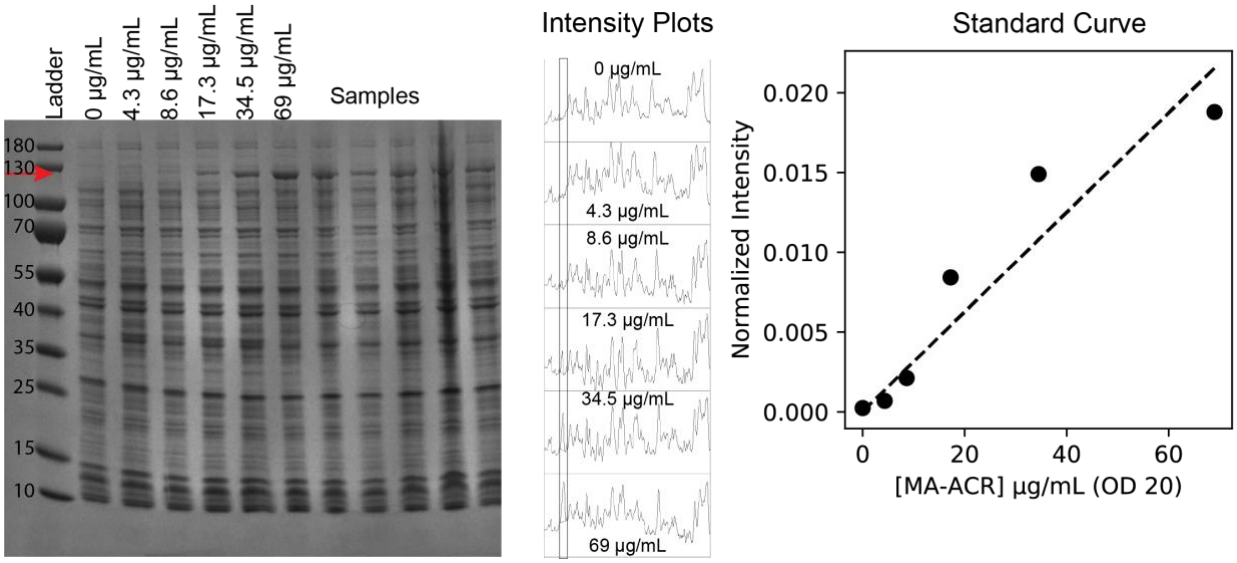

Figure 3.6: Representative SDS-PAGE gel for measuring ATR expression level (the molecular weight markers are in units of kDa), image analysis of the intensity plots, and standard curve. The red arrow in the gel panel shows the expected molecular weight of MA-ACR.

We first measured the level of enzyme expression in the production strain (Figure 3.6, Figure 3.5a). We found all sequences were expressed at high levels and there were no statistically significant differences between the natural and engineered sequences. Next, we purified the enzymes and measured their kinetic properties on palmitoyl-ACP (Figure 3.5b,c). ATR-83 and parent B displayed similar K_M values for palmitoyl-ACP, but ATR-83 had a substantially larger turnover number. ATR-83's increase in k_{cat} matches its improvements in fatty alcohol titer. Taken together with the enzyme expression data, this suggests that the engineered enzymes are increasing alcohol production by an enhanced catalytic rate.

We also analyzed the enzymes' activity on CoA substrates and found that ATR-83 has a lower activity than the parents on palmitoyl-CoA (Figure 3.7). This suggests that ATR-83 may not be a faster enzyme overall, but instead displays an altered preference

for ACP over CoA. This altered preference could be the result of changes in the protein surface that interacts with the ACP substrate.

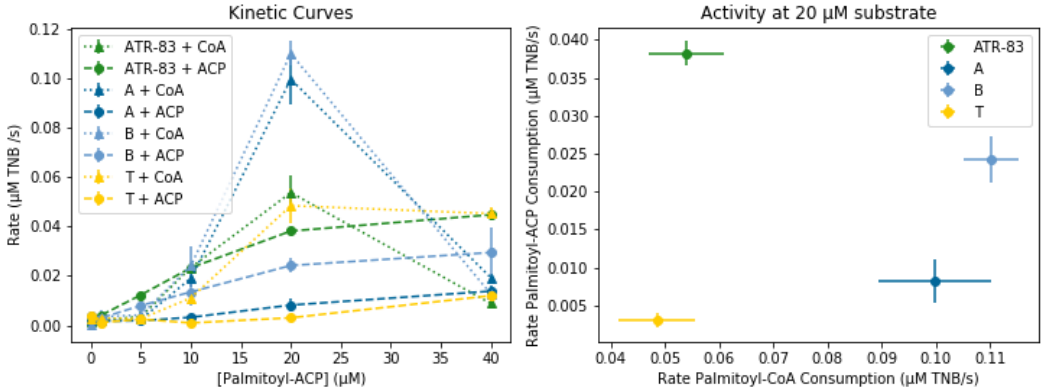

Figure 3.7: Comparison of enzyme activity on palmitoyl-CoA substrates and acyl-ACP substrates for the three parental enzymes (Parents A, B, and T) and ATR-83. Data are presented as the mean of four technical replicates ± SD.

3.3.4 Statistical analysis of the enzyme landscape reveals features that influence fatty alcohol production

Over the course of our UCB sequence optimization, we collected 96 data points mapping chimeric sequences to fatty alcohol titers. This sequence-function data can serve as a rich resource for understanding how protein sequence and structure impact *in vivo* enzyme activity. We trained a GP regression model to predict fatty alcohol titers from sequence. This model displayed excellent predictive ability in a cross-validation test (Figure 3.8).

We used this predictive model to assess how each chimera sequence block contributes to overall enzyme activity (Figure 3.9a). We see that most block positions influence activity and display a broad range of effects. The three sequence blocks with the largest positive contribution were block 7 from MA-ACR, block 3 from MB-ACR, and block 2 from MT-ACR. Substitution to any one of these blocks tends to increase alcohol

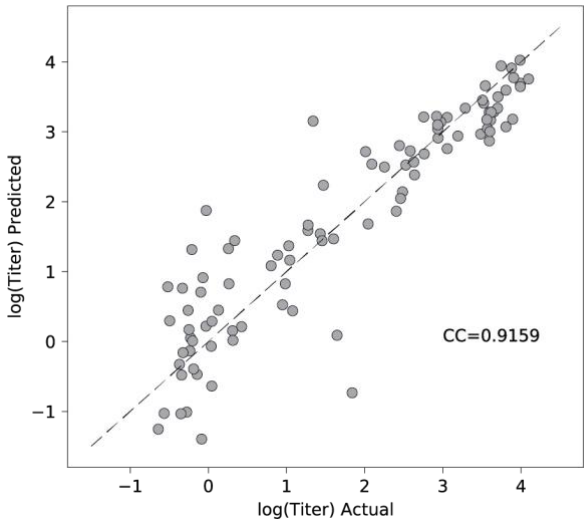

Figure 3.8: Cross-validated Gaussian process regression model used to study chimera landscape and determine contributions of blocks.

titers by over 70%. Block 8 from MB-ACR also strongly tends to increase the titers. The sequence blocks with the most negative contribution were blocks 3 and 7 from MT-ACR. Overall, most blocks from MT-ACR were deleterious for alcohol production.

We mapped the block effects onto MA-ACR's homology model to relate their contributions to structure and mechanism (Figure 3.9b). Block 2 likely forms extensive interactions with the enzyme's NADPH cofactor, and MT-ACR is the best parent at this position. While there are many amino acid differences in this block, it's notable that MT-ACR has a different NADPH binding motif than the other two parents (GGSSGIG vs.

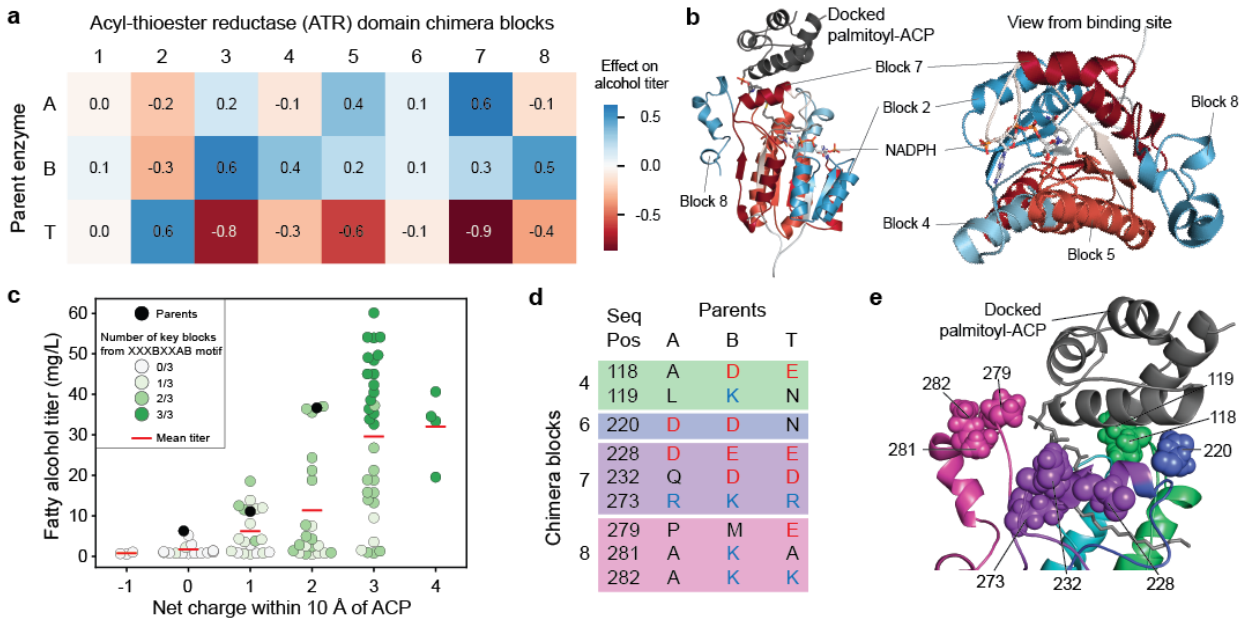

Figure 3.9: Statistical analysis of the fatty alcohol production landscape. (a) Contributions of each sequence block to the alcohol production of chimeras. (b) Mapping of the contributions to the structural model of MA-ACR. Blocks with strong effects (either positive or negative) line up with key structural features such as the NADPH binding domain, the active site and the ACP binding site. (c) Correlation between net charge near the binding site and fatty alcohol titer of chimeras. Chimeras with higher net charges tend to produce higher titers. Additionally, three key blocks were found to correlate with both activity and with charge. Combining all three of these blocks results in highly active enzymes. The statistics for the finalized dataset (mean, standard deviation and number of replicates) are available in Supplementary Table 3.5. (d) Positions in the parent sequence alignment that contain non-conserved charged residues within 10 Å of the putative ACP binding site (positively charged residues are shown as blue text, negatively charged residues are shown as red text). (e) Locations of key charged residues in the structural model of MA-ACR docked with palmitoyl-ACP. Optimal combinations of blocks could produce more favorable interactions with the ACP.

GATSGIG). MT-ACR's motif may provide more efficient NADPH utilization *in vivo*. Blocks 4-8 make up the binding pocket for the acyl-thioesters. Block 5 contains three of the catalytic residues (a Y, S and K), and block 6, whose sequence is highly conserved, appears to be involved in NADPH binding. Blocks 7 and 8 appear to contain surface residues; positively charged residues in these blocks are likely involved in docking the negatively charged acyl-ACP[29].

We hypothesized the net charge of the enzyme's substrate binding pocket may influence activity because the ACP substrate contains many negatively charged residues.

To examine the enzyme's charge distribution near the substrate binding site, we computationally docked ACP (from PDB entry 6DFL [https://www.rcsb.org/structure/6DFL]) to our homology model of MA-ACR using RosettaDock[30]. We then identified all interface positions within a 10 Å radius of the docked ACP and calculated the net charge of each chimera's interface residues. We found the net charge of an enzyme's substrate binding interface was positively correlated with the total fatty alcohol titer (Figure 3.9c). A chimera's substrate interface charge is dictated by nine sequence positions that are near the ACP substrate and that contain charged residues in at least one parent (Figure 3.9d,e). The charges at these sequence positions can largely explain the preferred blocks from Figure 3.9a.

3.4 Discussion

Engineering fatty acyl reductases (FARs) to have improved activity on acyl-ACP substrates could open routes to *in vivo* production of fatty alcohols, and other valuable bioproducts such as waxes and alkanes. In this work, we engineered enzymes with improved activity on acyl-ACP substrates. Our approach leveraged gene shuffling to broadly sample sequence space and ML-driven protein engineering to rapidly and efficiently identify optimized sequences. Our top identified enzyme, ATR-83, displayed two-fold higher *in vivo* fatty alcohol titers than the best natural sequence MB-ACR and nearly five-fold higher titers than MA-ACR. These increases in fatty alcohol titer are a result of ATR-83's enhanced turnover number on ACP substrates. The chimeric enzymes discovered in this work have potential to improve the efficiency of alcohol production from acyl-ACPs *in vivo*.

Shuffling the AHR and ATR domains between the three natural sequences generated chimeric enzymes that produce a broad range of fatty alcohol titers. From these results, it appears the ATR domain from MB-ACR has the highest activity on ACP substrates and the AHR domain from MA-ACR has the highest activity on the intermediate aldehyde substrate. Rather than directly affecting the catalytic rate, it's also possible that these domains could be enhancing activity through inter-domain interactions, especially since MA-ACR has been shown to be tetrameric[4].

Machine learning is rapidly advancing the fields of directed evolution and protein engineering[15–17,31]. Though some ML-based strategies (especially those involving deep learning or neural nets) require massive amounts of training data, active-learning approaches (such as UCB optimization) can be used to simultaneously explore the sequence-function landscape and identify improved sequences from relatively few data points. The reduced need for data enables protein engineering workflows that don't depend on high-throughput techniques, and thus overcomes major limitations of directed evolution approaches. Our design-test-learn cycle closely resembles the UCB optimization process previously used to engineer thermostable chimeric cytochrome P450s[14]. However, a key difference in this work was the introduction of an active/inactive binary classifier to filter out potential inactive sequences that provide little information regarding enzyme activity. Incorporating this classifier led to improved predictions by the GP regression model, especially in early UCB rounds when the number of active sequences was small (only 12 sequences were active from the first three rounds).

In the early rounds of our UCB sequence optimization, we found it was helpful to restrict the number of block exchanges from the parent sequences in order to bias the

search towards functional sequences. Sampling further away almost always resulted in non-functional sequences that provided little information about the fatty alcohol production landscape. We learned this trick during the course of the sequence optimization, which certainly limited the efficiency of our method. Future improvements to UCB algorithm could include an informative prior for the active/inactive binary classifier that encodes a preference to sample near the parent sequences when limited functional data is available.

In principle, our protein engineering framework is applicable whenever an underlying fitness landscape can be inferred via machine learning. There have been multiple previous studies demonstrating the effectiveness of machine learning to navigate the sequence-function landscape. A notable example used a similar UCB method to optimize cytochrome P450 thermostability[14]. A lower confidence bound (LCB) algorithm was used to predict chimeric channelrhodopsins that localize to the plasma membrane of mammalian cells, and UCB optimization was then used to identify chimeras with high localization[18]. GP classification and regression models were further used to engineer highly light sensitive channelrhodopsins for optogenetics[31]. Iterative searches through protein sequence-function landscapes such as UCB optimization and LCB minimization reduce dependency on large datasets, and enable engineering of more difficult protein targets.

ATR-83 produced 50% more fatty alcohols than parent B (A-BBBBBBBB) and 450% more than MA-ACR. It is difficult to interpret these *in vivo* results because intracellular acyl-ACP pools exist as a broad mixture from C4-C18, and each enzyme may have its own substrate preferences. We performed further kinetic characterization on the

enzymes and found ATR-83's increased *in vivo* alcohol production is the result of enhanced turnover number (k_{cat}) on ACP substrates, rather than enzyme expression or K_M effects. Interestingly, ATR-83 displays lower activity on acyl-CoA substrates than parent B and MA-ACR. Since both acyl-ACP and acyl-CoA substrates have the same thioester bond that is being reduced, one might expect substrate specificity to manifest as differences in K_m between the enzymes. However, we observed enzymes' k_{cat} to be the major determinant of substrate specificity. One possible explanation for the observed behavior could be that ACP is interacting with the enzyme surface to allosterically enhance the catalytic rate. Similar allosteric modulation by ACPs has been observed in the LovD enzyme[32].

We found a positive correlation between an enzyme's net charge near the putative substrate binding site and its activity on acyl-ACPs *in vivo*. This relationship may be expected because positive charges on the enzyme surface could enhance electrostatic interactions with the negatively charged ACP substrate. The chimeric enzymes' substrate interface charge is largely dictated by blocks 4, 7, and 8. Sequences with B at block 4, A at block 7, and B at block 8 (i.e. XXXBXXAB) can increase the net interface charge of a chimera by up to +4. The average alcohol titer of chimeras containing these three blocks is 42 mg/L, compared to an average of 8 mg/L for sequences without that combination. These results suggest future enzyme engineering directions to supercharge the substrate interface with positively charged residues to further enhance electrostatic interactions with ACP. A similar approach has been applied to acyl-ACP thioesterases, leading to improved enzyme activity[29].

While we demonstrated that our engineered enzymes have improved activity on palmitoyl-ACP both *in vivo* and *in vitro*, the activity of the enzymes on shorter and medium chain substrates is less clear. Production of medium chain fatty alcohols, such as octanol, remains a prime target for metabolic engineering, since medium chain fatty alcohols are more valuable than long chain fatty alcohols[33]. In order to explore the *in vivo* activity of these engineered enzymes on shorter chain acyl-ACPs, new methods would be needed to alter the acyl-ACP distribution in the cells without significantly disrupting pathways involving production of lipids for the cell membranes. Alternatively, pathways that utilize acyl-CoA pools show promise for making medium length alcohols selectively[7,9]. While our active-learning strategy focused on acyl-ACP activity, it could also be used to enhance activity on medium chain acyl-CoAs.

While our engineered ATRs were able to significantly boost fatty alcohol production from acyl-ACP substrates, the titers we achieved are still far below those from pathways that rely on acyl-CoA intermediates, such as the implementation of reverse beta oxidation in Mehrer et al. (1.8 g/L)[8] and the utilization of a thioesterase/acyl-coA ligase pair in Hernández-Lozada et al. (1.3 g/L)[9]. Our lower titers are expected since the acyl-ACP pool is considerably smaller than the acyl-CoA pools that can be achieved in these and similar pathways. Additionally, these previous works involved extensive strain optimization to boost acyl-CoA pools, while our current enzyme engineering results were achieved in an unmodified host strain. Importantly, the acyl-ACP route to produce fatty alcohols is more direct and has a lower energetic cost than pathways utilizing acyl-CoA intermediates. Future work could focus on strain engineering efforts to upregulate fatty acid biosynthesis

by modifying FadR expression or by relieving the pathway's feedback inhibition by longer chain acyl-ACPs.

Our ability to engineer microbes to produce high-value chemicals is often limited by the availability of enzymes to catalyze key chemical reactions. We have presented an enzyme engineering framework that leverages ML-based sequence-function models with iterative experimentation to rapidly identify improved enzymes. This approach can be generally applied to enzymes that lack a high-throughput functional assay or structural information, and therefore are challenging to engineer using traditional directed evolution and rational methods. Future advances in enzyme engineering will open routes to produce valuable chemicals from low-cost and renewable feedstocks.

3.5 Methods

3.5.1 Chemicals, reagents, and media

E. coli RL08ara[21] and CM24[8] assay media used for this study are the same composition as Miller LB, except with 10 g/L peptone instead of 10 g/L tryptone. CM24 media was supplemented with 1% w/v glucose, and sterile filtered using a 2 µM filter. *E. coli* RL08ara assay medium was sterilized by autoclaving. Both media were adjusted to a pH of 7.0 prior to sterilization.

Individual fatty alcohol standards were prepared at a concentration of 100 mg/mL by dissolving alcohols ranging from C3 to C17 in 200 proof ethanol. Then, alcohols were mixed to make 10 mg/mL standards of even-chain alcohols (C4, C6, C8, C10, C12, C14 and C16) and odd-chain alcohols (C3, C5, C7, C9, C11, C13, C15, C17). All unique biological materials are available upon request.

3.5.2 Measuring in vivo fatty alcohol titers

We measured *in vivo* alcohol titers produced by each enzyme variant using gas chromatography (GC). Overnight cultures started in LB + Kanamycin from individual colonies from the transformation were grown for 16-20 hours and diluted into a 50 mL culture of *E. coli* RL08ara Assay Medium + Kanamycin in a 250 mL baffled shake flask such that the final OD was about 0.01. The media had a 20% (10 mL) dodecane overlay, and we supplemented the media with 1 mL of 50% v/v glycerol. The cultures grew at 37 ˚C for 45 minutes at 250 rpm, and then we induced protein expression by adding 500 µL of 100 mM IPTG (final concentration 100 µM IPTG). As a control, each batch also included blank cultures that were prepared by mixing media, dodecane, glycerol and antibiotic in the same amounts as the expression cultures, but without any cells added. The expression cultures incubated for 18 hours at 30 ˚C after induction.

Afterwards, we cooled the expression cultures on ice to prevent evaporation. Then, we added 150 µL of 10 mg/mL odd-chain internal standard mixture to each culture flask and mixed them vigorously to make an emulsion. Immediately after mixing, we transferred 5 mL of the emulsion to a glass centrifuge tube pre-loaded with 1 mL of n-hexanes. We vortexed the tubes for 20 s, shook for 20 s, and vortexed for another 20 seconds. Then, we centrifuged the samples for about 10 minutes until the organic layer and aqueous layers separated and extracted about 900 µL of the organic layer to load into a GC vial for analysis on GC-FID.

We analyzed all GC samples using a Shimadzu Model 2010 GC-FID system with an AOC-20i autosampler and a 60 m 0.53 mm ID Stabilwax column (Restek 10658). The

oven temperature program used to analyze samples from RL08ara and CM24 samples was based on Mehrer et al.[8] and is as follows: 45 ˚C hold for 10 minutes, ramp to 250 ˚C at 12 ˚C/minute, hold at 250 ˚C for 10 minutes. In some individual experiments we shortened the hold time. Each run included standards of the odd-chain internal standard mixture and even-chain standard mixture to control for any changes in the retention times of the analytes. We estimated the concentrations of even-chain fatty alcohols by averaging the areas (A_{i-1} and A_{i+1}) and concentrations (C_{i-1} and C_{i+1}) of the odd-chain internal standards that bracketed the particular even-chain analyte. We used the resulting response factor to convert the area of the even-chain species (A_i) to the original media concentration (C_i) per Equation 3.1:

$$C_i = A_i * \frac{avg(C_{i-1}, C_{i+1})}{avg(A_{i-1}, A_{i+1})} \quad (i = 2,4,6,8,10,12,14,16) \tag{3.1}$$

3.5.3 Aerobic alcohol production in BL21(DE3)

We cloned the initial seed sample ACR chimeras into the pET28 backbone and transformed into BL21(DE3). Cultures were started in LB + Kanamycin from individual colonies from the transformation and grown overnight for 16-20 hours. We diluted the cultures 100-fold into 5 mL cultures of LB + Kanamycin in culture tubes. We grew the cultures for 2.5-3 hours, measured the ODs, and then induced with 5 µL of 100 mM IPTG and incubated for 24 hours at 20 ˚C with shaking at 250 rpm.

Following protein expression, we incubated the cultures ice for 1.5-2.5 hours. Nonanol (C9) and heptadecanol (C17) were used as internal standards; a solution that was 5 µM nonanol and 5 µM heptadecanol in hexanes was prepared and added (1 mL)

to each 5 mL expression culture. We then vortexed and spun down the sample in a centrifuge (1000x G for 10 minutes) to separate the phases. 900 µL of the organic layer was extracted for analysis on GC-FID. Titers of fatty alcohols were determined using an external standard curve with standards of each of the even chain fatty alcohols in hexanes and dividing by the extraction ratio (5) to convert from the concentration in the organic phase to the original concentration in the media.

3.5.4 Anaerobic alcohol production in CM24

ACR chimeras were cloned into the pBTRCK plasmid backbone and transformed into CM24 along with seFadBA (g130, pACYC-seFadBA) and tdTER (g131, pTRC99A-tdTER-fdh)[8]. We started overnight cultures from individual colonies in LB + Kanamycin + Carbenicillin + Chloramphenicol. The following day, after 16-20 hours, 600 µL of overnight cultures were diluted in 30 mL of CM24 Assay Medium + Kanamycin + Carbenicillin + Chloramphenicol with a 20 % (6 mL) dodecane overlay in a 50 mL serum vial, which was sealed. We grew the cultures for 2 hours at 30 ˚C, and then induced by injecting 300 µL of 100 mM IPTG (for a final IPTG concentration of approximately 100 µM) through the septum with a needle. Cultures were then incubated at 30 ˚C for 48 hours.

Following expression, we cooled the cultures on ice and added 180 µL of an internal standard mixture (the same fatty alcohol mixture used for quantitation of alcohols in RL08ara). We mixed the samples thoroughly and extracted 5 mL of the emulsion with 1 mL of hexane per the same protocol as RL08ara above.

3.5.5 Structural modeling and SCHEMA library design

We utilized the MODELLER[34] homology modeling software to build 100 models of each of the acyl-thioester reductase domains of MA-ACR, MB-ACR, and MT-ACR using the following PDB entries as templates: 3M1A-A [https://www.rcsb.org/structure/3M1A], 3RKR-A [https://www.rcsb.org/structure/3RKR], 3RIH-A [https://www.rcsb.org/structure/3RIH], 3AFM-B [https://www.rcsb.org/structure/3AFM], 3AFN-B [https://www.rcsb.org/structure/3AFN], and 4BMV-A [https://www.rcsb.org/structure/4BMV]. We built a contact map by determining which pairwise amino acid contacts (defined as two amino acids within a 4.5 Å radius based on any atoms in the amino acids) were present in each model, and weighted each contact by the percentage of models in which the contact was present.

We determined the crossover between the aldehyde-reductase domain and the acyl-thioester reductase (ATR) domain by aligning the sequences of MA-ACR, MB-ACR, and MT-ACR and selecting a crossover point at the conserved LDPDL, approximately 350-360 residues from the N-termini. Then, we used SCHEMA-RASPP to determine 7 additional crossover locations within the ATR domain that were compatible with Golden Gate assembly.

3.5.6 Gene assembly and strain construction

All ATR enzymes tested were cloned into the pBTRCK plasmid backbone and transformed into *E. coli* RL08ara[21]. We obtained the three natural parent sequences from prior studies[8,9]. We amplified the AHR and ATR domains of each of the natural sequences, as well as the plasmid backbone, by PCR using primers (Supplementary

Table 3.7) that contained Golden Gate overhangs. We used Phusion Hot Start Flex 2X Master-Mix (NEB) for all PCR reactions. Then, we used Golden Gate assembly to combine the pieces and synthesize the domain shuffled variants. Golden Gate reactions were carried out either using commercial Golden Gate assembly mix (NEB), or an in-house mixture of the components from NEB (T4 DNA ligase buffer, BsaI HF v2 and T4 DNA ligase).

We designed plasmids containing each of the 24 blocks determined by RASPP such that each block was flanked by BsaI restriction sites. The plasmids were synthesized by TWIST Biosciences. The blocks (including the BsaI site) were amplified by PCR and cloned into a backbone vector harboring the AHR domain of MA-ACR by Golden Gate assembly. For sequences that we studied *in vitro*, we amplified the whole FAR sequence and used Golden Gate assembly to add the insert into a pET 28 backbone.

3.5.7 Greedy algorithm to design an informative seed sample

We sought to identify the set of 20 chimera sequences that is maximally informative of the full chimera landscape. We quantify ‘informativeness’ as the Gaussian mutual information $I(S;L)$ between the chosen sequences S and the full landscape L. This mutual information simplifies to the Gaussian entropy $H(S)$ because S is a subset of L. Entropy is a submodular set function and can therefore be efficiently optimized using a greedy algorithm.

We started with our three parent sequences and scanned over all possible chimera sequences s_i to determine which resulted in the largest Gaussian entropy $H(S \cup \{s_i\})$. This

top sequence was added to the chosen set of sequences *S* and the greedy sequence selection process was repeated until 20 sequences were chosen.

3.5.8 Sequence-function machine learning

We modeled the sequence-function landscape using a combination of a Gaussian Naïve Bayes (GNB) classifier to distinguish inactive versus active sequences and Gaussian process (GP) regression to model a sequence's fatty alcohol titer.

The active/inactive classifier was trained on chimera sequence-function data using scikit-learn's Naïve Bayes classifier. We categorized sequences as active if their alcohol titer was above a certain threshold; otherwise, they were considered inactive. The amino acid sequences for each tested chimera were one-hot encoded and used as inputs for the classifier. The resulting model provides a prediction of the probability that a sequence is an active enzyme.

We also trained a GP regression model on the active sequences' fatty alcohol titers. Our GP regression model used a homogeneous linear kernel to define the covariance between pairs of sequences

$$k_{i,j} = \sigma^2 x_i \cdot x_j \tag{3.2}$$

where σ^2 is a tunable variance hyperparameter, and x_i and x_j are the encodings for sequences *i* and *j*, respectively. The Hamming kernel one-hot encoded each amino acid option at each sequence position, while our structure kernel one-hot encoded amino acid pairs at each residue-residue pair that was contacting in the three-dimensional structure.

We calculated the GP's posterior mean and variance following Algorithm 2.1 in Rasmussen & Williams[35] (section 3.5.17).

We used leave-one-out cross-validation to scan variance (σ^2) hyperparameter values ranging from 10^{-6} to 10^5 and selected values that maximized the correlation coefficient and minimized the mean squared error (Figure 3.10). When these two objectives could not be realized simultaneously, we chose σ^2 values that balanced them. We then used the chosen σ^2 value to fit the GP model on all the data and predict the activities of all untested sequences that the GNB classifier labeled as active.

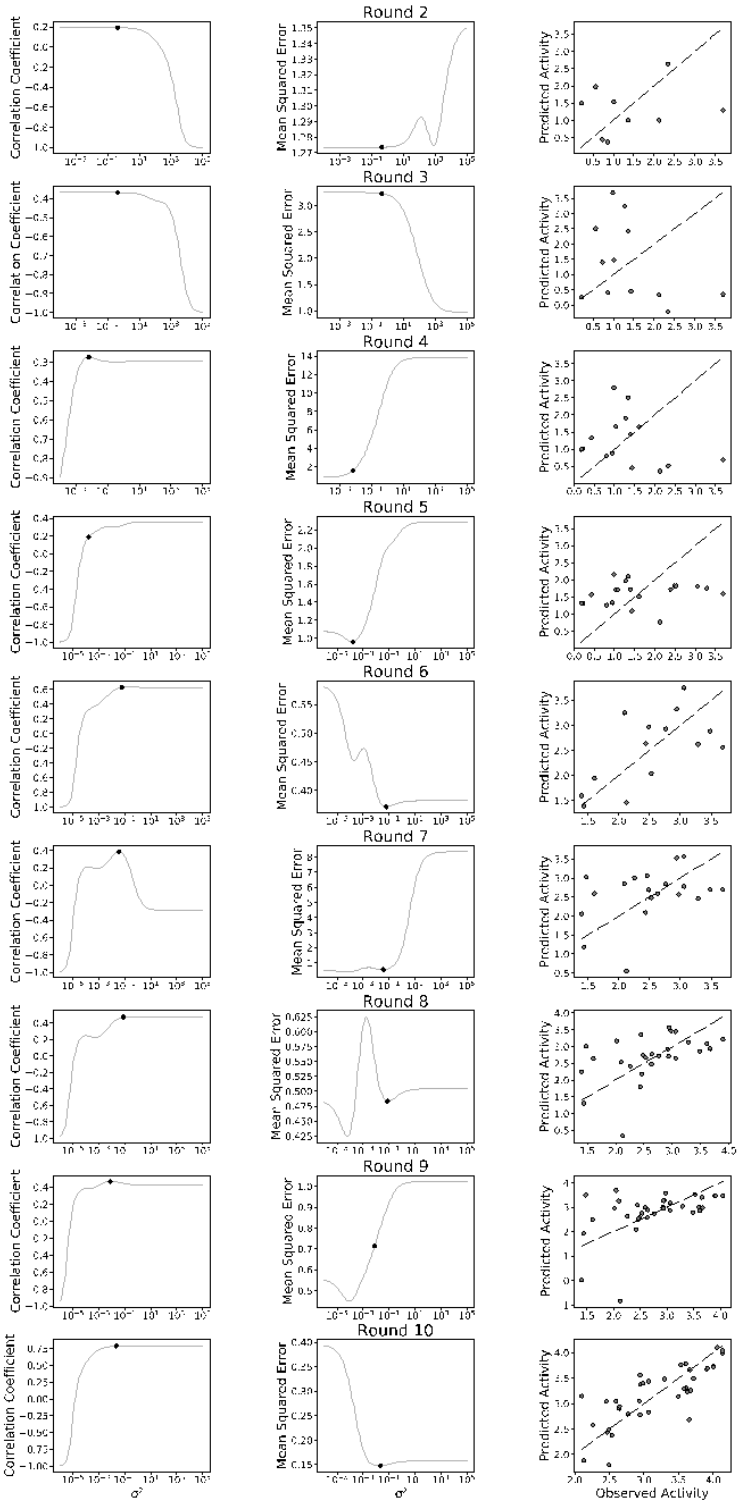

Figure 3.10: Cross validation scans by round during sequence optimization. The leftmost column shows the correlation coefficients as a function of σ^2, the middle shows the mean squared error, and the right shows the final cross validated result. The selected values of σ^2 are shown as black dots in the leftmost and middle columns.

3.5.9 Upper-confidence bound optimization

We utilized UCB optimization to select informative sequences to build and test for the next round. For UCB rounds 2-10, we trained the active/inactive GNB classifier and the alcohol titer GP regression model on all prior data. We then applied the GNB and GP models to make functional predictions over all untested chimeras. We choose a panel of sequences to test using a 'batch mode' UCB selection strategy[36], while excluding any sequences that were predicted to be inactive from the GNB classifier. We first chose the sequence that maximized the GP upper confidence bound (mean + one standard deviation). This is the UCB optimal sequence. We then retrained the GP model with the assumption that the UCB optimal sequence's true titer was equal to its predicted titer. We then recalculated the UCBs and chose the new UCB optimal sequence. This process enables selection of multiple UCB optimal sequences per round, and it was repeated until 10-12 sequences were chosen per batch. The details of each round of UCB optimization can be found in Supplementary Table 3.4.

The first UCB round was performed slightly differently than the others because we were still refining our method. For the first UCB round, we trained GP regression models on alcohol titers from both BL21(DE3) and CM24 strains. We chose sequences that maximized the sum of the BL21(DE3) and CM24 UCB scores and selected a panel of ten chimeras using the batch mode UCB approach described above.

3.5.10 Measuring in vivo enzyme expression levels using SDS-PAGE

To verify that increases in fatty alcohol titers were due to enzyme activity, we performed additional characterization of the protein expression levels for the parents and

selected chimeras. To estimate the expression level of the ATR enzyme, we performed additional replicates using the same expression conditions as were used during UCB optimization. Then, after extracting the fatty alcohols, we suspended the remaining 5 mL pellet in 1 mL of media. We normalized the ODs of the suspensions to an OD of 10 and pelleted and froze 500 µL of the OD 10 culture. We later thawed the frozen pellets and lysed them using 250 µL lysis buffer (3872 µL 100 mM Tris pH 7.4, 120 µL Bugbuster, 4 µL lysozyme and 4 µL DNAse I).

We prepared a standard curve using dilutions of purified MA-ACR. We added 3 µL of each MA-ACR dilution to 12 µL of SDS master mix (which consisted of 5 parts 2X SDS mix and 1 part 1 M DTT). and mixed them in a 1:1 ratio (volume:volume) with empty vector lysate. The other lysates were mixed with 2X SDS buffer and 3 µL 100 mM Tris pH 7.4 (to ensure equal volumes of lysate between the standards and the samples). We heat denatured the lysates (at 85 ˚C for 2-5 minutes) and analyzed them by SDS-PAGE.

We used FIJI, an image analysis software[37], to estimate the intensities of the ATR band in the MA-ACR standards and generate a standard curve (Supplementary Figure 3). We made new standard curves for each replicate to reduce gel to gel variability, and only compared samples to standards on the same page gel. Expression levels are reported as µg/mL of ATR (at an OD of 20).

3.5.11 Biosynthesis of fatty acyl-ACP substrates

We synthesized the acyl-ACP substrates by functionalizing purified *E. coli* ACP with a 4'-phosphopantetheine arm by the acyl-ACP synthetase from *Vibrio harveyi*[38], and

then attaching the acyl-chain to the thiol end of the arm using a phosphopantetheinyl transferase (SfP) from *Bacillus subtilis*.

3.5.12 Expression of V. harveyi AasS, B. subtilis SfP and E. coli ACP

The enzymes needed to functionalize palmitoyl-ACP were expressed using the method in Hernández-Lozada et al. with some minor modifications[39]. The cells were grown for 2 hours at 37 ˚C (200 rpm) and then induced with 1 mM IPTG (final concentration) without cooling the cultures as was done in Hernández-Lozada et al. AasS and SfP were expressed overnight at 18 ˚C for 18-24 hours, and ACP was expressed at 20 ˚C overnight (18-24 hours) and harvested by centrifugation. We also purified the proteins using the method from Hernández-Lozada et al., however rather than using dialysis, we used Amicon filter columns to carry out buffer exchange. The final concentrations of the proteins were determined using Bradford assays.

3.5.13 Functionalization of E. coli ACP

To cleave the His-tag from the Apo ACP, we added 700 uL of 2.1 mg/mL TEV protease to the 4 mL ACP solution. The reaction incubated overnight (16-20 hours) at 20 ˚C shaking at a speed of 250 rpm. At the conclusion of the digestion, we stored the mixture in 50% glycerol at -80 ˚C. Later, to purify the cleaved Apo ACP, we thawed the digestion and ran it over parallel gravity columns packed with Nickel Sepharose Fast Flow resin. We pooled the flow-through and buffer exchanged with 50 mM Na_2HPO_4 pH 8 + 10% glycerol using an Amicon filter unit (MWCO 3000 kDa). The concentration of the cleaved Apo-ACP was determined by a Bradford assay.

The conditions for the reactions to generate Holo-ACP were: 500 µM Apo ACP, 5 µM SfP, 5 mM Coenzyme A, and 10 mM $MgCl_2$ in 100 mM Na_2HPO_4 pH 8. The reactions took place in 500 uL aliquots in 1.5 mL Eppendorf tubes and shaken in a beaker at 37 ˚C for 1 hour.

We dissolved sodium palmitate in water heated to 65 ˚C to a concentration of 100 mM. After the holo-ACP reactions were finished, we added palmitate, ATP, and AasS to the reaction mixture, (along with enough buffer to double the reaction volume), to give final concentrations of 5 mM palmitate, 5 µM AasS and 10 mM ATP. The reactions incubated overnight (16-20 hours) at 37 ˚C. Then, we pooled the reactions, purified the palmitoyl-ACP by running the mixture through a gravity column packed with Nickel Sepharose Fast Flow Resin. We buffer exchanged the purified palmitoyl-ACP into 100 mM Na_2HPO_4 + 10% glycerol pH 8.

3.5.14 Purification of ATRs

We expressed parental ATRs (A-AAAAAAAA, A-BBBBBBBB, and A-TTTTTTTT) and purified them per the same method as *E. coli* ACP, except for the buffer exchange step. We buffer exchanged them into 20 mM Tris, 50 mM NaCl pH 7 using an Amicon filter unit (30,000 kDa MWCO). Then, we added glycerol to the proteins (about 15 % v/v for parents 1-3). We expressed ATR-83 at 30 ˚C rather than 20 ˚C but purified it in the same manner, though we added more glycerol to the purified ATR-83 (final concentration ~50 % v/v glycerol). We determined the concentration of the enzymes by Bradford assays.

3.5.15 In vitro enzyme kinetics on palmitoyl-ACP and palmitoyl-CoA

We determined the activity of the above ATRs in a 96 well plate based assay using 5'5 Dithiobis(2-nitrobenzoic acid) or DTNB to monitor the progress of the conversion of palmitoyl-ACP to hexadecanol and free holo-ACP (measuring the absorbance at a wavelength of 412 nm). We tested palmitoyl-ACP concentrations up to 40 µM (as this concentration should be within the physiological range within cells)[40]. Reactions contained 1 µM of the respective ATR and 200 µM NADPH in 20 mM Tris + 50 mM NaCl pH 7 and the total reaction volume was 100 µL. The concentration of DTNB was 250-252 µM (the difference is due to slightly different preparations of a NADPH/DTNB master mixes on different dates).

To gauge activity of the ATRs on CoAs *in vitro*, we carried out reactions using palmitoyl-CoA as a substrate. The *in vitro* assay used to determine CoA activity was identical to that used for ACP activity above.

3.5.16 Computational docking and analysis of interfacial charge

We used the RosettaDock[30,41] application to perform local docking simulations to dock a structure of palmitoyl-ACP (from PDB entry 6DFL) to MA-ACR. We did not include the acyl-chain in the docking simulations. We ran 1000 docking simulations and selected a model based on minimizing the total energy and the interface score. Then, using PyMOL, we determined which residues in the model of MA-ACR were within a 10 Å radius of the ACP molecule. The number of charged residues within that radius was then determined, and the net interface charge was defined as the number of positive residues minus the number of negative residues.

3.5.17 Gaussian process regression and UCB calculation

During the training step of each round, we preprocessed the data and mean centered the encoding of the training set ($\mathbf{x}_{trn}$) the test set encoding ($\mathbf{x}_{tst}$), and the activity values for the training set ($\mathbf{y}_{trn}$). Then, to train the model, we follow Algorithm 2.1 presented in Rasmussen and Williams[35]. First, we evaluate the kernel function between all pairs of training sequences (K) and between the training sequences and test sequences ($\mathbf{k}_*$):

$$K = \sigma^2 \mathbf{x}_{trn} \cdot \mathbf{x}_{trn}^T \tag{3.3}$$

$$\mathbf{k}_* = \sigma^2 \mathbf{x}_{trn} \cdot \mathbf{x}_{tst}^T \tag{3.4}$$

where σ^2 is the hyperparameter used to regularize the model. Then, we perform a Cholesky decomposition:

$$L = \text{Cholesky}(K + \sigma_n^2 I) \tag{3.5}$$

where σ_n^2 is the noise level (and in this case was set equal to 1) and I is the identity matrix. L is a lower triangular matrix that can be used to efficiently solve the following:

$$\boldsymbol{\alpha} = L^T(L \backslash \mathbf{y}_{trn}) \tag{3.6}$$

The predicted mean ($\mathbf{y}^*$) can then be calculated as:

$$\mathbf{y}^* = \mathbf{k}_*^T \boldsymbol{\alpha} + Y_{mean} \tag{3.7}$$

where Y_{mean} is the mean activity value of the training sequences. The confidence intervals (CI) are then determined by calculating the variance ($\mathbf{v}^*$**)** as follows:

$$\mathbf{v} = L \backslash \mathbf{k}_* \tag{3.8}$$

$$\mathbf{v}^* = diag(\sigma^2 \mathbf{x}_{tst} \cdot \mathbf{x}_{tst}^T) - \mathbf{v}^T \mathbf{v} \tag{3.9}$$

$$CI = \sqrt{\mathbf{v}^*} \tag{3.10}$$

Finally, the UCB can be calculated by summing $\mathbf{y}^*$ and CI for each sequence.

3.6 Data availability

The data supporting the findings of this work are available within the chapter and its Supplementary Information files. A reporting summary for this article is available as a Supplementary Information file. Structural data for the following PDB IDs from the protein databank were utilized: 6DFL [https://www.rcsb.org/structure/6DFL], 3M1A [https://www.rcsb.org/structure/3M1A], 3RKR [https://www.rcsb.org/structure/3RKR], 3RIH [https://www.rcsb.org/structure/3RIH], 3AFM [https://www.rcsb.org/structure/3AFM], 3AFN [https://www.rcsb.org/structure/3AFN], 4BMV [https://www.rcsb.org/structure/4BMV] for structural models. Additionally, all enzyme sequence function data collected in this work is available at the ProtaBank protein engineering database under ID nu9KXbjT4 [https://www.protabank.org/study_analysis/nu9KXbjT4/].

3.7 Code availability

All code for machine learning, UCB protein sequence optimization, and data analysis is available at the GitHub repository: https://github.com/RomeroLab/ML-Guided-Acyl-ACP-Reductase-Engineering (archived version: https://doi.org/10.5281/zenodo.5259326).

3.8 Acknowledgements

This work was funded by the National Institutes of Health (R35GM119854) and the National Science Foundation (CBET-1703504). J.C.G. is the recipient of a National Institutes of Health Biotechnology Training Program Fellowship (NGIMS T32GM008349). The authors would also like to acknowledge Dr. Néstor Hernández Lozada, Dr. Chris Mehrer, and Dr. Mark Politz for helpful discussions regarding assay development, and Bennett Bremer for helpful discussions regarding coding and algorithms.

3.9 Supplementary tables

Supplementary Table 3.1. Strain list.

Strain	Description	Source/Reference
Escherichia coli DH5α	n/a	Lucigen
Escherichia coli BL21 DE3	n/a	Lucigen
Escherichia coli 10 G Supreme	n/a	Lucigen
E. coli RL08ara	*E. coli* K-12 MG1655 ΔfadD ΔaraBAD ΔaraFGH Φ(ΔaraEp P_{CP18}-araE)	Lennen *et al*.[21]
E. coli CM24	*E. coli* LS5218 ΔfadE ΔatoC ΔldhA ΔackApta ΔadhE ΔpoxB ΔfrdABCD ΔydiO ΔfadBA ΔfadIJ ΔfadD	Mehrer *et al*.[3]

Supplementary Table 3.2. Key plasmids.

Name	Description	Source
pET 28 MA-ACR	ACR from *Marinobacter aqueolei* VT8 fused to maltose binding protein (MBP) on pET 28 backbone, T7 promoter KmR	Rung Yi Lai, (currently Suranaree University of Technology), while a post-doc at UW-Madison
pBTRKtrc	Ptrc promoter, pBBR1 origin, KanR	Youngquist *et al*.7
pBTRCK MA-ACR	ACR from *Marinobacter aqueolei* VT8 fused to maltose binding protein (MBP) on pBTRCKtrc backbone	Youngquist *et al*.7
pBTRCK MB-ACR	ACR from *Marinobacter BSs20148* fused to maltose binding protein (MBP) on pBTRCKtrc backbone	Mehrer *et al*.[8]
pBTRCK MT-ACR	ACR from *Methylibium Sp. T29* fused to maltose binding protein (MBP) on pBTRCKtrc backbone	Mehrer *et al*.[8]
pACYC-seFadBA	FadBA from *Salmonella enterica*, pACYC origin, Trc promoter, CmR	Mehrer *et al*.[8]
pTRC99A-vhTER-fdh	Trans-enyol-CoA reductase (TER) from *Vibrio harveyi* and formate dehydrogenase from *Candida boindinii*, pBR322 origin, Trc Promoter, AmpR	Mehrer *et al*.[8]
pET 28 Ec-ACP	Apo-Acyl Carrier Protein (ACP) from *E. coli*	Hernández-Lozada *et al*.[39]
pET 28 Vh-AasS	AasS from *V. harveyi*	Hernández-Lozada *et al*.[39]
pET 28 Bs-Sfp	Sfp from Bascillus subtilis	Hernández-Lozada *et al*.[39]
pLacIRARE rTEV (pQE60)	TEV Protease	Hernández-Lozada *et al*.[39] (Originally from Hazel Holden Lab)

Supplementary Table 3.3. Protein structure templates used for homology models.

Template (PDB)	**Protein function**	**%ID to MA-ACR**
3m1a.A	Short-chain dehydrogenase	40
3rkr.A	Short-chain oxidoreductase	37
3rih.A	Putative short-chain dehydrogenase or reductase	36
3afm.B	Aldose reductase	36
3afn.B	Aldose reductase	36
4bmv.A	Short-chain dehydrogenase	36

Supplementary Table 3.4. Details of UCB optimization rounds.

	Data used for model training	Model used for design	Sequence design criteria	# of sequences designed	# of sequences tested	Allowed block subs	Classifier activity threshold (mg/L)	# of sequences predicted to be active	GNB AUC	Method for σ^2 selection	σ^2	GP model correlation coefficient	Sum of squared error
Initialization	None	None	Max MI	20	20	5	N/A	N/A	N/A	N/A	N/A	N/A	N/A
UCB1*	CM24 + BL21	GPR with Hamming kernel	UCB	10	5	5	N/A	N/A	None	Inspection	0.01	Not calculated	Not calculated
UCB2**	RL08ara	NB, GPR with Hamming kernel	UCB positive	10	7	5	1.1	1268	Not calculated	Inspection	0.433	0.192	11.46
UCB3	RL08ara	NB, GPR with Hamming kernel	UCB positive	10	7	5	1.1	1414	Not calculated	Inspection	0.433	-0.368	38.77
UCB4	RL08ara	NB, GPR with Hamming kernel	UCB positive	10	5	5	1.1	1341	0.7086	Maximize correlation coefficient	0.00658	-0.275	23.91
UCB5	RL08ara	NB, GPR with structure kernel	UCB positive	10	4	2	1.1	1951	0.7472	Minimize sum of squared error	0.000167	0.189	19.2
UCB6***	RL08ara	NB, GPR with	UCB positive	10	9	2	4	715	0.7886	Minimize sum of squared error	0.0599	0.632	5.2

		structure kernel											
UCB7	RL08ara	NB, GPR with structure kernel	UCB positive	10	10	2	4	714	0.81	Maximize correlation coefficient	0.0359	0.384	11.17
UCB8	RL08ara	NB, GPR with structure kernel	UCB positive	10	7	2	4	587	0.869	Inspection	0.0774	0.472	13.06
UCB9	RL08ara	NB, GPR with structure kernel	UCB positive	10	10	5	4	614	0.8799	Maximize correlation coefficient	0.00774	0.466	25
UCB10	RL08ara	NB, GPR with structure kernel	UCB positive	12	9	5	8	323	0.9757	Minimize sum of squared error	0.0215	0.786	5.47

*Cross validation was not used for hyperparameter tuning during the first round

**Sequences in this round were non-UCB-optimal due to a coding error, but the data generated from these sequences was still valuable for training models.

***The labels for two sequences in this round were accidentally swapped during model training

Supplementary Table 3.5. Fatty alcohol titers of chimeric ATRs characterized in RL08ara both during the UCB phase and final validation.

	Block_seq	UCB titer (mg/L)	n (UCB)	Finalized titer (mg/L)	n (finalized)
ATR-01	A-ATAATTBB	1.5 ± 0.3	3	1.5 ± 0.3	3
ATR-02	A-TATTTTAB	0.8	1	0.8	1
ATR-03	A-TTTTBTBA	0.9	1	0.9	1
ATR-04	A-ATTBAATB	0.6	1	0.6	1
ATR-05	A-ABTATTTA	0.9	1	0.9	1
ATR-06	A-TAATBTTB	0.9	1	0.9	1
ATR-07	A-AATAATBT	0.8	1	0.8	1
ATR-08	A-ATTABTAT	0.8	1	0.8	1
ATR-09	A-TBATATAB	1	1	1	1
ATR-10	A-TTBAATTA	2.8 ± 0.1	3	2.8 ± 0.1	3
ATR-11	A-ATBTATTA	0.7	1	0.7	1
ATR-12	A-BAABTTAA	0.9	1	0.9	1
ATR-13	A-BTTAAATB	0.8	1	0.8	1
ATR-14	A-TBBTTABT	1	1	1	1
ATR-15	A-TBABATTT	0.7	1	0.7	1
ATR-16	A-BBATBATT	0.7	1	0.7	1
ATR-17	A-BBTTATBA	0.7	1	0.7	1
ATR-18	A-TTABTABA	2.2 ± 0.2	3	2.2 ± 0.2	3
ATR-19	A-TBTABAAA	0.8	1	0.8	1
ATR-20	A-BTBATTAT	2.6 ± 0.2	3	2.6 ± 0.2	3
ATR-21	A-AABTTAAT	0.8	1	0.8	1
ATR-22	A-BABBTATA	1	1	1	1
ATR-23	A-ATBBBAAT	3.8 ± 0.3	3	3.8 ± 0.3	3
ATR-24	A-TABATABT	0.8	1	0.8	1
ATR-25	A-BTBTTAAB	5 ± 0.9	3	5 ± 0.9	3
ATR-26	A-TAAABTBB	3.6 ± 0.6	3	3.6 ± 0.6	3
ATR-27	A-TTBTTTAA	4 ± 1	3	4 ± 1	3
ATR-28	A-TBAAATAA	2.7 ± 0.2	3	2.7 ± 0.2	3
ATR-29	A-TBTTATTB	0.5 ± 0.1	3	0.5 ± 0.1	3
ATR-30	A-TAATAATA	0.6 ± 0.1	3	0.6 ± 0.1	3
ATR-31	A-TBTTTATB	0.6 ± 0.2	3	0.6 ± 0.2	3
ATR-32	A-TBTATATA	0.7 ± 0.2	3	0.7 ± 0.2	3
ATR-33	A-BTTAATBA	0.9 ± 0.2	3	1 ± 0.2	5
ATR-34	A-BTATBTTA	4 ± 0.1	3	5.2 ± 1.6	5
ATR-35	A-ABBATABA	0.8 ± 0.1	3	1 ± 0.3	5

ATR-36	A-AABBTTBA	1 ± 0.1	3	1.3 ± 0.4	5
ATR-37	A-BABTTTBA	1.2 ± 0.2	3	1.4 ± 0.2	5
ATR-38	A-AABTTTBA	1.2 ± 0.1	3	1.4 ± 0.3	5
ATR-39	A-AABATABA	1 ± 0.2	3	1.1 ± 0.3	5
ATR-40	A-BTBBBABB	21 ± 2	3	21 ± 2	3
ATR-41	A-ATAAAAAB	27 ± 7	3	27 ± 7	3
ATR-42	A-AABABAAA	12 ± 1	3	12 ± 1	3
ATR-43	A-TTBTBTTT	2.9 ± 0.3	3	2.9 ± 0.3	3
ATR-44	A-AAABBAAA	12.5 ± 1	2	12.5 ± 1	2
ATR-45	A-AAABAAAB	33 ± 6	3	33 ± 6	3
ATR-46	A-AABAAAAB	19 ± 2	3	19 ± 2	3
ATR-47	A-BBBTBABB	8.1 ± 0.8	3	8.1 ± 0.8	3
ATR-48	A-BAAAAAAB	15.8 ± 2	3	15.8 ± 2	3
ATR-49	A-ABAAAAAT	3.6 ± 0.1	2	3.6 ± 0.1	2
ATR-50	A-BBBBBATB	1.4 ± 0.1	2	1.4 ± 0.1	2
ATR-51	A-BBBBBTAB	20 ± 3	2	20 ± 3	2
ATR-52	A-ABBAAAAA	12 ± 3	2	12 ± 3	2
ATR-53	A-ABBBBTBB	21 ± 1	2	21 ± 1	2
ATR-54	A-AABAAAAT	4.4 ± 0.1	2	4.4 ± 0.1	2
ATR-55	A-BBBBBTTB	1.3 ± 0.2	2	1.3 ± 0.2	2
ATR-56	A-ATBAAAAA	14 ± 4	2	14 ± 4	2
ATR-57	A-BBBABTBB	9.5 ± 0.6	2	9.5 ± 0.6	2
ATR-58	A-AAAABATA	0.8 ± 0.1	3	0.8 ± 0.1	3
ATR-59	A-BBBBBAAB	49.1 ± 0.9	4	49.1 ± 0.9	4
ATR-60	A-ATABAAAA	19 ± 1	2	19 ± 1	2
ATR-61	A-AABBAAAA	12 ± 2	3	12 ± 2	3
ATR-62	A-BABBBABB	37 ± 5	2	37 ± 5	2
ATR-63	A-ABBBBABB	36 ± 4	3	36 ± 4	3
ATR-64	A-AAAABAAB	13.2 ± 0.6	2	13.2 ± 0.6	2
ATR-65	A-ABAAAAAB	14 ± 2	2	14 ± 2	2
ATR-66	A-BTBBBTBB	18.8 ± 0.7	2	18.8 ± 0.7	2
ATR-67	A-BBBABABB	7 ± 5	2	7 ± 5	2
ATR-68	A-BTBBBATB	7.7 ± 0.4	2	7.7 ± 0.4	2
ATR-69	A-BTBBBAAB	54 ± 6	4	54 ± 6	4
ATR-70	A-BBBBAAAB	39 ± 4	2	39 ± 4	2
ATR-71	A-ABBBBAAB	50 ± 4	2	45 ± 14	4
ATR-72	A-BABBBAAB	34 ± 6	2	34 ± 6	2
ATR-73	A-BBABBATB	2.8 ± 0.4	2	2.8 ± 0.4	2
ATR-74	A-ATBBBABB	19 ± 2	2	19 ± 2	2

ATR-75	A-ABBBBATB	2.4 ± 0.3	2	2.4 ± 0.3	2
ATR-76	A-BABBBATB	4.3 ± 0	2	4.3 ± 0	2
ATR-77	A-AABBBABB	35.5 ± 0	2	35.5 ± 0	2
ATR-78	A-ATBBATAB	41 ± 10	2	41 ± 10	2
ATR-79	A-BTABBAAB	54 ± 1	2	54 ± 1	2
ATR-80	A-BTBBAAAB	60 ± 4	3	60 ± 4	3
ATR-81	A-ATBTBAAB	24 ± 24	4	24 ± 24	4
ATR-82	A-BTABAAAB	54 ± 3	2	42 ± 21	3
ATR-83	A-ATBBAAAB	61 ± 6	4	54 ± 11	11
ATR-84	A-AABBBAAB	36 ± 10	2	36 ± 10	2
ATR-85	A-ATABBAAB	48 ± 7	2	48 ± 7	2
ATR-86	A-ATBBBAAB	56 ± 4	2	45 ± 19	3
ATR-87	A-ATABAAAB	50 ± 12	2	50 ± 12	2
ATR-88	A-ATBBBTAB	33 ± 3	2	33 ± 3	2
ATR-89	A-AABBAAAB	40 ± 4	2	40 ± 4	2
ATR-90	A-BTBAAAAB	37 ± 2	2	37 ± 2	2
ATR-91	A-ATBABAAB	16 ± 3	2	16 ± 3	2
ATR-92	A-ABBBAAAB	35 ± 4	2	35 ± 4	2
ATR-93	A-BTBBATAB	35	1	35	1
Fusion B-A	B-AAAAAAAA	N/A	N/A	10.6 ± 3	4
Fusion B-T	B-TTTTTTTT	N/A	N/A	6 ± 2	4
Fusion T-A	T-AAAAAAAA	N/A	N/A	1 ± 0	4
Fusion T-B	T-BBBBBBBB	N/A	N/A	2 ± 0	4
MA-ACR (Parent A)	A-AAAAAAAA	12 ± 3	13	11 ± 3	26
MB-ACR	B-BBBBBBBB	23 ± 1	3	26 ± 8	7
MT-ACR*	T-tTTTTTT	5 ± 1	3	4 ± 1	7
Parent B (Fusion A-B)	A-BBBBBBBB	39 ± 5	5	37 ± 8	13
Parent T (Fusion A-T)	A-TTTTTTTT	8 ± 1	3	6 ± 2	10

The average titer is represented as the mean ± SD. The number of replicates (n) indicates the number of cultures derived from individual colonies tested. Titers reported in 'Finalized Titer' column of Supplementary Table 3.5 are the final titers averaged over all experimental replicates, including additional validation experiments. For this reason, some of the final titers reported in Supplementary Table 3.5 differ slightly from the titers shown in the Figure 3.3 of the main text, which reflect average titers based on fewer replicates during the UCB optimization process (the titers presented in Figure 3.3 are shown in the 'UCB Titer' column of Supplementary Table 3.5). Specifically, the titer of ATR-83 was originally 61 mg/L (n = 4), making it the top enzyme sequence during the

UCB optimization. This is why ATR-83 was chosen for further characterization. Additional *in vivo* experiments performed on ATR-83 caused its average to drift down to 54 mg/L (n = 11) and below the next highest enzyme's (ATR-80) titer of 60 mg/L (n = 3). We performed a Welch's T-test and found the average titers of ATR-80 and ATR-83 are not statistically significant (p=0.18). The WT MT-ACR sequence in the first block (1t) of the ATR domain differs from the sequence of the block used to make chimeric sequences (1T) containing that block by a one amino acid swap (N to H).

Supplementary Table 3.6. Amino-acid sequences of blocks and sequence elements.

	Element/ block	Amino acid sequence
MBP Tag	**Maltose binding protein (MBP) tag**	MKIEEGKLVIWINGDKGYNGLAEVGKKFEKDTGIKVTVEHPDKLEEKFPQVAA TGDGPDIIFWAHDRFGGYAQSGLLAEITPDKAFQDKLYPFTWDAVRYNGKLIA YPIAVEALSLIYNKDLLPNPPKTWEEIPALDKELKAKGKSALMFNLQEPYFTWP LIAADGGYAFKYENGKYDIKDVGVDNAGAKAGLTFLVDLIKNKHMNADTDYSI AEAAFNKGETAMTINGPWAWSNIDTSKVNYGVTVLPTFKGQPSKPFVGVLSA GINAASPNKELAKEFLENYLLTDEGLEAVNKDKPLGAVALKSYEEELVKDPRIA ATMENAQKGEIMPNIPQMSAFWYAVRTAVINAASGRQTVDEALKDAQTNSSS NNNNNNNNNNLGIEGRISEF
AHR Domains	**AHR domain - A**	NYFLTGGTGFIGRFLVEKLLARGGTVYVLVREQSQDKLERLRERWGADDKQV KAVIGDLTSKNLGIDAKTLKSLKGNIDHVFHLAAVYDMGADEEAQAATNIEGTR AAVQAAEAMGAKHFHHVSSIAAAGLFKGIFREDMFEEAEKLDHPYLRTKHES EKVVREECKVPFRIYRPGMVIGHSETGEMDKVDGPYYFFKMIQKIRHALPQW VPTIGIEGGRLNIVPVDFVVDALDHIAHLEGEDGNCFHLVDSDPYKVGEILNIFC EAGHAPRMGMRIDSRMFGFIPPFIRQSIKNLPPVKRITGALLDDMGIPPSVMSF INYPTRFDTRELERVLKGTDIEVPRLPSYAPVIWDY
	AHR domain B	NYFVTGGTGFIGRFLIAKLLARGAIVHVLVREQSVQKLADLREKLGADEKQIKA VVGDLTAPSLGLDKKTLKQLSGKIDHFFHLAAIYDMSASEESQQAANIDGTRA AVAAAEALEAGIFHHVSSIAVAGLFKGTFREDMFAEAGKLDHPYFRTKHESER VVRDDCKVPFRIYRPGLVIGDSATGDMDKVDGPYYFFKMIQKIRGALPQWVP TIGIEGGRLNIVPVNFVADALDHIAHLPNEDGKCFHLVDSDPYKVGEILNIFCEA GHAPKMGMRIDSRMFGFVPPFIRQSLKNLPPVKRMGRALLDDLGIPASVLSFI NYPTRFDARETERVLQGTGIEVPRLPDYAPVIWDY
	AHR domain T	QYFVTGATGFIGKRLVRKLLDRRGSTVHFLLRPESERKLPELLAYWGLSGAAK ARAVPVYGDLTAKKLGVAADAIKALKGRIDAIYHLAAVYDLGADEAAQVQVNIE GTRSAVEFAQAIQAGHFHHVSSIAAAGLYEGVFREDMFDEAEGLDHPYFMTK HESEKIVRKECKLPWTVFRPAMVVGDSTTGEMDKIDGPYYFFKLIQRMRQLL PPWMPAVGLEGGRVNIVPVDFVVAALDHISHAKLELDRRCFHLVDPVGYRVG DVLDIFGKAAHAPKMNLFVNAALLGFIPKSVKKGLMALAPVRRIRNAVMKDLG LPEDMLTFVNYPTRFDCRDTQAALKGSGIECPNLKDYAWRLWDY
ATR Domain Blocks	**1A**	WERNLDPDLFKDRTLKGTVEGKVCV
	1B	WERNLDPDLFKDRTLRGTVEGKVCV
	1T	WERNLDPDLFIDRSLRGTVGGKVVL
	1t*	WERHLDPDLFIDRSLRGTVGGKVVL
	2A	VTGATSGIGLATAEKLAEAGAILVIGARTKETLDEVAASLEAKGGNVHAYQCD FS
	2B	VTGATSGIGLATAEKLADAGAILVIGARTQETLDQVSAQLNARGADVHAYQCD FA
	2T	VTGGSSGIGLAAACKFAEAGAVTVICARDADKLDEAVKEIKAFAGKEARVFSY SVDIA
	3A	DMDDCDRFVKTVLDNHGHVDVLVNN
	3B	DMDACDRFIQTVSENHGAVDVLINN
	3T	DEAGCKAFLEALQAEHGGVDFLINN
	4A	AGRSIRRSLALSFDRFHDFERTMQLNY
	4B	AGRSIRRSLDKSFDRFHDFERTMQLNY
	4T	AGRSIRRAIENSYERFHDFERTMQLNY
	5A	FGSVRLIMGFAPAMLERRRGHVVNISSIGVLTNAPRFSAYVSSKSALDAFSRC AAAEWSDRNVTF
	5B	FGSLRLIMGFAPAMLERRRGHIINISSIGVLTNSPRFSAYVASKSALDSFSRCA AAEWSDRRVCF

	5T	FGCLRVTMGVLPGMVAKRKGHVVNISSIGVLTNAPRFSAYVASKAALDAWTR CASSEYADTGISF
	6A	TTINMPLVKTPMIAPTKIYDSVPT
	6B	TTINMPLVKTPMIAPTKIYDSVPT
	6T	TTINMPLVRTPMIAPTKIYNNVPT
	7A	LTPDEAAQMVADAIVYRPKRIATRLGVFAQVLHALAPKMGEIIMNTGYRM
	7B	LSPEEAADMVVNAIVYRPKRIATRMGVFAQVLNAVAPKASEILMNTGYKM
	7T	LAPEEAADMIAQACVYKPVRIATRLGTAGQVLHALAPRVAQIVMNTSFRM
	8A	FPDSPAAAGSKSGEKPKVSTEQVAFAAIMRGIYW*
	8B	FPDSMPKKGKEVSAEKGASTDQVAFAAIMRGIHW*
	8T	FPDSEAAKGEKGAKPQLSAEAVALQQMMRGIHF*

Supplementary Table 3.7. Primer list.

Name	Sequence	Purpose
rJCG5	CTGGACCCGGACCTGTTCATCGATCGTAGCTTGCGT	Clone MT-ACR (domain 2) onto domain 1 of MA-ACR and backbone via Gibson assembly
rJCG6	GAACAGGTCCGGGTCCAG	Amplify MA-ACR domain 1 and pET 22 vector backbone for Gibson assembly of MT- and MB-ACR fusions
rJCG7	TTTGTTAGCAGCCGGATCTTAAAAATGAATACCGCG	Clone MT-ACR (domain 2) onto domain 1 of MA-ACR and backbone via Gibson assembly
rJCG8	CTGGACCCGGACCTGTTCAAAGATCGCACTCTCAGA	Clone MB-ACR (domain 2) onto domain 1 of MA-ACR and backbone via Gibson assembly
rJCG9	TTTGTTAGCAGCCGGATCTTACCAGTGGATACCACG	Clone MB-ACR (domain 2) onto domain 1 of MA-ACR and backbone via Gibson assembly
rJCG10	GATCCGGCTGCTAACAAA	Amplify MA-ACR domain 1 and pET 22 vector backbone for Gibson assembly of MB-- and MT-ACR fusions
rJCG11	GGCTATAACGGGCTCGCTGAAG	Remove BsaI site from MBP
rJCG12	TTTATCGCCGTTAATCCAGAT	Remove BsaI site from MBP
rJCG13	GGTCTCGGTAGTCCCAGATAACCGG	Put BsaI site into MA-ACR domain 1
rJCG14	GGTCTCGTAAGATCCGGCTGCTAA	Put BsaI site onto pET 28 after MA-ACR domain 2
rJCG15	CCGGATGCTCAACGG	For verifying sequence of MBP w/o BsaI site
rJCG16	GTGAAATCATGCCGAACATC	Sequencing MBP-ACR Junction
rJCG17	AATTACCCGACCCGT	For sequencing ATR domain
rJCG18	CCCATCGGTCTCGTAAGATCCGGCTGCTAA	For amplifying pET 28/ ATR 1 backbone with extra bases after the BsaI sites
rJCG19	GCCTGAGGTCTCGGTAGTCCCAGATAACCGG	For amplifying pET 28/ ATR 1 backbone with extra bases after the BsaI sites
rJCG29	TGTAAAACGACGGCCAGT	Amplifying inserts from Twist plasmids containing ATR blocks. Similar to M13 Forward (-20) sequencing primer
rJCG30	CACACAGGAAACAGCTATGAC	Amplifying inserts from Twist plasmids containing ATR blocks. Similar to M13 Reverse (-27) sequencing primer
rJCG36	GGC TAT AAC GGG CTC GC	Fixing deletion in MBP
rJCG37	TTT ATC GCC GTT AAT CCA GAT T	Fixing deletion in MBP
rJCG38	CATACTTGAACGCATAAC	Sequencing MBP

rJCG39	GGTATATCTCCTTCTTAAAGTTA	Making empty pET 28 vector
rJCG40	TAAGATCCGGCTGCTAAC	Making empty pET 28 vector
rJCG41	CCCCACACTACCATCGG	Eliminating BsaI site in 5S RNA terminator region of pBTRCk plasmid
rJCG42	GCTCCCCATGCGAGAGTAG	Eliminating BsaI site in 5S RNA terminator region of pBTRCk plasmid
rJCG43	GCTCGCTGAAGTCGGTAAGAA	Eliminating BsaI site in MBP in MA-ACR on the pBTRCk plasmid
rJCG44	CCGTTATAGCCTTTATCGCC	Eliminating BsaI site in MBP in MA-ACR on the pBTRCk plasmid
rJCG45	AAGACAGGTCTCGGTAGTCCCAGATAACCGGG	Putting Golden-Gate sites in MA-ACR on the pBTRCk plasmid
rJCG46	GTCCGAGGTCTCGTAAGTAGACCATCATCACCATCATCA	Putting Golden-Gate sites in MA-ACR on the pBTRCk plasmid
rJCG47	TCTGGTCTCGCTACTGGGAGCGCAATCT	Put Golden-Gate sites back on chimeric ATR sequence for transferring from pET 28 to pBTRCk
rJCG48	CGAGGTCTCGCTTACCAGTATATCCCCCGC	Put Golden-Gate sites back on chimeric ATR sequence for transferring from pET 28 to pBTRCk
rJCG49	CGAGGTCTCGCTTACCAGTGGATACCACGC	Put Golden-Gate sites back on chimeric ATR sequence for transferring from pET 28 to pBTRCk
rJCG50	CGAGGTCTCGCTTAAAAATGAATACCGCGCATC	Put Golden-Gate sites back on chimeric ATR sequence for transferring from pET 28 to pBTRCk
rJCG51	ATGCTATGGTCCTTGTTGGT	Make empty pBTRCK plasmid
rJCG52	TAAGCTGTTTTGGCGGATG	Make empty pBTRCK plasmid
rJCG53	TGGCAGGGTCTCGTCTGAAATCCTTCCCTCGATC	Make new Golden-Gate backbone for rebuilding parent enzymes
rJCG54	ATACTAGGTCTCGCAGAATTCCAATATTTCGTTACCG	Re-clone WT MT-ACR to get rid of His Tag
rJCG55	ATACTAGGTCTCGCAGAATTCAATTATTTTGTGACCG	Re-clone WT MB-ACR to get rid of His Tag
rJCG56	GTAAAGGGTCTCGTGCGTTATTCACCAGTACATCC	Fix block 3 parent A (MA-ACR) overhangs
rJCG57	GTAAAGGGTCTCGTGCGTTGTTAATCAGTACGTCC	Fix block 3 parent B overhangs
rJCG58	GTAAAGGGTCTCGTGCGTTGTTGATTAAGAAATCTACTC	Fix block 3 parent T overhangs
rJCG59	CCTACAGGTCTCGCGCAGGTCGCTCCATCCGC	Fix block 4 parent A (MA-ACR) overhangs
rJCG60	CCTACAGGTCTCGCGCAGGTCGCAGCATCCGT	Fix block 4 parent B overhangs
rJCG61	CCTACAGGTCTCGCGCAGGACGTAGCATCCGTCG	Fix block 4 parent T overhangs
rJCG62	ATTGTGCCGGTGGATTTC	Sequencing ACR Golden-Gate junction
rJCG63	GTAAGGAATGTAAGCTGCCATG	Sequencing MT-ACR (internal)

rJCG64	AAGACAGGTCTCGGTAGTCCCAGATAACGGGGG	Cloning MB-ACRs AHR into a backbone vector
rJCG65	AAGACAGGTCTCGGTAGTCCCATAAGCGCCACG	Cloning MT-ACRs AHR into a backbone vector
rJCG66	GTCGAAGGTCTCGCTA CTG GGA GCG CAA TCT G	Cloning ACR chimeras from pBTRCk vector back into pET 28 vector, AHR side
rJCG67	GTCGAAGGTCTCGCTT ACC AGT ATA TCC CCC GCA TAA TCG	Cloning ATR chimeras from pBTRCK vector back into pET 28 vector, ATR side parent 1
rJCG68	GTCGAAGGTCTCGCTT ACC AGT GGA TAC CAC GC	Cloning ATR chimeras from pBTRCK vector back into pET 28 vector, ATR side parent 2
rJCG69	GTCGAAGGTCTCGCTT AAA AAT GAA TAC CGC GCA TC	Cloning ATR chimeras from pBTRCK vector back into pET 28 vector, ATR side parent 3
rJCG70	TAATACGACTCACTATAGGG	T7 Promoter (sequencing)
rJCG71	GCTAGTTATTGCTCAGCGG	T7 Terminator (sequencing)
rJCG72	CTACAGGGCGCGTCCCATTCGC	pET50downstream_rev (sequencing)
rJCG73	GTCGAAGGTCTCGTAAGTGCTGTTTTGGCGGATGA	Amplifying MB-ACR backbone to make constructs with MB-ACRs AHR domain in pBTRCK
rJCG74	GTCGAAGGTCTCGGTAGTCCCAGATAACGGGGGC	Amplifying MB-ACR backbone to make constructs with MB-ACRs AHR domain in pBTRCK
rJCG75	GTCGAAGGTCTCGCAGAATTCAATTATTTTGTGACCGG	Amplifying MB-ACR Aldehyde reductase domain along with rJCG74 in pBTRCK
rJCG76	GTCGAAGGTCTCGGTAGTCCCATAAGCGCCACG	Amplifying MT-ACR aldehyde reductase domain along with rJCG 77 in pBTRCK
rJCG77	GTCGAAGGTCTCGCAGAATTCCAATATTTCGTTACCG	Amplifying MT-ACR aldehyde reductase domain along with rJCG 76 in pBTRCK

References:

(1) Shirmer, A.; Rude, M. A.; Li, X.; Popova, E.; del Cardayre, S. B. Microbial Biosynthesis of Alkanes. *Science (80-.).* **2010**, *329* (July), 559–562. https://doi.org/10.1126/science.1187936.

(2) Hofvander, P.; Doan, T. T. P.; Hamberg, M. A Prokaryotic Acyl-CoA Reductase Performing Reduction of Fatty Acyl-CoA to Fatty Alcohol. *FEBS Lett.* **2011**, *585* (22), 3538–3543. https://doi.org/10.1016/j.febslet.2011.10.016.

(3) Vioque, J.; Kolattukudy, P. E. Resolution and Purification of an Aldehyde-Generating and an Alcohol-Generating Fatty Acyl-CoA Reductase from Pea Leaves (Pisum SativumL.). *Arch. Biochem. Biophys.* **1997**, *340* (1), 64–72. https://doi.org/10.1006/abbi.1997.9932.

(4) Willis, R. M.; Wahlen, B. D.; Seefeldt, L. C.; Barney, B. M. Characterization of a Fatty Acyl-CoA Reductase from Marinobacter Aquaeolei VT8: A Bacterial Enzyme Catalyzing the Reduction of Fatty Acyl-CoA to Fatty Alcohol. *Biochemistry* **2011**, *50* (48), 10550–10558. https://doi.org/10.1021/bi2008646.

(5) Metz, J. G.; Pollard, M. R.; Anderson, L.; Hayes, T. R.; Lassner, M. W.; Campus, C.; Street, F. Purification of a Jojoba Embryo Fatty Acyl-Coenzyme A Reductase and Expression of Its CDNA in High Erucic Acid Rapeseed. *Plant Physiol.* **2000**, *122* (3), 635–644. https://doi.org/10.1104/pp.122.3.635.

(6) Rowland, O.; Zheng, H.; Hepworth, S. R.; Lam, P.; Jetter, R.; Kunst, L. CER4 Encodes an Alcohol-Forming Fatty Acyl-Coenzyme A Reductase Involved in Cuticular Wax Production in Arabidopsis. *Plant Physiol.* **2006**, *142* (3), 866–877. https://doi.org/10.1104/pp.106.086785.

(7) Youngquist, J. T.; Schumacher, M. H.; Rose, J. P.; Raines, T. C.; Politz, M. C.; Copeland, M. F.; Pfleger, B. F. Production of Medium Chain Length Fatty Alcohols from Glucose in Escherichia coli. *Metab. Eng.* **2013**, *20*, 177–186. https://doi.org/10.1016/j.ymben.2013.10.006.

(8) Mehrer, C. R.; Incha, M. R.; Politz, M. C.; Pfleger, B. F. Anaerobic Production of Medium-Chain Fatty Alcohols via a β-Reduction Pathway. *Metab. Eng.* **2018**, *48* (April), 63–71. https://doi.org/10.1016/j.ymben.2018.05.011.

(9) Hernández Lozada, N. J.; Simmons, T. R.; Xu, K.; Jindra, M. A.; Pfleger, B. F. Production of 1-Octanol in Escherichia coli by a High Flux Thioesterase Route. *Metab. Eng.* **2020**, *61*, 352–359. https://doi.org/10.1016/j.ymben.2020.07.004.

(10) Opgenorth, P.; Costello, Z.; Okada, T.; Goyal, G.; Chen, Y.; Gin, J.; Benites, V.; De Raad, M.; Northen, T. R.; Deng, K.; et al. Lessons from Two Design-Build-Test-Learn Cycles of Dodecanol Production in Escherichia coli Aided by Machine Learning. *ACS Synth. Biol.* **2019**, *8* (6), 1337–1351. https://doi.org/10.1021/acssynbio.9b00020.

(11) Liu, A.; Tan, X.; Yao, L.; Lu, X. Fatty Alcohol Production in Engineered E. coli Expressing Marinobacter Fatty Acyl-CoA Reductases. *Appl. Microbiol. Biotechnol.* **2013**, *97* (15), 7061–7071. https://doi.org/10.1007/s00253-013-5027-2.

(12) Steen, E. J.; Kang, Y.; Bokinsky, G.; Hu, Z.; Schirmer, A.; McClure, A.; del Cardayre, S. B.; Keasling, J. D. Microbial Production of Fatty-Acid-Derived Fuels and Chemicals from Plant Biomass. *Nature* **2010**, *463* (7280), 559–562. https://doi.org/10.1038/nature08721.

(13) Liu, R.; Zhu, F.; Lu, L.; Fu, A.; Lu, J.; Deng, Z.; Liu, T. Metabolic Engineering of

Fatty Acyl-ACP Reductase-Dependent Pathway to Improve Fatty Alcohol Production in Escherichia coli. *Metab. Eng.* **2014**, *22*, 10–21. https://doi.org/10.1016/j.ymben.2013.12.004.

(14) Romero, P. a; Krause, A.; Arnold, F. H. Navigating the Protein Fitness Landscape with Gaussian Processes. *Proc. Natl. Acad. Sci. U. S. A.* **2013**, *110* (3), E193-201. https://doi.org/10.1073/pnas.1215251110.

(15) Yang, K. K.; Wu, Z.; Arnold, F. H. Machine-Learning-Guided Directed Evolution for Protein Engineering. *Nature Methods*. Nature Publishing Group August 1, 2019, pp 687–694. https://doi.org/10.1038/s41592-019-0496-6.

(16) Saito, Y.; Oikawa, M.; Nakazawa, H.; Niide, T.; Kameda, T.; Tsuda, K.; Umetsu, M. Machine-Learning-Guided Mutagenesis for Directed Evolution of Fluorescent Proteins. *ACS Synth. Biol.* **2018**, *7* (9), 2014–2022. https://doi.org/10.1021/acssynbio.8b00155.

(17) Alley, E. C.; Khimulya, G.; Biswas, S.; AlQuraishi, M.; Church, G. M. Unified Rational Protein Engineering with Sequence-Based Deep Representation Learning. *Nat. Methods* **2019**. https://doi.org/10.1038/s41592-019-0598-1.

(18) 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 Comput. Biol.* **2017**, *13* (10), 1–21. https://doi.org/10.1371/journal.pcbi.1005786.

(19) Liao, J.; Warmuth, M. K.; Govindarajan, S.; Ness, J. E.; Wang, R. P.; Gustafsson, C.; Minshull, J. Engineering Proteinase K Using Machine Learning and Synthetic Genes. *BMC Biotechnol.* **2007**, *7*, 1–19. https://doi.org/10.1186/1472-6750-7-16.

(20) Fox, R. J.; Davis, S. C.; Mundorff, E. C.; Newman, L. M.; Gavrilovic, V.; Ma, S. K.; Chung, L. M.; Ching, C.; Tam, S.; Muley, S.; et al. Improving Catalytic Function by ProSAR-Driven Enzyme Evolution. *Nat. Biotechnol.* **2007**, *25* (3), 338–344. https://doi.org/10.1038/nbt1286.

(21) Lennen, R. M.; Braden, D. J.; West, R. M.; Dumesic, J. A.; Pfleger, B. F. A Process for Microbial Hydrocarbon Synthesis: Overproduction of Fatty Acids in *Escherichia coli* and Catalytic Conversion to Alkanes. *Biotechnol. Bioeng.* **2010**, *106* (2), 193–202. https://doi.org/10.1002/bit.22660.

(22) Voigt, C. A.; Martinez, C.; Wang, Z. G.; Mayo, S. L.; Arnold, F. H. Protein Building Blocks Preserved by Recombination. *Nat. Struct. Biol.* **2002**, *9* (7), 553–558. https://doi.org/10.1038/nsb805.

(23) Silberg, J. J.; Endelman, J. B.; Arnold, F. H. SCHEMA-Guided Protein Recombination. *Methods Enzymol.* **2004**, *388* (2003), 35–42. https://doi.org/10.1016/S0076-6879(04)88004-2.

(24) Endelman, J. B.; Silberg, J. J.; Wang, Z.; Arnold, F. H. Site-Directed Protein Recombination as a Shortest-Path Problem. *Protein Eng. Des. Sel.* **2004**, *17* (7), 589–594. https://doi.org/10.1093/protein/gzh067.

(25) Srinivas, N.; Krause, A.; Kakade, S. M.; Seeger, M. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. *IEEE Trans. Inf. Theory* **2009**, *58* (5), 3250–3265. https://doi.org/10.1109/TIT.2011.2182033.

(26) Auer, P. *Using Confidence Bounds for Exploitation-Exploration Trade-Offs*; 2002; Vol. 3.

(27) Davis, M. S.; Cronan, J. Inhibition of Escherichia coli Acetyl Coenzyme A Carboxylase by Acyl-Acyl Carrier Protein. *J. Bacteriol.* **2001**, *183* (4), 1499–1503. https://doi.org/10.1128/JB.183.4.1499-1503.2001.

(28) Rock, C. O.; Jackowski, S. Regulation of Phospholipid Synthesis in Escherichia coli Composition of the Acyl-Acyl Carrier Protein Pool in Vivo. *J Biol Chem* **1982**, *257* (18), 10759–10765.

(29) Sarria, S.; Bartholow, T. G.; Verga, A.; Burkart, M. D.; Peralta-Yahya, P. Matching Protein Interfaces for Improved Medium-Chain Fatty Acid Production. *ACS Synth. Biol.* **2018**, *7* (5), 1179–1187. https://doi.org/10.1021/acssynbio.7b00334.

(30) Gray, J. J.; Moughon, S.; Wang, C.; Schueler-Furman, O.; Kuhlman, B.; Rohl, C. A.; Baker, D. Protein-Protein Docking with Simultaneous Optimization of Rigid-Body Displacement and Side-Chain Conformations. *J. Mol. Biol.* **2003**, *331* (1), 281–299. https://doi.org/10.1016/S0022-2836(03)00670-3.

(31) Yang, K. K.; Elliott Robinson, J. Machine Learning-Guided Channelrhodopsin Engineering Enables Minimally Invasive Optogenetics. *Nat. Methods*. https://doi.org/10.1038/s41592-019-0583-8.

(32) Jiménez-Osés, G.; Osuna, S.; Gao, X.; Sawaya, M. R.; Gilson, L.; Collier, S. J.; Huisman, G. W.; Yeates, T. O.; Tang, Y.; Houk, K. N. The Role of Distant Mutations and Allosteric Regulation on LovD Active Site Dynamics. *Nat. Chem. Biol.* **2014**, *10* (6), 431–436. https://doi.org/10.1038/nchembio.1503.

(33) Pfleger, B. F.; Gossing, M.; Nielsen, J. Metabolic Engineering Strategies for Microbial Synthesis of Oleochemicals. *Metab. Eng.* **2015**, *29*, 1–11. https://doi.org/10.1016/j.ymben.2015.01.009.

(34) Fiser, A.; Kinh Gian Do, R.; Sali. Modeling Loops in Protein Structures. *Protein Sci.* **2000**, *9*, 1753–1773. https://doi.org/10.1002/9780470882207.ch13.

(35) Rasmussen, C. E.; Williams, C. *Gaussian Processes for Machine Learning*; Adaptive Computation and Machine Learning; MIT Press: Cambridge, MA, 2006; Vol. 14.

(36) Desautels, T.; Krause, A.; Burdick, J. W. *Parallelizing Exploration-Exploitation Tradeoffs in Gaussian Process Bandit Optimization*; 2014; Vol. 15.

(37) Schindelin, J.; Arganda-Carreras, I.; Frise, E.; Kaynig, V.; Longair, M.; Pietzsch, T.; Preibisch, S.; Rueden, C.; Saalfeld, S.; Schmid, B.; et al. Fiji: An Open-Source Platform for Biological-Image Analysis. *Nature Methods*. Nature Publishing Group July 28, 2012, pp 676–682. https://doi.org/10.1038/nmeth.2019.

(38) Beld, J.; Finzel, K.; Burkart, M. D. Versatility of Acyl-Acyl Carrier Protein Synthetases. *Chem. Biol.* **2014**, *21*, 1293–1299. https://doi.org/10.1016/j.chembiol.2014.08.015.

(39) Néstor, N.; Hernández, J.; Lozada, H.; Lai, R.-Y.; Simmons, T. R.; Thomas, K. A.; Chowdhury, R.; Maranas, C. D.; Pfleger, B. F. Highly Active C 8-Acyl-ACP Thioesterase Variant Isolated by a Synthetic Selection Strategy. *ACS Synth. Biol.* **2018**, *7*, 2205–2215. https://doi.org/10.1021/acssynbio.8b00215.

(40) Heath, R. J.; Rock, C. O. Inhibition of β-Ketoacyl-Acyl Carrier Protein Synthase III (FabH) by Acyl-Acyl Carrier Protein in Escherichia coli. *J. Biol. Chem.* **1996**, *271* (18), 10996–11000. https://doi.org/10.1074/jbc.271.18.10996.

(41) Marze, N. A.; Roy Burman, S. S.; Sheffler, W.; Gray, J. J. Efficient Flexible Backbone Protein-Protein Docking for Challenging Targets. *Bioinformatics* **2018**,

34 (20), 3461–3469. https://doi.org/10.1093/bioinformatics/bty355.

Chapter 4

Engineering 1-Deoxy-D-Xylulose 5-Phosphate Synthase (DXS) for improved MEP pathway flux using a high-throughput growth selection

Authors and Contributors:

- Jonathan C. Greenhalgh designed and performed experiments, analyzed the data, wrote code, made figures, and wrote the manuscript.
- Dr. Jyun-Liang Lin designed and performed experiments, wrote code, analyzed data and made figures.
- Ben Bremer performed experiments, wrote code and analyzed data.
- Sam Gardner performed experiments and analyzed data.
- Haiyang Zheng performed experiments and analyzed data.
- Jerry Duan wrote code and analyzed data.
- Dr. Brian Pfleger provided guidance.
- Dr. Philip A. Romero, provided guidance, wrote code, and edited and reviewed the manuscript.

This work was funded by the Great Lakes Bioenergy Research Center (GLBRC).

4.1 Introduction

Protein engineering[1] is a very useful tool for solving various problems in biology. For example, engineered proteins, such as antibodies, can be used as therapeutics and engineered enzymes can be used in industrial processes or in cells to perform specific chemical reactions or degradations. Protein engineering is an especially useful tool in the field of metabolic engineering, where an engineered protein can be used to introduce a novel[2] biochemical pathway, or to remove a bottleneck in an existing pathway[3] with the goal of improving the yield of a specific product or chemical degradation.

One pathway that can be improved by protein engineering efforts is the 2-methyl-3-erythritol 4-phosphate (MEP) pathway. The MEP pathway is an important pathway for bacterial production of terpenoid compounds[4–7]. Terpenoids, also called isoprenoids, are useful as advanced biofuels such as farnesane[12,13], as precursors for polymers[14],and as precursors to pharmaceuticals, such as the anti-malarial drug artemisinin[8–10] or the anti-cancer compound taxol[11],

However, the first enzyme in the pathway, 1-Deoxy-D-xylulose 5-phosphate (DXS) is a major bottleneck for the pathway[5,15–18]. As such, DXS is a prime target for protein engineering, since increasing its activity could alleviate the bottleneck in the pathway and enable higher *in vivo* flux towards isoprenoids[19]. There have been many studies that have used DXSs from various species of bacteria and plants to improve flux through the MEP pathway (to make products such as lycopene[4,20,21], isoprene[5] or other terpenoid or carotenoids[22]), but there have not been as many attempts to engineer the DXS enzyme itself. One notable attempt to engineer DXS was work done by Banerjee et al.[19], where they used rational protein design to engineer the DXS from *Populus trichocarpa* to have

lower feedback inhibition effects from isopentenyl pyrophosphate (IPP), but the mutations in some cases reduced the catalytic activity as well.

In this work, we build a library of chimeric DXSs and design and utilize a high throughput growth-based selection and Oxford Nanopore sequencing to identify variants with improved *in vivo* characteristics. We also employ machine learning techniques to design improved DXS sequences and then characterize DXS activity in depth with an *in vitro* coupled-enzyme assay. The designed DXSs generally show substantially higher enrichment in our selection than the wild-type parental enzymes, though the *in vitro* assays suggest their catalytic efficiency is near the wild-type. Interestingly, we also observe that the designed DXSs display a sensitivity to downstream metabolites (IPP and DMAPP) that was not observed in the parental sequences. These results demonstrate a workflow that could be adapted to engineer other enzymes in the MEP pathway.

4.2 Results

4.2.1 Library design

Protein recombination is an extremely useful tool in protein engineering. Proteins in a protein family can be split into functional modules or sequence blocks, which can then be swapped for equivalent modules from a different homolog in the family to form a chimeric protein. To determine the location and size of these modules for our chimeric protein library, we made a multiple sequence alignment of DXSs using jackhmmer[23] (Figure 4.1a), and then used PSI-COV[24] (Figure 4.1c) to identify pairs of covarying amino acid resides. We selected four DXSs as parental enzymes: EcDXS, BsDXS, ZmDXS

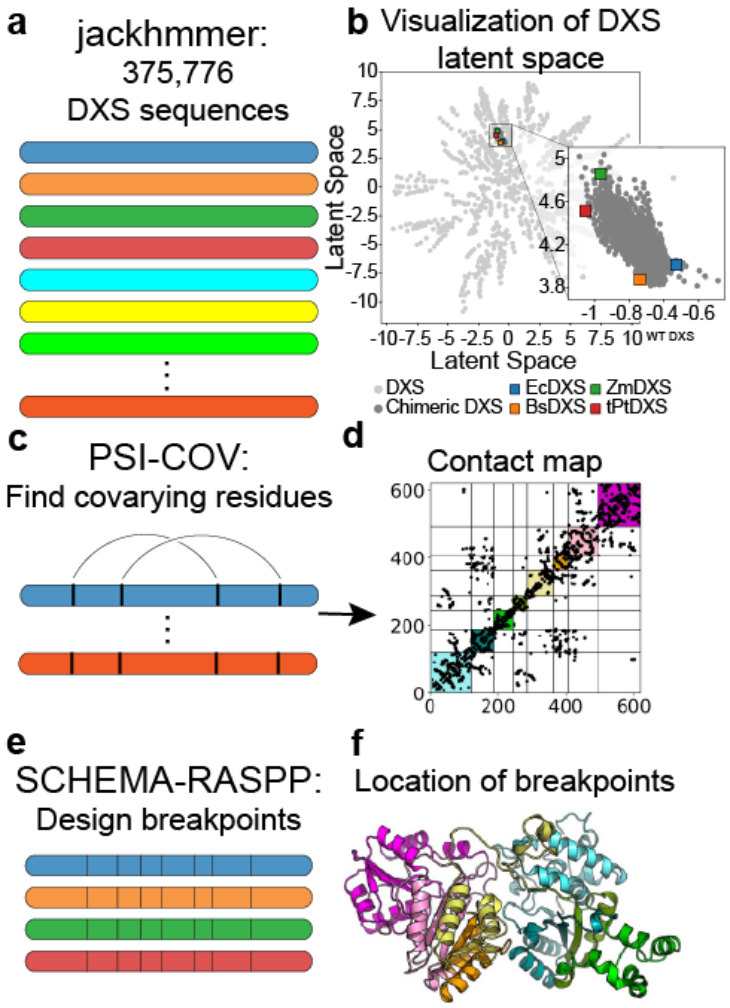

Figure 4.1: Overview of DXS library design. a) We used the jackhmmer algorithm to generate a multiple sequence alignment (MSA) of 375,776 DXS sequences. b) We visualized the DXS sequence space by trainining a variational autoencoder (VAE) on the MSA. Chimeric DXSs in our final library are shown in the inset as dark gray, the parents are shown as colored squares. c) We used PSI-COV to analyze the MSA and generate a contact map (d) that can be a proxy to a structure or homology model. Then, we used the contact map as an input for the SCHEMA RASPP algorithm (e) to determine the most effective breakpoints. f) Crystal structure of DXS from E. coli with each block in a different color. The colors in (f) correspond to the colors in the contact map (d).

(which are wild-type sequences), and tPtDXS, (which is a truncation of the double-mutant DXS from *Populus trichocarpa*)[19]. We then determined a contact map (Figure 4.1d) using these covarying residues and used it as an input for the SCHEMA-RASPP algorithm. We used SCHEMA-RASPP to calculate breakpoints that would split the parental enzymes into eight blocks (Figure 4.1e and f), resulting in a library size of 6,5536 possible DXS

sequences. We denote chimeric sequences as a string of numbers, such as 43231311, where 1, 2, 3 and 4 correspond to EcDXS, BsDXS, ZmDXS and tPtDXS respectively.

4.2.2 High throughput mapping of the DXS activity landscape

To learn the DXS sequence function mapping, we designed a growth-based selection. DXS is an essential enzyme in *E. coli*, but by expressing enzymes from the heterologous mevalonate pathway in *E. coli* it is possible to grow DXS knockout strains by supplementing with mevalonate[8] or functionally equivalent molecules (such as mevalonolactone). The Jennifer Reed lab generously provided us with the ΔdxsMB strain[8] (originally developed by Martin et al.) which we used as the base strain for our selection. We then cloned the full chimeric DXS library into ΔdxsMB, supplementing the media with either mevalonic acid or mevalonolactone (Figure 4.2a). In supplemented conditions, the cells can grow regardless of whether the DXS that they're expressing is functional. However, by transferring them to media without mevalonate, a selection pressure is applied that favors cells with active DXS variants. In order to determine DXS fitness, we grew the DXS library with mevalonate, washed the cells to remove extracellular mevalonate and transferred them to media without mevalonate. Then we added a chemical inducer (IPTG) to initiate protein expression. We grew this culture longer to ensure any remaining intercellular mevalonate was depleted, and then used the cells to start a final selection culture. We sequenced fractions of the cells from prior to the wash step (pre-selection) and after the induced culture was allowed to grow (post-selection). Comparing the sequencing reads, we then calculated the frequencies each sequence

appeared in each dataset and determined their enrichment values, which we use as an indicator of overall fitness in the selection (Figure 4.2a and b).

The enrichment values that we determined from the sequencing reads can be used to map the DXS sequences to their function (Figure 4.2b and c). We found that the enrichment values were strongly correlated between our experimental replicates, demonstrating the robustness of the method. The enrichment values, as well as machine learning models trained on them, were then used to identify DXS sequences with especially high fitness. Two DXS sequences were designed by simply maximizing the enrichment values (site-wise and sequence based), and two other DXS sequences were designed using a positive-unlabeled (PU) learning strategy[25] (Figure 4.2d). Each of these designed sequences showed a very high enrichment in our *in vivo* selection and were then studied further to determine what traits led to their high fitness.

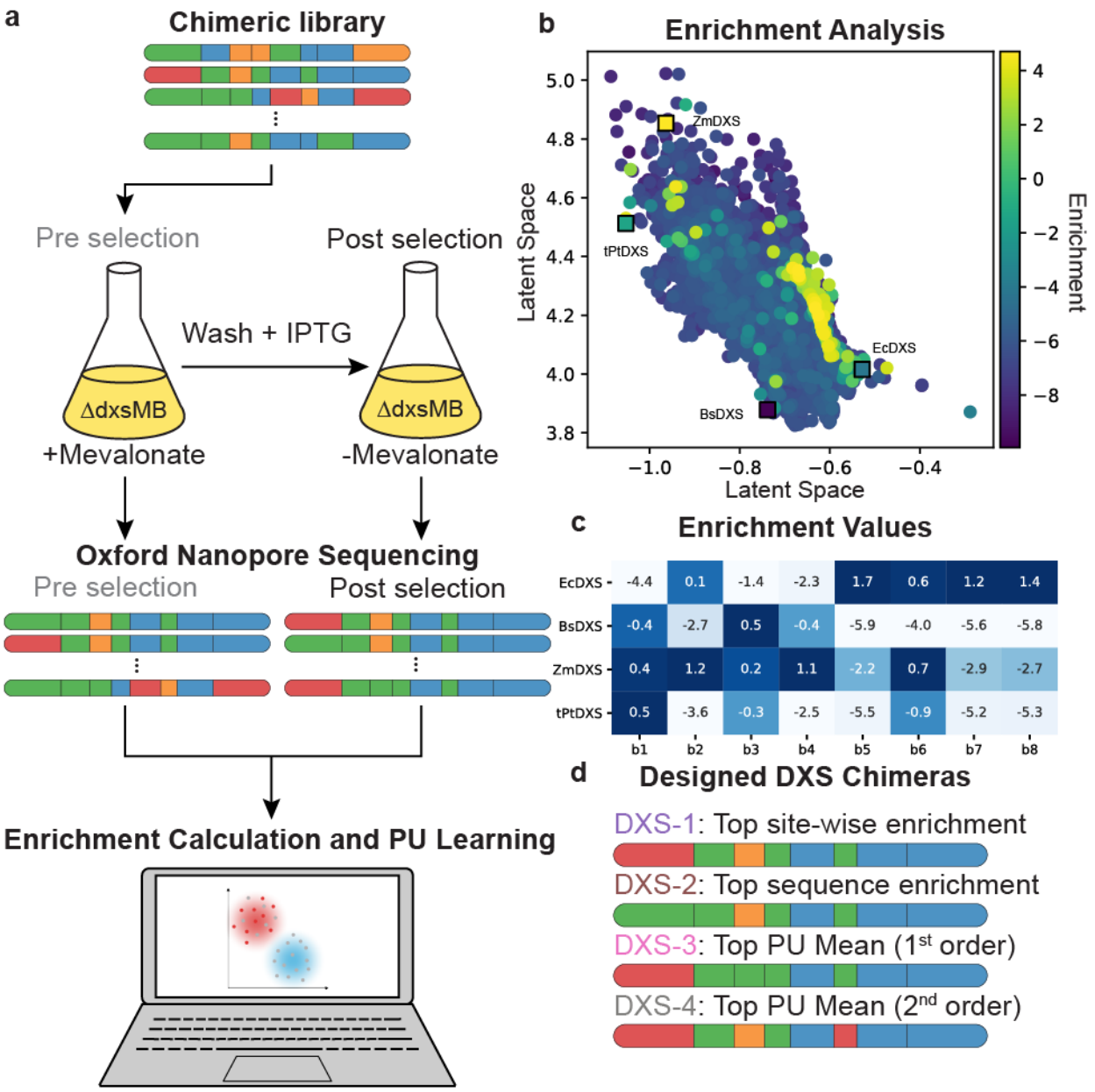

Figure 4.2: a) We transformed our library of chimeric DXS enzymes into the strain ΔdxsMB and carried out a selection by removing mevalonate from the media. We isolated the DNA from cultures before and after the selection for Oxford Nanopore sequencing, and then used the data to determine which DXS variants were enriched or predicted to have high activity (using PU learning models). b) We mapped the enrichment values to the VAE visualization to generate a depiction of the sequence function landscape. We observed a cluster of highly active DXS variants near the bottom right corner of the chimera space. c) Site-wise enrichment values. d) Designed DXS chimeras block representations and methods of design. Blue, orange, green and red correspond to EcDXS, BsDXS, ZmDXS and tPtDXS respectively.

4.2.3 Experimental characterization of designed DXSs

To characterize our designed DXS variants, we first performed growth assays designed to mimic the conditions of the selection (Figure 4.3), except for the sequences were assayed individually and not in a pooled manner. We found that all the designed DXSs and parental DXSs grew under selective conditions, but we did not observe a correlation between the growth rates and the enrichment values from the pooled selection.

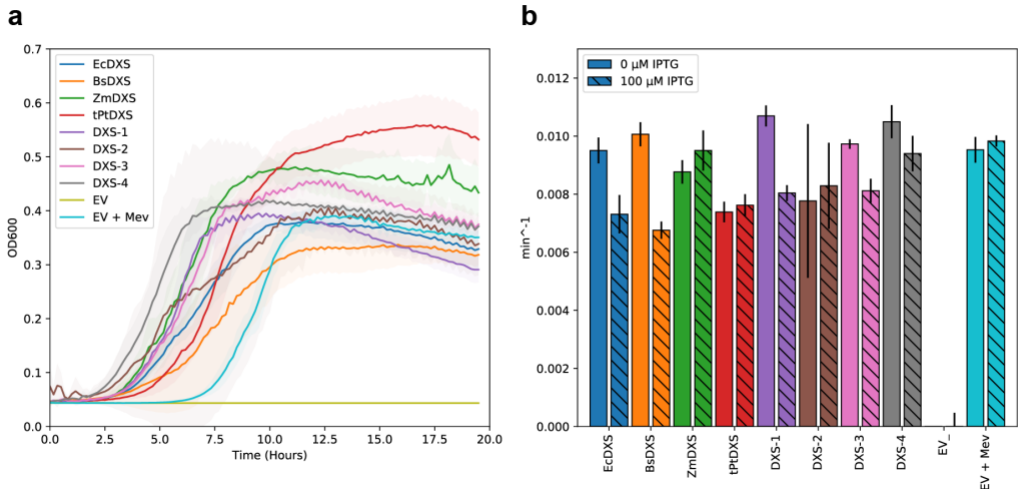

Figure 4.3: a) Growth curves of DXSs transformed into ΔdxsMB in parallel individual (i.e. non-pooled) growth assays. EV is an empty vector control EV + Mev is the empty vector control supplemented with mevalonate, which was used as a positive control. b) Growth rates with and without IPTG. We observed that the growth rates in our selection conditions were generally quite similar between the parental enzymes and the designed chimeras. Interestingly, we also observed high growth rates with no IPTG, suggesting that there is some leaky expression.

We also attempted to determine whether differences in protein expression could be the cause of increased enrichment (Figure 4.4). We grew *E. coli* cells expressing each of the parental and designed DXSs in selective conditions, and then harvested the cells in the exponential growth phase for analysis. We then lysed the cells and used SDS-PAGE gels to study the expression levels of each of the DXSs. We did not observe a

clear correlation with the enrichment and expression; EcDXS was the one of the most highly expressed DXSs, yet it has some of the lowest enrichment values, whereas ZmDXS expression was very low, yet it has the highest enrichment of all the wild-type enzymes. However, it does appear that the chimeric DXSs were expressed better than the other parental DXSs.

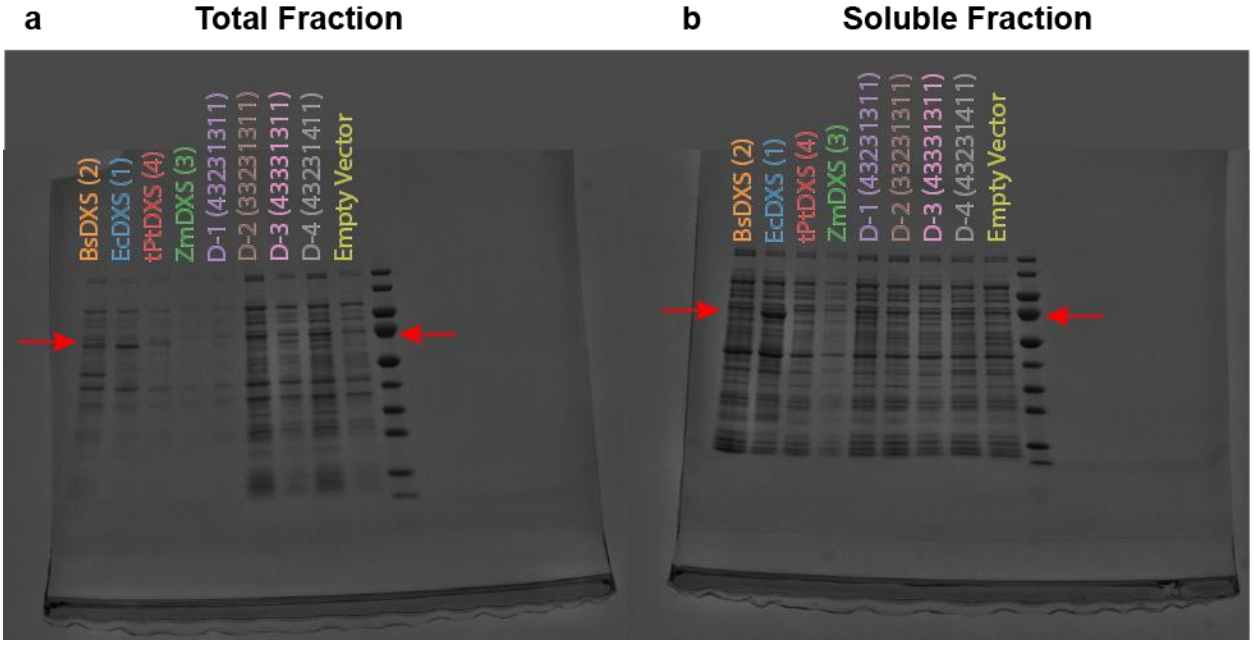

Figure 4.4: a) SDS-PAGE gel of total fractions from DXS expression experiment. b) Gel of soluble fractions. In general, the chimeric DXSs were expressed quite well, as was EcDXS. The other parental DXSs were more poorly expressed.

Due to the lack of correlation between expression and enrichment, , we next set up a miniaturized pooled selection to verify the trends we had initially observed in the enrichment values. We repeated the pooled selection process as described earlier, but instead of the full DXS library, we only transformed eight sequences (four designed DXSs and the four parental DXSs). The sequencing results from this experiment showed a very strong correlation with the results from the full library (Figure 4.5), validating the method and the observations.

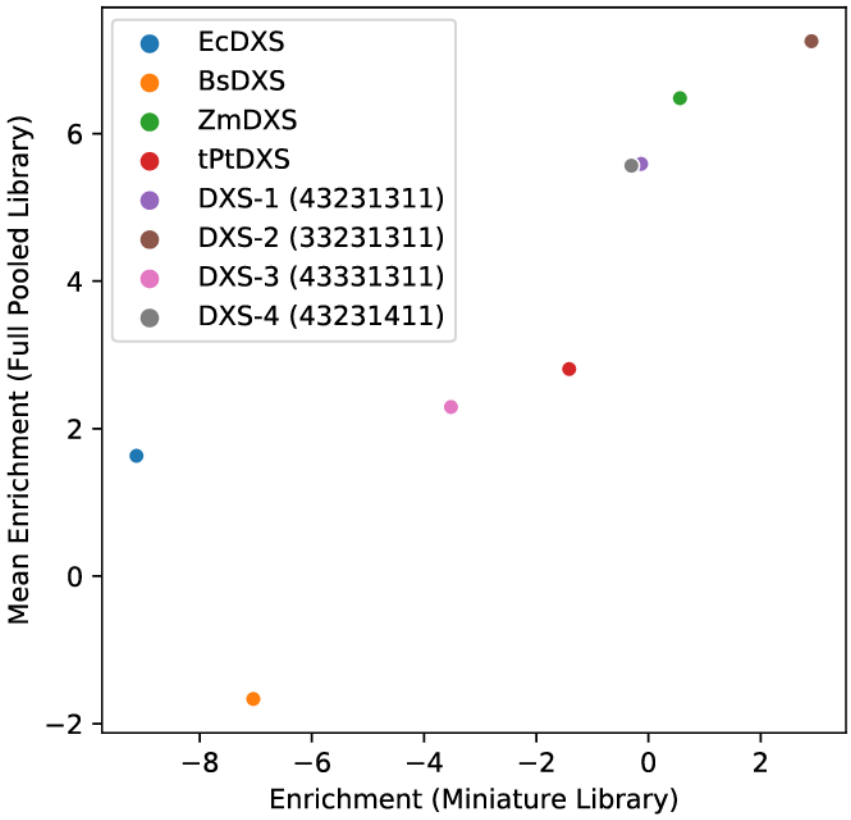

Figure 4.5: Comparison of pooled and non-pooled (miniature) enrichment experiments. The overall trend in enrichment was reproduced when we tested the smaller pool of eight enzymes and them to the results from the full library. There was a mismatch in the precise enrichment values, but this was expected since the full library would have had manyfold lower counts of the parents and designed enzymes due to competition with all the other sequences in the library.

To further study the designed DXSs, and probe the origin of their increased enrichment, we performed *in vitro* assays to determine kinetic parameters for each parental and designed DXS. We quantified reaction rates using a coupled-enzyme assay with DXR (Table 4.1) and fit the rates to a Michaelis-Menten model. Surprisingly, we found that, overall, the designed DXSs had similar activity to the wild type sequences. The K_M

values for the DXSs were all very similar to each other as well. One DXS sequence (DXS-1) did appear to have an improved k_{cat} over the best parental DXS (EcDXS) in some assays, but the difference was small.

Table 4.1: Kinetic constants for DXS enzymes determine using coupled enzyme assays

Enzyme	GAP kcat (/s)	Pyruvate kcat (/s)	TPP kcat (/s)	GAP KM (μM)	Pyruvate KM (μM)	TPP KM (μM)	IPP Ki (μM)	DMAPP Ki (μM)
EcDXS	3.63 ± 4.36	2.01 ± 0.4	0.88 ± 0.37	260.83 ± 149.06	80.49 ± 34.98	1.48 ± 0.66	471.91 ± 102.55	954 ± 79.68
BsDXS	1.56 ± 0.49	1.32 ± 0.49	0.81 ± 0.35	136.33 ± 34.45	100.85 ± 28.4	1.3 ± 0.34	>1000	>1000
ZmDXS	0.8 ± 0.5	0.92 ± 0.61	2.14 ± 0.18	226.46 ± 147.37	239.41 ± 144.14	0.83 ± 0.67	165.26 ± 48.34	602.95 ± 343.97
tPtDXS	0.13 ± 0.08	0.19 ± nan	0.74 ± nan	7.71 ± 3.51	8.23 ± nan	0.16 ± nan	NM	NM
DXS-1	2.09 ± 1.09	2.21 ± 0.38	1.76 ± 0.32	237.71 ± 155.86	213.79 ± 72.55	2.03 ± 0.38	29.79 ± 9.02	74.75 ± 2.5
DXS-2	1.52 ± 0.39	1.46 ± 0.64	2.52 ± 0.35	227.17 ± 66.7	122.04 ± 36.82	1.61 ± 0.37	21.54 ± 1.68	55.47 ± 5.34
DXS-3	0.75 ± 0.17	0.79 ± 0.36	1.54 ± 0.56	101.66 ± 35.29	108.19 ± 46.23	0.72 ± 0.37	11.15 ± 2.61	35.29 ± 17.92
DXS-4	1.05 ± 0.36	1.03 ± 0.81	1.65 ± 0.41	119.45 ± 61.82	93.25 ± 74.47	1.07 ± 0.14	28.74 ± 1.83	98.79 ± 23.64

We also performed inhibition studies to determine whether there were differences in how the designed DXSs were impacted by feedback inhibition. Feedback inhibition involving downstream prenyl phosphates is known to affect some DXSs, and our experiments probed whether there was competition between these inhibitors and the cofactor thiamine pyrophosphate (TPP), whose activation to an ylide is essential to catalyzing the DXS reaction[26]. Surprisingly, we observed feedback inhibition from isopentenyl pyrophosphate (IPP) and dimethylallyl pyrophosphate (DMAPP) in all the designed DXS sequences, while the inhibition was markedly less in the wild-type sequences (Figure 4.6).

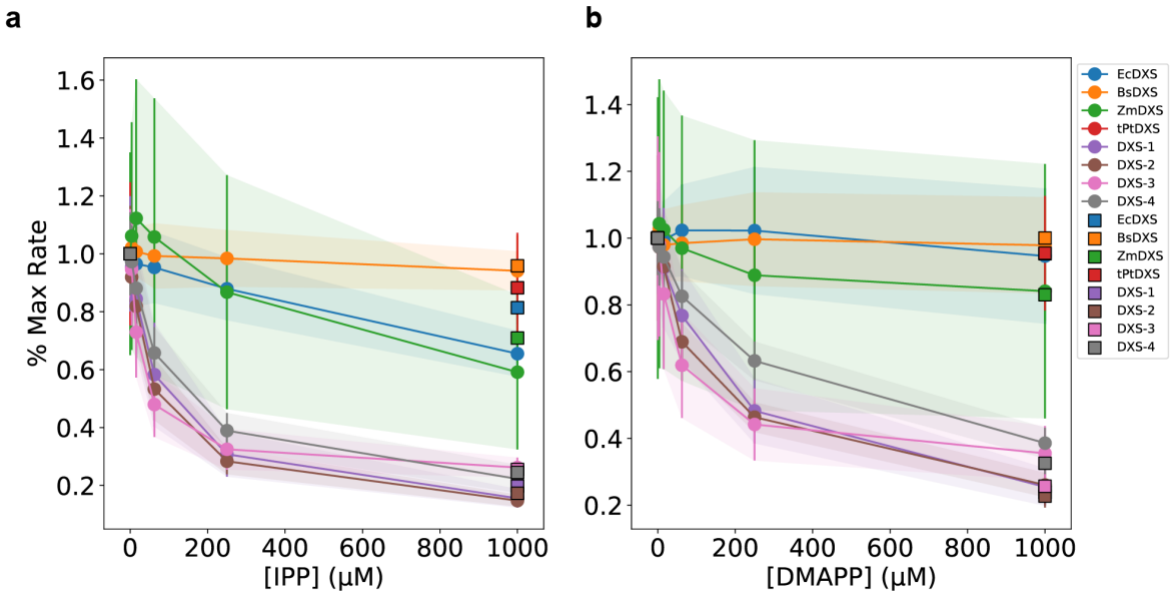

Figure 4.6: IPP (a) and DMAPP (b) inhibition curves. We titrated IPP and DMAPP into coupled-enzyme assays to determine their effects on enzyme activity. Interestingly we observed a very strong effect on the chimeric enzymes and only a small effect on the parental enzymes.

4.3 Discussion

4.3.1 Recombination

A major advantage of our recombination strategy is that it generally preserves the most important features of the proteins structure and makes it more likely that the recombinant protein will retain its function. Interestingly, recombination strategies often result in new proteins that have properties that are superior to any of the parental enzymes used to make it. Other advantages of module-based recombination approaches are that they are highly amenable to training machine learning models, and that the size of the sequence block is usually large enough that it can be correctly identified in next generation sequencing workflows without base-level accuracy. These two factors make comprehensive screening of a chimeric protein library and learning the sequence - function landscape a possibility. In this study, we had good success of designed enzymes

both expressing and exhibiting the desired function. Combined with the PU machine learning strategy, we were able to design DXS enzymes with enhanced enrichment scores.

4.3.2 Advantages of coupling chimeric library with ONT

Our strategy for coupling a growth-based selection with Oxford Nanopore (ONT) long read sequencing resulted in generation of a very large library of chimeric proteins containing tens of thousands of sequencing reads. Machine learning modeling efforts involving chimeric proteins have typically been limited in size to a few hundred sequences, due to the need to clone chimeras individually for testing. Because we have access to a growth selection and ONT sequencing, we were able to screen a large library of over 65,000 sequences. In some cases, we were unable to aquire adequate sequencing data, however we acquired enough sequencing data to calculate enrichment values for about 25% of these sequences (roughly 16,000 sequences), making this a very thorough characterization of the chimeric landscape. Having access to such a large volume of data also enables training of more advanced machine learning models to better identify important sequence features for activity.

4.3.3 Discussion of kinetic parameters and correlations with enrichment

It was very surprising that we didn't observe improvements in either the growth rates or k_{cat}/K_M for the designed DXS sequences given their high enrichment values. We expected that DXS variants with higher enrichment values would correspond to higher activity, but we found that that was not the case. There appears to be no correlation

between the k_{cat} of the DXS or the enrichment values, suggesting that there are other factors influencing the enrichment values than enzyme activity alone. To determine what other factors could be influencing the enrichment values, we performed growth rate experiments in conditions selective for DXS function, and also probed the role of protein expression. However, there were also no trends relating enrichment to either growth rate in selective conditions or to protein expression.

Our finding that the designed chimeras were inhibited more strongly by IPP and DMAPP was even more surprising. The results of the inhibition experiments suggest that feedback regulation is playing an important role in DXS fitness. It is possible that because the wild type enzymes are less strongly inhibited, cells expressing them accumulate toxic levels of IPP and/or DMAPP, whereas cells expressing the DXSs that are susceptible to feedback inhibition can reduce flux through the MEP pathway when IPP and DMAPP levels are too high. IPP has been shown to have varying detrimental effects ranging from slowing down nutrient uptake to potentially disrupting nucleotide synthesis[27,28]. Since we observed both better protein expression for the designed DXS enzymes, and an increased inhibition in the presence of IPP and DMAPP, it is likely that these two observations are linked.

4.3.4 Effects of feedback inhibition by IPP and DMAPP

While DXS variants that are feedback inhibited may seem contrary to the engineering objective of achieving high flux, it could be an advantage *in vivo*, allowing the cells expressing the DXS to temporarily turn down the flux when there is too much IPP, and allowing the flux to increase again once excess IPP has been depleted by

downstream enzymes. Because there are multiple other bottleneck enzymes downstream of DXS in the MEP pathway, having a DXS that can switch between high and low flux states could be a beneficial trait *in vivo* when there is a selective pressure applied. To contrast, cells expressing a DXS without the inhibition trait would maintain a high flux until the concentration of IPP was too high and then would effectively shut off.

It is possible that because there was not a deliberate IPP/DMAPP sink in our base selection strain, that we unintentionally selected for the feedback inhibition trait in our DXS sequences. However, the fact that this trait was identified in chimeras consisting of parents that lack the inhibition trait is very interesting in and of itself and demonstrates the capability of recombination strategies to generate proteins with characteristics not seen, or weakly seen, in the parents. Future engineering efforts could utilize our selection method with an IPP/DMAPP sink (by adding an isoprene synthase for example)[5] that would enable higher fluxes through the pathway and potentially identify enzymes capable of sustaining those fluxes. Despite this, our selection strategy was successful in identifying enzymes that demonstrate high fitness *in vivo* and is a powerful method for identifying active enzymes in the MEP pathway, and the methodology could be easily modified to engineer other bottlenecks in the pathway as well.

4.3.5 ML modeling to understand sequence function landscape

Machine learning tools are useful for understanding key features of the sequence function landscape. We trained machine learning models on the enrichment data to attempt to determine key structural features for DXS activity. Models trained on only the sequence (in block representation) generally underpredicted DXS activity. Adding

interaction terms to the encoding greatly improved the model performance, suggesting that pairwise and higher order interactions between blocks of sequence could be playing a significant role in DXS fitness (Figure 4.7b). Interestingly, the model predictions for the top block and top pairwise interactions don't match up with the enrichment values (Figure 4.7a), but the PU learning model did predict a top sequence that was very similar to the actual top sequence (43331311 for PU learning, 43231311 for the actual best sequence). However, despite the closeness of the PU learning model to the top enrichment sequence, the two DXS sequences designed by optimizing the enrichment performed better *in vivo* and *in vitro* than the two DXSs that we designed using PU learning. PU learning works best for sparse matrices, but with so much enrichment data, using PU learning wasn't strictly necessary. The fact that there was a difference in the PU learning prediction and the sequence with the top enrichment suggests that the PU learning model underpredicted the performance of the top variants. It is also interesting to note that the two sequences designed based on their enrichment values had higher k_{cat}s than the two PU learning-designed sequences. Because of the amount of enrichment data available in this case, enrichment appears to be the better metric to use for designing sequences.

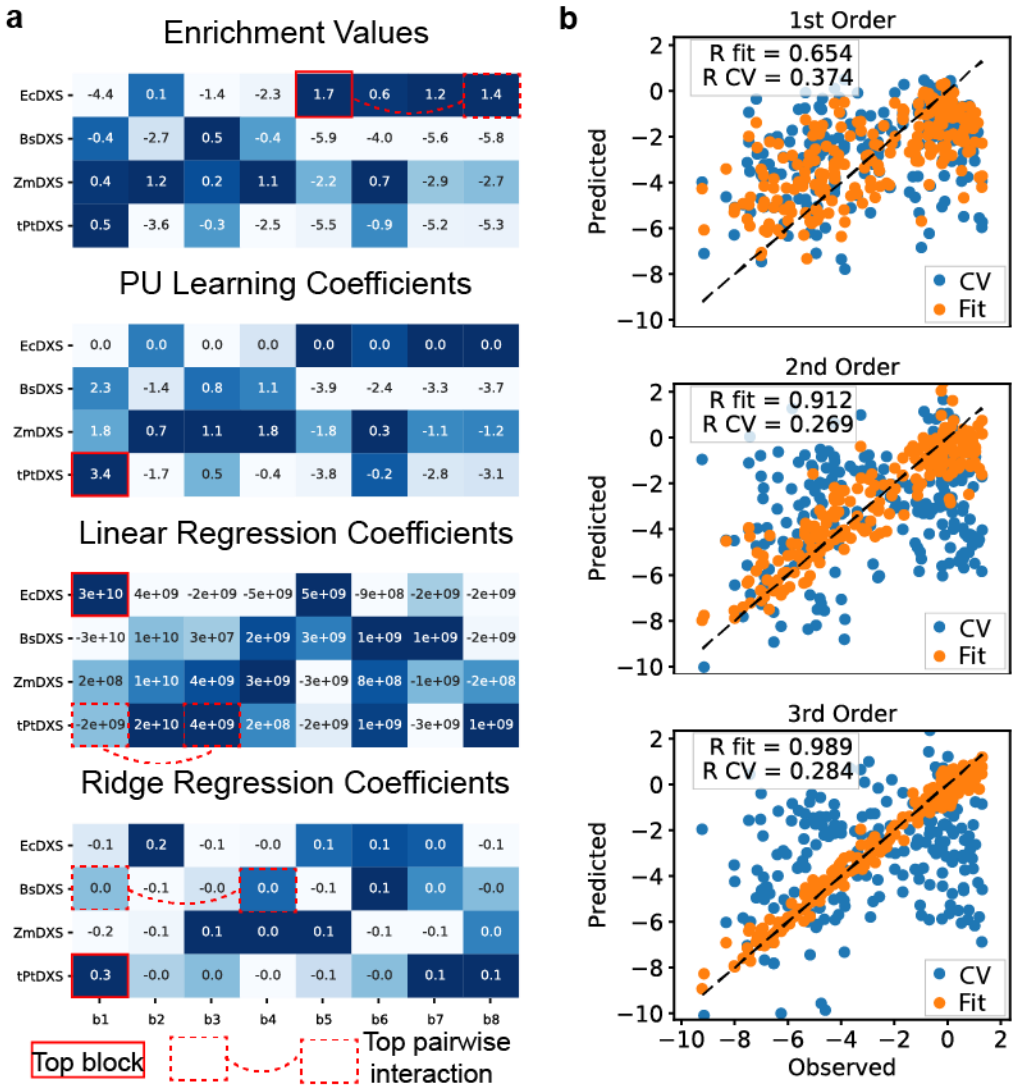

Figure 4.7: a) Heatmaps showing the site-wise enrichment values, PU learning coefficients, linear regression coefficients (3rd order), and ridge regression coefficients (3rd order). The top individual block predicted by each method is highlighted in solid red. The top pairwise interaction is highlight by dashed red lines with a connection between the two blocks. b) Ridge regression models that account for higher order interactions fit the data better but may also be slightly less generalizable based on their low cross validation scores.

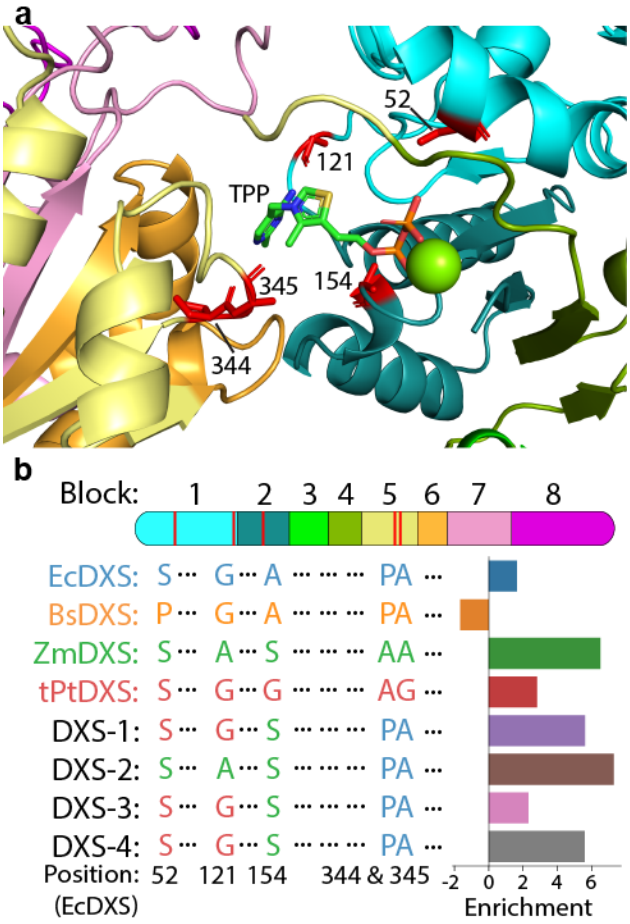

Figure 4.8: a) Structure of EcDXS zoomed in on the TPP binding pocket with non-conserved residues that contact TPP highlighted in red. b) Location of the positions in the 1D sequence and a small alignment showing which chimeras contain which residues. The enrichment values are shown on the side for reference.

4.3.6 Discussion of sequence factors which could impact activity and inhibition

Banerjee et al. described the engineering of PtDXS to reduce its inhibition by IPP[19]. We used a truncated form of the PtDXS double mutant from that study as one of our parental DXSs, and, in line with Banerjee et al.'s findings, we observed that it had low activity, and very low inhibition by IPP and DMAPP compared to other DXSs. EcDXS and ZmDXS had more inhibition than the other parents, while BsDXS had very low inhibition. Banerjee et al. point to a pair of specific alanine residues in PtDXS that play a role in

binding IPP, and in their engineered PtDXS these alanine residues were both mutated to glycine. However, EcDXS and BsDXS both have the native alanine's at these positions. ZmDXS has a serine residue in place of the first alanine. Differences at this position could influence how the enzymes are affected differently by inhibitors.

We used PyMOL to identify all residues in the crystal structure of EcDXS (PDB 2o1sA)[29] that were within 5 Å of the bound thiamine pyrophosphate (TPP) cofactor (Figure 4.8a). Of these 22 residues, all but five are conserved in the parental DXSs (and therefore in the chimeras as well). Four of the five non-conserved positions contain either a glycine or an alanine in at least one of the parents, and at three positions the difference between the parents is simply a swap of whether the residue is glycine or alanine. The findings of Banerjee et al. suggest that the activity and inhibition can be modified by modifying the non-polar residues involved in binding TPP, A154 and A345 (as numbered in EcDXS), so recombining the blocks containing these residues could change the activity. Interestingly, all four designed DXS variants contained serine residues at position 154 instead of alanine (Figure 4.8b). Likely, mutating this alanine to serine increases the fitness of DXSs in our selection, potentially by influencing how well the DXS binds and is feedback inhibited by IPP. The importance of A345 is emphasized by the fact that the block that contains it (block 5) is especially important when considering the site-wise enrichment values.

Block 5 from EcDXS was the most highly enriched block based on the enrichment analysis. It is notable that all four designed chimeras contain the sequence from EcDXS at this position. Additionally, block 5 contains a histidine (H299) residue that could have a role in catalysis[26], which could explain its importance for fitness, though the residue

itself is conserved in all the parents. Block 5 from EcDXS is also involved in an important pairwise interaction with block 8 from EcDXS, though the nature of the interaction is unclear. The combination of these two blocks is the highest site-wise enrichment score when pairwise interactions are considered. Interestingly, the machine learning models predictions of the top block and the top pairwise interaction don't match up (Figure 4.7a), which suggests that even the third order models trained on the enrichment data failed to capture important information about the fitness landscape.

4.4 Conclusions

In this work, we develop a pipeline for engineering DXS to have improved *in vivo* characteristics. The pipeline consists of generating a chimeric enzyme library, carrying out a high-throughput growth selection, and sequencing the library at different stages (before and after the selection) using long-read sequencing. While originally designed for engineering DXS, this pipeline could easily be modified to engineer either other bottleneck enzymes in the MEP pathway, such as IspG and IspH[15]. The nature of the selection could also enable combinatorial engineering of the entire MEP pathway, which would otherwise be much more difficult to study.

With our approach we identified chimeric DXS sequences that had significantly higher enrichment values than their parental enzymes. Surprisingly, we found that the improvement in enrichment did not correlate with enzymatic activity, improved substrate binding, or growth rate in selective conditions. We also observed no correlation between enrichment and protein expression levels. We further discovered that the chimeric DXSs that we designed based on enrichment values and PU learning were much more strongly

inhibited by IPP and DMAPP than the parental enzymes, suggesting that the selection conditions favored enzymes that could be regulated by feedback inhibition. Future experiments could leverage the findings of this work to make further improvements to DXS enzyme activity and MEP pathway flux.

4.5 Materials and methods

4.5.1 DXS library design

To design the library we first built a large multiple sequence alignment (MSA) by using jackhmmer[23] to align 375,776 DXS sequences from the Uniref database[30]. We then used a program called PSI-COV[24] to identify pairs of covarying residues. We used the information to build a contact map that could be used as an input for the SCHEMA/RASPP algorithm[31–33] along with an alignment of the sequences of DXSs from *E. coli*, *B. subtilis*, *Z. mobilis*, and *P. trichocarpa* (truncated) to design crossover points for chimeric DXSs.

4.5.2 Library construction and cloning

After designing the crossover points for the chimeric library, we ordered two sets of pre-cloned insert plasmids containing the each of the blocks designed by SCHEMA/RASPP from Twist Biosciences. One set contained blocks 1-4 (one plasmid for each parent enzyme), and the other set contained blocks 5-8 for each of the parental enzymes. Each block was flanked by BbsI restriction enzyme sites. The pre-cloned insert plasmids from each set were combined into one reaction and then a Golden Gate assembly was used to build two combinatorial sub libraries, one for blocks 1-4 and the other for blocks 5-8. The reaction conditions were 75 µg of the destination plasmid, 75 ng

of the each of the pre-cloned insert plasmids, 2.5 µL 10X T4 DNA ligase buffer (NEB), 1 µL or 5 µL (400 or 2000 units) T4 DNA ligase (NEB), and 1 µL (10 units) of BbsI-HF (NEB) to a total reaction volume of 25 µL in water. The reactions were carried out using a thermocycler, alternating between 37 ˚C and 16 ˚C (5 minutes each) for 30 cycles, followed by a final 5-minute step at 60 ˚C. The Golden Gate product was transformed into *E. coli* 10G classic electrocompetent cells from Lucigen (1 µL product into 25 µL of cells) and plated on LB + 50 µg/mL kanamycin agar plates. The final products were flanked by BsaI restriction enzyme sites on the destination plasmid.

We made the final 8 block library by digesting the sub library plasmids with BsaI and then ligating them together into a new destination vector (pET22_Ptrc_GG site TrrnB) with a 1:3 molar ratio of vector to sub library inserts. The ligation was carried out for 16 hours at 16 ˚C and 1 µL of the ligation product was transformed into 10 G classic competent cells and plated on LB + 100 µg/mL carbenicillin + 0.5% glucose agar plates.

4.5.3 High throughput DXS growth selection and sequencing

We miniprepped the DXS library and transformed the DNA into the strain ΔdxsMB[8] in ten parallel transformations by adding 50 ng of the miniprepped library into 50 µL of electrocompetent ΔdxsMB cells. We grew the cells at 37 ˚C for 1 hour in 1 mL of SOB + 1 mM mevalonolactone + 10 µg/mL tetracycline. We pooled these outgrowths and transferred the cells to a 200 mL culture of LB + 1 mM mevalonolactone + 10 µg/mL tetracycline + 100 µg/mL carbenicillin + 0.5% glucose and incubated it at 37 ˚C for 16 hours. We set aside 10 mL of the culture for the post selection and maxiprepped the DNA from the remaining preselection culture. Then we washed the 10 mL of culture that had

been set aside three times with LB and used it to inoculate a 200 mL culture of LB + 10 µg/mL tetracycline + 100 µg/mL carbenicillin + 100 µM IPTG at a starting density of OD = 0.05. We grew this culture for 5 hours at 37 ˚C to deplete the intercellular mevalonate, and then transferred it to a 200 mL culture of LB + 100 µM IPTG + 100 µg/mL carbenicillin. We maxiprepped the culture after 16 hours for sequencing. We linearized the DNA from both the preselection and post selection culturesby digesting with HpaI and submitted the linear DNA for sequencing using Oxford Nanopore Technology (ONT). We repeated this process in triplicate.

We also prepared a miniature library using the same workflow consisting of the four parental DXS enzymes and the four enzymes that we designed. We transformed 1 µL (50 ng) of each plasmid containing each DXS individually into 50 µL aliqhots of ΔdxsMB competent cells. We grew the cells at 37 ˚C for 1 hour in 1 mL of SOB + 1 mM mevalonolactone + 10 µg/mL tetracycline and then pooled 625 µL of each transformation outgrowth for a total of about 5 mL culture. We added this 5 mL of culture to an 100 mL culture of LB + 1 mM mevalonolactone + 10 µg/mL tetracycline + 0.5 % glucose and grew it for 16 hours at 37 ˚C. Afterwards we washed 5 mL of the preselection culture 3 times with LB and used it to inoculate a 100 mL culture of LB + 100 µM IPTG + 100 µg/mL carbenicillin (at a starting OD of 0.05). Then, we grew this culture for 16 hours, midiprepped the DNA, and digested it with HpaI prior to submission for sequencing with ONT.

4.5.4 Analysis of sequencing data

We converted the raw sequencing reads from the selection experiments to block based sequences. We did this by aligning each of the 24 sequence blocks to each of the reads. We set an alignment threshold, and if a block aligned to a sequencing read with a percent identity above this threshold, then we labelled that section of the sequence as the block in question. We only kept sequences that could be completely determined for downstream processing of the dataset.

We used the files containing the block-based sequences to calculate enrichment values for each of the sequences (sequence-wise) and for each of the blocks (site-wise). The enrichment (E) of a sequence is given by the following equation:

$$E = \log_2\left(\frac{f_{post\ selection}}{f_{pre\ selection}}\right) \tag{4.1}$$

where *f* corresponds to the frequency the sequence or block appears in the pre or post selection dataset respectively.

Once we calculated enrichment values, we used the enrichment values to train positive-unlabeled (PU) learning models and design chimeric DXS sequences that would have high fitness. PU learning models were trained using the methods described in Song et al.[25] We designed four chimeric DXS sequences. DXS-1 contained the combination of blocks with the highest site-wise enrichment scores. DXS-2 was the sequences with the highest sequence-wise enrichment score. DXS-3 and DXS-4 maximized the PU learning scores; DXS-3 had the highest main effects mean and DXS-4 had the highest pairwise effects mean.

4.5.5 Growth validation of DXS chimeras

We transformed each of the parental and designed DXS enzymes and an empty vector control into individual aliquots of ΔdxsMB electrocompetent cells per methods described above. The outgrowth was plated on LB + 1 mM mevalonolactone + 100 µg/mL carbenicillin + 10 µg/mL tetracycline agar plates. We selected individual colonies from the plates and used them to inoculate 5 mL cultures of LB + 1 mM mevalonolactone + 10 µg/mL tetracycline + 100 µg/mL carbenicillin + 0.5% glucose, which we grew for about 16 hours overnight. We then washed 1 mL from each culture 3 times and started 3 mL cultures of LB + 10 µg/mL tetracycline + 100 µg/mL carbenicillin + 100 µM IPTG with an initial OD of 0.05. We grew these cells for 5 hours and then inoculated new cultures in wells of a 96-well microtiter plate such that the starting OD was 0.012 in 100 µL LB + 10 µg/mL tetracycline + 100 µg/mL carbenicillin + 100 µM IPTG. We monitored the OD continuously for 20-24 hours using a Tecan Spark plate reader.

4.5.6 Estimation of DXS expression level

We transformed plasmids (Ptrc-TrrnB backbone) into ΔdxsMB cells and started 5 mL cultures from individual colonies in LB + 1 mM mevalonolactone + 100 µg/mL carbenicillin + 10 µg/mL tetracycline. We grew these cultures for 15 hours, and then washed the cells three times with 5 mL of LB. We then used the washed cells to inoculate 50 mL cultures of LB + 100 µg/mL carbenicillin + 10 µg/mL tetracycline + 100 µM IPTG with a starting OD of 0.05. We grew these cultures for about 5 hours at 37 ˚C, and then passaged them to start new 50 mL cultures with an initial OD of 0.05. We grew these cultures until the OD was between 0.4 and 0.5 and harvested the cells.

To analyze the expression levels, we resuspended the cells and adjusted the volumes such that the ODs would be uniform, and then added a fixed amount of cells to Eppendorf tubes and pelleted them. Then we resuspended the cells in 120 µL of 100 mM Tris pH 7.4 and lysed using 120 µL of lysis buffer (5 µL DNAse I, 5 µL rLysozyme, 150 µL BugBuster, 2,340 µL 100 mM Tris pH 7.4). We incubated the mixtures at room temperature for about a half hour and then collected the total fraction for SDS-PAGE analysis. Finally, we collected the soluble fraction after we clarified the lysate by centrifugation, and analyzed the total and soluble lysate fractions by SDS-PAGE.

4.5.7 DXS and DXS expression and purification

To express the DXS enzymes of interest for purification, we subcloned the DXS sequences from the Ptrc-TrrnB backbone into pET 22 and transformed them into BL21 (DE3) competent cells. We grew either 5 or 50 mL cultures for 16-18 hours overnight from individual colonies in LB + 100 µg/mL carbenicillin and the following day diluted the cultures 100 fold into larger culture volume (ranging from 50 mL to 1 L in scale). We grew these cultures until the OD was between 0.4 and 0.6 (usually 2.5 to 3 hours) and then induced protein expression by adding IPTG to a final concentration of 100 µM. Then we incubated the cells at 20 ˚C overnight, for about 20 to 24 hours and harvested the cells.

To express DXR we started a 50 mL culture in LB + carbenicillin from a glycerol stock and grew it for about 18 hours. The following day, we used 40 mL of the culture to inoculate two 1 L cultures in a 4 L baffled shake flask (20 mL culture to each). We grew these until the OD was about 0.4 (1.75 hours) and added IPTG to a final concentration of

100 μM. We then grew the cells at 20 ˚C for about 20 hours and harvested the cells. We kept harvested cell pellets at -80 ˚C until we were ready to purify the proteins.

We resuspended DXS and DXR cell pellets using 50 mM NAH_2PO_4, 300 mM NaCl, 10 mM imidazole, 10% glycerol (v/v) pH 8, and lysed the cells by sonication. In some cases we added hen egg lysozyme and DNAse as well. We equilibrated gravity flow columns with Nickel Sepharose Fast Flow Resin (GE healthcare) with 3 column volumes (about 15 mL each) of water and 3 column volumes of binding buffer (20 mM NaH_2PO_4, 500 mM NaCl, 20 mM imidazole pH 7.4). We loaded the clarified lysates and collected the flow through, and then washed the column with two column volumes of binding buffer. Then we eluted with 5- 10 mL of 20 mM NaH_2PO_4, 500 mM NaCl, 500 mM imidazole pH 7.4. We verified the success of the protein purifications by running SDS PAGE gels. Following the elution, we concentrated the protein using Amicon spin columns (30000 MWCO) and diluted the protein in 100 mM Tris-HCl pH 7.4. Last, glycerol was added in a roughly equal volume to the Tris-HCl buffer to make the stocks roughly 50% glycerol. Stocks were frozen and stored at -80 ˚C until use. We measured DXS and DXR concentrations by Bradford assay (using a Tecan Spark plate reader and Biorad Protein Assay Reagent), and sometimes by measuring the absorbance at 280 nM (using a Nanodrop).

4.5.8 Coupled enzyme assays for kinetic assays

To assess the activity of DXSs in various conditions we used a coupled enzyme assay. We made a master mix of DXR, NADPH, $MgCl_2$ and DTT in 100 mM Tris-HCl pH 7.4, which we then mixed with solutions of DXS (normalized to a concentration of 1.48

μM). Mixtures of the one of the substrates (GAP and pyruvate) and the cofactor TPP were prepared, and then several dilutions of the other substrate were added to this mixture. Finally, 10 μL of the substrate mixture was mixed with 90 μL of the master mix + DXS mixture in wells of 96-well microtiter plates and the absorbance at 340 nm was observed for 30-60 minutes. To study the effects of inhibition, the same procedure as above was used, except that inhibitor (IPP or DMAPP) was added to the substrate mixture and rather than using a dilution series of GAP or pyruvate we used a dilution series of TPP. We calculated the initial reaction rates and fit the results to the Michaelis-Menten equation to obtain estimates of the K_M and k_{cat} for each enzyme.

4.5.9 Machine learning modeling for data analysis

To use machine learning to understand sequence features, we attempted to train a variety of models on enrichment data. We pre-pocessed the sequencing data by filtering out sequences with fewer than ten reads in either the pre- or post-selection data sets and then calculated enrichment as was done in section 4.5.4. We trained linear and ridge regression models on the datasets using different numbers of interaction terms (1st, 2nd and 3rd order interactions) and calculated the Pearson's R value for the fit and for the cross validated predictions. After training the models, we analyzed the regression coefficients to identify the top predicted effects on activity.

4.6 Supplementary information

Table 4.2: Amino acid sequences of DXS blocks

DXS	Block 1
Ec	MSFDIAKYPTLALVDSTQELRLLPKESLPKLCDELRRYLLDSVSRSSGHFASGLGTVELTVALHYVYNTPFDQLIWDVGHQ AYPHKILTGRRDKIGTIRQKGGLHPFPWRGESEYDVLSVGH
Bs	------- MLDLLSIQDPSFLKNMSIDELEKLSDEIRQFLITSLSASGGHIGPNLGVVELTVALHKEFNSPKDKFLWDVGHQSYVHKLL TGRGKEFATLRQYKGLCGFPKRSESEHDVWETGH
Zm	- MFPNDKTPLLDKIKTPAELRQLDRNSLRQLADELRKETISAVGVTGGHLGSGLGVIELTVALHYVFNTPKDALVWDVGH QTYPHKILTGRRDRIRTLRQRDGLSGFTQRAESEYDAFGAAH
Pt	-------- MPLLDTINYPIHMKNLSVKELKQLADELRSDVIFNVSKTGGHLGSSLGVVELTVALHYVFNAPQDKILWDVGHQSYPHK ILTGRRDKMHTIRQTNGLAGFTKRSESEYDCFGTGH

	Block 2
Ec	SSTSISAGIGIAVAAEKEGKNRRTVCVIGDGAITAGMAFEAMNHAGDIRPDMLVILNDN-EMSIS--------ENV
Bs	SSTSLSGAMGMAAARDIKGTDEYIIPIIGDGALTGGMALEALNHIGDEKKDMIVILNDN-EMSIA--------PNV
Zm	SSTSISAALGFAMASKLSDSDDKAVAIIGDGSMTAGMAYEAMNNAKAAGKRLIVILNDN-EMSIS--------PPV
Pt	SSTTISAGLGMAVGRDLKGGTNKVVAVIGDGGMTAGQAYEAMNNAGYLDSDMIVILNDNKQVSLPTANLDGPIPPV

	Block 3
Ec	GALNNHLAQLLSGKLYSSLREGGKKVFSGVP----PIKELLKRTEEHIKGMV--VPGTLF
Bs	GAIHSMLGRLRTAGKYQWVKDELEYLFKKIPAVGGKLAATAERVKDSLKYML--VSGMFF
Zm	GALSSYLSRLISSRPFMNLRDIMRGVVNRMP---KGLATAARKADEYARGMA--TGGTFF
Pt	GALSSALSRLQSNRPLRELREVAKGVTKQIG---GPMHELAAKVDEYARGMISGSGSTLF

	Block 4
Ec	EELGFNYIGPVDGHDVLGLITTLKNMRDLK--GPQFLHIMTKKGRG
Bs	EELGFTYLGPVDGHSYHELIENLQYAKKTK--GPVLLHVITKKGKG
Zm	EELGFYYVGPVDGHNLDQLIPVLENVRDAK-DGPILVHVVTRKGQG
Pt	EELGLYYIGPVDGHNIDDLIAILKEVKSTKTTGPVLIHVVTEKGRG

	Block 5
Ec	YEPAEKDPI-TFHAVPKFDPSSGCLPKSSGGLPSYSKIFGDWLCETAAKDNKLMAITPAMREGSGMVEFSRKFPDRYF
Bs	YKPAETDTIGTWHGTGPYKINTGDFVKPKAAAPSWSGLVSGTVQRMAREDGRIVAITPAMPVGSKLEGFAKEFPDRM F
Zm	YAPAEAAKD-KYHAVQRLDVVSGKQAKAPPGPPSYTSVFSEQLIKEAKQDDKIVTITAAMPTGTGLDRFQQYFPERMF
Pt	YPFAERAAD-KYHGVAKFDPATGKQFKASPSTQSYTTYFAEALTAEAEADKDIVAIHAGMGGGTGLNLFLRRFPTRC

	Block 6
Ec	DVAIAEQHAVTFAAGLAIGGYKPIVAIYSTFLQRAYDQVLH
Bs	DVGIAEQHAATMAAAMAMQGMKPFLAIYSTFLQRAYDQVVH
Zm	DVGIAEQHAVTFAAGLAAAGYKPFCCLYSTFLQRGYDQLVH
Pt	DVGIAEQHAVTFAAGLACEGLKPFCAIYSSFLQRAYDQVVH

Block 7

Ec	DVAIQKLPVLFAIDRAGIVGADGQTHQGAFDLSYLRCIPEMVIMTPSDENECRQMLYTGYHYNDGPSAVRYPRGNAV GVELTP-LEKLP--I
Bs	DICRQNANVFIGIDRAGLVGADGETHQGVFDIAFMRHIPNMVLMMPKDENEGQHMVHTALSYDEGPIAMRFPRGN GLGVKMDEQLKTIP--I
Zm	DVAIQNLPVRFAVDRAGLVGADGATHAGSFDLAFMVNLPNMVVMAPSDERELANMVHSMAHYDQGPISVRYPRG NGVGVSLEGEKEILP--I
Pt	DVDLQKLPVRFAMDRAGLVGADGPTHCGAFDVTFMACLPNMVVMAPSDEAELFHMVATATAIDDRPSCFRYPRGN GVGVQLPPGNKGVPLEV

Block 8

Ec	GKGIVKRRGEKLAILNFGTLMPEAAKVAESLNA----- TLVDMRFVKPLDEALILEMAASHEALVTVEENAIMGGAGSGVNEVLMAH--- RKPVPVLNIGLPDFFIPQGTQEEMRAELGLDAAGMEAKIKAWLA----------
Bs	GTWEVLRPGNDAVILTFGTTIEMAIEAAEELQKEGLSVRVVNARFIKPIDEKMMKSILKEGLPILTIEEAVLEGGFGSSILEF AHDQG--EYHTPIDRMGIPDRFIEHGSVTALLEEIGLTKQQVANRIRLLMP--PKTHKGIGS---------
Zm	GKGRLIRRGKKVAILSLGTRLEESLKAADRLDAQGLSTSVADMRFAKPLDEALTRQLLKSHQVIITIEEGAL- GGFATQVLTMASDEGLMDDGLKIRTLRLPDRFQPQDKQERQYAEAGLDADGIVAAVISALHRNSKPVEVVEMANMG SIARA
Pt	GKGRMLIEGERVALLGYGTAVQSCLAAASLVERHGIRLTVADARFCKPLDHALIRSLAKSHEILITVEEGSI- GGFGSHVVQFLALDGLLDGKLKWRPVVLPDRYIDHGSPADQLVEAGLTPSHIAATVFSILGQRREALEIMSS---------

References:

(1) Grisewood, M. J.; Hernández-Lozada, N. J.; Thoden, J. B.; Gifford, N. P.; Mendez-Perez, D.; Schoenberger, H. A.; Allan, M. F.; Floy, M. E.; Lai, R. Y.; Holden, H. M.; et al. Computational Redesign of Acyl-ACP Thioesterase with Improved Selectivity toward Medium-Chain-Length Fatty Acids. *ACS Catal.* **2017**, *7* (6), 3837–3849. https://doi.org/10.1021/acscatal.7b00408.

(2) Miller, D. C.; Athavale, S. V.; Arnold, F. H. Combining Chemistry and Protein Engineering for New-to-Nature Biocatalysis. *Nat. Synth. 2022 11* **2022**, *1* (1), 18–23. https://doi.org/10.1038/s44160-021-00008-x.

(3) Li, C.; Zhang, R.; Wang, J.; Wilson, L. M.; Yan, Y. Protein Engineering for Improving and Diversifying Natural Product Biosynthesis. *Trends Biotechnol.* **2020**, *38* (7), 729–744. https://doi.org/10.1016/J.TIBTECH.2019.12.008.

(4) Kim, S. W.; Keasling, J. D. Metabolic Engineering of the Nonmevalonate Isopentenyl Diphosphate Synthesis Pathway in Escherichia Coli Enhances Lycopene Production. *Biotechnol. Bioeng.* **2001**, *72* (4), 408–415. https://doi.org/10.1002/1097-0290(20000220)72:4<408::AID-BIT1003>3.0.CO;2-H.

(5) Zhao, Y.; Yang, J.; Qin, B.; Li, Y.; Sun, Y.; Su, S.; Xian, M. Biosynthesis of Isoprene in Escherichia Coli via Methylerythritol Phosphate (MEP) Pathway. *Appl. Microbiol. Biotechnol.* **2011**, *90* (6), 1915–1922. https://doi.org/10.1007/S00253-011-3199-1/FIGURES/6.

(6) Klein-Marcuschamer, D.; Ajikumar, P. K.; Stephanopoulos, G. Engineering Microbial Cell Factories for Biosynthesis of Isoprenoid Molecules: Beyond

Lycopene. *Trends Biotechnol.* **2007**, *25* (9), 417–424. https://doi.org/10.1016/J.TIBTECH.2007.07.006.

(7) Li, Z.; Sharkey, T. D. Metabolic Profiling of the Methylerythritol Phosphate Pathway Reveals the Source of Post-Illumination Isoprene Burst from Leaves. *Plant. Cell Environ.* **2013**, *36* (2), 429–437. https://doi.org/10.1111/J.1365-3040.2012.02584.X.

(8) Martin, V. J. J.; Piteral, D. J.; Withers, S. T.; Newman, J. D.; Keasling, J. D. Engineering a Mevalonate Pathway in Escherichia Coli for Production of Terpenoids. *Nat. Biotechnol. 2003 217* **2003**, *21* (7), 796–802. https://doi.org/10.1038/nbt833.

(9) Paddon, C. J.; Keasling, J. D. Semi-Synthetic Artemisinin: A Model for the Use of Synthetic Biology in Pharmaceutical Development. **2014**. https://doi.org/10.1038/nrmicro3240.

(10) Dhingra, V.; Vishweshwar Rao, K.; Lakshmi Narasu, M. Current Status of Artemisinin and Its Derivatives as Antimalarial Drugs. *Life Sci.* **1999**, *66* (4), 279–300. https://doi.org/10.1016/S0024-3205(99)00356-2.

(11) Danishefsky, S. J.; Masters, J. J.; Young, W. B.; Link, J. T.; Snyder, L. B.; Magee, T. V; Jung, D. K.; Isaacs, R. C. A.; Bornmann, W. G.; Alaimo, C. A.; et al. *Total Synthesis of Baccatin III and Taxol*; 1996.

(12) Peralta-Yahya, Pamela P., Zhang, Fuzhong, del Cardayre, Stephen B., Keasling, J. D. Microbial Engineering for the Production of Advanced Biofuels. *Nature* **2012**, *488*. https://doi.org/10.1038/nature11478.

(13) Liao, J. C.; Mi, L.; Pontrelli, S.; Luo, S. Fuelling the Future: Microbial Engineering

for the Production of Sustainable Biofuels. *Nat. Rev. Microbiol.* **2016**, *14* (5), 288–304. https://doi.org/10.1038/nrmicro.2016.32.

(14) Straathof, A. J. J. Transformation of Biomass into Commodity Chemicals Using Enzymes or Cells. *Chem. Rev.* **2014**, *114* (3), 1871–1908. https://doi.org/10.1021/cr400309c.

(15) Volke, D. C.; Rohwer, J.; Fischer, R.; Jennewein, S. Investigation of the Methylerythritol 4-Phosphate Pathway for Microbial Terpenoid Production through Metabolic Control Analysis. *Microb. Cell Fact.* **2019**, *18* (1), 1–15. https://doi.org/10.1186/S12934-019-1235-5/FIGURES/8.

(16) Wang, Q.; Quan, S.; Xiao, H. Towards Efficient Terpenoid Biosynthesis: Manipulating IPP and DMAPP Supply. *Bioresour. Bioprocess. 2019 61* **2019**, *6* (1), 1–13. https://doi.org/10.1186/S40643-019-0242-Z.

(17) Estévez, J. M.; Cantero, A.; Reindl, A.; Reichler, S.; León, P. 1-Deoxy-d-Xylulose-5-Phosphate Synthase, a Limiting Enzyme for Plastidic Isoprenoid Biosynthesis in Plants. *J. Biol. Chem.* **2001**, *276* (25), 22901–22909. https://doi.org/10.1074/JBC.M100854200.

(18) Kim, S. J.; Kim, M. D.; Choi, J. H.; Kim, S. Y.; Ryu, Y. W.; Seo, J. H. Amplification of 1-Deoxy-d-Xyluose 5-Phosphate (DXP) Synthase Level Increases Coenzyme Q10 Production in Recombinant Escherichia Coli. *Appl. Microbiol. Biotechnol.* **2006**, *72* (5), 982–985. https://doi.org/10.1007/S00253-006-0359-9/FIGURES/1.

(19) Banerjee, A.; Preiser, A. L.; Sharkey, T. D. Engineering of Recombinant Poplar Deoxy-D-Xylulose-5-Phosphate Synthase (PtDXS) by Site-Directed Mutagenesis Improves Its Activity. *PLoS One* **2016**, *11* (8).

https://doi.org/10.1371/journal.pone.0161534.

(20) Kim, M. J.; Noh, M. H.; Woo, S.; Lim, H. G.; Jung, G. Y. Enhanced Lycopene Production in Escherichia Coli by Expression of Two MEP Pathway Enzymes from Vibrio Sp. Dhg. *Catalysts* **2019**, *9* (12), 1–12. https://doi.org/10.3390/catal9121003.

(21) Alper, H.; Miyaoku, K.; Stephanopoulos, G. Construction of Lycopene-Overproducing E. Coli Strains by Combining Systematic and Combinatorial Gene Knockout Targets. *Nat. Biotechnol.* **2005**, *23* (5), 612–616. https://doi.org/10.1038/nbt1083.

(22) Leonard, E.; Ajikumar, P. K.; Thayer, K.; Xiao, W.-H.; Mo, J. D.; Tidor, B.; Stephanopoulos, G.; Prather, K. L. J. Combining Metabolic and Protein Engineering of a Terpenoid Biosynthetic Pathway for Overproduction and Selectivity Control. *Proc. Natl. Acad. Sci.* **2010**, *107* (31), 13654–13659. https://doi.org/10.1073/pnas.1006138107.

(23) Potter, S. C.; Luciani, A.; Eddy, S. R.; Park, Y.; Lopez, R.; Finn, R. D. HMMER Web Server: 2018 Update. *Web Serv. issue Publ. online* **2018**, *46*. https://doi.org/10.1093/nar/gky448.

(24) Jones, D. T.; Buchan, D. W. A.; Cozzetto, D.; Pontil, M. PSICOV: Precise Structural Contact Prediction Using Sparse Inverse Covariance Estimation on Large Multiple Sequence Alignments. *Bioinformatics* **2012**, *28* (2), 184–190. https://doi.org/10.1093/BIOINFORMATICS/BTR638.

(25) Song, H.; Bremer, B. J.; Hinds, E. C.; Raskutti, G.; Romero, P. A. Inferring Protein Sequence-Function Relationships with Large-Scale Positive-Unlabeled Learning.

Cell Syst. **2021**, *12* (1), 92-101.e8. https://doi.org/10.1016/j.cels.2020.10.007.

(26) White, J. K.; Handa, S.; Vankayala, S. L.; Merkler, D. J.; Woodcock, H. L. Thiamin Diphosphate Activation in 1-Deoxy- d -Xylulose 5-Phosphate Synthase: Insights into the Mechanism and Underlying Intermolecular Interactions. *J. Phys. Chem. B* **2016**, *120* (37), 9922–9934. https://doi.org/10.1021/ACS.JPCB.6B07248/SUPPL_FILE/JP6B07248_SI_001.PDF.

(27) George, K. W.; Thompson, M. G.; Kim, J.; Baidoo, E. E. K.; Wang, G.; Benites, V. T.; Petzold, C. J.; Chan, L. J. G.; Yilmaz, S.; Turhanen, P.; et al. Integrated Analysis of Isopentenyl Pyrophosphate (IPP) Toxicity in Isoprenoid-Producing Escherichia Coli. *Metab. Eng.* **2018**, *47*, 60–72. https://doi.org/10.1016/J.YMBEN.2018.03.004.

(28) Sivy, T. L.; Fall, R.; Rosenstiel, T. N. Evidence of Isoprenoid Precursor Toxicity in Bacillus Subtilis. *Biosci. Biotechnol. Biochem.* **2011**, *75* (12), 2376–2383. https://doi.org/10.1271/bbb.110572.

(29) Xiang, S.; Usunow, G.; Lange, G.; Busch, M.; Tong, L. Crystal Structure of 1-Deoxy-d-Xylulose 5-Phosphate Synthase, a Crucial Enzyme for Isoprenoids Biosynthesis. *J. Biol. Chem.* **2007**, *282* (4), 2676–2682. https://doi.org/10.1074/JBC.M610235200.

(30) Suzek, B. E.; Wang, Y.; Huang, H.; Mcgarvey, P. B.; Wu, C. H.; Consortium, U. UniRef Clusters: A Comprehensive and Scalable Alternative for Improving Sequence Similarity Searches. https://doi.org/10.1093/bioinformatics/btu739.

(31) Voigt, C. A.; Martinez, C.; Wang, Z. G.; Mayo, S. L.; Arnold, F. H. Protein Building

Blocks Preserved by Recombination. *Nat. Struct. Biol.* **2002**, *9* (7), 553–558. https://doi.org/10.1038/nsb805.

(32) Silberg, J. J.; Endelman, J. B.; Arnold, F. H. SCHEMA-Guided Protein Recombination. *Methods Enzymol.* **2004**, *388* (2003), 35–42. https://doi.org/10.1016/S0076-6879(04)88004-2.

(33) Endelman, J. B.; Silberg, J. J.; Wang, Z.; Arnold, F. H. Site-Directed Protein Recombination as a Shortest-Path Problem. *Protein Eng. Des. Sel.* **2004**, *17* (7), 589–594. https://doi.org/10.1093/protein/gzh067.

Chapter 5

Conclusions and future directions

Author: Jonathan C. Greenhalgh

5.1 Summary of dissertation research

Machine learning guided protein engineering is a rapidly expanding discipline and is being used to solve problems in many fields, ranging from healthcare to chemical production to environmental issues. Over the course of this dissertation, we reviewed the history and outlook of this field and applied the strategies in the engineering of enzymes two different systems.

In Chapter 1, we reviewed the field of protein engineering as it relates to solving problems in chemical production. Directed evolution and rational design are historically the most common protein engineering techniques, but machine learning approaches are broadening the scope of how these techniques are used and making it possible to engineer proteins that were previously too difficult or cost intensive to engineer otherwise. Advances in machine learning-guided protein engineering will continue to accelerate protein engineering discoveries and applications in all fields, and these advances can be used in tandem with other metabolic engineering approaches to develop sustainable production routes for valuable chemical products.

Machine learning-guided protein engineering is a relatively young, but very rapidly expanding field. In Chapter 2[1], we examined different strategies for encoding or representing the information contained in proteins (1-D sequence or 3-D structure for example) and how machine learning can be used to not only learn from these different encodings, but also learn new ways to encode information about proteins. We also described how different kinds of machine learning strategies (supervised, unsupervised and semi-supervised) can be used in realizing various protein engineering objectives. Finally, we reviewed key applications of machine learning methods in protein engineering

efforts and described the outlook of the field.

Improving fatty alcohol production in cells is a very promising application of metabolic engineering guided by machine learning. Fatty alcohols are very valuable bioproducts that are used extensively in surfactants and detergents, as well as many other commercial applications[2,3]. Due to environmental shortcomings associated with industrial production of fatty alcohols, many strategies have been employed to enhance microbial production of fatty alcohols. In Chapter 3, we used an iterative, active-learning approach to engineer acyl-CoA and acyl-ACP reductases (ACRs and AARs respectively) to improve their activity on acyl-ACP substrates and enhance the production of fatty alcohols *in vivo*[4]. We used Gaussian process regression coupled with upper confidence bound optimization to design sets of chimeric acyl-ACP reductase enzymes that helped us search the sequence-function landscape and optimize the activity of the enzymes simultaneously. Using this approach, we identified a chimeric enzyme that had a nearly threefold higher titer *in vivo* than the best wild-type sequence. We used *in vitro* kinetics assays to verify that the improvement in fatty alcohol production stemmed from an improvement in the catalytic efficiency of the enzyme, and we used machine learning models to help identify key features that enabled the increase in activity. Among these features were a specific set of sequence blocks that resided near the ACP binding interface, and docking simulations showed that optimizing the sequence blocks in these positions resulted in optimal numbers of positively charged amino acid residues that could help facilitate binding of acyl-ACPs.

Another important class of molecule that can be made in microbes is terpenoids. Terpenoids (also known as isoprenoids) are valuable natural products that are built up of

isoprene subunits. They can form very large and complex molecules that have many applications, ranging from medicines[5] to biomaterials[6,7] to advanced fuels[8]. There are two main pathways to making terpenoids; the mevalonate pathway, which is widely used in eukaryotes[9], and the MEP pathway, which is more common in microbes (although some plants use both pathways)[10]. One of the rate-limiting enzymes in the MEP pathway is 1-Deoxy-D-xylulose 5-phosphate (DXS). In Chapter 4, we developed a high throughput selection that coupled growth to DXS functionality, which could lead to downstream increase in terpenoid production. We then used it to screen a combinatorial library of DXS chimeras and generate a large sequence-function dataset. We used positive-unlabeled (PU) learning and enrichment data to design four chimeric DXS sequences. These four designed DXS sequences were predicted to have very high activity based on their high enrichment scores. However, we found that in non-pooled growth assays that there was not an observable difference between the growth rates of *E. coli* cells expressing the chimeric DXSs compared to the wild-type sequences when we grew the cells in selective conditions. Additionally, when we tested the four designed enzymes and the four parental enzymes *in vitro* we found that the enzymes had comparable turnover numbers (k_{cat}) and Michaelis-Menten constants (K_M). Interestingly however, we did find that our designed DXS variants displayed a significantly stronger reaction to inhibitors (IPP and DMAPP) than the wild-type enzymes when we performed inhibition studies *in vitro*. It is likely that stronger feedback inhibition from IPP was advantageous when selective pressure was applied because it could enable the DXS to work like a valve that could turn down the flux of the pathway when IPP levels are too high (preventing a toxic buildup), but then allow the flux levels to increase once the IPP level drops again. Finally, we used machine

learning methods and structural models to analyze the dataset and determine potential explanations for the inhibition effects that we observed, such as the role of non-polar residues in the TPP binding pocket.

5.2 Future directions

5.2.1 Machine learning for acyl-ACP and acyl-CoA reductase engineering

While our results in Chapter 3 were very exciting in terms of the specific goal that we were aiming to achieve (improving the activity of an acyl-CoA reductase on acyl-ACP), the practicality of the pathway that we used to make fatty alcohols is still quite low compared to other metabolic strategies. The highest titers we observed in our pathway were about 60 mg/L, but alternative routes, even though they theoretically require more cellular energy, can produce closer to 1.8 g/L of fatty alcohol[11,12]. However, there are still applications in which our engineered acyl-ACP reductase (ATR-83) could prove useful. Metabolic engineering approaches, coupled with protein engineering strategies, could be used to design improved strains of *E. coli* that produce more flux through fatty acid biosynthesis, and increasing the amount of acyl-ACP in the cell would make ATR-83 a much more viable enzyme for high-titer production of alcohols.

Additionally, the methodology we employed to design ATR-83 is very useful and applicable to efforts to engineering acyl-CoA/acyl-ACP reductases. The iterative UCB optimization-based strategy that we used to design each experimental round could theoretically be employed for any enzyme[4,13], and this strategy could easily be employed to rapidly design custom acyl-CoA/acyl-ACP reductases for optimizing specific pathways. Furthermore, chain length specificity of acyl-CoA reductases remains an unsolved

problem[3,14]. It is strongly believed that acyl-CoA reductases dictate the final chain-length distribution and similar machine learning-guided approaches could be used to identify enzymes that produce specific chain-length alcohols. Advances in machine learning-based protein structure prediction (such as AlphaFold[15,16]) enable rational engineering of ACRs and AARs to limit space in the active site pocket, favoring production of short and medium chain fatty alcohols, similar to work that was done with thioesterases[17]. Enabling higher throughput assays of ACR and AAR function, such as developing an analogous selection to that done by Hernández-Lozada et al.[18], would also rapidly accelerate attempts to engineer specific AARs and ACRs.

5.2.2 Using machine learning-guided enzyme engineering to enhance MEP pathway flux

Our results in Chapter 4 align with the aphorism commonly applied to biological engineering projects, “you get what you select for”. We designed the selection to link DXS function to growth, which it did. However, the selection also favored enzymes that displayed inhibition in the presence of IPP, potentially because the strain that we used for the selection did not contain a downstream sink to consume excess IPP or DMAPP,. It is very possible that altering the selection conditions somewhat could lead to discovery of enzymes with even higher activity. One strategy that could be used to change the selection conditions to be less biased towards selecting for inhibition would be to add a downstream pathway or enzyme to consume IPP. One option would be to add the pathway to produce the carotenoid lycopene, which has the advantage of doubling as a qualitative visual readout of activity or flux. However, because lycopene production can result in oxidative stress, it is not always quantitatively correlated with high flux. Another

option would be to add an isoprene synthase. The isoprene synthase would convert IPP to isoprene gas, which would then exit the cellular system[8]. This would be a very advantageous way to pull flux through the MEP pathway and impose a selective pressure for enabling higher flux. Capturing isoprene to quantify flux would be technically challenging (due to the nature of quantifying gases), but possible, and potentially more reliable than colorimetric strategies dependent on carotenoid production[19].

Despite potential flaws in the design of the selection itself, the workflow that we used in this experiment could be applied to engineering other bottlenecks in the MEP pathway such as DXR, IspG, and IspH[20]. The mevalonate supplementation strategy could be used to sustain growth in knockout strains of either of these enzymes, and a similar chimeric library selection could be performed to determine functionality or flux. The overall strategy of building a chimeric library, subjecting it to a high-throughput selection, and using long-read sequencing (Oxford Nanopore) to identify the chimeras, is also highly generalizable and could be used to engineer many other kinds of proteins whose function can be linked to growth via a selection strategy.

5.3 Accelerating protein and metabolic engineering with machine learning

Biological sciences are currently experiencing a revolutionary phase of growth due to advances in DNA sequencing and DNA synthesis. These advances are resulting in an acceleration in the acquisition of biological data. Machine learning plays a crucial role in managing this explosion of biological data. For example, the number of solved protein structures has always lagged far behind the number of known protein sequences. However, advances in machine learning have let to AlphaFold[15,16], which is expected to

revolutionize protein structure prediction and bridge the gap between solved structures and known sequences. This massive growth in biological data makes it possible to use machine learning models in new ways, such as using unsupervised models to classify proteins or generating information rich encodings of protein sequences based on evolutionary information. Advances in screening technologies will further increase the number of labeled datasets that can be used to engineer proteins.

To conclude this dissertation, I will briefly outline examples of how advances in machine learning based protein engineering have accelerated and continue to accelerate protein and metabolic engineering.

5.3.1 Machine learning-assisted directed evolution

Machine learning dovetails naturally with directed evolution. Directed evolution is normally limited in its requirement for large protein libraries, but machine learning strategies can be used to design intelligent libraries[21–23] that reduce the screening burden. Recently, machine learning-assisted directed evolution has been used to engineer nitric oxide deoxygenase enzymes performs carbene Si-H insertion in an enantioselective manner[23], improve the transpeptidase activity of Sortase A[24], and even engineering a PETase to be more thermostable for degrading plastics[25].

In particular, adaptive machine learning strategies work especially well with directed evolution[26], and have been used to engineer proteins such as cytochrome P450s, channelrhodopsins[27,28], adeno-associated virus 2 capsids[29], GFP[30], and acyl-ACP reductases[4]. Adaptive machine learning strategies have also been applied at the pathway level for making chemicals; the automated recommendation tool (ART) is a

valuable machine learning based tool for designing optimal metabolic pathways and it has been used to optimize production of limonene, dodecanol[31,32] and tryptophan[32,33] in microbial cells. A similar tool, ActiveOpt, was used to optimize valine production[34]. The combination of machine learning and directed evolution is sure to lead to substantial advances in a diverse set of fields.

5.3.2 Sequence-function relationships and rational strategies

Machine learning strategies can also complement rational design strategies to engineer proteins by making more efficient use of structural, evolutionary, or functional data. Advances in deep learning have made it possible to leverage a wealth of sequence and structural data to design more advanced ways to encode protein sequences such as UniRep[35]. UniRep is a very data rich method for encoding protein information, and it has recently been used to demonstrate how proteins (such as GFP and β-lactamase) can be engineered with minimal numbers of experiments[36].

There have also been significant advances in how machine learning is used to predict protein structures and design new proteins. AlphaFold is a ground-breaking new algorithm that uses deep learning strategies to accurately predict protein structures[15]. It has already been used to predict protein structures for the entire human proteome[16], and efforts to use it to predict other important protein structures are currently underway. Deep learning strategies have also accelerated design of proteins. In one study, deep learning was used to solve the inverse of the problem that AlphaFold was designed to solve; rather than learning a protein structure, a large set of protein sequences was designed based on structure information[37]. Without the aid of machine learning models, protein design,

especially *de novo* design, is extremely challenging, but with machine learning models, the pace at which novel functional protein sequences can be designed is accelerating.

5.3.3 Generative models and neural networks

Early work in machine learning guided-protein design focused on regression models[13,22,38]. However, to enable learning from larger and more complex datasets new machine learning tools have emerged and are being commonly utilized to engineer proteins. Among these are neural networks[39] and generative models, such as autoregressors[40,41] and autoencoders[42,43]. Generative models have received a lot of attention recently; they have applications in designing new protein sequences[40,43] and learning about protein fitness[42] and structure[41]. Neural networks have also been used widely in machine learning-assisted directed evolution, both on their own and as part of ensembles of models[26]. Neural nets can also be used to understand protein fitness landscapes[39] and protein structure[37], and they are key to the success of AlphaFold[15] and other advances in machine learning-assisted evolution[44] and protein design[35,37]. These kinds of models can be trained on the vast amount of sequence available about protein families and used to generate or suggest new sequences that are not found in nature but that still retain the same structure or function.

5.4 Conclusion

The usage of machine learning strategies in protein and enzyme engineering has exploded in recent years, and as scientific advances enable curation of larger datasets and more accurate models, their usage will continue to expand. As illustrated in this

dissertation, many enzymes involved in biological pathways that are difficult to engineer using directed evolution or rational design can be more easily engineered with the help of machine learning methods. The ability to precisely engineer proteins and metabolic pathways will help solve problems in chemical production, energy, the environment and human health, and machine learning is a highly promising tool for enabling further advances in protein and metabolic engineering.

References:

(1) Greenhalgh, J.; Saraogee, A.; Romero, P. A. Data-Driven Protein Engineering. *Protein Eng.* **2021**, 133–151. https://doi.org/10.1002/9783527815128.CH6.

(2) Lennen, R. M.; Pfleger, B. F. Microbial Production of Fatty Acid-Derived Fuels and Chemicals. *Curr. Opin. Biotechnol.* **2013**, *24* (6), 1044–1053. https://doi.org/10.1016/j.copbio.2013.02.028.

(3) Yan, Q.; Pfleger, B. F. Revisiting Metabolic Engineering Strategies for Microbial Synthesis of Oleochemicals. *Metab. Eng.* **2020**, *58*, 35–46. https://doi.org/10.1016/J.YMBEN.2019.04.009.

(4) 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.* **2021**, *12* (1), 1–10. https://doi.org/10.1038/s41467-021-25831-w.

(5) Martin, V. J. J.; Piteral, D. J.; Withers, S. T.; Newman, J. D.; Keasling, J. D. Engineering a Mevalonate Pathway in Escherichia Coli for Production of Terpenoids. *Nat. Biotechnol. 2003 217* **2003**, *21* (7), 796–802. https://doi.org/10.1038/nbt833.

(6) Bohlmann, J.; Keeling, C. I. Terpenoid Biomaterials. *Plant J.* **2008**, *54* (4), 656–669. https://doi.org/10.1111/J.1365-313X.2008.03449.X.

(7) Van Beilen, J. B.; Poirier, Y. Production of Renewable Polymers from Crop Plants. *Plant J.* **2008**, *54* (4), 684–701. https://doi.org/10.1111/J.1365-313X.2008.03431.X.

(8) Zhao, Y.; Yang, J.; Qin, B.; Li, Y.; Sun, Y.; Su, S.; Xian, M. Biosynthesis of

Isoprene in Escherichia Coli via Methylerythritol Phosphate (MEP) Pathway. *Appl. Microbiol. Biotechnol.* **2011**, *90* (6), 1915–1922. https://doi.org/10.1007/S00253-011-3199-1/FIGURES/6.

(9) Xiang, S.; Usunow, G.; Lange, G.; Busch, M.; Tong, L. Crystal Structure of 1-Deoxy-d-Xylulose 5-Phosphate Synthase, a Crucial Enzyme for Isoprenoids Biosynthesis. *J. Biol. Chem.* **2007**, *282* (4), 2676–2682. https://doi.org/10.1074/JBC.M610235200.

(10) Li, Z.; Sharkey, T. D. Metabolic Profiling of the Methylerythritol Phosphate Pathway Reveals the Source of Post-Illumination Isoprene Burst from Leaves. *Plant. Cell Environ.* **2013**, *36* (2), 429–437. https://doi.org/10.1111/J.1365-3040.2012.02584.X.

(11) Hernández Lozada, N. J.; Simmons, T. R.; Xu, K.; Jindra, M. A.; Pfleger, B. F. Production of 1-Octanol in Escherichia Coli by a High Flux Thioesterase Route. *Metab. Eng.* **2020**, *61*, 352–359. https://doi.org/10.1016/j.ymben.2020.07.004.

(12) Mehrer, C. R.; Incha, M. R.; Politz, M. C.; Pfleger, B. F. Anaerobic Production of Medium-Chain Fatty Alcohols via a β-Reduction Pathway. *Metab. Eng.* **2018**, *48* (April), 63–71. https://doi.org/10.1016/j.ymben.2018.05.011.

(13) Romero, P. a; Krause, A.; Arnold, F. H. Navigating the Protein Fitness Landscape with Gaussian Processes. *Proc. Natl. Acad. Sci. U. S. A.* **2013**, *110* (3), E193-201. https://doi.org/10.1073/pnas.1215251110.

(14) Pfleger, B. F.; Gossing, M.; Nielsen, J. Metabolic Engineering Strategies for Microbial Synthesis of Oleochemicals. *Metab. Eng.* **2015**, *29*, 1–11. https://doi.org/10.1016/j.ymben.2015.01.009.

(15) Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; et al. Highly Accurate Protein Structure Prediction with AlphaFold. *Nature* **2021**, *596*, 583. https://doi.org/10.1038/s41586-021-03819-2.

(16) Tunyasuvunakool, K.; Adler, J.; Wu, Z.; Green, T.; Zielinski, M.; Žídek, A.; Bridgland, A.; Cowie, A.; Meyer, C.; Laydon, A.; et al. Highly Accurate Protein Structure Prediction for the Human Proteome. *590* | *Nat.* | **2021**, *596*. https://doi.org/10.1038/s41586-021-03828-1.

(17) Grisewood, M. J.; Hernández-Lozada, N. J.; Thoden, J. B.; Gifford, N. P.; Mendez-Perez, D.; Schoenberger, H. A.; Allan, M. F.; Floy, M. E.; Lai, R. Y.; Holden, H. M.; et al. Computational Redesign of Acyl-ACP Thioesterase with Improved Selectivity toward Medium-Chain-Length Fatty Acids. *ACS Catal.* **2017**, *7* (6), 3837–3849. https://doi.org/10.1021/acscatal.7b00408.

(18) Néstor, N.; Hernández, J.; Lozada, H.; Lai, R.-Y.; Simmons, T. R.; Thomas, K. A.; Chowdhury, R.; Maranas, C. D.; Pfleger, B. F. Highly Active C 8-Acyl-ACP Thioesterase Variant Isolated by a Synthetic Selection Strategy. *ACS Synth. Biol.* **2018**, *7*, 2205–2215. https://doi.org/10.1021/acssynbio.8b00215.

(19) Bongers, M.; Chrysanthopoulos, P. K.; Behrendorff, J. B. Y. H.; Hodson, M. P.; Vickers, C. E.; Nielsen, L. K. Systems Analysis of Methylerythritol-Phosphate Pathway Flux in E. Coli: Insights into the Role of Oxidative Stress and the Validity of Lycopene as an Isoprenoid Reporter Metabolite. *Microb. Cell Fact.* **2015**, *14* (1), 1–16. https://doi.org/10.1186/S12934-015-0381-7/TABLES/3.

(20) Volke, D. C.; Rohwer, J.; Fischer, R.; Jennewein, S. Investigation of the

Methylerythritol 4-Phosphate Pathway for Microbial Terpenoid Production through Metabolic Control Analysis. *Microb. Cell Fact.* **2019**, *18* (1), 1–15. https://doi.org/10.1186/S12934-019-1235-5/FIGURES/8.

(21) Wittmann, B. J.; Yue, Y.; Arnold, F. H. Informed Training Set Design Enables Efficient Machine Learning-Assisted Directed Protein Evolution. *Cell Syst.* **2021**, *12* (11), 1026-1045.e7. https://doi.org/10.1016/J.CELS.2021.07.008.

(22) Fox, R. J.; Davis, S. C.; Mundorff, E. C.; Newman, L. M.; Gavrilovic, V.; Ma, S. K.; Chung, L. M.; Ching, C.; Tam, S.; Muley, S.; et al. Improving Catalytic Function by ProSAR-Driven Enzyme Evolution. *Nat. Biotechnol.* **2007**, *25* (3), 338–344. https://doi.org/10.1038/nbt1286.

(23) Wu, Z.; Jennifer Kan, S. B.; 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.* **2019**, *116* (18), 8852–8858. https://doi.org/10.1073/pnas.1901979116.

(24) Saito, Y.; Oikawa, M.; Sato, T.; Nakazawa, H.; Ito, T.; Kameda, T.; Tsuda, K.; Umetsu, M. Machine-Learning-Guided Library Design Cycle for Directed Evolution of Enzymes: The Effects of Training Data Composition on Sequence Space Exploration. *ACS Catal.* **2021**, *11* (23), 14615–14624. https://doi.org/10.1021/ACSCATAL.1C03753/SUPPL_FILE/CS1C03753_SI_007.XLSX.

(25) Gupta, A.; Agrawal, S. Machine Learning-Based Enzyme Engineering of PETase for Improved Efficiency in Degrading Non-Biodegradable Plastic. *bioRxiv*. 2022, p 2022.01.11.475766.

(26) Hie, B. L.; Yang, K. K. Adaptive Machine Learning for Protein Engineering. *Curr. Opin. Struct. Biol.* **2022**, *72*, 145–152. https://doi.org/10.1016/J.SBI.2021.11.002.

(27) 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 Comput. Biol.* **2017**, *13* (10), 1–21. https://doi.org/10.1371/journal.pcbi.1005786.

(28) Yang, K. K.; Elliott Robinson, J. Machine Learning-Guided Channelrhodopsin Engineering Enables Minimally Invasive Optogenetics. *Nat. Methods*. https://doi.org/10.1038/s41592-019-0583-8.

(29) Bryant, D. H.; Bashir, A.; Sinai, S.; Jain, N. K.; Ogden, P. J.; Riley, P. F.; Church, G. M.; Colwell, L. J.; Kelsic, E. D. Deep Diversification of an AAV Capsid Protein by Machine Learning. *Nat. Biotechnol. 2021 396* **2021**, *39* (6), 691–696. https://doi.org/10.1038/s41587-020-00793-4.

(30) Biswas, S.; Kuznetsov, G.; Ogden, P. J.; Conway, N. J.; Adams, R. P.; Church, G. M. Toward Machine-Guided Design of Proteins. *bioRxiv* **2018**, 337154. https://doi.org/10.1101/337154.

(31) Opgenorth, P.; Costello, Z.; Okada, T.; Goyal, G.; Chen, Y.; Gin, J.; Benites, V.; De Raad, M.; Northen, T. R.; Deng, K.; et al. Lessons from Two Design-Build-Test-Learn Cycles of Dodecanol Production in Escherichia Coli Aided by Machine Learning. *ACS Synth. Biol.* **2019**, *8* (6), 1337–1351. https://doi.org/10.1021/acssynbio.9b00020.

(32) Radivojević, T.; Costello, Z.; Workman, K.; Garcia Martin, H. A Machine Learning Automated Recommendation Tool for Synthetic Biology. *Nat. Commun.* **2020**, *11*

(1). https://doi.org/10.1038/S41467-020-18008-4.

(33) Zhang, J.; Petersen, S. D.; Radivojevic, T.; Ramirez, A.; Pérez-Manríquez, A.; Abeliuk, E.; Sánchez, B. J.; Costello, Z.; Chen, Y.; Fero, M. J.; et al. Combining Mechanistic and Machine Learning Models for Predictive Engineering and Optimization of Tryptophan Metabolism. *Nat. Commun. 2020 111* **2020**, *11* (1), 1–13. https://doi.org/10.1038/s41467-020-17910-1.

(34) Kumar, P.; Adamczyk, P. A.; Zhang, X.; Andrade, R. B.; Romero, P. A.; Ramanathan, P.; Reed, J. L. Active and Machine Learning-Based Approaches to Rapidly Enhance Microbial Chemical Production. *Metab. Eng.* **2021**, *67*, 216–226. https://doi.org/10.1016/J.YMBEN.2021.06.009.

(35) Alley, E. C.; Khimulya, G.; Biswas, S.; AlQuraishi, M.; Church, G. M. Unified Rational Protein Engineering with Sequence-Based Deep Representation Learning. *Nat. Methods* **2019**. https://doi.org/10.1038/s41592-019-0598-1.

(36) Biswas, S.; Khimulya, G.; Alley, E. C. Low-N Protein Engineering with Data-Efficient Deep Learning. *Nat. Methods*. https://doi.org/10.1038/s41592-021-01100-y.

(37) Anishchenko, I.; Pellock, S. J.; Chidyausiku, T. M.; Ramelot, T. A.; Ovchinnikov, S.; Hao, J.; Bafna, K.; Norn, C.; Kang, A.; Bera, A. K.; et al. De Novo Protein Design by Deep Network Hallucination. *Nat.* | **2021**, *600*. https://doi.org/10.1038/s41586-021-04184-w.

(38) Romero, P. A.; Stone, E.; Lamb, C.; Chantranupong, L.; Miklos, A.; Hughes, R. A.; Fechtel, B.; Ellington, A. D.; Arnold, F. H.; Georgiou, G. SCHEMA Designed Variants of Human Arginase I & II Reveal Sequence Elements Important to

Stability and Catalysis. **2013**, *1* (6), 221–228. https://doi.org/10.1021/sb300014t.SCHEMA.

(39) Gelman, S.; Fahlberg, S. A.; Heinzelman, P.; Romero, P. A.; Gitter, A. Neural Networks to Learn Protein Sequence–Function Relationships from Deep Mutational Scanning Data. *Proc. Natl. Acad. Sci.* **2021**, *118* (48), e2104878118. https://doi.org/10.1073/PNAS.2104878118.

(40) Shin, J.-E.; Riesselman, A. J.; Kollasch, A. W.; McMahon, C.; Simon, E.; Sander, C.; Manglik, A.; Kruse, A. C.; Marks, D. S. Protein Design and Variant Prediction Using Autoregressive Generative Models. *Nat. Commun. 2021 121* **2021**, *12* (1), 1–11. https://doi.org/10.1038/s41467-021-22732-w.

(41) Trinquier, J.; Uguzzoni, G.; Pagnani, A.; Zamponi, F.; Weigt, M. Efficient Generative Modeling of Protein Sequences Using Simple Autoregressive Models. *Nat. Commun. 2021 121* **2021**, *12* (1), 1–11. https://doi.org/10.1038/s41467-021-25756-4.

(42) Hsu, C.; Nisonoff, H.; Fannjiang, C.; Listgarten, J. Learning Protein Fitness Models from Evolutionary and Assay-Labeled Data. *Nat. Biotechnol. 2022* **2022**, 1–9. https://doi.org/10.1038/s41587-021-01146-5.

(43) Hawkins-Hooker, A.; Depardieuid, F.; Baurid, S.; Couairon, G.; Chen, A.; Bikardid, D. Generating Functional Protein Variants with Variational Autoencoders. **2021**. https://doi.org/10.1371/journal.pcbi.1008736.

(44) Wittmann, B. J.; Johnston, K. E.; Wu, Z.; Arnold, F. H. Advances in Machine Learning for Directed Evolution. *Curr. Opin. Struct. Biol.* **2021**, *69*, 11–18. https://doi.org/10.1016/j.sbi.2021.01.008.

Chapter 6

Application of machine learning-based protein engineering to make an improved acyl-ACP reductase enzyme

Author: Jonathan Greenhalgh

I believe science can be applied in ways that make a difference in people's lives. But that can only happen if science is communicated in a way that people can understand the findings and see ways to apply them. I wrote this chapter to present the results of my scientific work in a more accessible manner so that scientists and non-scientists alike can understand the findings and the process. In some respects, that makes this the most important chapter of my dissertation. I am grateful to the Wisconsin Initiative for Science Literacy (WISL) at UW-Madison for encouraging and enabling communication of science to broader audiences, and I am especially grateful to Professor Bassam Shakhashiri, Elizabeth Reynolds and Cayce Osborne for their support and feedback as I've worked on this chapter.

6.1 Introduction

6.1.1 Background and motivation

Ever since I took biochemistry class in college, I have found the chemistry of proteins fascinating. As a chemical engineering student, I was really interested in how proteins could be used to make chemicals in cells. I joined the lab of Phil Romero who studies protein engineering and machine learning, and collaborated very closely with Brian Pfleger, who studies ways to engineer microbial organisms (like bacteria and yeast) to make valuable chemicals. Combining these two disciplines led to the project that I worked on for most of my graduate studies, which is engineering an enzyme that can be used to make the fatty alcohol molecules that are commonly found in lotions and detergents. Protein engineering is a really exciting field of study and combining protein engineering and machine learning (basically using computers to help engineer proteins better), is even more exciting. In this chapter I'll explain in as simple terms as I can what protein engineering is, why we used it, and how machine learning helps the process using a specific example from my research[1].

6.1.2 Proteins and enzymes

To understand protein engineering, first I must explain a little bit about proteins. Proteins are molecules that are made up of smaller pieces called amino acids. There are twenty common amino acids, each with unique properties. This set of amino acids is like an alphabet; the amino acid alphabet contains twenty letters (A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y), where each letter corresponds to a specific amino acid (A for Alanine, C for Cysteine etc., except it's not always the first letter of the amino acids

name). Each amino acid typically only sticks to two other amino acids, so when combined they form long chains that are kind of like words when we write out their one letter abbreviations (things like “WERNLPLDL…”). The order of amino acids in a protein also determines what the protein looks like, and importantly what it does, just like the order of letters in a word determines the word’s meaning. Enzymes are a class of proteins that carry out chemical reactions in cells to convert one molecule to another.

6.1.3 How protein engineering works

Sometimes, for whatever reason, it’s desirable to change what a protein or enzyme does, or make it do a specific task better. Making changes in the protein sequence will usually result in changes to the protein itself (though not always good ones). But how do we change the protein sequence? It turns out there are a lot of ways, but they all involve making changes to DNA in a cell. DNA also has an alphabet (only four letters, A, C, G, and T), but the letters in the DNA alphabet and the letters in the amino acid alphabet are quite different (DNA letters are called nucleotides or bases). DNA contains the instructions for making proteins in cells, and cells make proteins by translating messages from the language of DNA to the language of proteins. In this way, DNA is almost like a coding language, and proteins are kind of like a software application.

Humans have come a long way in terms of understanding DNA, to the point where editing DNA code is becoming much easier to do. Any edits made to instructions for making a protein will result in a modified protein (sometimes called a mutant or variant). There are lots of ways to approach editing the DNA; randomly changing one DNA base at a time, systematically changing very specific bases to target specific amino acids in a

protein, or combining or shuffling large fragments of DNA (this is called recombination)[2–4]. All these methods will affect the protein sequence, and in turn the protein function.

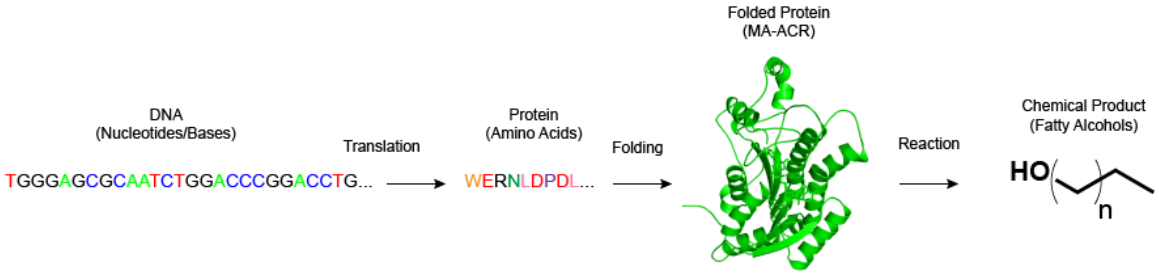

Figure 6.1: DNA determines a protein's sequence, which determines what the protein does or makes. DNA is transcribed to a similar molecule, RNA, which is then translated to make the proteins. Then a protein will fold into a 3D structure, which is uniquely suited for carrying out its designated function (for enzymes this would be carrying out a chemical reaction).

Once a strategy has been selected for editing the protein sequence (via editing the DNA), there are multiple ways to approach engineering. The Nobel Prize in 2018 was awarded to Frances Arnold for developing an approach called directed evolution. Directed evolution is a way to engineer proteins by copying how evolution in nature works[5]. Large sets of altered proteins are tested for a specific trait (for example, heat tolerance, ability to use a specific chemical as a starting material, or reaction speed) and the best one is used as the starting template for the next round. This process is repeated over and over and can result in new protein sequences with massive improvements. Another strategy is to use information about a protein's shape, or structure, and pick specific amino acids to change to accomplish a specific goal. This strategy is called rational design. Both directed evolution and rational design can be used to alter similar properties, the choice of which method to use really depends on how much is known about the structure (more

knowledge favors rational design), or whether it is easy to test large numbers of proteins quickly (the rough number of sequences that can be tested is called throughput; high throughput favors directed evolution).

Though rational design and directed evolution are the main ways to engineer proteins, they have drawbacks too. Rational design requires accurate models of protein structures, which are not always available. Directed evolution requires being able to test a huge number of proteins, tens of thousands to millions (usually all at once, though some very fast technologies enable individual testing), so it is limited by the capacity and speed of the test. However, new methods are emerging to use machine learning to accelerate protein engineering. Machine learning can overcome bottlenecks or shortcomings of directed evolution and rational design to help design better protein sequences in a more efficient manner[6].

6.1.4 Protein engineering for chemical production

Protein engineering has a lot of uses, ranging from therapeutic antibodies, which can be used to treat viral infections, to improving the enzymes used in liquid laundry detergent. In my studies, I am using protein engineering to increase chemical production of fatty alcohols (a kind of chemical commonly used in detergents, cosmetics and flavorings)[7] in cells. Making a chemical product in cells is similar to navigating from a location to a destination on a map. The location is the starting material (usually for cells this would be a sugar like glucose) and the destination is the product (in this case, fatty alcohols). Each step in the path is carried out by an enzyme. The enzyme controls how fast and how well the reaction occurs; this is kind of like controlling what kind of road the

path is and enforcing a speed limit. Sometimes the most direct route to the product goes through a slow enzyme, and so protein engineering can be used to speed that step up.

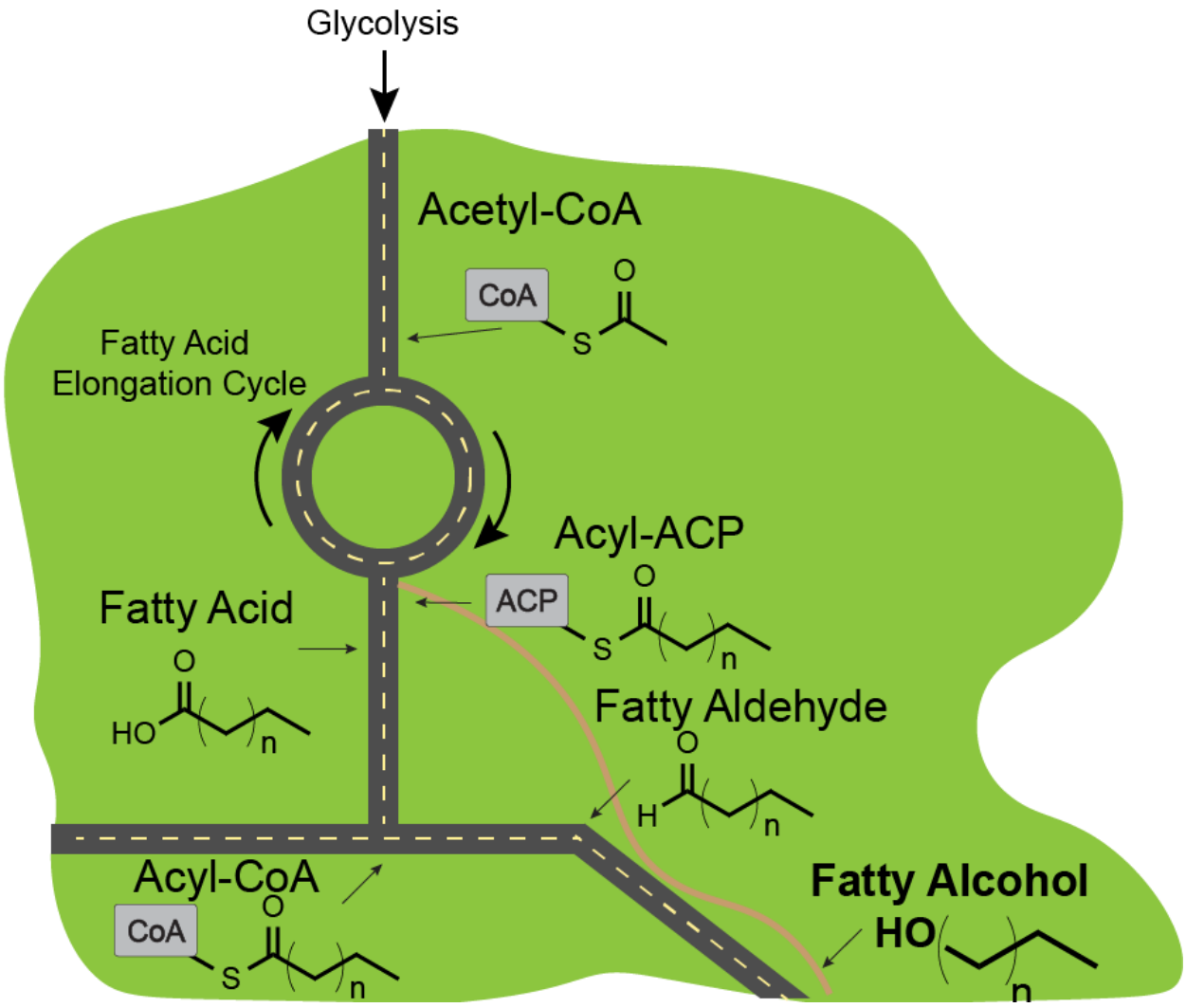

Figure 6.2: The roadmap to fatty alcohols in cells. Glycolysis is the process that breaks down sugars, fatty acid elongation is how fatty acyl-ACPs of different sizes get made. There are multiple possible routes to get from sugars to fatty alcohols. The direct route from acyl-ACPs (shown as a narrow path) is currently less used, but potentially more efficient. More commonly, the longer path through fatty acids and acyl-CoAs is used, which requires more energy.

6.1.5 The acyl-CoA route

There are multiple routes to fatty alcohols on our map. The most used routes go through a specific kind of intermediate called acyl-CoA (ay-seel-co-ay)[7–9]. Acyl-CoAs are

a lot like fatty alcohols in that they come in a range of lengths. They consist of two parts, an acyl chain, and a large molecule called coenzyme A or CoA, (which is a very important molecule elsewhere in metabolism too) linked together. A type of enzyme called an acyl-CoA reductase (or ACR) basically breaks the link between the acyl chain and the CoA. When the link breaks, the acyl chain gets converted first to a fatty aldehyde, and then the ACR transforms the fatty aldehyde to a fatty alcohol (in a reaction called a reduction)[10].

6.1.6 The acyl-ACP route

Acyl-CoA reductases are very good at converting acyl-CoAs to fatty alcohols, but acyl-CoAs aren't necessarily the most direct route to the destination. There is another kind of intermediate called acyl-acyl-carrier proteins (acyl-ACPs) that can be converted to fatty alcohols, and that could potentially be more efficient. Acyl-ACPs are like acyl-CoAs, the acyl chain is just attached to a small protein (ACP) instead of CoA. Because in many cases, the cells need to make acyl-ACPs to get to CoAs anyway, having an ACR that can make fatty alcohols from acyl-ACPs could save the cell energy and resources and enable better results. Some ACRs can also convert acyl-ACPs to fatty alcohols, but they are typically very bad at it[11], so we decided to engineer an ACR to be able to do it better. The ACR called MA-ACR from a species of bacteria called *Marinobacter Aqueolei* showed a lot of promise in scientific work done by others[7–9], so we decided to use it as the starting point for our engineering effort.

6.2 Engineering ACRs to Acyl-ACP Reductases: design, build, test, learn

6.2.1 Designing the library

To engineer MA-ACR, we started by designing a sequence library. Just like a physical library is a place where documents or books are stored, a protein library is a place where a large set of potential protein sequences are gathered. Each protein sequence in the library is like an individual book or document. This metaphor can extend even further, chapters in the book could be like important motifs or modules in the protein sequence and the words could be the individual amino acids. But the library is where we figure out what sequences (or books) are available.

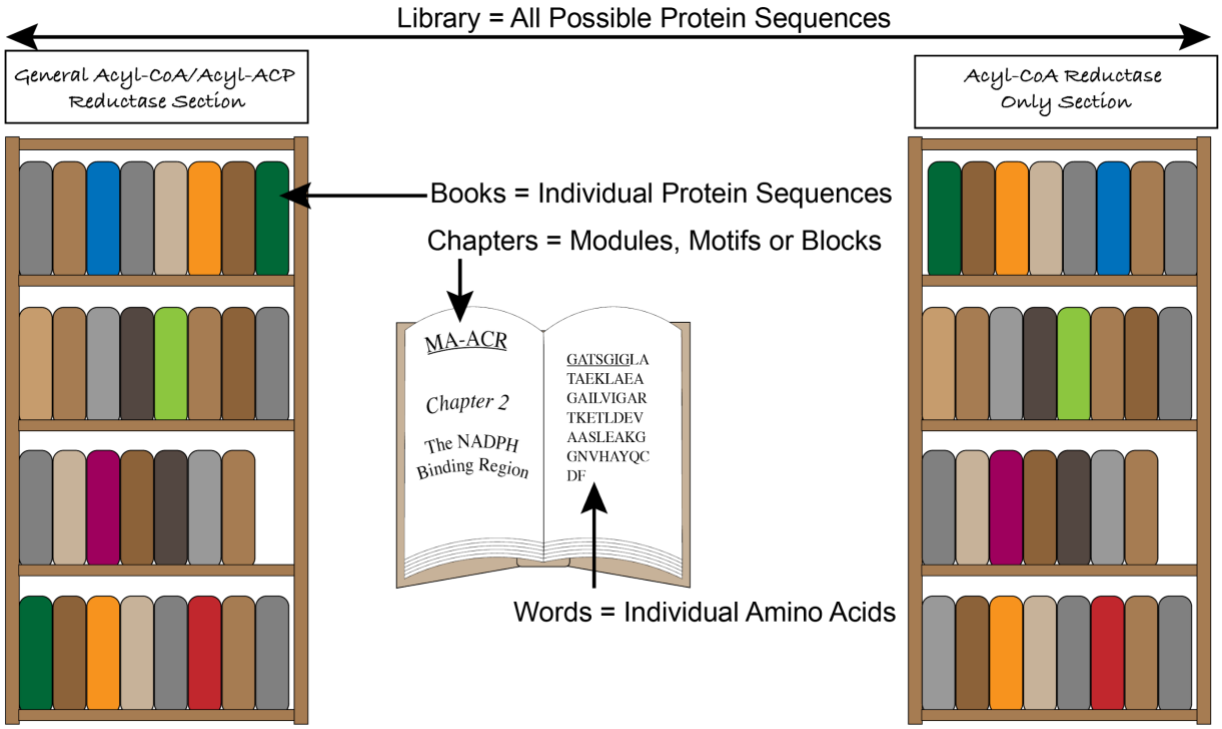

Figure 6.3: Large groups of protein sequences are often referred to as “libraries” in the scientific literature. In this analogy, the library refers to the place where all the sequences are stored, and each protein sequence can be thought of as an individual document. Though literature examples typically don’t extend the library metaphor further, we can think of the parts that make up the protein (modules or motifs), as chapters and the amino acids as words.

The way we chose to design our ACR library was using a trick called recombination[4]. The idea of recombination is that proteins are modular and have modules or blocks of amino acids that can be interchanged for other similar blocks. Because different kinds of organisms often have slightly different versions of the same proteins, these modules can be swapped out to generate new enzymes that work better, and because evolution tends to generate proteins that actually work, it is more likely that changing the protein sequence in this way will result in a functional protein than by just making random changes (a technique that is actually very commonly used). First, since we already knew that each ACR had two parts (or domains)[10], we selected three ACR sequences from different bacteria (MA-ACR, MB-ACR and MT-ACR), and made all nine combinations of the pieces. The results were surprising. First, we found that MB-ACR (BB in Figure 6.4) actually worked better than MA-ACR (AA in Figure 6.4). Second, and more surprisingly, when we made an enzyme that was half MA-ACR and half MB-ACR (AB in Figure 6.4), it worked even better than both of them. It was really exciting and encouraging to have a positive result so early.

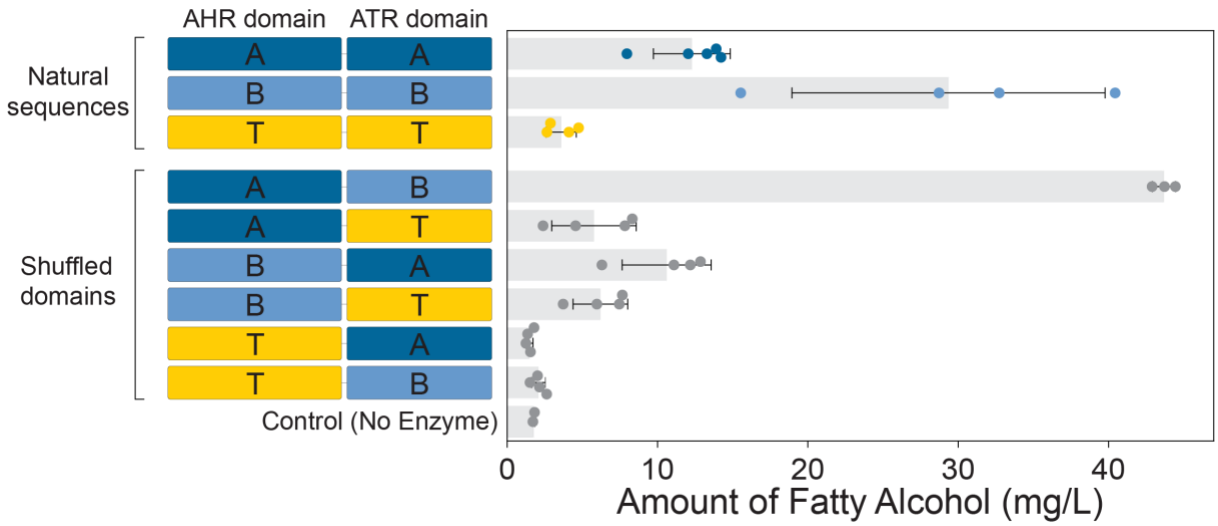

Figure 6.4: Recombining parts from natural enzymes resulted in even better enzymes (compare AB to AA and BB). Note, AA is MA-ACR and BB is MB-ACR.

Next, we started to work on studying the half of the enzyme that carries out the first reaction step (the conversion of acyl-ACP to a fatty aldehyde). To do this, we kept the other half of the enzyme constant and chose the three enzymes that had it as "parents" for recombination (I'll refer to the versions we used as Parent A, B and T), and then used a mathematical algorithm to help us identify a set of eight blocks from each sequence that line up with the other sequences[12]. This would normally give 3^8 = 6,561 possible combinations of blocks, but in this case because two blocks were exactly the same there are 4,374.

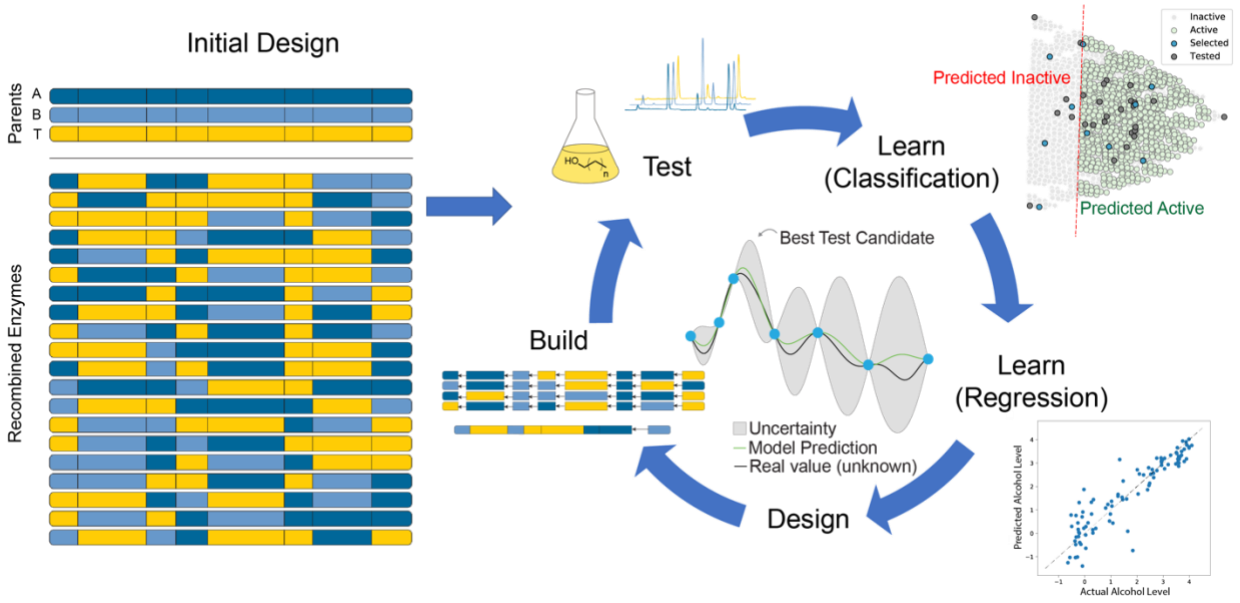

Figure 6.5: The design-build-test-learn cycle. We designed a test set to try as many combinations of the three parent enzymes as possible. Then we tested the sequences experimentally to figure out how much fatty alcohol each one made. After that we used two different kinds of machine learning models to help design sequences to study in future iterations of the cycle. Finally, we assembled the sequences using pieces of DNA and repeated the process.

Normally, with this many potential combinations, we would have to test all the sequences, either individually (which would be prohibitive with our detection methods), or all at once (by linking alcohol production to cell survival or growth). However, because we didn't have reasonable ways to use either of these approaches, we chose to use machine learning to help us learn about these protein sequences more efficiently and bypass the bottleneck in the number of sequences that we could test experimentally.

6.2.2 Machine learning compliments protein engineering

Just like mapping apps can use data and algorithms to help us find directions, similar strategies can be applied to protein engineering. Machine learning is the term for computer algorithms that learn from data. Examples of machine learning in everyday life

are abundant; one common and important example is the spam filter that detects junk emails. Machine learning has the potential to drastically improve efforts to engineer proteins and biological pathways. Biological pathways are complex, and the number of mathematical variables involved in engineering them can be staggering. But machine learning models can deal with all those variables and can help us identify the most important ones. Also, just like machine learning tools can be used in advertisements to recommend specific products for specific people, machine learning can be used to recommend specific “books” or sequences in our library of ACRs that could be good for making fatty alcohols. This means that we don’t have to test all four thousand sequences to find the best enzymes, instead we can test a smaller set and use what we learn to improve our models and find better enzymes.

6.2.3 Building the sequences

From our set of 4,374 sequences, I identified twenty sequences that as a set, gave me the most information about all the potential variables (using statistics). Then, I started working on building them. This is kind of like building Legos, just instead of plastic blocks, we use tiny fragments of DNA and a special enzyme (called ligase) that sticks them together[13]. I found this process fascinating. The idea that I could get on my computer, design a large circular piece of DNA (called a plasmid), and then use that DNA to build something else was very exciting. Of course, it is not quite that easy, and I had my share of troubles getting it to work, but at the end of the day I figured things out and was able to successfully build all twenty sequences I needed for initial testing. I would use the same

method again later once we started learning more about which sequences were good at making fatty alcohols.

6.2.4 Testing

To test our ACR sequences, we would put the DNA sequences into *E. coli* and grow the *E. coli* for about 18 hours. In order to signal the cells to make more ACR protein (i.e. expressing ACR), we would add a chemical called IPTG, which signals the cells to start making the protein, and to make sure the cells had enough resources to make extra fatty alcohols we added glycerol (which serves as a source of carbon, similar to the sugar glucose). After turning on protein production, the cells would naturally start making fatty alcohols as well. We then used an instrument called a Gas Chromatograph to measure the amount of fatty alcohol produced by the cells (the amount or concentration of fatty alcohols in cells is also called the titer).

6.2.5 Learning

After figuring out how much fatty alcohol each of our ACRs made, we then started using machine learning models to learn as much as we could about what sequences worked best. The machine learning models take the protein's amino acid sequences as an input and try to find a mathematical relationship between the sequence and the fatty alcohol levels. This process is called model training. Once a model is trained, it can also be used to make predictions about other sequences, even sequences that have not been tested yet.

I used two kinds of models. First, I used a classifier to learn the difference between ACRs that worked and ACRs that didn't. The classifier input is the protein sequences I tested and whether they were active in my experiments, and it produces a binary output or prediction (active/inactive) for all the sequences that I hadn't tested yet. The classifier is kind of similar to how a spam filter works (predicting spam/not spam based on content of an email). Using the classifier helped me pre-filter out bad sequences so that I wouldn't have to worry about them for the next modeling step.

The next step was using regression models. Unlike the classifier, which outputs categories, regression has a continuous output, meaning that it outputs numerical values. For example, a classifier can be used to classify based on meteorological data, whether it will be hot or cold, but a regression model can be used to predict the temperature. Regression allowed me to make predictions about which ACR sequences would work, and how well they would work compared to others. Additionally, the regression model that I used also outputs estimates of uncertainty[14]. This was very useful, because it allowed me to use the model to suggest (or design) new sequences to build that would be both rich in information and likely to work well at the same time.

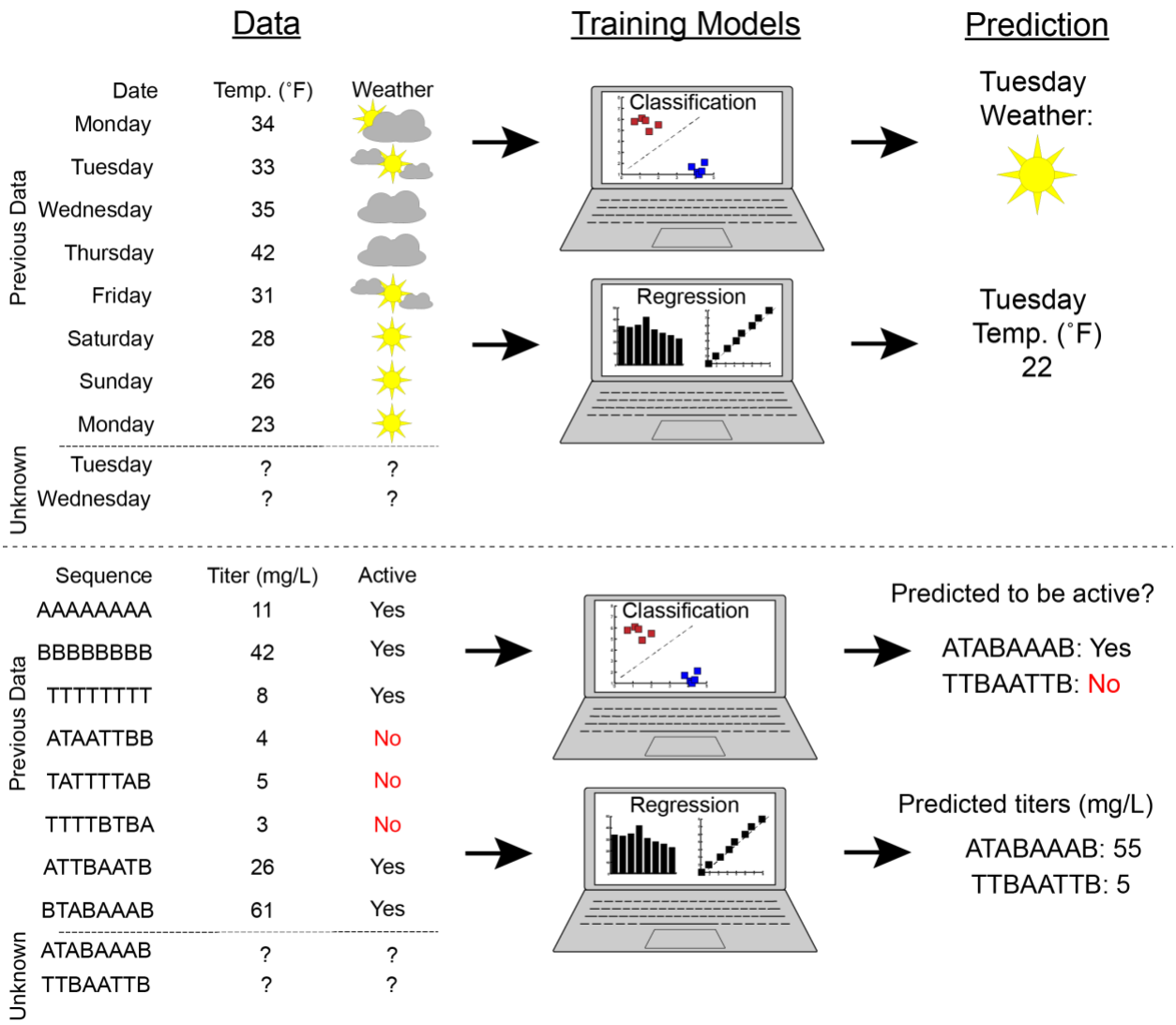

Figure 6.6: Schematic depicting how different kinds of machine learning models work (regression and classification). The top half shows an example of how each kind of model could be used to predict different weather-related properties. The bottom example is an example of how the models could be used to predict whether an enzyme works (i.e., whether it is active) and how much fatty alcohol it can make (i.e. it's titer).

6.2.6 Completing the cycle

Searching for the best enzyme is kind of like hiking to the top of a mountain. From the base of the mountain, the top is not always visible, so we can make an estimate (like training a model and making a prediction) to guess where it might be. As we climb higher and higher, we test out our theory and gain more information. This helps us get a clearer picture of where the summit is (like updating a model). Occasionally this journey leads to the top of false summits or secondary peaks, but if we keep updating our objective as we learn from the landscape, we can make it to the top.

Finding the best ACR sequence was very similar, the first batch of sequences that were suggested by the machine learning models didn't work very well (we had no idea where the summit was). But that was ok. We never intended to stop after the first attempt, and when it comes to machine learning, any data we could get would be useful for future rounds. After the sequences in the first round failed, we just updated the machine learning models and tried again. We repeated the process of building and testing sequences, training machine learning models on the data, and using the models to design new sequences, over and over until we finally achieved ACR sequences that worked much better than the original enzymes we started with. All in all, we tested about 96 ACR sequences over ten turns (or rounds) of this iterative design-build-test-learn cycle.

Each full cycle took about two weeks: one week to build the sequences and verify that they were correct, and another week to grow the cells containing the sequences and test them. Updating the machine learning models was the fastest part of the workflow. It only took a few minutes to update the models and make new predictions. Over time, as

there was more and more data to train the models, the models took slightly longer to train, but on the flipside, they got much more accurate too.

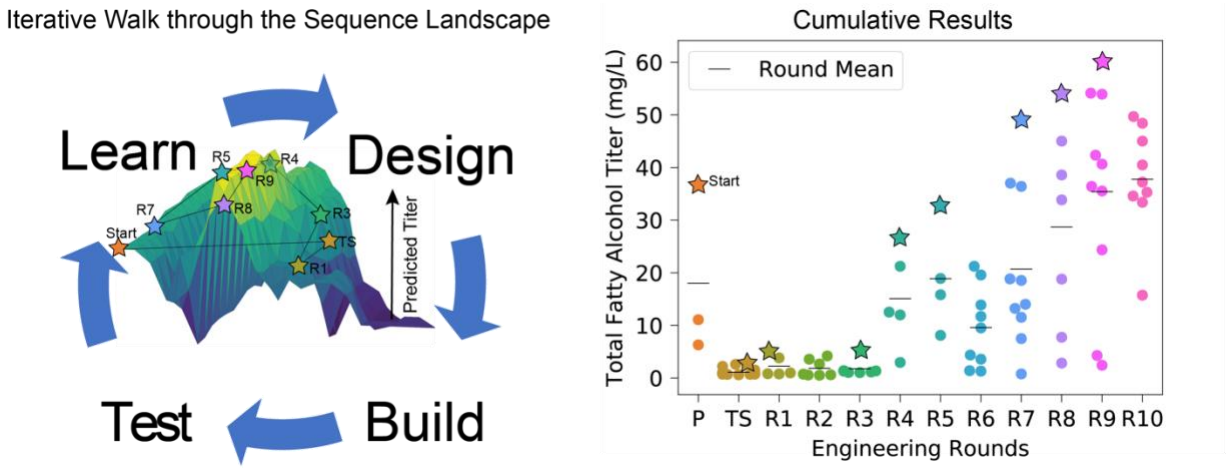

Figure 6.7: The search for the optimal enzyme can be thought of as a hike. By gradually working your way uphill, you can make your way to the summit. Similarly, by gradually searching for better and better enzymes you can progress towards the best one. The left side of this figure shows a visualization of the sequence landscape in 3D, where the "elevation" is like the predicted titer from the models. The stars show key points along the way (basically top sequence in every round where there was an improvement). Further distances between sequences suggest that the sequences are more different from each other. The figure on the right shows the amounts of fatty alcohols in each round (each round is shown as a different color). P is the parents (AA, AB, and AT) and TS is the initial test set of twenty ACRs. The best sequence was identified in round 9. The black bars in this figure show the average titer for all the sequences in that round.

6.3 Conclusions

Our best ACR sequence (which we called ATR-83) made about twice as much fatty alcohol as the best natural enzyme that we studied. Experiments in test tubes validated these results and verified that they were due to the enzyme being a better catalyst, rather than simply being easier for the cell to make. The next question we wanted to answer was why it worked better. Again, we turned to machine learning models. This time instead of making predictions about sequences we hadn't tested yet; we used the models to try to understand as much of the sequence as we could. The outputs of the

models suggested that a few very specific blocks were very important. Some of these were expected to play a big role (the blocks that actually carry out the reaction), but some of the important blocks were surprising. As we studied the protein sequence further, we found that there were a large number of positively charged amino acid residues near the site where we believed the acyl-ACP should bind. Interestingly, ACP has a lot of negatively charged amino acids, so the two proteins can stick together similar to a pair of oppositely charged magnets[15]. ATR-83, our best sequence, had even more positively charged amino acids than the natural proteins.

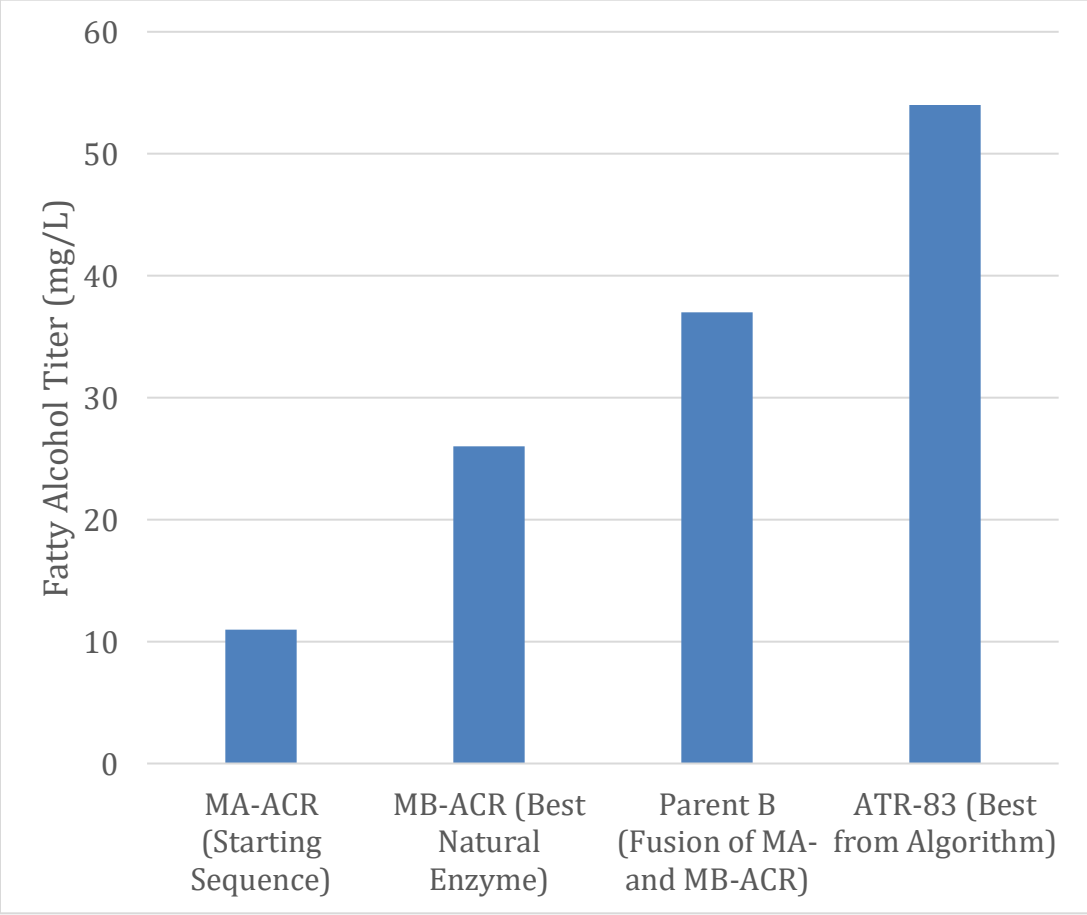

Figure 6.8: Summary of Improvements to ACRs. Testing out new natural enzymes helped us find a better ACR for converting acyl-ACPs to alcohols. Recombining that sequence with our starting sequence resulted in another boost in activity and using our machine learning-guided approach we were able to design a sequence (ATR-83) that could make even more fatty alcohols.

In summary, we were able to over double the amount (or titer) of fatty alcohols that we produced in cells by using our designed ACR sequence, ATR-83. This result shows that machine learning can be used to help engineer proteins without needing to test thousands of sequences or without knowing a precise structure. The most exciting thing about this approach is that it can be used to engineer almost any enzyme, as long as there is a way to test the enzyme's function. Hopefully this result will not only enable further improvements in production of chemicals in cells but accelerate the use of machine learning in protein engineering workflows.

References:

(1) 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.* **2021**, *12* (1), 1–10. https://doi.org/10.1038/s41467-021-25831-w.

(2) Coco, W. M.; Levinson, W. E.; Crist, M. J.; Hektor, H. J.; Darzins, A.; Pienkos, P. T.; Squires, C. H.; Monticello, D. J. DNA Shuffling Method for Generating Highly Recombined Genes and Evolved Enzymes. *Nat. Biotechnol.* **2001**, *19* (4), 354–359. https://doi.org/10.1038/86744.

(3) Stemmer, W. P. Rapid Evolution of a Protein in Vitro by DNA Shuffling. *Nature*. 1994, pp 389–391. https://doi.org/10.1038/370389a0.

(4) Silberg, J. J.; Endelman, J. B.; Arnold, F. H. SCHEMA-Guided Protein Recombination. *Methods Enzymol.* **2004**, *388* (2003), 35–42. https://doi.org/10.1016/S0076-6879(04)88004-2.

(5) Arnold, F. H.; Volkov, A. A. Directed Evolution of Biocatalysts. *Curr. Opin. Chem. Biol.* **1999**, *3* (1), 54–59. https://doi.org/10.1016/S1367-5931(99)80010-6.

(6) Yang, K. K.; Wu, Z.; Arnold, F. H. Machine-Learning-Guided Directed Evolution for Protein Engineering. *Nature Methods*. Nature Publishing Group August 1, 2019, pp 687–694. https://doi.org/10.1038/s41592-019-0496-6.

(7) Hernández Lozada, N. J.; Simmons, T. R.; Xu, K.; Jindra, M. A.; Pfleger, B. F. Production of 1-Octanol in Escherichia Coli by a High Flux Thioesterase Route. *Metab. Eng.* **2020**, *61*, 352–359. https://doi.org/10.1016/j.ymben.2020.07.004.

(8) Youngquist, J. T.; Schumacher, M. H.; Rose, J. P.; Raines, T. C.; Politz, M. C.; Copeland, M. F.; Pfleger, B. F. Production of Medium Chain Length Fatty Alcohols from Glucose in Escherichia Coli. *Metab. Eng.* **2013**, *20*, 177–186. https://doi.org/10.1016/j.ymben.2013.10.006.

(9) Mehrer, C. R.; Incha, M. R.; Politz, M. C.; Pfleger, B. F. Anaerobic Production of Medium-Chain Fatty Alcohols via a β-Reduction Pathway. *Metab. Eng.* **2018**, *48* (April), 63–71. https://doi.org/10.1016/j.ymben.2018.05.011.

(10) Willis, R. M.; Wahlen, B. D.; Seefeldt, L. C.; Barney, B. M. Characterization of a Fatty Acyl-CoA Reductase from Marinobacter Aquaeolei VT8: A Bacterial Enzyme Catalyzing the Reduction of Fatty Acyl-CoA to Fatty Alcohol. *Biochemistry* **2011**, *50* (48), 10550–10558. https://doi.org/10.1021/bi2008646.

(11) Opgenorth, P.; Costello, Z.; Okada, T.; Goyal, G.; Chen, Y.; Gin, J.; Benites, V.; De Raad, M.; Northen, T. R.; Deng, K.; et al. Lessons from Two Design-Build-Test-Learn Cycles of Dodecanol Production in Escherichia Coli Aided by Machine Learning. *ACS Synth. Biol.* **2019**, *8* (6), 1337–1351.

https://doi.org/10.1021/acssynbio.9b00020.

(12) Endelman, J. B.; Silberg, J. J.; Wang, Z.; Arnold, F. H. Site-Directed Protein Recombination as a Shortest-Path Problem. *Protein Eng. Des. Sel.* **2004**, *17* (7), 589–594. https://doi.org/10.1093/protein/gzh067.

(13) Engler, C.; Gruetzner, R.; Kandzia, R.; Marillonnet, S. Golden Gate Shuffling: A One-Pot DNA Shuffling Method Based on Type Ils Restriction Enzymes. *PLoS One* **2009**, *4* (5). https://doi.org/10.1371/journal.pone.0005553.

(14) Romero, P. a; Krause, A.; Arnold, F. H. Navigating the Protein Fitness Landscape with Gaussian Processes. *Proc. Natl. Acad. Sci. U. S. A.* **2013**, *110* (3), E193-201. https://doi.org/10.1073/pnas.1215251110.

(15) Sarria, S.; Bartholow, T. G.; Verga, A.; Burkart, M. D.; Peralta-Yahya, P. Matching Protein Interfaces for Improved Medium-Chain Fatty Acid Production. *ACS Synth. Biol.* **2018**, *7* (5), 1179–1187. https://doi.org/10.1021/acssynbio.7b00334.