UNDERSTANDING GENETIC SYSTEMS THROUGH MULTIPLEXED DESIGN, SYNTHESIS, AND MEASUREMENT

by

DANIEL B. GOODMAN

BS Bioengineering & Bioinformatics
The University of California at San Diego, 2008

Submitted to the
MEDICAL ENGINEERING AND MEDICAL PHYSICS PROGRAM
HARVARD-MIT DIVISION OF HEALTH SCIENCES AND TECHNOLOGY

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY
BIOINFORMATICS AND INTEGRATIVE GENOMICS

Massachusetts Institute of Technology

June 2016

Signature redacted

Signature of Author: ....
Daniel B. Goodman
Harvard-MIT, Division of Health Sciences and Technology
April 29, 2016

Certified By: Signature redacted ....
George M. Church, PhD
Professor of Genetics
Harvard Medical School
Thesis Supervisor

Signature redacted

Accepted By: ....
Emery N. Brown, MD, PhD
Director, Harvard-MIT Program in Health Sciences and Technology
Professor of Computational Neuroscience and Health Sciences and Technology

UNDERSTANDING GENETIC SYSTEMS THROUGH MULTIPLEXED DESIGN, SYNTHESIS, AND MEASUREMENT

DANIEL B. GOODMAN

Submitted to the Harvard-MIT Division of Health Sciences and Technology
on 29 April 2016 in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Bioinformatics and Integrative Genomics

ABSTRACT

Next-generation DNA sequencing has allowed us to extract vast quantities of functional information from genetic systems. However, natural systems represent only a fraction of all possible DNA sequences. Our understanding of how genomes function is limited by our ability to make modifications and test hypotheses. Multiplexed DNA synthesis now allows us to generate thousands of computationally designed sequences, each representing a physical hypothesis to test. Here, we combine DNA sequencing and synthesis technologies to design, make, and measure the behavior of thousands of new genetic elements in the bacterium *E. coli.*

We begin by quantifying the interactions between regulatory elements that control transcription and translation and show that these interactions create large deviations from the predicted behavior of individual elements. Regulatory elements also interact with the codons of the genes they control. We show that rare codon usage at the beginning of genes unexpectedly leads to a strong increase in protein translation due to the relationship between codon rarity, genomic nucleotide bias, and mRNA structure. We next examine the behavior of regulatory elements that bind transcription factors by designing and synthesizing over 100,000 transcriptional circuits. From each circuit we measure repression, activation, and small-molecule induction, deriving relationships between DNA sequence features and functional properties including cooperativity, sensitivity, and dynamic range of gene expression response.

Finally, as the scale and speed of DNA synthesis and functional readout continues to increase, our ability to computationally design and analyze genetic systems has become the bottleneck. We have built software to predict and design individual genetic elements in high throughput (*Promuter*) as well as software to analyze and compare hundreds of evolved or engineered bacterial whole genomes (*Millstone*). As generating high dimensional datasets becomes exponentially easier than designing experiments and extracting knowledge, bioinformatics, machine learning, and data science will become the primary tools we use to pose new hypotheses and build models of biology.

Thesis Supervisor: George M. Church
Title: Professor, Department of Genetics, Harvard Medical School

To my whole family; past, present, and future. You make me who I am.

"What lies at the heart of every living thing is not a fire,
not warm breath, not a 'spark of life.'
It is information, words, instructions.
If you want to understand life,
don't think about vibrant, throbbing gels and oozes,
think about information technology."

— Richard Dawkins

"Torture the data, and it will confess to anything."

— Ronald Coase

ACKNOWLEDGMENTS

Graduate school has been the most amazing time of my life. It would not have been so without all of people who have shown me love, kindness, patience, and friendship throughout.

One of the most fateful decisions I have ever made was joining George Church's lab. Throughout graduate school he has constantly challenged me to think bigger and expand my horizons. I feel incredibly lucky to be able to sample his mind at regular intervals, and he has altered my scientific outlook in ways I'm sure I won't appreciate fully until after I've left his lab.

One of George's most amazing feats is somehow assembling and tending to this thing we call the Church Lab. I can't imagine that there exists a more vibrant and collborative place to do science, where everyone is constantly learning from and helping one another. All of my ideas I owe in part to its special zeitgeist. It is by far the most remarkable community I've ever been a member of.

From Sri Kosuri I've learned what it actually takes to be successful as a scientist. I cannot imagine navigating gradaute school without his guidance, patience, and friendship. Thank you for helping to steer me through this crazy corner of science, and showing me what mentorship is all about.

Marc Lajoie taught me so many things in lab for the first time, and because of him I learned them the right way. He will always be an unparalleled scientific role model, enthusiastic collaborator, and a kind friend.

The times spent getting to know and work with Gleb Kuznetsov have been some of my favorite in the lab. I have learned a lot of things from him, but especially how to be a deep, careful, and practical thinker. Through it all, he has become one of my best friends.

Believing in your own research is much easier and less frightening when you have someone talented and enthusiastic who believes in it too. Chapter 4 would not have been possible without the tireless and patient efforts of Casper Enghuus, who I am proud to call my friend and colleague.

Tara Gianoulis was an incredible person who showed me how to get the most out of science and life. I still think about her all the time.

Vatsan Raman taught me the value of having regular coffee conversations, thinking hard about hard problems, and being a well-rounded scientist. He was also the glue that held our bay together.

In addition to others I have mentioned, I want to thank some of the many other friends and colleagues I've made during my time in the lab, including Noah Taylor, Jamie Rogers, Alex Garruss, Kevin Esvelt, Raj Chari, Adrian Briggs, Evan Daugharthy, Dima Ter-Ovanesyan, Max Schubert, Pierce Ogden, David Thompson, Dan Mandell, Harris Wang, Jay Lee, Michael Napolitano, Matthieu Landon, Joe Davis, Noah Davidsohn, and Rigel Chan.

I am also very grateful to have an amazing family. Their love and support has made me who I am.

By far the best part of graduate school has been sharing it with my wife Linda. She's always willing to listen to crazy ideas, offer brilliant suggestions, entertain outlandish scientific discussions, and to laugh with me. She keeps me sane and stable in times of stress and uncertainty. She is my closest friend, teammate, and confidant, and I'm a better person because of her.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1

INTRODUCTION

Next-generation DNA sequencing has revolutionized the scale at which we study molecular biological systems. In addition to directly reading out the sequences of genes and genomes, functional genomics tools like RNA-seq, CHIP-seq, DNase-seq, ART-seq, NET-seq, bisulfite sequencing, and dozens more all couple measurement of various cellular functions to high-throughput DNA readout[104]. By turning a variety of different cellular information into a DNA signal, we can now extract vast quantities of functional information from genetic systems.

However, natural systems represent only a fraction of all possible DNA sequences, and understanding the function of unknown and novel sequence - new viral strains, de novo mutations in genetic disease and cancer, new bacteria resistant to antibiotics - are often of primary importance to biology. From single base changes to newly sequenced genomes, our understanding of how genotypes map to phenotypes is limited by our ability to make modifications and test hypotheses. Until recently, *de novo* DNA synthesis has been limited to slow, expensive, and relatively short column-based oligonucleotide synthesis. Recently, technological advances[51] now allow us to design and generate libraries of thousands of computationally designed DNA sequences that are hundreds of nucleotides in length. Each of these DNA sequences represents the physical instance of a hypothesis to test[42] (Figure 1.1).

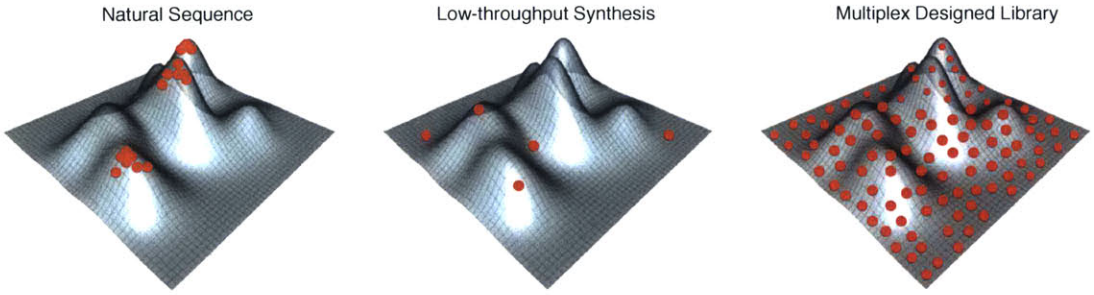


Figure 1.1: Left: Natural sequence is insufficient for understanding complex sequence-behavior relationships because of coalescence and evolutionary constraints. Middle: Low-throughput synthetic biology is constrained by the speed of assembly and measurement. Right: Rational synthesis and measurement of large DNA libraries can efficiently extract emergent properties and iteratively test models.

Here, we combine oligonucleotide libraries with a novel method we developed called *FlowSeq*, which can accurately measure protein expression from large libraries of genetic variants using a fluorescent reporter construct. *FlowSeq* combines fluorescence activated cell sorting (FACS) and a multiplexed barcoded DNA sequence readout. In the first part of this thesis, composing 3 chapters, we combine DNA synthesis with *FlowSeq* to design, make, and measure the behavior of thousands of new genetic elements in the bacterium *E. coli*. First, in **Chapter 2**, we synthesize combinations of regulatory elements that control transcription and translation and examine their composability. We show that the interactions between these elements create large deviations from the predicted behavior of individual elements, suggesting that regulatory sequences are not easily separated into composable parts. Though we found that these interactions were complex and context sensitive, the scale of our assay allowed us to begin unraveling the mechanisms by which they occur. The work also suggests that while we can use these sequences to understand the molecular mechanisms of gene regulation, designing and testing many DNA sequences simultaneously lets us make multiple 'shots on goal' for achieving the desired function.

The genetic code is redundant—multiple codons can code for the same amino acid. So-called synonymous codon changes within genes can nonetheless have substantial affects on protein expression, which have been attributed to changes in the structure of 5′ messenger RNAs, among other factors. In **Chapter** 3, we use oligonucleotide library synthesis and *FlowSeq* to show that rare codon usage at the beginning of genes unexpectedly leads to a strong increase in protein translation. We used a synthetic combinatorial library of promoters, ribosome binding sites, and codon variants to separate out the effects of codon adaptation and mRNA folding, and show the increase in gene expression is due not to rare codons directly, but due to the relationship between codon rarity, genomic nucleotide bias, and 5′ mRNA structure.

In **Chapter** 4, we examine the behavior of regulatory elements that bind transcription factors by designing and synthesizing over 100,000 transcriptional circuits consisting of a transcription factor and a fluorescent reporter output. From each circuit we measure repression, activation, and small-molecule induction, deriving relationships between DNA sequence features and functional properties including cooperativity, sensitivity, and dynamic range of gene expression response. For all 8 different transcription factors and 3 promoters, we show that new functional regulatory elements that can reached in as few as 5 single nucleotide changes. In addition to being useful in their own right as synthetic biological parts, a quantitative understanding of the behavior of these circuits and their constitutive elements helps us to understand the evolution and functional landscape of genetic regulation.

Finally, as the scale and speed of DNA synthesis and functional readout continues to increase, our ability to computationally design and analyze genetic sytstems has become the bottleneck. To this end, we have built software to predict and design individual genetic elements in high throughput (*Promuter*, described as part of our regulatory circuit study in Chapter 4). In the second part of the thesis,

Chapter 5, we describe *Millstone*, another software package we have built to analyze and compare hundreds of evolved or engineered bacterial whole genomes. This software comes out of several years of genome evolution and engineering experiments undertaken by our lab[27, 48, 49]. The cost of modifying genomes using synthetic DNA and then sequencing them has become faster and cheaper, to the extent that individual *E. coli* genomes, when sequenced in bulk, cost under $20 USD each. At the scale of hundreds of bacterial genomes, indentifying and annotating the functions of genomic changes, comparing mutations among multiple clones, and tracing the genotypes and phenotypes of many iteratively engineered genomic versions becomes extremely computationally unwieldly without an integrated computational tool. Built to solve this isssue, *Millstone* is a web-based platform for multiplex mutation analysis and iterative genome engineering. Millstone integrates alignment, variant-calling, genotype comparison, and visualization for hundreds of microbial genomic samples. We show how *Millstone* has been used in in-house genome engineering projects and also describe its use in reanalyzing a published genome evolution dataset composed of over 100 experimentally evolved strains[93].

As studies like those described here generate ever increasing amounts of high-dimensional data, the challenge will become the design of these large experiments and the extraction of knowledge from them. Implicit in this work is the notion of a high-throughput "design, build, test, and learn" cycle where each step is computationally driven and each cycle extends our ability to understand, predict, and ultimately design genetic systems across all scales. Not only will bioinformatics, machine learning, and data science rapidly become essential for understanding genetic systems, but they will increasingly be the tool with which we pose hypotheses and build models of biology.

Part I

SYNTHESIS AND MEASUREMENT OF SYNTHETIC GENETIC ELEMENTS

2

COMPOSABILITY OF REGULATORY SEQUENCES CONTROLLING TRANSCRIPTION AND TRANSLATION IN E. COLI

This chapter is reproduced with permission from its initial publication:

Sriram Kosuri[1], **Daniel B. Goodman**[1], Guillaume Cambray, Vivek K. Mutalik, Yuan Gao, Adam P Arkin, Drew Endy, and George M Church. *Composability of regulatory sequences controlling transcription and translation in Escherichia coli.* **PNAS**, August 2013.

Author Contributions and Acknowledgements:

S.K., D.B.G., and G.M.C. conceived the project, analyzed the data with input from all authors. All authors reviewed the manuscript. S.K. and D.B.G. designed and performed all experiments and wrote the manuscript. D.B.G. and S.K. performed all computational analyses.

This work was supported by US Department of Energy Grant DE-FG02-02ER63445 (to G.M.C.), National Science Foundation (NSF) Synthetic Biology Engineering Research Center Grant SA5283-11210 (to G.M.C.), and Office of Naval Research Grant N000141010144 (to G.M.C. and S.K.), as well as by Agilent Technologies and Wyss Institute. D.B.G. is supported by an NSF Graduate Research Fellowship.

[1] (co-first author)

ABSTRACT

The inability to predict heterologous gene expression levels precisely hinders our ability to engineer biological systems. Using well-characterized regulatory elements offers a potential solution only if such elements behave predictably when combined. We synthesized 12,563 combinations of common promoters and ribosome binding sites and simultaneously measured DNA, RNA, and protein levels from the entire library. Using a simple model, we found that RNA and protein expression were within twofold of expected levels 80% and 64% of the time, respectively. The large dataset allowed quantitation of global effects, such as translation rate on mRNA stability and mRNA secondary structure on translation rate. However, the worst 5% of constructs deviated from prediction by 13-fold on average, which could hinder large-scale genetic engineering projects. The ease and scale this of approach indicates that rather than relying on prediction or standardization, we can screen synthetic libraries for desired behavior.

2.1 INTRODUCTION

Organisms can be engineered to produce chemical, material, fuel, and medical products that are often superior to non-biological alternatives [63]. Biotechnologists have sought to discover, improve, and industrialize such products through the use of recombinant DNA technologies [18, 38]. In recent years, these efforts have increased in complexity from expressing a few genes at once to optimizing multi-component circuits and pathways [19, 90, 92, 98]. To reliably attain desired system-level function, careful and time-consuming optimization of individual components is required [13, 57, 71, 88].

To mitigate this slow trial-and-error optimization, two dominant approaches have taken hold. The first approach seeks to predict expression levels by elucidating the biophysical relationships between sequence and function. For example, several groups have modified promoters [64] and ribosome binding sites [8, 61, 76] (RBSs) to see how small sequence changes affect transcription or translation. Such studies are fundamentally challenging due to the vastness of sequence space. In addition, because these approaches mostly look at either transcription or translation individually, they are rarely able to investigate interactions between these processes.

The second approach uses combinations of individually characterized elements to attain desired expression without directly considering their DNA sequences [2, 5, 6, 17, 25, 32, 74, 77, 105]. Current efforts have focused on approaches to limit the number of time-consuming steps required to characterize potential interactions and on identifying existing or engineered elements that act predictably when used in combination [59, 60]. However, these studies still suggest there are enough idiosyncratic interactions and context effects that it will be necessary to construct and measure many variants of a circuit to achieve desired function [40]. For larger circuits, such approaches are necessarily limited in scope due to the difficulty in measuring large numbers of combinations [59, 60].

Here we overcome previous limitations in generating and measuring large numbers of regulatory elements by combining recent advances in DNA synthesis with novel multiplexed methods for measuring DNA, RNA, and protein levels simultaneously using next-generation sequencing. We use the method to characterize all combinations of 114 promoters and 111 ribosome binding sites and quantify how often simple measures of promoter and RBS strengths can accurately predict gene expression when used in combination. In addition, since we measure both RNA and protein levels across the library, we can also quantify how translation affects mRNA levels and how mRNA secondary structure affects translation efficiency. Finally, the size of the characterized library also provides a resource for researchers seeking to achieve particular expression levels. In lieu of using standardized elements or prediction-based design, library synthesis and screening allows precise tuning of expression in arbitrary contexts.

2.2 RESULTS

2.2.1 *Library Design, Construction, and Initial Characterization.*

To systematically explore the effects of regulatory element composition, we designed and synthesized all combinations of 114 promoters with 111 RBSs (12,653 constructs in total; one combination resulted in an incompatible restriction site). We used 90 promoters from an existing library from BioFAB, 17 promoters from the Anderson promoter library on the BioBricks registry, 6 promoters from common cloning vectors, and a spacer sequence chosen as a negative control. From RBSs, we used 55 RBSs from the BioFAB library, 31 from the Anderson BioBrick library, 13 from the Salis RBS Calculator expected to give a range of expression, 12 commonly used RBSs from cloning vectors and the BioBrick Registry, and one sequence chosen as a negative control (reverse complement of canonical RBS sequence). We synthesized the construct library using Agilent's OLS technology [51] and cloned at ~50x coverage into a custom medium-copy vector (pGERC) where the constructs drive expression of super-folder GFP [66] (Fig. A.1). pGERC also contains an mCherry [84] reporter under constant expression by $P_{LTetO-1}$ [22] to act as a control for extrinsic noise (Fig. 2.1A). We grew the library to early exponential phase and characterized expression levels by flow cytometry. As expected, cells in the library expressed constant levels of mCherry, while expression levels of GFP varied over four orders of magnitude (Fig. 2.1B). We sequence-verified 282 colonies and found that 154 (55%) were error-free. We measured fluorescence levels of 144 of the unique error-free colonies individually to act as a defined set of controls (Fig. 2.1C).

2.2.2 *Multiplexed Measurements of DNA, RNA and Protein Levels.*

We grew the entire pooled library to early exponential phase and performed multiplexed measurements of the steady-state DNA, RNA, and protein levels. We used DNASeq and RNASeq to obtain steady-state DNA and RNA levels across the library [64]. For obtaining protein levels, we used FlowSeq, which combines fluorescence activated cell sorting and high-throughput DNA sequencing and is similar in design to recently published work [69, 81]. Briefly, we sorted cells into 12 log-spaced bins of varying GFP/mCherry ratios, isolated, amplified, and barcoded DNA from each of the bins, and then used high-throughput sequencing to count the number constructs that fell into each bin (Fig. 2.1A and D). Using the read counts from each of the bins, we reconstructed the average expression level for each construct. Because our library contains a mixture of perfect and imperfect constructs, we only use reads that match the full designed sequences perfectly and thus filter out the effects of synthesis error.

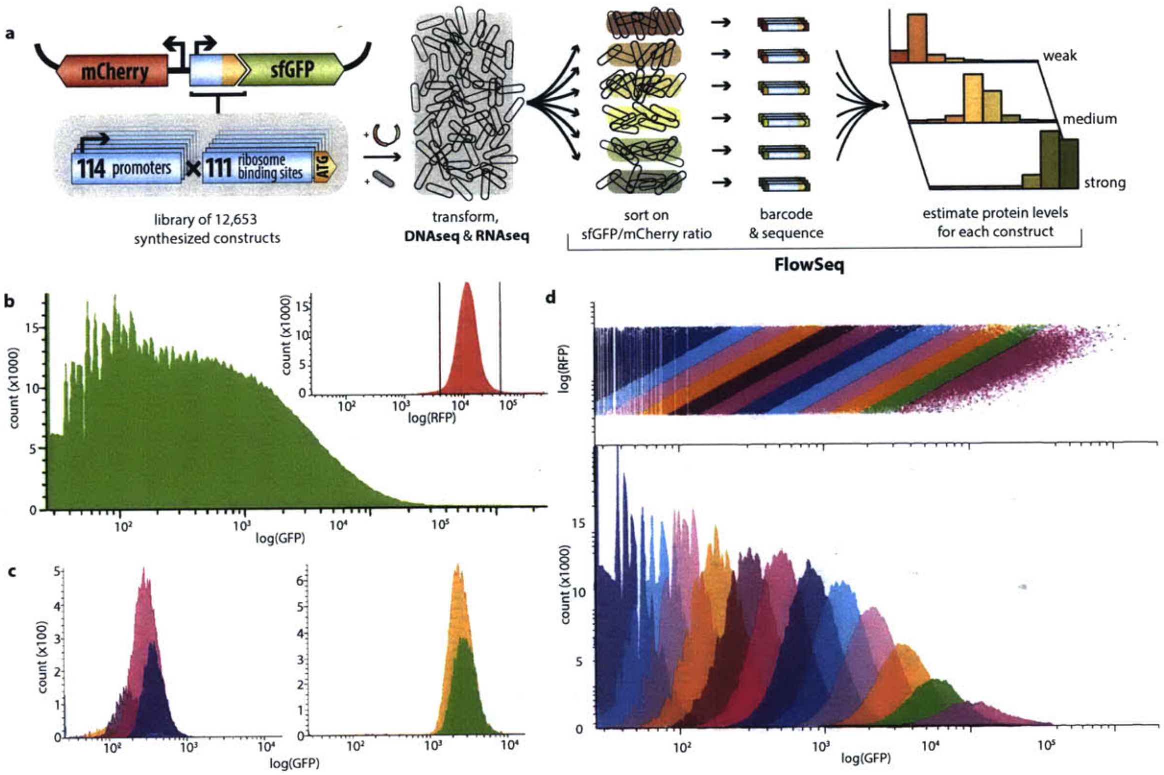


Figure 2.1: *Library characterization and workflow.* (a) We synthesized all combinations of 114 promoters and 111 ribosome binding sites to create a 12,653 construct library. The library was then cloned into an expression plasmid to express super-folder GFP; mCherry was also independently expressed from a constitutive promoter to act as an intracellular control. The cell library was harvested for DNASeq, RNASeq, and FlowSeq to quantify DNA, RNA, and protein levels for each construct. In FlowSeq, cells were first sorted into bins of varying GFP to mCherry ratios, barcoded, and sequenced to reconstruct protein levels for each individual construct. (b) GFP expression levels for the library varied over ~4 order of magnitude compared to a relatively constant red fluorescence (inset). (c) 144 sequence-verified clones were individually subjected to flow cytometry analysis to act as controls. Displayed are GFP levels of two representative clones, P007-R065 (left) and P081-R062 (right), which show that individual constructs generally fall into 2-3 bins. (d) The library is split into 12 log-spaced bins based on GFP to RFP ratio (top). Individual bins have large differences in the number of cells that fall into each one (bottom).

Using DNASeq, we detected 98.5% of constructs and displayed high concordance between technical replicates ($R^2 = 0.997$; Fig. A.2). Most of the missing constructs and constructs with few DNA reads (which prevented accurate RNA level measurements) were expected to have very high expression levels indicating either growth defects or cloning issues (Fig. A.4 and Fig. A.5). RNA level calculations also showed high concordance between technical replicates ($R^2 = 0.995$; Fig. A.5). Overall, RNA levels varied by 3 orders of magnitude, but within a single promoter the coefficient of variation was only 0.63 (2.2 (left) and S7). RNASeq data also allowed us to identify dominant transcriptional start sites for most promoters (Fig. A.6). 87% of all promoters had one dominant start position (>60% of all mapped reads). Two promoters (marked with a * in Fig. A.6) had very few uniquely mapping reads, did not show a strong start site, and showed unrealistic translation efficiency calculations. These observations indicated that we were missing most of the RNA (but not protein) reads from these promoters possibly because of transcription starting after the end of the barcode sequence preventing unique identification. The 222 constructs (1.7%) containing these promoters were removed from all analyses.

Using FlowSeq, we were able to reconstruct expected protein levels for 94% of the constructs (2.2 (right)). As expected, individual constructs mostly fell into 1-3 contiguous flow-sorted bins (Fig. A.8). The average protein expression levels displayed a large range and were highly correlated with the independently characterized constructs ($R^2 = 0.94$; 2.3A and Fig. A.9). Due to the boundaries of our sorted bins, we determined that accurate quantitation was limited within a maximum and minimum range; 6.5% of the constructs were above and 14% were below this range (Fig. A.9). Again, most constructs with missing measurements (insufficient or zero reads) contained combinations of strong promoters and RBSs. We calculated average promoter and RBS strengths by averaging transcription levels and translation efficiency (protein/RNA), respectively. Promoter and RBSs were ordered and named based on their relative deviation from the average element (see Chapter A).

Finally, we spiked in 42 of the individual clones into a separate library (not analyzed here) and performed DNASeq, RNASeq, and FlowSeq to test reproducibility in biological replicates. Once again, protein levels were highly correlated with the individual measurements ($R^2 = 0.91$; Fig. 2.3C). Reconstructed values for RNA and protein levels also matched well between independent runs ($R^2 = 0.89$ and 0.90, respectively; 2.3B and D).

2.2.3 *Composability of Gene Expression.*

Our large dataset allows us to measure the extent to which combining regulatory elements led to predictable outcomes. Using a simple model for gene expression where promoter strengths determine RNA levels and RBS strengths determine

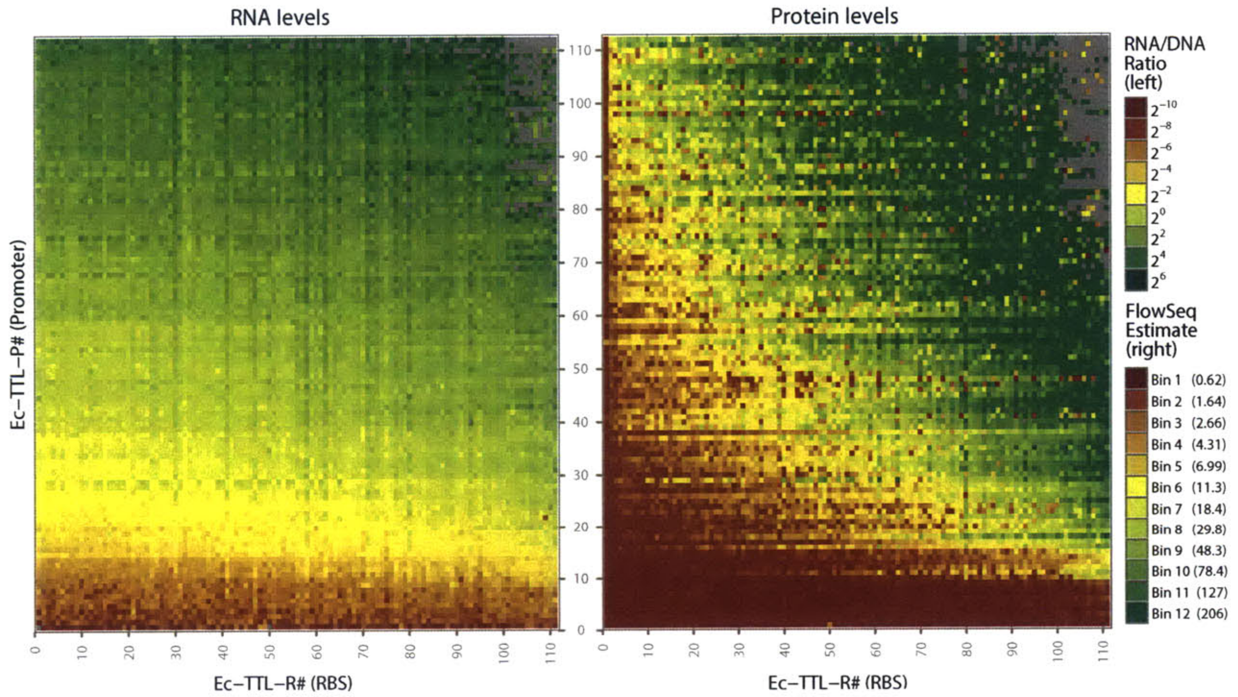


Figure 2.2: *RNA and Protein Level Grids.* The RNA (left) and protein (right) levels for all 12,653 constructs are plotted on a grid according to the identity of construct's promoter (y-axis) and RBS (x-axis). Promoters and RBSs are sorted by average RNA and protein abundance, respectively. Grey boxes indicate constructs that were below empirically determined cutoffs. Scale bars for RNA (RNA:DNA ratio) and protein (RFU of GFP:RFP ratio) levels are shown at the right.

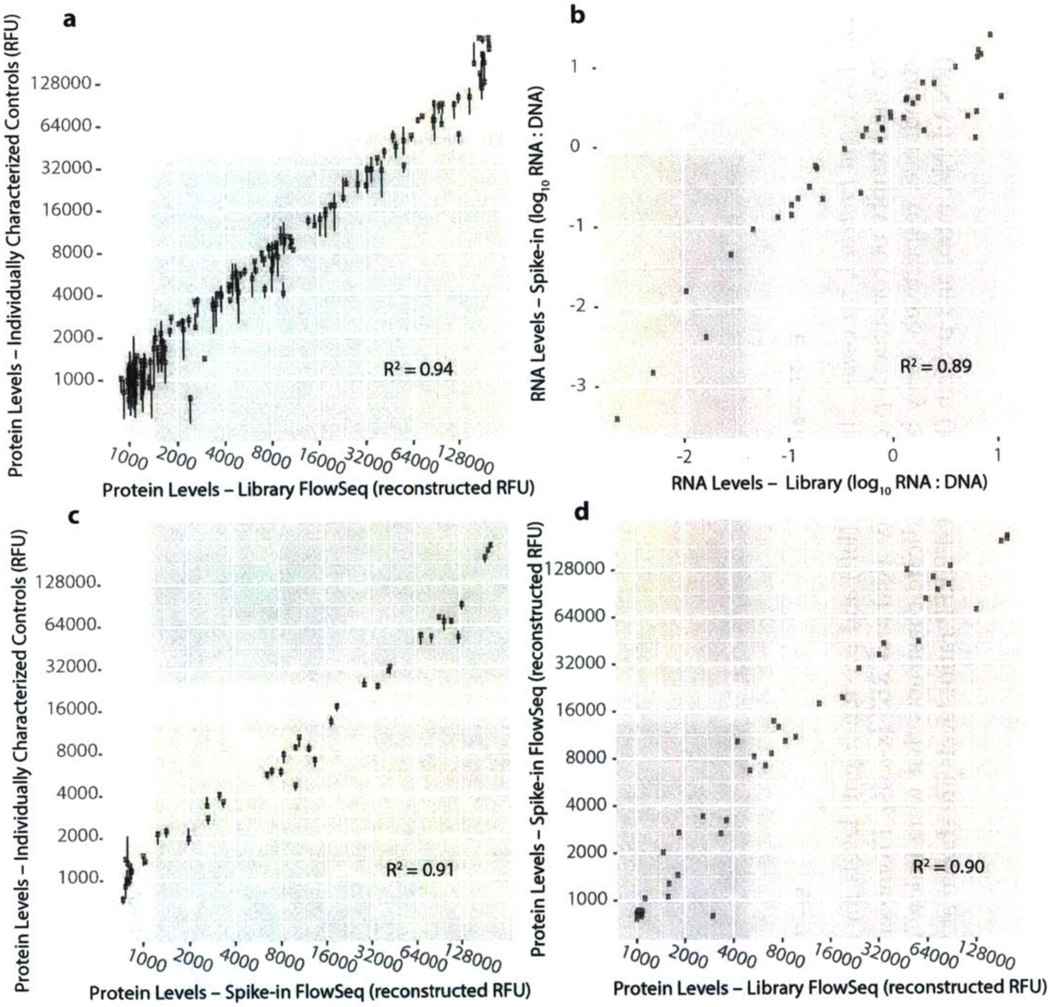


Figure 2.3: *Library measurements versus individual colony and spike-in controls.* (a) Protein levels for 141 sequence verified constructs characterized by at least two flow cytometry measurements plotted against their FlowSeq-estimated protein levels. One construct out of 142 is missing because it had insufficient reads in the FlowSeq analysis. (b) RNA levels for 41 constructs as measured in our library plotted against control constructs spiked into a separate library. One construct out of 42 is missing because it had no reads in the spike-in data. (c) Protein levels for 42 control constructs spiked into a separate library plotted against protein levels for those same constructs measured at least twice by flow cytometry. (d) Protein levels for 42 control constructs spiked into a separate library are plotted against protein level measurements as measured in our Promoter + RBS library. (All R2 for linear regressions pass an F-test with p-value: < 2.2e-16.)

	Low RNA 0.5 ± 0.13	Med RNA 2.1 ± 0.53	High RNA 6.9 ± 1.73
Low Protein 7,393 ± 1848	**107** P041-R034 P051-R032 P042-R013	**69** P084-R002 P070-R006 P061-R040	**23** P092-R022 P095-R002 P097-R039
Med Protein 39,450 ± 9863	**95** P055-R032 P017-R107 P022-R096	**178** P070-R031 P035-R107 P060-R089	**157** P086-R028 P109-R015 P094-R006
High Protein 152,484 ± 38,121	**3** P018-R110 P029-R108 P031-R102	**252** P055-R055 P049-R090 P056-R086	**338** P089-R052 P077-R100 P086-R055

Table 2.1: *Lookup table of regulatory elements for given RNA and protein levels.* We chose three levels of low (17th percentile), medium (50th percentile), and high (83th percentile) RNA and protein levels and determined how many promoter-RBS combinations fall within 25% of those desired levels. The total number of combinations that fall within each range is shown, along with three examples from each group. RNA levels are given as the measured RNA:DNA ratio and protein levels in relative fluorescence units.

translation efficiencies, we reconstructed expected expression across all constructs and compared them to measurements (2.4). We find that 80% of RNA and 64% of protein levels fall within 2x of the model predictions, and display an R^2 of 0.92 and 0.76 for RNA and protein respectively (Fig. A.10A,B).

When unexpected levels of expression do occur, they can be quite large; the largest 5% of protein model deviations are off by an average of 13-fold. Such unpredictability makes precise engineering of large systems intractable. The ease and scale of these measurements indicate that rather than using prediction or standardization to construct a single design, we can construct a library to screen for desired expression levels when optimizing large genetic systems. Desired RNA and protein levels for an entire pathway of genes could be chosen from measurements across subsets of promoters and RBSs for each gene. For example, given a desired protein level, we can choose from many sequence-divergent promoter-RBS combinations that achieve desired transcription and translation strengths of GFP (Table 2.1).

2.2.4 *Interactions between RNA and protein levels.*

We conducted a more detailed ANOVA [60] where both RNA and protein levels are independently determined by both promoter and RBS identity. This model is able to take into account effects such as the dependency of RNA levels on

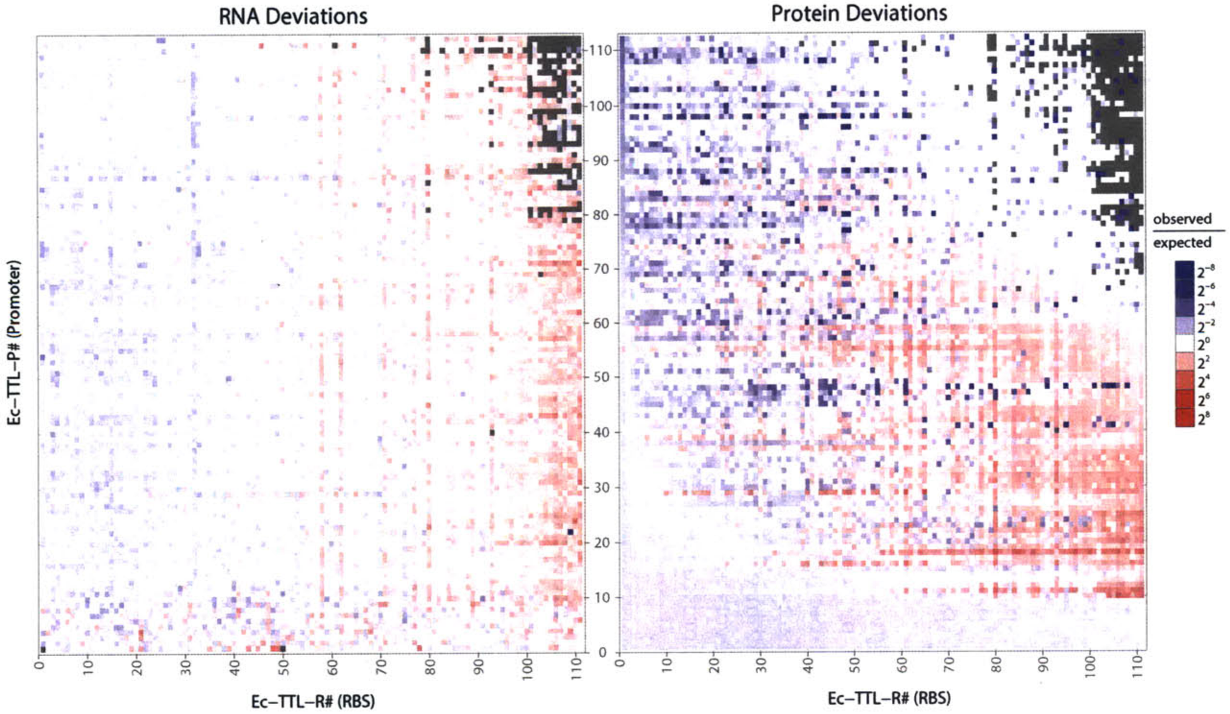


Figure 2.4: *RNA & Protein Model Deviations*. Based the promoter and RBS strengths, we calculated expected RNA (left) and protein (right) levels for each construct. Red and blue denote measured values below and above expectation and are plotted on the same scale for both plots. For constructs where expected protein levels are above or below the empirically determined thresholds, we set the prediction to be at the threshold level.

translation rate. We found that the model resulted in a modestly better fit (RNA $R^2 = 0.96$; Protein $R^2 = 0.82$, Fig. A.10C,D). Analysis of explained variance showed that 92% of the RNA levels can be explained by the promoter choice, while only 4% by the RBS choice, and the remaining 4% is unexplained (2.5A). For protein levels, both promoter choice (54% explained variation) and RBS choice (30%) are important, but a larger portion remains unexplained (16.7%). To better understand how factors such as RBS choice can affect RNA levels, we examined interactions between RNA and protein levels. For example, several previous studies in E. coli and B. subtilis have shown that for particular model transcripts, increased ribosome binding or occupancy may enhance mRNA stability [7, 102]. Such studies have been hard to interpret due to the complex interactions between the ribosome, RNA degradation machinery, and the transcript. We indeed find a significant and prevalent correlation between mRNA stability and RBS strength across all promoters. Given the size and sequence diversity of our library, it is likely that RBS strength is responsible for increased mRNA levels. Overall, we find a ~10-fold increase in translation efficiency correlates to a ~3-fold increase in RNA abundance (2.5B). However, the effect is limited at the extremes; the difference between the weakest and strongest RBSs (an 87x increase in translation efficiency) corresponds to only a ~4.3-fold increase in mRNA. As another example, many groups have found that secondary structure across the 5′ UTR and initial coding sequence can hinder effective translation [1, 28, 47, 76, 100]. In our data, we find that the correlation between secondary structure free energy across the UTR-GFP interface is significant (2.5C). However, this metric of secondary structure is neither necessary nor sufficient, as many sequences with high secondary structure do not display reductions in expected expression and vice-versa. Improved models for how secondary structure interacts with ribosome binding could increase this correlation [76].

2.3 DISCUSSION

Here we developed a method to characterize transcription and translation rates of thousands of synthetic regulatory elements simultaneously. We used this method to characterize the extent to which promoters and RBSs can be naturally composed. This large library can be used to as a basis for titrating expression using sequence divergent promoter RBS pairs for recombinant expression in E. coli and the expression data can be used to further refine models of how sequence composition determines levels of gene expression.

Here we do not examine how expression is altered by a gene?s amino acid composition and codon usage, which are known to have large effects [1, 28, 47, 60, 100]. In follow-up work we explore the influence of these two factors across a matrix of coding sequences, promoters and ribosome binding sites2. Another limitation of our current approach is that we do not examine how expression affects cellu-

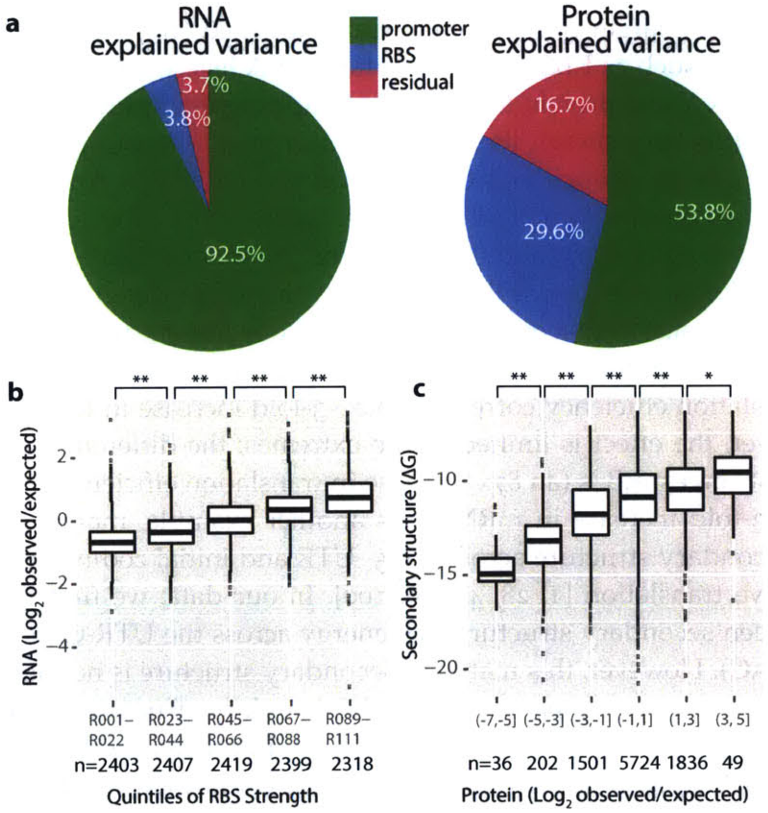


Figure 2.5: *ANOVA Explained Variance and Composition Effects of Promoter/RBS pairs.* (a) Explained variance (as percentages of sum of squared deviations) for RNA and protein measurements using ANOVA. Left pie chart shows partitioned variance for RNA measurements, while the right chart is for protein measurements. 'Residual' indicates the unexplained variance in the model. (b) Deviation from expected RNA level is correlated with RBS strength. RBSs are partitioned into 5 groups based on increasing average translation strength. (c) Free energy of a transcript's 5′ secondary structure (TSS to +30 of sfGFP) is correlated with average deviation from expected protein level. Average deviations are partitioned into 6 equal ranges. Brackets at top indicate 2-sample Student t-tests with p-values < 2e-5 (**) and < 0.02 (*). Box plot displays median with hinges indicating the first and third quartiles. Whiskers extend to farthest point within 1.5 times the inter-quartile range, with outliers shown as points.

lar growth rate. Highly expressed constructs might impair the growth rate and decrease steady-state dilution of cellular contents, which would lead to an overestimation of transcription and translation strengths. We analyze only promoter and RBS pairings here, but future studies can test large numbers of any composable genetic designs to broadly assess their effectiveness [60].

The methods developed here should be extendable to any organism that is amenable to FACS and RNASeq, such as other bacteria, yeast, and mammalian cell lines. In addition, our methods can also used to optimize more complex phenomena including inducible expression, gene circuits, and time-dependent responses. Finally, improvements in the quality and length of synthetic oligo pools can also extend such analyses to the characterization of regulatory protein variants or longer-range interactions.

2.4 MATERIALS AND METHODS

2.4.1 *Strains, Library Construction, and Growth Conditions*

We used E. coli MG1655 (Yale CGSC No. 6300) for all experiments. The oligo library was constructed by Agilent Technologies (USA) using their Oligo Library Synthesis (OLS) process [51]. The design of pGERC is based on the synthetic plasmid pZS-123 [22], which allows independent expression from three promoters, and was synthesized by DNA 2.0 (USA). The amplified OLS pool was first subcloned into 5-alpha Electrocompetent E. coli (NEB) (giving an initial library size of ~600,000 colonies), purified, and re-transformed into MG1655 and several aliquots were frozen. Overnight cultures from both pooled experiments and individual clones were first diluted 1000x grown at 30°C in LB-Miller media shaking at 250 RPM for 2-3 hours until reaching an OD(600nm) of 0.15-0.25. Detailed information can be found in ChapterA.

2.4.2 *DNASeq and RNASeq*

From a single 300 mL culture of the library, pellets from four 50 mL aliquots of culture were frozen in liquid nitrogen with the remaining culture saved for FlowSeq. Two technical replicates of DNA and RNA were isolated by Qiagen DNA and RNA Midiprep Kits (USA). Ribosomal RNA was removed by Ribo-Zero rRNA removal kit for meta-bacteria (Epicentre, USA). 5′ triphosphates were monophosphorylated by 5′ polyphosphatase (Epicentre) and then ligated to an RNA adaptor using T4 RNA Ligase (Epicentre). First strand cDNA was made from a specific primer in sfGFP. Both DNA and cDNA were amplified and monitored by real-time PCR to prevent over-amplification. Illumina adaptors and barcodes were then added, and sequencing was performed on a HiSeq 2000 in two separate PE100 lanes. A separate library that contained spike-ins from the 42 colonies underwent the same procedure. Detailed information can be found in ChapterA.

2.4.3 *FlowSeq*

We used 50mL of the library culture as prepared above for analysis by FlowSeq. We flow-sorted the cells into 12 log-spaced bins in three sequential runs sorting four bins each. Cells were then grown overnight to saturation and plasmid prepped by Qiagen Miniprep kit. A small aliquot was diluted, regrown, and subjected to flow cytometry to verify proper sorting. All data from library measurements are reported in GFP:RFP ratio units, which range from 1 to 255,000. The 12 minipreps were amplified again by real-time PCR, barcoded, and sequenced on a single lane PE100 on a HiSeq 2000. Detailed information can be found in ChapterA.

2.4.4 *Data Analysis*

Reads from all experiments were first aligned using SeqPrep [87] to form paired-end contigs for improved accuracy. Custom software was written to identify unique contigs and map them to library members using bowtie [50] and grep. DNASeq and RNASeq contigs were counted where reads mapped uniquely and contained less than 3 mismatches. In addition, DNA contamination from RNASeq reads were identified and removed. Statistics, graphs, and tables were all generated using custom software written in Python, R, and the ggplot2 package. Detailed information can be found in ChapterA.

3

CAUSES AND EFFECTS OF N-TERMINAL CODON BIAS IN BACTERIAL GENES.

This chapter is reproduced with permission from its initial publication:

Daniel B. Goodman, George M. Church, and Sriram Kosuri. *Causes and Effects of N-Terminal Codon Bias in Bacterial Genes.* **Science** (New York, NY), September 2013.

Author Contributions and Acknowledgements:

D.B.G, G.M.C. and S.K. conceived the project, analyzed the data, and reviewed the manuscript. D.B.G. and S.K. designed and performed all experiments and wrote the manuscript. D.B.G. performed all computational analyses.

We thank J. C. Way, E. R. Daugharthy, and R. T. Sauer for comments. The research was supported by the U.S. Department of Energy (DE-FG02-02ER63445 to G.M.C.), NSF SynBERC (SA5283-11210 to G.M.C.), Office of Naval Research (N000141010144 to G.M.C. and S.K.), Agilent Technologies, Wyss Institute, and an NSF Graduate Research Fellowship to D.B.G. Data can be accessed on the National Center for Biotechnology Information, NIH, Sequence Read Archive (SRA) (SRP029609). pGERC reporter can be obtained from AddGene (#47441). Accession numbers: The Project accession at the SRA is SRP029609. The sample accession is SRS477429. There are three experiments, one for DNA, one for RNA, one for FlowSeq: RNA, SRX346948; DNA, SRX346944; and FlowSeq, SRX346268.

ABSTRACT

Most amino acids are encoded by multiple codons, and codon choice has strong effects on protein expression. Rare codons are enriched at the N terminus of genes in most organisms, although the causes and effects of this bias are unclear. Here, we measure expression from >14,000 synthetic reporters in Escherichia coli and show that using N-terminal rare codons instead of common ones increases expression by ~14-fold (median 4-fold). We quantify how individual N-terminal codons affect expression and show that these effects shape the sequence of natural genes. Finally, we demonstrate that reduced RNA structure and not codon rarity itself is responsible for expression increases. Our observations resolve controversies over the roles of N-terminal codon bias and suggest a straightforward method for optimizing heterologous gene expression in bacteria.

3.1 INTRODUCTION

Codon usage is biased in natural genes and can strongly affect heterologous expression [67]. Many organisms are enriched for poorly-adapted codons at the N-terminus of genes[1, 11, 65, 94]. Several studies suggest that these codons slow ribosomal elongation during initiation and lead to increased translational efficiency [52, 65, 94]. Most organisms also display reduced mRNA secondary structure at the N-terminus [28], and studies using synthetic codon gene variants have resulted in conflicting theories on which mechanisms are causal for expression changes [47]. Information about the causes and effects of codon bias has been restricted to relationships inferred from natural sequences using genome-wide correlation[1, 11, 70, 78, 94], conservation among species [65], or relatively small libraries of synthetic genes with synonymous codon changes [1, 47, 62, 89, 95, 100, 107]. Here, we separate and quantify the factors controlling expression at the N-terminus of genes in E. coli by building and measuring expression from a large synthetic library of defined sequences.

3.2 RESULTS

We used array-based oligonucleotide libraries [51] to generate 14,234 combinations of promoters, ribosome binding sites (RBSs), and 11 N-terminal codons in front of super-folder GFP (sfGFP) on a plasmid that constitutively co-expresses mCherry (Fig. B.1) [43, 66, 79]. The sequences for the N-terminal peptides correspond to the first 11 amino acids (including the initiating methionine) of 137 endogenous E. coli essential genes [101] that utilize the entire codon repertoire (Fig. B.2). We expressed these sfGFP fusions from two promoters and three RBSs of varying strengths [43]. We also included the natural RBS for each endogenous gene. For each combination of promoter, RBS, and peptide sequence, we designed a set of 13 codon variants to represent a wide range of codon usages and secondary structure free energies across the translation initiation region. We studied the interactions between the 5′ untranslated region (UTR) and N-terminal codon usage because initiation is thought to be the rate-limiting step for translation [67], this region has been previously implicated in determining most expression variation [47], N-terminal codons are more highly conserved [31], and rare codons are enriched at the N-terminus of natural genes and especially those that are highly expressed [94].

3.2.1 *Library Measurement*

We measured DNA, RNA, and protein levels from the entire library using a multiplex assay (Figs. 3.1C, B.3, B.4) [43]. DNA and RNA levels were determined using DNASeq and RNASeq. Protein levels were determined by FlowSeq; 7327

(51.5%) constructs were within the quantitative range of our assay ($R^2 = 0.955$, $p < 2 \times 10^{-16}$; Fig. B.5). We normalized the expression measurements across each 13-member codon variant set as fold change from log-average to control for changes in promoters, RBSs, and peptide sequence (Fig B.6).

3.2.2 *Rare codons correlate with increased expression*

Changing synonymous codon usage in the 11-aa N-terminal peptide resulted in a mean 60-fold increase in protein abundance from the weakest to strongest codon variant even though >96% of the gene remained unchanged. For over 160 codon variant sets (25% of sets within range), the difference was >100-fold. For each codon variant set, we included sequences encoding the most common or rare synonymous codon in E. coli for every amino acid. The rare codon constructs displayed a mean 14-fold (median 4-fold) increase in protein abundance compared to common codon constructs (Fig. 3.1A; $p < 2 \times 10^{-16}$, two-tailed T-test) even though common codons are generally thought to increase protein expression and fitness [26, 67, 70, 82].

3.2.3 *Codon Usage Metrics, Codon Ramp, and Motif Analysis*

To understand why rare codons cause increased expression, we first examined several codon usage metrics, but they could only explain <5% of expression differences (Fig B.7A). New metrics that take into account both tRNA availability and usage (nTE) show stronger N-terminal enrichment [65]. We calculated nTE scores for E. coli and found that nTE scores were similar to the tRNA adaptation index (tAI) ($R^2 = 0.847$, $p < 2 \times 10^{-16}$), did not correlate well with N-terminal codon enrichment in the E. coli genome ($R^2 = 0.107$, $p = 0.00654$), and did not significantly correlate with codons that increased protein expression in our data set ($R^2 = 0.024$, $p = 0.124$). Others have proposed that slow ribosome progression at the N-terminus due to rare codons increases translational efficiency [62, 94, 95]. This 'codon ramp' hypothesis should apply primarily in the context of strong translation, but we found that using rare codons at the N-terminus increases expression regardless of translation strength (Fig. 3.1B). Finally, ribosome occupancy profiling in E. coli has shown that tRNA abundance does not correlate to translation rate, but that specific rare codons can create internal Shine-Dalgarno-like motifs that can alter translational efficiency [52]. We looked for an association between the presence of internal Shine-Dalgarno-like motifs and changes in expression, and found it to be weak but statistically significant ($R^2 = 0.002$, $p < 1.3 \times 10^{-5}$).

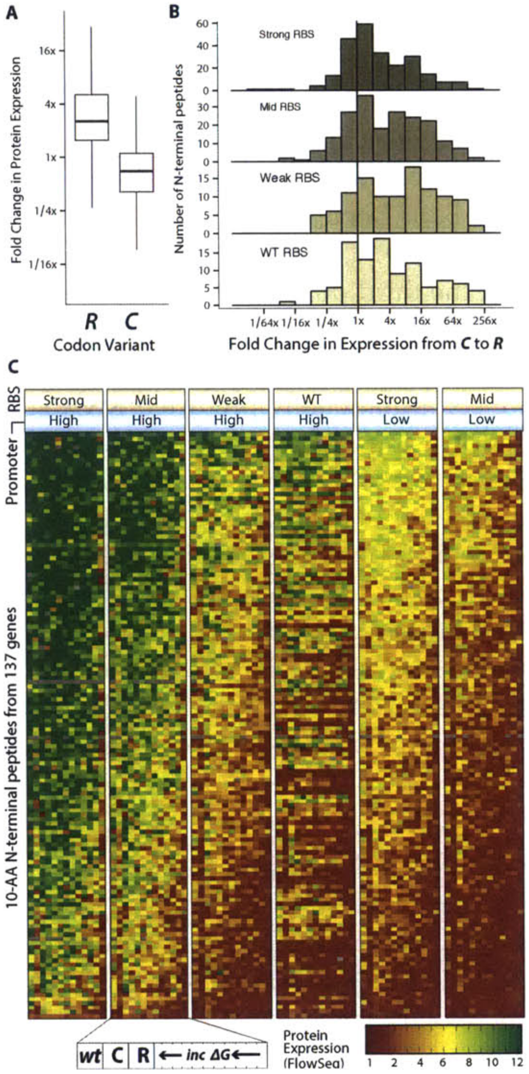


Figure 3.1: *Gene expression measurements of the reporter library.* (A) N-terminal peptide sequences encoding the most rare (R) codon variants show increased expression when compared to the most common ones (C). (B) Fold change in expression between C and R codon variants is largely independent of RBS strength. (C) Protein expression of the library (as measured by the sfGFP:mCherry ratio) covers a ~200-fold range. 13-member codon variant sets are grouped into columns by promoter/RBS combination (right). Codon variants include C, R, wild-type sequence (wt), and 10 sequences with varying secondary structure (ΔG). Not shown are two additional low promoter panels, which were mostly outside the quantitative FlowSeq range. Dark gray squares had insufficient data, and light gray squares correspond to duplicate constructs.

3.2.4 *Individual Codon Effects*

We built a simple linear regression model correlating the use of each individual synonymous codon with expression changes (Fig. 3.2A, Fig. B.8). For most amino acids, we found a link between the rarity of the codon and increased expression (Fig. 3.2B). There is a strong correlation between codons that affected expression and their relative N-terminal enrichment in E. coli ($R^2 = 0.73$, $p < 2.3 \times 10^{-9}$; Fig. 3.2C). Using relative translation efficiency instead of relative expression produced similar results (Fig. B.9). Decreased GC-content correlated with increased protein expression ($R^2 = 0.12$, $p < 2 \times 10^{-16}$; Fig. 3.3A). Rare codons in E. coli are frequently A/T-rich at the third position, and codons ending in A/T more frequently correlate with increased expression than synonymous codons ending in G/C. (Fig. B.10). This association suggested a link to mRNA transcript secondary structure [47], and so we computationally predicted RNA structure over the first 120 bases of each transcript using NUPACK [103]. We found that increased secondary structure was correlated with decreased expression, explaining more variation than any other variable we measured ($R^2 = 0.34$, $p < 2 \times 10^{-16}$; Fig. 3.3A). We made a similar linear regression model relating individual codon substitution to change in secondary structure free energy rather than expression levels, and found a strong correlation between codons that decreased secondary structure and those that increased protein expression ($R^2 = 0.87$, $p < 2 \times 10^{-16}$; Fig. 3.3B). Additionally, codon adaptation metrics at the N-terminus correlate as well to change in secondary structure free energy as they do to change in protein expression (Fig. B.7B).

3.2.5 *mRNA secondary structure at N-terminus is responsible*

We used multiple regression to control for the secondary structure changes between codon variants and found that no relationship remained between N-terminal codon adaptation and increased expression ($R^2 = 0.05$, $p = 0.197$; Fig. 3.3D). Additionally, constructs with constant tAI still show a correlation between expression and secondary structure, but constructs with constant secondary structure have no correlation between tAI and expression. (Fig. 3.3E, F). Finally, if secondary structure is the dominant factor, we would expect a disproportionate enrichment of A over T due to G-U wobble pairing. Indeed, nucleotide triplets with A at the wobble position were more consistently correlated with expression our dataset and with enrichment at the N-terminus of E. coli genes (Fig. B.3).

Kudla et al. show that local RNA structure in the region between -4 to +38 of translation start is most correlated with expression change [47]. Our data indicate that the region centered on +10 is most correlated with expression changes (Fig. 3.4, Figs. B.12,B.13,B.14), closely matching in-vitro translation studies [96]. This region remained the most correlated for the subset of constructs with no change in

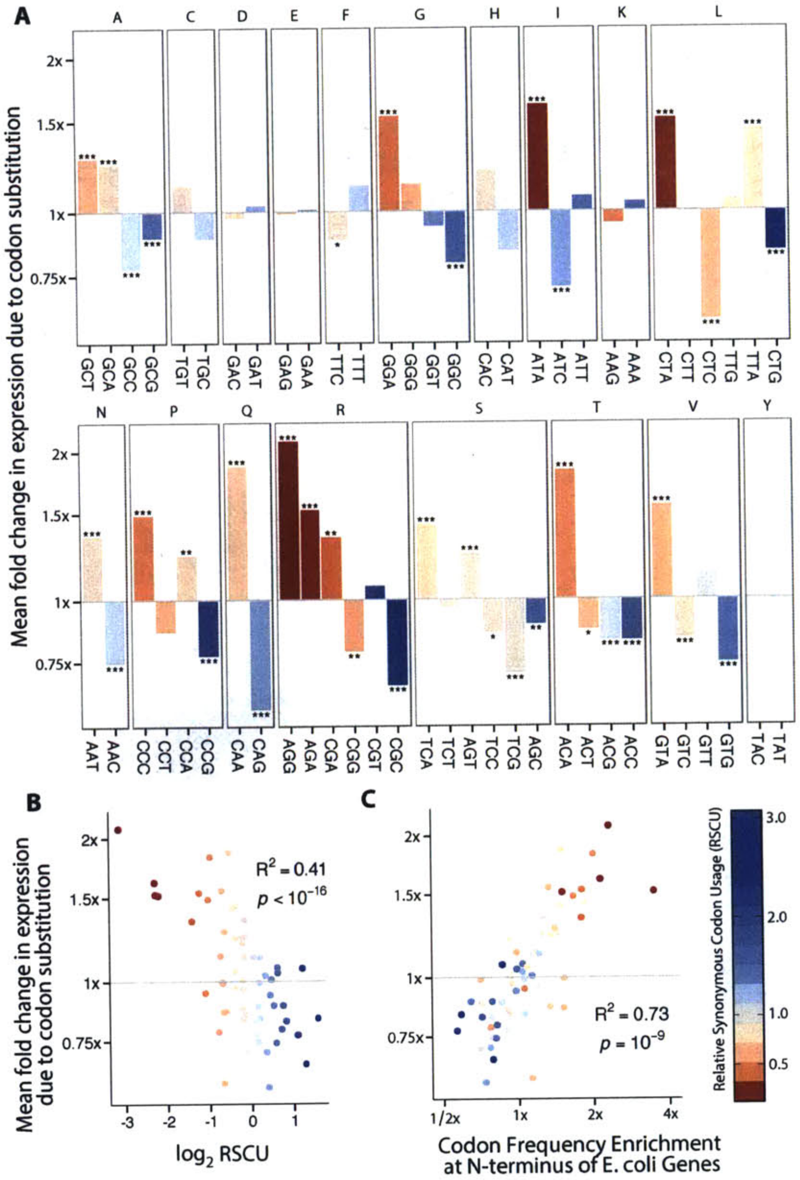


Figure 3.2: *The average fold change in expression is correlated with the choice of codon.* The y-axis is the slope of a linear model linking codon use to expression change. Codons are sorted left to right by increasing genomic frequency, and colored according to their relative synonymous codon usage (RSCU) in E. coli. (p-values after Bonferroni correction: *: p < 0.05, **: p < 0.005, ***: p < 0.001). (B) The individual codon slopes (y-axis) as in (A) show an inverse relationship with RSCU (x-axis). (C) The individual codon slopes correlate with enrichment of codons at the N-terminus of genes in E. coli.

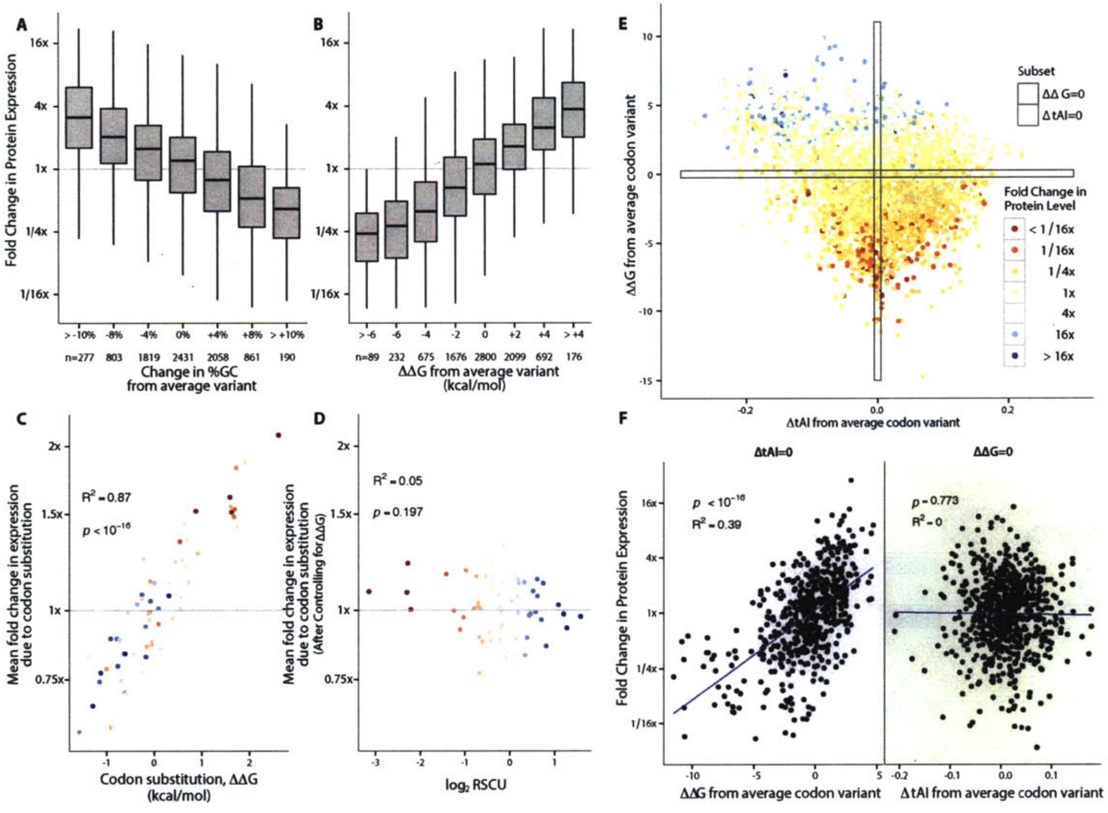


Figure 3.3: *Rare codons alter expression by reducing mRNA secondary structure.* (A) Expression changes are correlated with relative changes in %GC content. Each boxplot includes +/- 2% of centered value. (B) Expression increases correlate to relative increases in free energy of folding at the front of the transcript ($\Delta\Delta G$). Each boxplot includes +/- 2 kcal/-mol of centered value. (C) Individual codon slopes (same as Fig. 3.2A y-axis) correlate with the $\Delta\Delta G$ per individual codon substitution. (D) After controlling for $\Delta\Delta G$ with a multiple linear regression, there is no longer any relationship between individual codon slopes and RSCU (compare with Fig. 3.2B). (E) The $\Delta\Delta G$ versus change in tAI is plotted for all constructs within the quantitative range. Constructs are colored by their relative fold change in expression from the average codon variant within the set. (F) The two lower panels show subsets of constructs corresponding to the shaded boxes in (E). The left panel shows points with constant codon adaptation and varied secondary structure, while the right panel shows points with constant secondary structure and varied codon adaptation.

total free energy of folding across the N-terminal region (Figs. B.15,B.16). While secondary structure is known to affect the RBS [86], when altering only codon usage, RNA structure after the start codon, and not at the RBS, is the major contributor to expression differences. A multiple linear regression model that combines promoter and RBS choice, as well as N-terminal secondary structure and GC content still explains only 54% of variation in expression levels. Amino acid composition effects on sfGFP folding and inadequacies in computational RNA structure prediction could be partially responsible. However, there are likely additional effects left to uncover, and the extent to which codon usage beyond the N-terminal region alters gene expression remains unresolved [47, 95].

3.3 DISCUSSION

The N-terminus of genes in almost all bacteria display reduced secondary structure, but enrichment of poorly-adapted N-terminal codons are only found in bacteria with GC content of at least 50% [1]. Recent work further shows that AT-rich codons as opposed to rare codons themselves are preferentially selected, thus implicating secondary structure as the driving force for N-terminal codon selection in most bacteria [11]. Despite mechanistic differences in translation between prokaryotes and eukaryotes, both single- and multi-cell eukaryotes also have reduced N-terminal secondary structure [28]. For synthetic GFP templates in yeast, secondary structure is more correlated with expression changes than codon adaptation metrics [78]. Here, we do not examine other factors that might shape natural sequence such as codon pair bias [20, 67], co-translational folding [41, 65, 107], or growth conditions [89, 100]. Natural genomic sequence is often not suited to distinguish between conflicting hypotheses of how sequence affects function; multiplexed assays of large synthetic DNA libraries provide a powerful method to examine such hypotheses in a controlled manner.

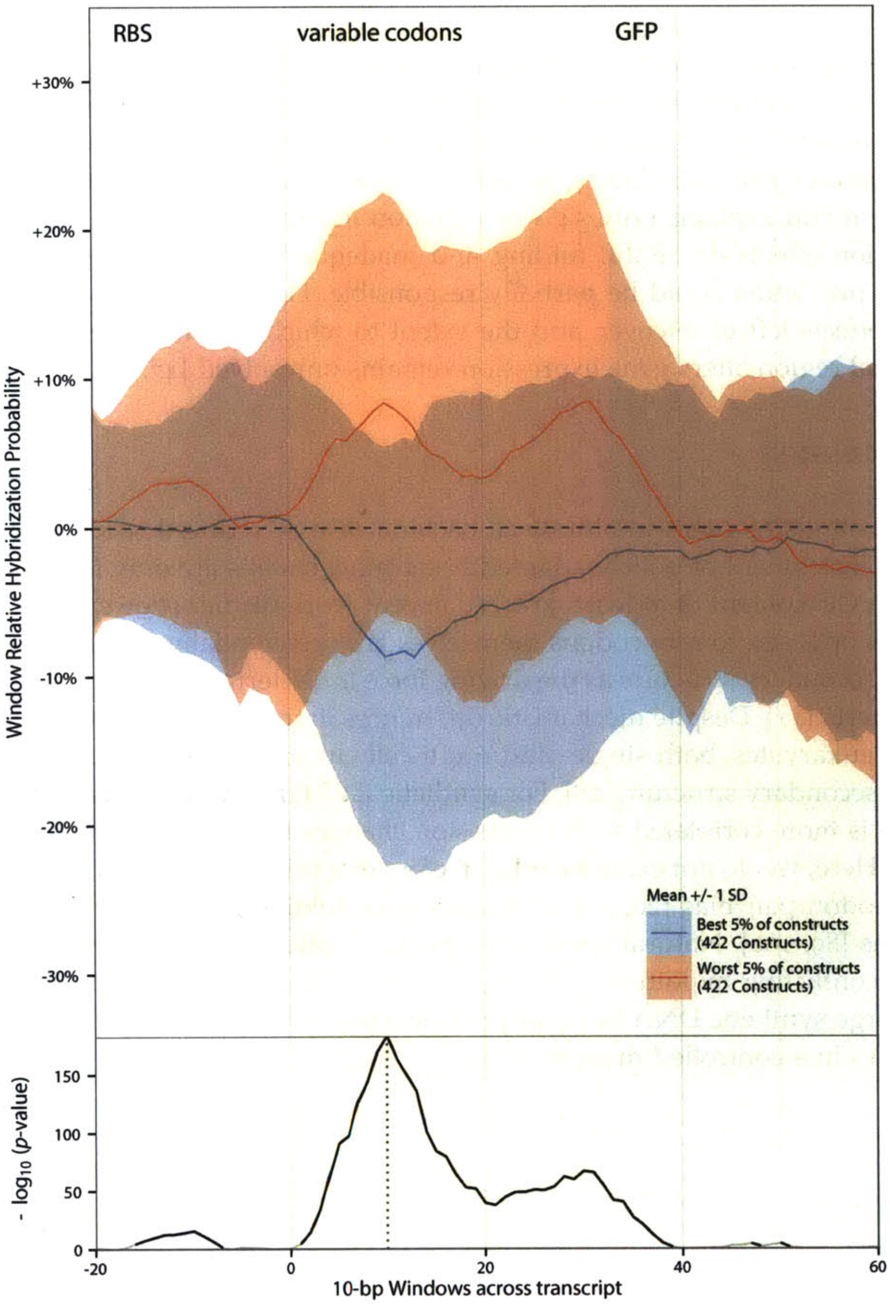


Figure 3.4: *mRNA structure downstream of start codon is most correlated with reduced expression.* Relative hybridization probabilities averaged in 10nt windows are plotted against their correlation with expression change as a function of position (-20 to +60 from ATG). In the top panel, the best and worst 5% of constructs – as ranked by relative expression within a codon variant set – are grouped and plotted as blue and red ribbons, respectively. The ribbon tops and bottoms are one standard deviation from the mean, which is shown as a solid line. The bottom panel shows the p-value for linear regressions correlating hybridization probabilities within each window to expression fold change in all constructs.

4

HIGH-THROUGHPUT DESIGN AND MEASUREMENT OF TRANSCRIPTIONAL REGULATION

This chapter contains in progress and currently unpublished work.

Daniel B. Goodman[1], Casper Enghuus[1], Max Schubert, George M. Church. *High-Throughput Design and Measurement of Transcriptional Regulation.* In preparation.

Author Contributions and Acknowledgements:

D.B.G. devised the study with input from C.E. and G.M.C. D.B.G, C.E., and M.S performed the research. D.B.G and C.E. analyzed the data with input from all authors.

[1] (co-first-author)

ABSTRACT

Despite our deep knowledge of the individual DNA motifs and proteins involved in transcriptional regulation, our ability to identify and predict the function of transcriptional regulatory elements remains limited. Here we computationally designed and synthesized 135,016 divergent dual-promoter transcriptional circuits for 8 different prokaryotic transcription factors (TFs), and then measured repression, activation, and small-molecule induction across the entire library in a single experiment. Our system allowed measurement of input/output induction and repression curves for over 5,000 cis-regulatory regions in *E. coli*, each containing one or more TF binding sites. We quantify how binding site strength, location, and coordination affect functional properties like cooperativity, sensitivity, and dynamic range of gene expression response. Across all transcription factors and promoter backgrounds we found functional regulatory elements that can be created from existing sequence by as few as 5 single nucleotide changes, suggesting that new transcriptional regulatory elements are shallow in sequence space and easily accessible to evolution.

4.1 INTRODUCTION

Transcriptional cis-regulatory elements are the primary means by which cells control gene expression and direct information flow. The architecture and interactions of these elements underpin the behavior of natural genetic networks and the adaptation of organisms to new environments. In bacteria, such elements include transcription factor binding sites and promoters. Bacterial genetic elements choreograph all aspects of the cell, including genetic programs for virulence, metabolism, and antibiotic resistance. In addition to studying the evolution and function of natural elements, knowledge of how to quantitatively design and tune synthetic genetic elements is of interest to biotechnology. For example, they are used in small-molecule sensors in diagnostics and biomanufacturing, where precise measurement of proteins and metabolites or control of input-output relationships are required.

As non-coding DNA, cis-regulatory elements lack the defined codon-based structure of the genes they control, and are instead loosely treated as ensembles of short sequence motifs to which various proteins bind, recruiting RNA polymerase or inhibiting the initiation of transcription. The strength and relative positioning of these motifs is the primary means by which they are identified as functional. However, transcriptional regulation and initiation is an intricate process that relies on the interactions of numerous DNA and protein components[14, 75]. These DNA sequences and their protein partners have been studied since the discovery of the *lac* operon by Jacob and Monod [34, 35] and the dissection of the *cI* repressor in the lambda phage by Ptashne and colleagues[68]. However, even in humble *E. coli,* new insights on the relationships between regulatory sequence and function are still being discovered[97]. Though many previous studies have attempted to understand the architecture of regulatory elements by randomizing short sequences[39, 64, 85] and combinatorially assembling small pre-existing DNA fragments[29, 54], we still do not possess the knowledge to quantitatively predict the strength and function of genetic elements from sequence[43] nor can we effectively engineer new functional elements from knowledge of motifs alone.

4.2 EXPERIMENTAL DESIGN

4.2.1 *Design of the Expression System and Choice of Transcription Factors*

Here, we use a large library of computationally-designed dual-promoter regions to explore the quantitative relationship between cis-regulatory sequence, transcription factor expression, and gene expression output in a single experiment. Each cis-regulatory region is synthesized as an oligonucleotide which is cloned into a plasmid containing one of 8 distinct transcription factor genes and a superfolder green fluorescent protein (sfGFP). Each oligonucleotide contains an upstreamfacing promoter (the P_L promoter) driving the transcription factor gene while a second downstream-facing promoter (the P_R promoter) simultaneously drives the sfGFP reporter construct. A constitutively-expressed mCherry gene elsewhere on the plasmid serves as a copy-number control (Fig.4.1A). By adding a transcription factor binding site to the right promoter and varying the strength of the left promoter, we measure a response function which relates the expression of the transcription factor to the expression of sfGFP (Fig.4.1C) so that repression or activation of sfGFP expression by the TF can be measured. We chose 8 different transcription factors, including one built from a synthetic zinc-finger array, four different *cI* repressors from lambda-like phages[10, 15, 16, 45, 58], the well-studied inducible transcription factors *lacI* and *tetR*, and the transcription factor *acuR*, which responds to the presence of acrylate, a chemical monomer used in the manufacture of bioplastics[72, 73]. For 3 transcription factors which respond to small molecules (lacI, tetR, and acuR), we additionally examine the response of the systems to various concentrations of inducer molecule (Fig.4.1B, Table C.1).

4.2.2 *Multiplex Sequence Design using* Promuter

To design our cis-regulatory regions, we took a high-throughput rational forward-engineering approach. We built a software package called *Promuter* which computationally generates and scores promoters to meet specified sequence constraints (Section C.1.1 and Figure C.1). We generated 8,430 P_R promoters which vary in their basal transcription strength as well as the identity, strength, location and multiplicity of their transcription factor binding sites. These promoters were then combinatorially paired with P_L promoter variants in silico to generate 135,016 230bp oligonucleotide sequences, so that each oligo encodes two divergent promoters separated by a strong terminator (Fig.4.1A). These oligos were then synthesized in a pooled library using Agilent OLS[51]. We include a variety of control constructs to test that there is minimal interaction between the P_R and P_L regulatory regions and to separately measure P_R and P_L expression outside of the divergent promoter context. We also include sublibraries of all single base pair mutants for the three right-facing 'base' promoters to which the TF binding sites are added (P_{R0-R2}) and all single base pair mutants for a separate set of natural promoters, each known to respond to one TF.

4.3 RESULTS

4.3.1 *Library Measurement using FlowSeq*

We simultaneously measured sfGFP expression level from all constructs using *FlowSeq*, a multiplex assay described previously[43]. *FlowSeq* allows us to perform discretized flow cytometry measurements from a large library of distinct DNA

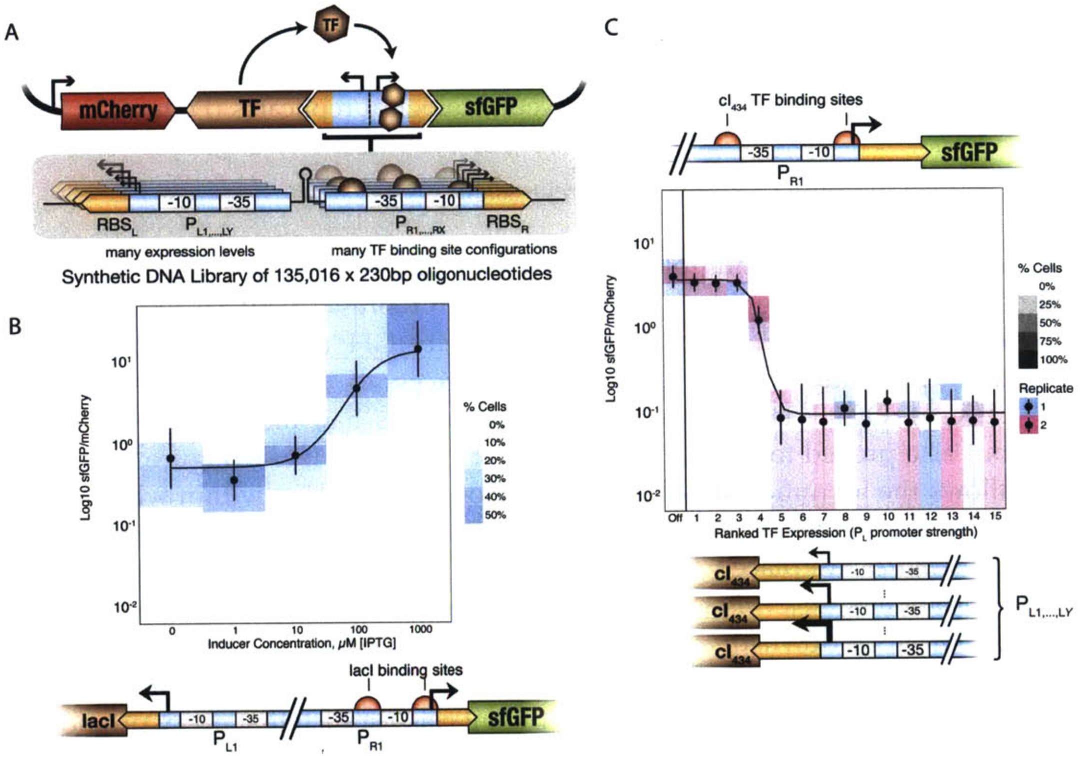


Figure 4.1: Measuring repression and` induction across a library of synthesized dual-promoter constructs. (A) Each synthesized oligonucleotide contains two divergent promoters cloned into a plasmid with a sfGFP reporter and a transcription factor (TF). An upstream P_L promoter of varying transcriptional strength drives the TF and an upstream P_R promoter drives sfGFP, and each P_R promoter variant is computationally designed to add one or more TF binding sites. Cells are transformed in a library so that each contains copies of a single plasmid with an oligonucleotide containing a $P_L P_R$pair. The ratio of sfGFP to mCherry expression is measured to control for plasmid copy number. (B) Constructs are measured by FlowSeq, in which cells are FACS-sorted into quantitative bins with different sfGFP/mCherry ratios (colored regions along y axis) followed by pooled plasmid sequencing per bin. Performing the library measurement in different concentrations of a small molecule inducer (x axis) generates an induction response curve for each $P_L P_R$ library member. (C). Comparing multiple oligonucleotide constructs with a single P_R promoter and a series of increasingly strongP_L promoters generates a repression curve.

constructs simultaneously by separating cells into quantitative bins using FACS and then sequencing the bins separately. FlowSeq *was performed* in two distinct biological replicates. For 3 transcription factors which respond to small molecules (lacI, tetR, and acuR), we additionally performed *FlowSeq* in media containing 4 different concentrations of the inducer molecule (IPTG, aTC, and acrylic acid, respectively). Our measurements were highly accurate across a ~2000-fold range of sfGFP expression, both compared to a subset of individually measured flow-cytometry controls ($R^2 = 0.957$) and between biological replicates (Figure C.2).

4.3.2 *Factors controlling transcriptional repression*

This in silico design and multiplex measurement approach successfully identified over 700 synthetic cis-regulatory sequences which repress sfGFP expression with ≥10-fold dynamic range. 300 promoters repressed sfGFP expression over 30-fold, and several promoters were identified with an excess of 300-fold repression (Figures 4.2, C.4). Of all TFs measured, the*cI* repressors had the highest dynamic range but also proved to be toxic at the highest expression levels (Figure C.3); our assay allows the identification of the window of TF expression within which they maximally repress the promoter but do not incur a fitness cost to the cell.

Using *Promuter*, we placed TF binding sites at various locations throughout the 3 'base' promoters, $P_{R0} - P_{R2}$, and categorized each individual placement as upstream, central, or downstream, relative to the core -35 and -10 motifs of theσ_{70} RNA polymerase subunit. 'Double' TF placements were generated from combinations of individual placements in these different classes, such that every promoter with two TFBS is made by combining the mutations required to generate two individual binding sites. Across all 8 TFs, we identified strong repression in all binding configurations, but saw clear trends in the effects of individual placements and their combinations (Figures 4.2, C.4). For example, TFBS placement upstream of the -35 do not strongly repress transcription by themselves, but when placed in combination with a secondary binding site either central to or upstream of the core promoter, they increase the fold-repression in a cooperative manner. As expected, adding two strong TF binding sites generate promoters that repress more strongly, are more sensitive, and are more cooperative in their response to TF concentration (Figure C.6).

Fold repression correlates well to motif strength as compared to the consensus (calculated from literature and natural sequences), although some promoters seem to successfully repress despite their weak TFBS motifs, while for others, strong motifs are necessary but not sufficient. Indeed, many motifs predicted to be strong do not repress expression, or repress well when located in some positions but not others. Across all TFs there are no major trends for TFBS positioning (Figure 4.3A), but individual TFs seem to have positional preferences when only strong motifs are considered (Figure C.7).

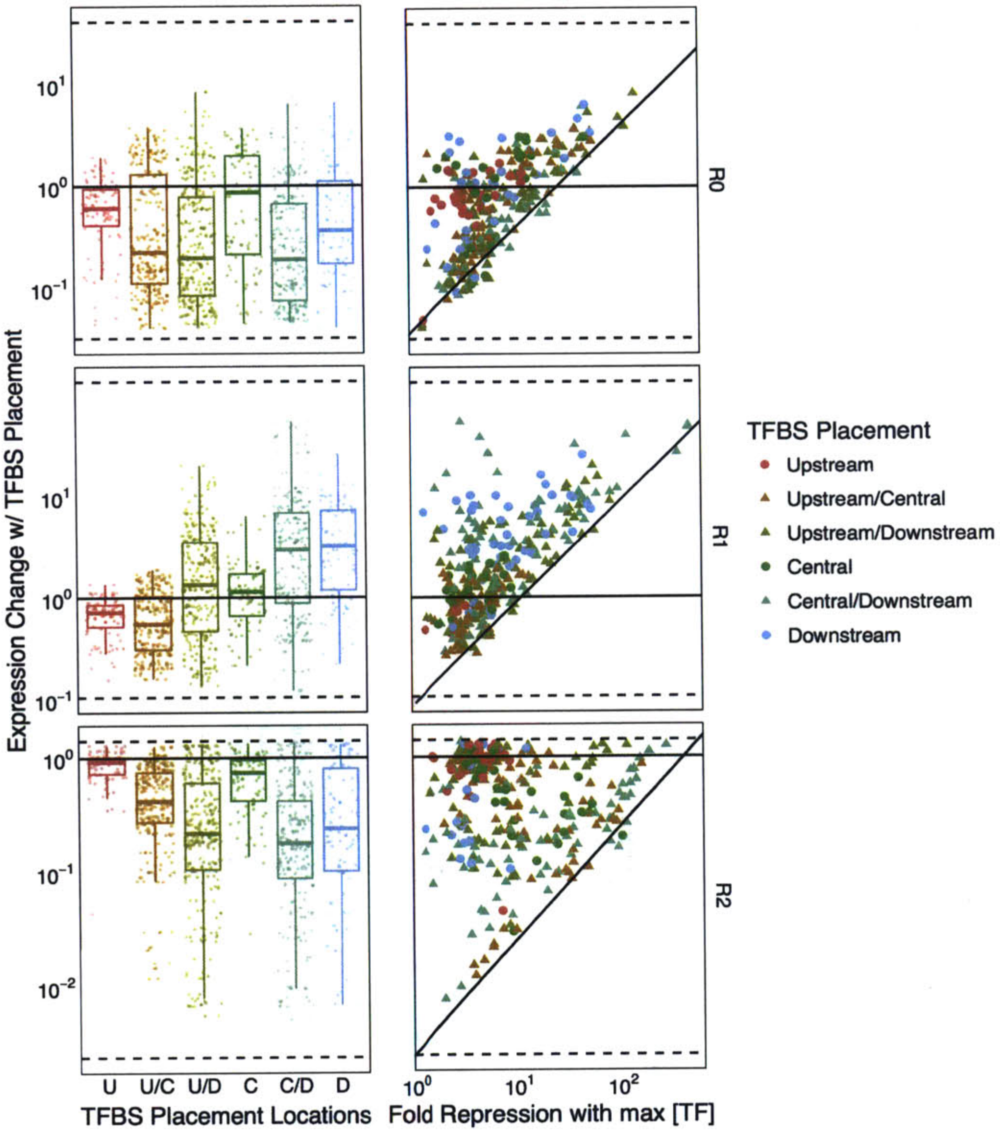


Figure 4.2: *Influence of TF Binding Site Placement on Basal Promoter Strength.* (Left) Placing single transcription factors has different effects on basal promoter strength (y axis), which in turn alters the dynamic range of potential repression. These effects depend more on promoter context than TF identity. U: Upstream, C: Central, D: Downstream. Upstream/Downstream etc. refer to combinations of multiple binding sites. (Right) TF binding sites can successfully repress transcription when placed in almost all locations, but the most successful configurations are the ones that do not negatively impact basal transcription. The dotted lines correspond to the maximal fold-change detectable vs. background fluorescence. The diagonal line corresponds to the maximum fold-repression detectable vs. background fluorescence.

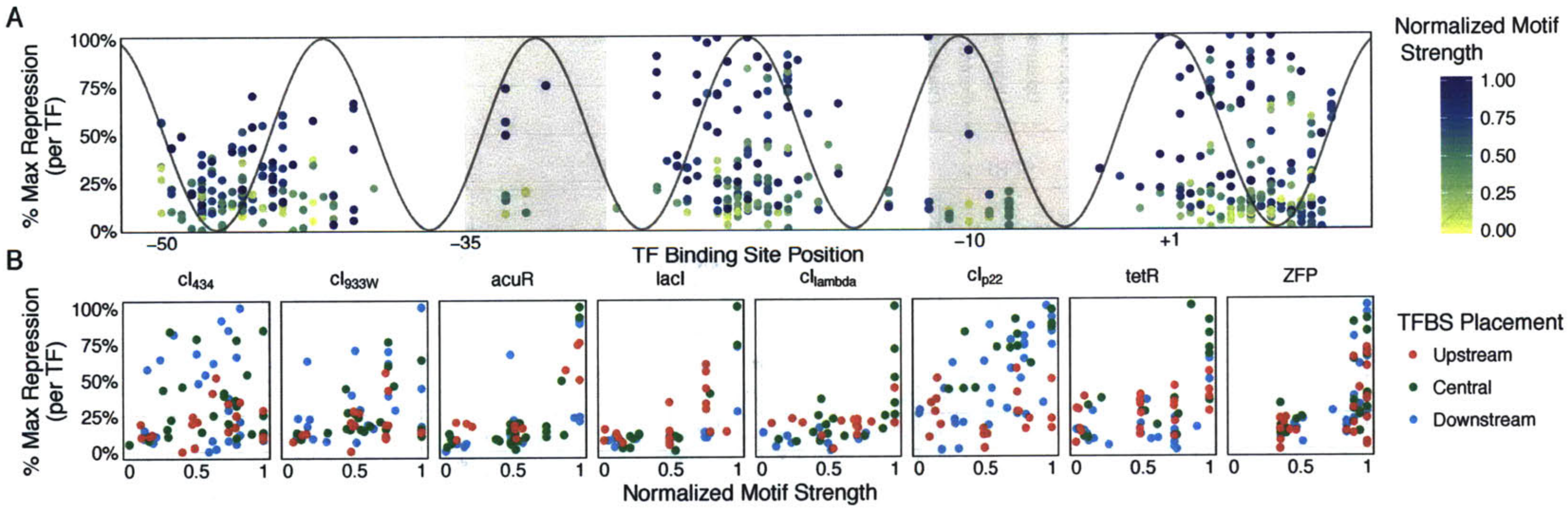


Figure 4.3: *Interaction between TF Binding Site Positioning, Motif Strength, and Repression.* (A) Center of TF binding site versus normalized repression, as a percentage of the maximal fold-repression observed for that TF. Only promoters that have the potential to be repressed by at least 10-fold through our dynamic range are plotted. Generally downstream and central TF placements repress more strongly than upstream. (B) TF binding site motif strength plotted versus normalized repression separately for each transcription factor, and colored by TF region.

4.3.3 *Transcriptional Activation from cI_{434} and a ZFP-ω fusion*

Lambdoid phage cI repressors have the capacity to activate transcription when their binding site is properly placed with respect to the -35 motif, [16] and we generated several promoters using the precise binding locations specified in the literature. For the cI_{434} transcription factor we identified several promoters that can be activated in this manner, but only when placed upon the weakest right promoter, R_0. When combined with additional binding sites at different locations within the promoters, these activating sites generally ceased to function (Figure C.8). We did not identify TF binding site placements that worked as transcriptional activators for the other three lambdoid cI repressors, suggesting that factors beyond the strength and location of the binding motif influence their ability to activate transcription.

In addition to activation from cI_{434}, we also tested a synthetic transcriptional activator built by fusing a synthetic zinc finger array to the ω subunit of RNA polymerase[24]. This zinc finger was also able to recruit the RNA polymerase when its binding site was placed far upstream of the -35, but acted as a modest repressor (maximum 10-fold) when placed in locations closer to the core promoter.

4.3.4 *Evolvability of Transcriptional Regulation*

Promuter uses the motif constraints to build a mutation landscape and conservatively mutates the 'base' P_R promoters to generate transcription factor binding sites with as few mutations as possible. As we intelligently sample the local sequence space around each promoter for predicted regulatory elements, we build a map of the functional landscape of each promoter in a manner analogous to an evolutionary search, posing questions of evolvability and epistasis in cis-regulation. In all three promoters and across all 8 transcription factor motifs, we find that it is possible to generate TF binding sites which repress transcription at least 5-fold with as few as 5 single base pair changes. We also see that the maximum level of repression for each factor is achievable with approximately 10 nucleotide mutations (Figure 4.4). It has been shown through transcriptional profiling that experimentally evolved strains of *E. col*i are capable of 'rewiring' regulatory connections in a widespread manner [21]. However, it was not clear how this rewiring occurred, or if it was possible to generate new cis-regulation across the genome on such a relatively short time scale. This finding highlights that functional regulatory elements are shallow in sequence space and accessible to evolution. Further, it allows us to pinpoint regions of non-coding sequence which are most capable of being converted into new regulatory elements.

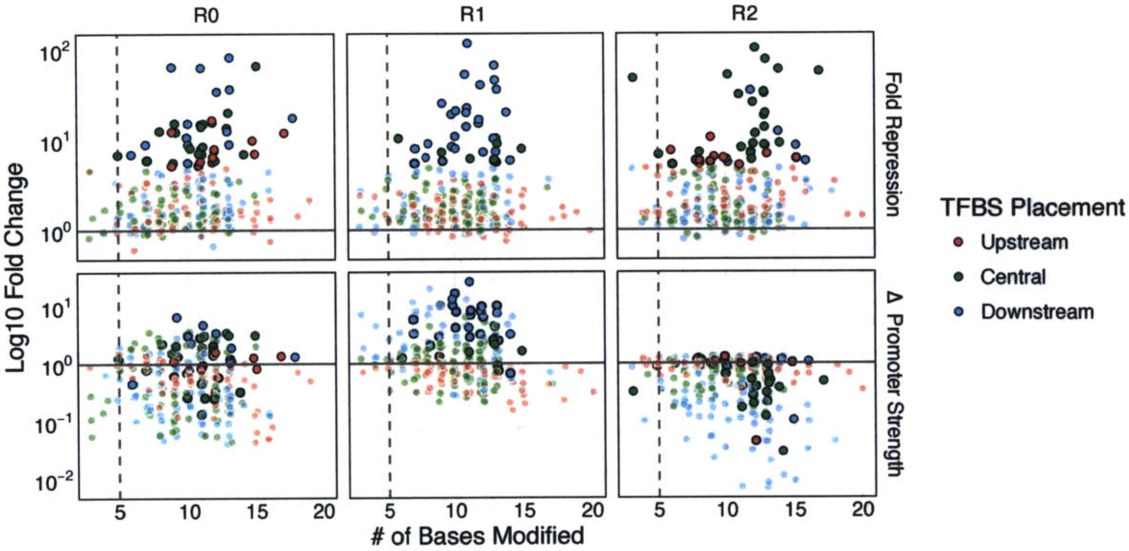


Figure 4.4: *Mutations required to create new transcriptional regulation.* Effect of TFBS placement on promoter strength and repression is compared to the number of single base changes required to place the site. 5-fold or greater repression (dots outlined in black) can be achieved with only 5 mutations, and maximal repression is achieved in 10 mutations.

4.3.5 *Changes in Basal Promoter Strength Accounts for the Majority of Repression Differences*

TF binding site placement changed basal expression in a promoter-specific manner. The -35, -10 and UP elements were maintained by *Promuter* when adding TF binding sites, which shows that despite a lack of sequence conservation in these regions, even small modifications can strongly effect expression. Promoter R_0 is sensitive to mutations across the entire cis-regulatory sequence, and based on single-base-pair scanning mutagenesis, strengthening the -35 and -10 σ_{70} binding motifs strongly increase expression, suggesting that RNA polymerase recruitment is the major limit to expression level in R_0 . However, promoter R_1 is relatively insensitive to single base pair mutations, especially mutations expected to affect the -35 and -10 regions. It also shows a marked increase in strength when the region downstream of its -10 is modified (Figure 4.5). Taken together these observations suggest that expression from $R_1$is limited not by recruitment of the RNA polymerase but instead by formation of the open complex and selection of the TSS, which is highly dependent upon this region[97]. Based on average per-base effects from both the TFBS placements and single base mutations, removing a trinucleotide T repeat upstream of the TSS and reducing high GC content downstream of the TSS both seem to strongly increase basal transcription.

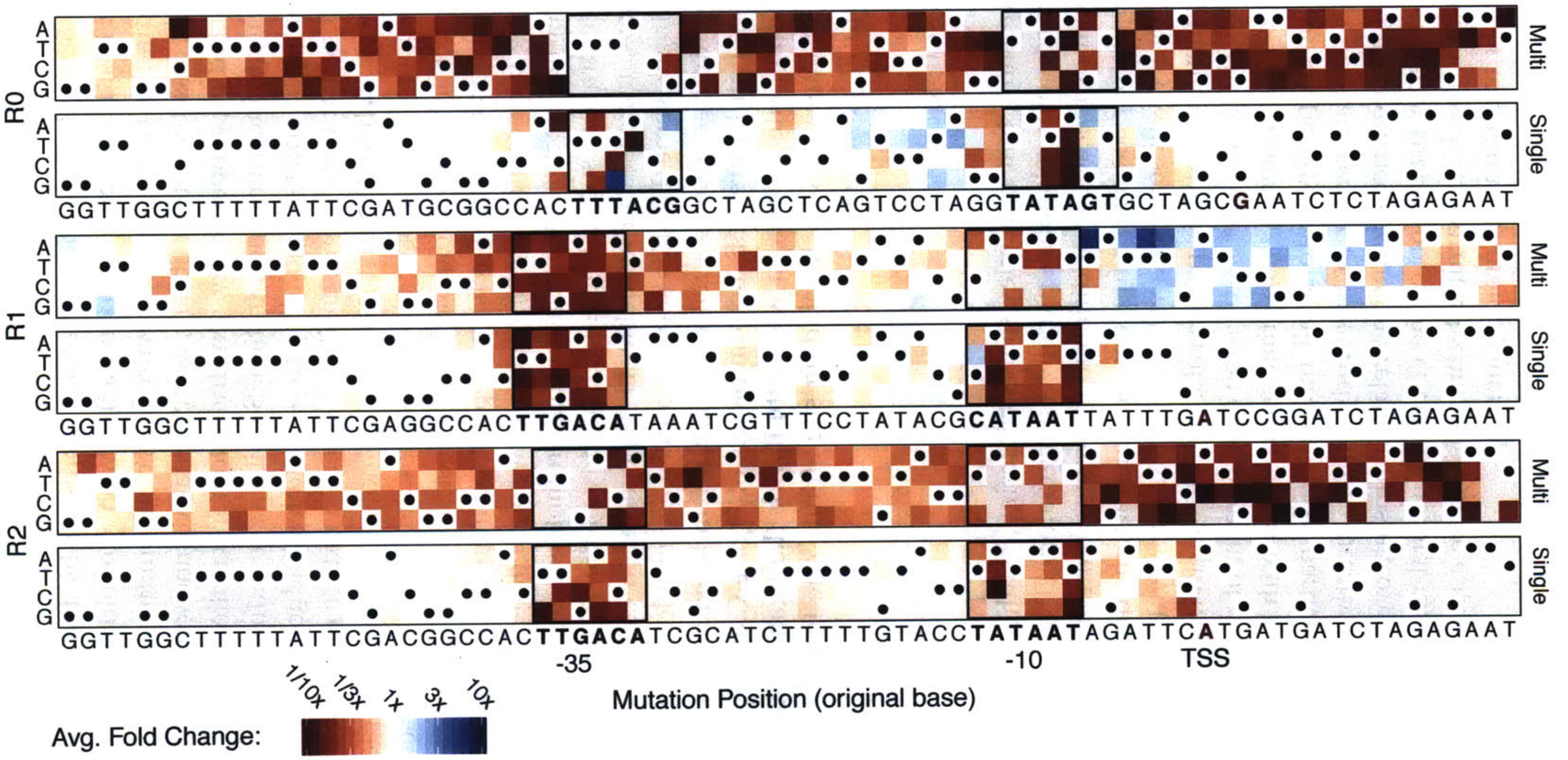


Figure 4.5: *Per-base effects of mutation on basal transcription strength vary between promoters.* Single base pair changes and their fold-change effects on transcription are plotted for each P_R promoter in the 'Single' rows. 'Multi' rows correspond to the average effect of changing a base when all TF placements that modify that base are considered. The predicted TSS base is shown in red and the -35 and -10 regions are boxed.

4.3.6 *Induction Strength, Sensitivity, and Cooperativity*

Of the 6,247 dual-promoter circuits repressed by the three small molecule inducible TFs, 130 responded to induction with >10-fold increase in expression, corresponding to 34 P_R sequences with distinct TF binding site configurations (Figure 4.6A). Across the entire library, we calculated functional properties of these circuits including cooperativity, sensitivity to transcription factor concentration, and sensitivity to inducer concentration. Sensitivity to inducer molecule varied widely, with some promoters responding to 100-fold less inducer than others (Figure 4.6B). Promoters that were more sensitive tended to induce expression linearly across a wide range, while the least sensitive promoters responded rapidly, showing that more cooperative repressor binding makes the promoter less sensitive to inducer (Figure 4.6C). Maximal fold induction closely matched fold-repression, suggesting that induction completely relieved repression in most cases. Strikingly, the effect of increasing transcription factor concentration varied depending on TF identity and promoter architecture. Many inducible promoters were insensitive to increases in TF concentration, suggesting a high affinity of the TF for the binding site, while for others, increases in available TF concentration caused increased repression. Surprisingly, some TF sites showed a decrease in fold-repression with increased TF concentration, due to a reduction in fully induced expression (increasing [TF] decreases b in Figure 4.6B, bottom panels of Figure C.9). This could be the result of a binding site that has a weak affinity for the ligand-bound conformation of the TF or alterations in the binding site affinity due to DNA supercoiling at the upstream P_L promoter.

4.4 DISCUSSION

Here we show that mutliplex sequence design, synthesis, and measurement can uncover complex relationships between regulatory DNA architecture and detailed functional properties. In addition to improving our mechanistic understanding of how regulatory sequences function and interact, the diverse regulatory behaviors in this library will serve as a valuable resource for synthetic biology, including the development of synthetic prokaryotic circuits for environmental sensing and actuation[44]. This work also enables the development of more quantitative metabolic sensor-selector systems[73, 91] that can respond with increased or decreased sensitivity or cooperativity to small molecules. In particular, we chose to examine the *acuR* acrylate sensor in this study as it is directly applicable to biomaterials production. Sequence-to-function relationships uncovered by this library also will allow for better prediction of the effects of natural mutation on gene expression, especially if machine learning techniques can be applied to these data and compared back to natural genomic sequence.

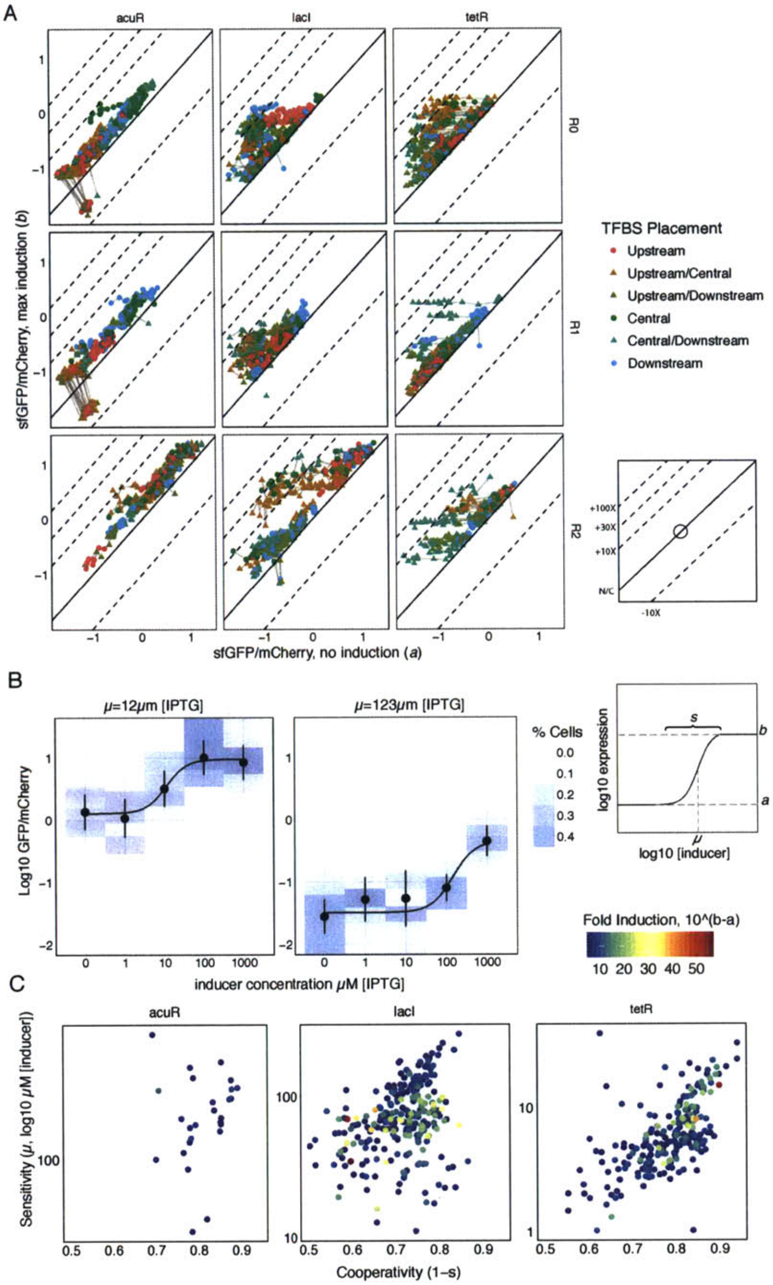


Figure 4.6: *Induction Strength, Sensitivity, and Cooperativity.* (A) Dynamic Ranges of Induction for 6,247 dual-promoter constructs. Each point represents one dual promoter construct combining both a P_L and P_R promoter set, generated using *Promuter*. The shape of each point corresponds to a single or double motif, and the color corresponds to the location(s) of the motif(s). The X axis measures sfGFP in the absence of TF expression, and the Y axis measures sfGFP under maximal TF expression. Points along the diagonal are unaffected by TF expression, while points below are repressed and points above are activated. An open circle corresponds to the unmodified promoter. Dotted diagonal lines correspond to 10x activation, and 10, 30, and 100x repression.

Going forward, we can use the knowledge gained from these measurements to improve sequence-to-function design and and prediction software like *Promuter*. By performing multiple rounds of this design, build, and test process, we can iteratively learn new rules and improve the predictive power and design accuracy of these algorithms. *Promuter* could be expanded into a general tool to make conservative mutations to natural sequences that have desired effects on regulatory function in the context of larger genomic systems or whole re-engineered genomes. By performing an informed and targeted search of sequence space instead of an exhaustive one, we avoid the curse of dimensionality inherent in the exploration of DNA sequence space. This allows us to achieve a broader understanding of regulatory element evolution, function, and design than would possible by an exhaustive sequence screen. Targetted design and synthesis of sequence libraries can be applied to a diverse set of biological problems beyond regulatory elements, including protein structure, exon splicing, transcriptional network perturbation, and genome engineering.

Part II

HIGH-THROUGHPUT MODIFICATION AND MEASUREMENT OF BACTERIAL GENOMES

5

MILLSTONE: SOFTWARE FOR MULTIPLEX MICROBIAL GENOME ANALYSIS AND ENGINEERING

This chapter contains in progress and unpublished work.

Daniel B. Goodman[1], Gleb Kuznetsov[1], Marc Lajoie, Brian W. Ahern, Michael Napolitano, Kevin Chen, Changping Chen, George Church. *Millstone: Software for Multiplex Microbial Genome Analysis and Engineering.* In preparation.

Acknowledgements:

Millstone was designed by D.B.G., G.K., and M.J.L. The code was written primarily by G.K., D.B.G., B.W.A., K.C., and C.C. Documentation, troubleshooting, and testing was performed by M.N., D.B.G., G.K., B.W.A., K.C., and C.C with assistance from other authors. D.B.G., G.K., and M.J.L. wrote the manscript with input from all authors. G.M.C. supervised the research.

We would like to thank Amazon Web Services for providing us with an AWS in Education Research Grant and the MIT Undergraduate Research Opportunities Program for providing funding for B.W.A., K.C., and C.C.

[1] (co-first-author)

ABSTRACT

Inexpensive DNA sequencing and advances in genome editing have made computational analysis a rate-limiting step in microbial genome engineering. We describe Millstone, a web-based platform for multiplex mutation analysis and iterative genome engineering. Millstone integrates alignment, variant-calling, genotype comparison, and visualization for hundreds of microbial genomic samples. To facilitate iterative genome editing, Millstone can design targeted mutations and reversions and generate and track new reference genomes. Millstone is open source and is easily deployable on a desktop, a cluster, or an Amazon Machine Image (AMI), making it a scalable solution for any lab.

5.1 INTRODUCTION

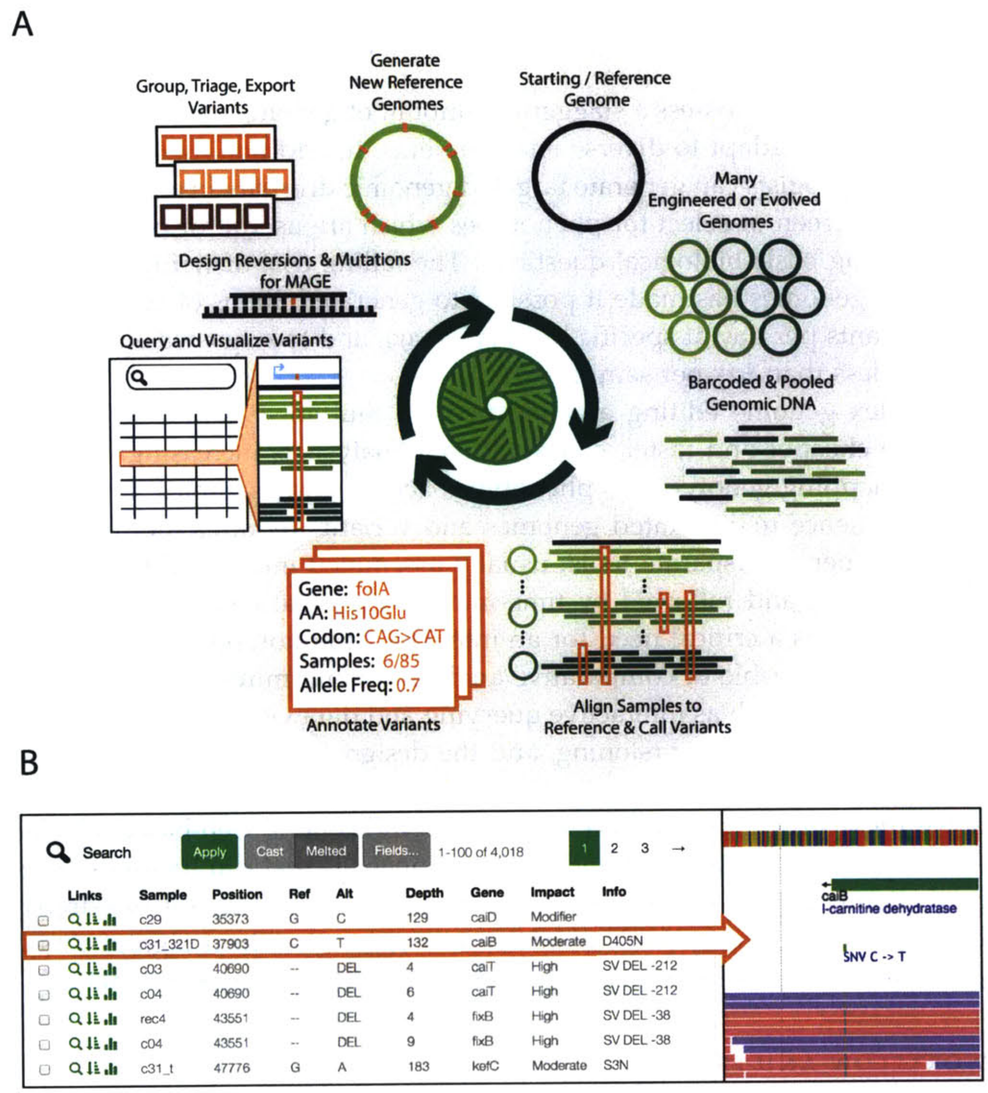
Microbial populations possess a staggering amount of genomic diversity, enabling them to evolve and adapt to diverse environments. In addition to studying natural evolution, biologists can generate targeted genomic diversity in a population of cells and then screen or select for phenotypes which are useful for biotechnology or for answering basic biological questions. The falling cost of writing and reading microbial genomes has made it possible to generate billions of combinatorial genomic variants per day at specific loci [33, 37, 99] and to sequence entire E. coli genomes for less than $25 per sample [9, 80].

As multiplex genome editing and inexpensive multiplex sample preparation have become cheaper and faster, computational analysis is increasingly a bottleneck when mapping genotypes to phenotypes across many samples. Going from raw DNA sequence to annotated genomes and variants requires the integration of a large number of disparate tools, usually assembled into an ad-hoc pipeline by individual labs and followed by time-intensive manual confirmation of variants. There remains a critical need for an integrated solution optimized for large amounts of data, capable of comparative analysis among multiple genomes, and supporting features such as interactive querying and data visualization, collaboration, iteration and genome versioning, and the design of additional mutations or reversionsD.1.

To solve these problems, we have developed Millstone, a web-based platform for iterative genome engineering and mutation analysis that can handle the complexity of alignment, variant-calling, comparison, and versioning for hundreds of evolved or rationally modified microbial genomes, as well as the design of oligonucleotide libraries for targeting mutations in the next round of experiments.

5.2 WORKFLOW AND FEATURES

Millstone was designed to eliminate the complicated infrastructure, installation, and configuration requirements for multiplex whole-genome sequencing analyses, minimizing the time between sequence upload and delivery of searchable, annotated, and visualizable results. After a researcher uploads .fastq reads derived from evolved or engineered genome sequences, Millstone automates read alignment and variant calling, automatically processing hundreds of microbial genomes. One hundred genomes can be aligned and analyzed in 2 hours at the cost of $5 in compute resources (D.6.2.2). The researcher can then explore and compare variants across samples. A rich user interface and query language facilitate drilling down into evidence for individual variants and visualizing raw read alignments. Variants can be grouped into semantically-related variant sets. Finally, a variant set can be used to generate oligonucleotides for follow-up experiments and to create new versions of reference genomes. (Fig. 5.1).


A
Group, Triage, Export Variants
Generate New Reference Genomes
Starting / Reference Genome
Many Engineered or Evolved Genomes
Design Reversions & Mutations for MAGE
Query and Visualize Variants
Barcoded & Pooled Genomic DNA
Gene: folA
AA: His10Glu
Codon: CAG>CAT
Samples: 6/85
Allele Freq: 0.7
Annotate Variants
Align Samples to Reference & Call Variants
B
Search
Apply
Cast
Melted
Fields...
1-100 of 4,018
1
2
3
→
calB
l-carnitine dehydratase
SNV C -> T

	Links	Sample	Position	Ref	Alt	Depth	Gene	Impact	Info
□		c29	35373	G	C	129	caiD	Modifier	
□		c31_321D	37903	C	T	132	caiB	Moderate	D405N
□		c03	40690	--	DEL	4	caiT	High	SV DEL -212
□		c04	40690	--	DEL	6	caiT	High	SV DEL -212
□		rec4	43551	--	DEL	4	fixB	High	SV DEL -38
□		c04	43551	--	DEL	9	fixB	High	SV DEL -38
□		c31_t	47776	G	A	183	kefC	Moderate	S3N

Figure 5.1: *Millstone enables rapid iterative genome analysis and engineering.* (A). To use Millstone, a researcher provides a reference genome and multiplexed next-generation sequencing data for many individual genomic clones, arrived at either via long-term evolution or targeted genome engineering. Millstone performs alignment and variant calling for both short polymorphisms and large structural variants and then assigns predicted effets based on provided genome annotations. A unified data model relates sample genotype, phenotype, and variant annotation data. Variants can then be queried, filtered, and grouped into sets for export, triage, and analysis. These variant sets can be used to create additional MAGE oligos to recreate or revert mutations or be used to generate new reference genomes for further rounds of the cycle. (B). A combined screenshot of the Millstone analysis and visualize views (condensed and cropped for clarity). A custom query language allows searching and filtering over the data. As with all genomics pipelines, variant calls sometimes require visual inspection and comparison. Millstone's variant analysis view provides programmatically-generated links to visualizations of the relevant read alignments in Jbrowse.

5.2.1 *Deployment*

Researchers can provision a fully-configured private instance of Millstone running on Amazon Web Services (AWS) in minutes and can specify compute, memory, and disk requirements to match project needs. The software can also be deployed on a laptop or an in-house cluster. We recommend AWS for most users and maintain a public release of an Amazon Machine Image (AMI) preconfigured with the latest stable version of Millstone which allows a lab to provision a private instance of Millstone in minutes D.6.2.2.

5.3 APPLICATIONS

5.3.1 *Genome engineering to reassign the genetic code*

Millstone provides end-to-end support for the iterative process of genome-scale engineering, from confirming designed mutations to debugging fitness defects. In Lajoie et al.[49], we engineered a strain of E. coli for which all 321 UAG stop codons in the genome were replaced with a synonymous UAA stop codon. Initially, custom scripts were used to analyze the next-generation sequencing data from 76 genomes produced along the UAG replacement lineage. These analyses were slow and error-prone, and it became challenging to quickly visualize and compare evidence for mutational events. This crude pipeline provided the impetus for developing Millstone. Here, using the latest version of Millstone, we re-analyzed the initial, intermediate and final genomes in a single day (Fig. 5.2A). Further, Millstone allowed us to annotate and rank the 355 off-target mutations in the final strain according to predicted effect. In Kuznetsov et al. (in preparation) we describe how we tested combinatorial reversions of the highest ranked targets and iteratively used Millstone to improve the strain fitness. Millstone allowed us to rapidly compare genomes constructed in each MAGE experiment to each other and to previous generations, enabling the rapid iteration of this rational strain improvement process (Fig. 5.2A).

5.3.2 *Rapid analysis of long-term evolution experimental data*

Millstone can also be used to analyze genomic variation arising from samples undergoing directed laboratory evolution. In Tenaillon et al.[93], 115 strains of E. coli were grown at high temperature for over 2,000 generations in an attempt to identify convergent evolutionary responses to this environmental challenge. This impressive effort required their team to develop a custom sequencing analysis pipeline involving over half a dozen tools combined with custom software, followed by painstaking manual validation and visual confirmation of all 1331 variants. We reanalyzed the raw data from this project in Millstone and identified

99.7% of SNVs and 83% of structural variants identified in the original study, as well as 2 additional mutations that were unidentified in the original work (Fig. 5.2B). On an Amazon AWS instance, the entire process from sample upload to the triaging of variants across all strains took a single day.

5.4 DISCUSSION

New technologies for constructing, screening, and selecting genomes now allow for increasingly complex functional genomics studies and bioengineering endeavors. As the sequence constraints of the genome come into focus, the promise of designing new organisms that can address humanity's medical and material needs [30, 49] is becoming a reality. The path forward requires rapid construction and characterization of successive versions of genomes. Millstone's analysis and exploration features are complemented by features for refining reference assemblies to accurately represent all SNPs and structural events in lab-generated strains and allow maintaining a version history of these refined genomes.

Whether analzying rational designs or evolved strains, researchers can use all or a subset of Millstone's features. For example, researchers who already have raw sequencing data from as many as hundreds of genomes can use Millstone to identify and explore mutations. We have reduced the barrier for other labs to get started with Millstone by making the software deployable on Amazon Web Services (AWS). Instructions and an online demo are available at `http://churchlab.github.io/millstone`.

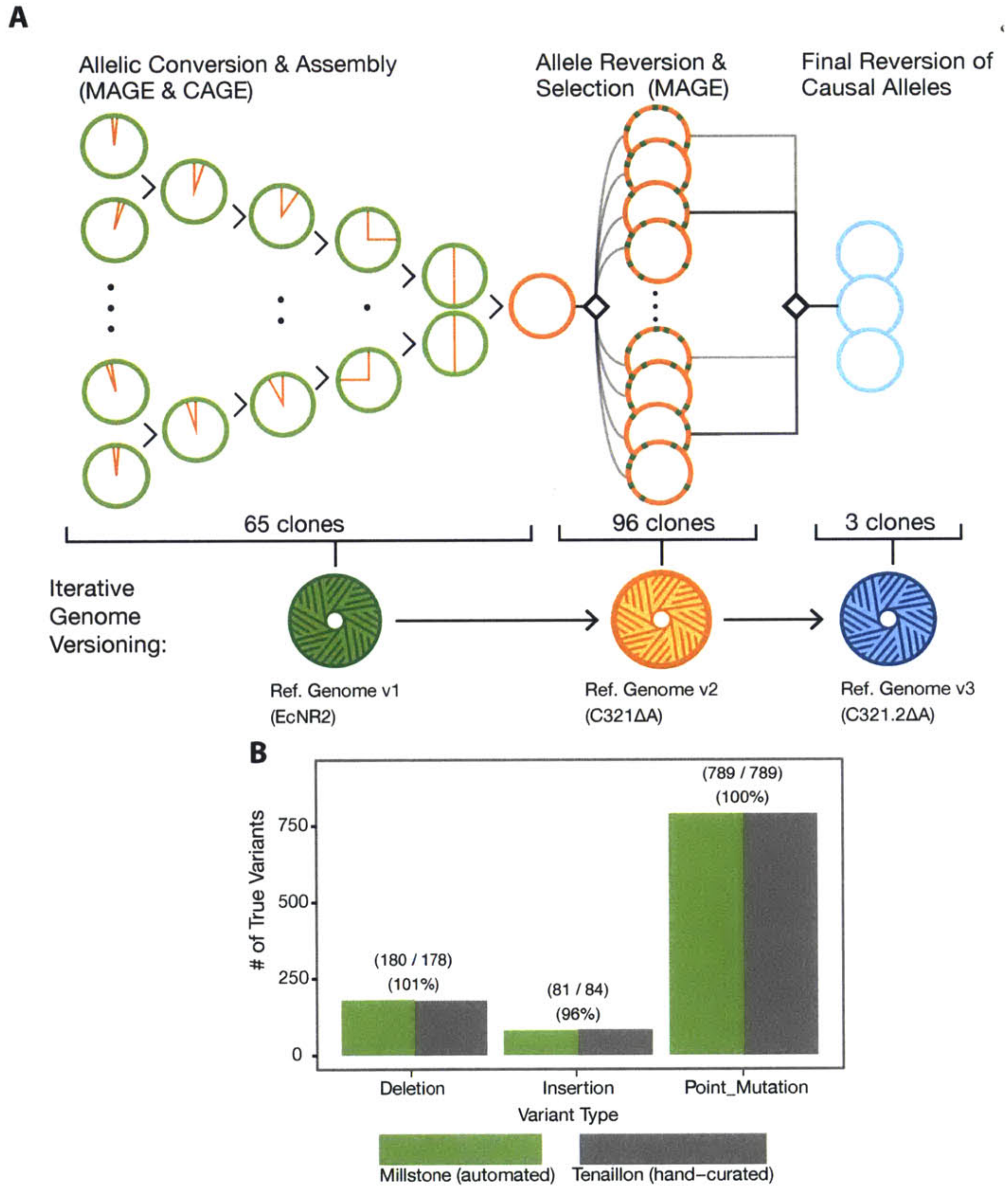


Figure 5.2: *Millstone accurately detects genomic variants and can iteratively version genomes.* (A). Millstone was used to analyze genomic clones involved in generating a genomically recoded organism. MAGE and CAGE were used to generate the C321ΔA strain of *E. coli*[49]. With sequencing data from these strains, Millstone confirmed designed mutations, identified and annotated unintended ones, and generated a new reference genome. Further reversion of annotated variants was performed with MAGE to improve the strain's fitness, and a final reference genome was generated. (B). Millstone automates finding variants across hundreds of strains. Applied to the Tenaillon et al. dataset[93], Millstone reports all SNVs found by the Tenaillon pipeline, including some not identified in the original study.

Part III

APPENDIX

A SUPPLEMENTAL INFORMATION FOR CHAPTER 2

This appendix contains the extended experimental methods, extended data analytical methods, and additional figures for Chapter 2. It is adapted from the supplemental information included with Kosuri et al.

A.1 EXTENDED EXPERIMENTAL METHODS

A.1.1 *Reporter Construction*

The gene expression reporter construct (pGERC) used in all experiments follows the design of the pZS2-123 plasmid that drives independent expression of three fluorescent proteins from Cox et al[22]. Briefly, we began with the divergent promoter portion of pZS2-123, which has insulated sequences to express CFP with PLtetO-1 and YFP with PLlacO-1. We replaced the CFP, with a codon-optimized version of mCherry[84], and replaced the YFP with a codon-optimized version of superfolder GFP (sfGFP)[66]. We replaced PLlacO-1 with the EM7 promoter to avoid issues of endogenous regulation by the Lac repressor in MG1655. We also removed an AscI recognition site in the intergenic space, and placed an AscI recognition site directly upstream of the EM7 promoter and an NdeI recognition site at the start of the sfGFP sequence. These sites are used for cloning library components upstream of the sfGFP sequence. The whole construct is flanked by XhoI and NotI on the left, and PacI and XbaI on the right, and was constructed by DNA 2.0 (USA) in their pJ251 backbone that has a low-copy number p15A origin of replication and a kanamycin resistance marker.

A.1.2 *Library Design, Construction, and Cloning*

The library was constructed by combining 114 promoter sequences with 111 RBS sequences. Promoter sequences were chosen from existing libraries such as the BIOFAB [25], a few control promoters (including an inactive spacer), and a set of promoters from Chris Anderson's promoter library from the BioBricks registry (53). We added a five base barcode and then checked for restriction site compatibility (AscI and NdeI) to generate the final promoter library. The ribosome binding site library contains RBSs from BioFAB [4], control RBSs, Chris Anderson's RBS library from the BioBricks registry [3], and sequences generated by Howard Salis's RBS Library calculator [76]. The promoters and RBSs were filtered for restriction sites and to ensure that all pairwise Levenshtein distances are greater than 1. In addition, all RBSs have bases 'CAT' replacing the terminal 3 bases prior to the coding sequence to allow for cloning using the NdeI site for a total of 111 RBSs. Finally, each promoter is crossed by all RBSs to form a final library of 12,653 promoter + RBS combinations. One combination was removed because the junction resulted in a disallowed restriction site. All constructs were flanked by restriction enzyme sites (AscI and NdeI) and the following PCR primer binding sites:

```
skpp-202-F AATCCTTGCGTCAATGGTTC
skpp-202-R GGGTTCTCGGATTTTACACG
```

The oligo library was constructed by Agilent Technologies (USA) using their Oligo Library Synthesis (OLS) process [51], and was delivered as a ~1 picomole lyophilized oligo pool. The library was amplified from the oligo pool using biotinylated primers, digested with AscI and NdeI (New England Biolabs, USA), and the resulting ends were removed by Invitrogen (USA) M-270 streptavidin beads. The plasmid backbone was also amplified by PCR using biotinylated primers, digested with the same restriction enzymes, and cleaned again by streptavidin beads. We then ligated the library and plasmid backbone using T4 DNA Ligase (NEB) and cloned into 5-alpha electrocompetent cells (NEB) resulting in ~600,000 clones. The library was grown under kanamycin selection, and plasmids were isolated using Qiagen Miniprep kit. The plasmid library was re-transformed into E. coli MG1655 (Yale CGSC No. 6300) (>3 million clones). We froze several aliquots of this library and used these aliquots for all subsequent experiments.

A.1.3 *Control colonies and flow cytometry*

We plated the transformed MG1655 library and Sanger sequenced 282 clones. 154 of 282 (55%) of these clones matched the designed sequences exactly. 144 sequence-perfect clones (2 clones were duplicates) were inoculated from glycerol stocks into 200 µL of LB with kanamycin and grown overnight at 30 °C with shaking in 96-well culture plates. The cells were then backdiluted 1:1000 into 200 µL LB with kanamycin and grown for 3.5 hours, until the cells reached an OD600 of ~0.15-0.25. The cells were then immediately put on ice, pelleted by centrifugation, and diluted 1000-fold in ice-cold PBS. We measured RFP and GFP fluorescence levels using a BD FACS LSRFortessa flow cytometer with a high throughput sampling attachment (30,000 events per observation). Events were gated on forward and side scatter to exclude debris, dead cells, and doublets. The overnight growth, back-dilution, and flow cytometry procedure was performed four times from different back-dilutions on two separate days.

A.1.4 *Library growth and FlowSeq*

A 300 mL culture was inoculated with 1 mL of overnight library culture grown overnight at 30°C from a frozen aliquot. The culture was grown 3.5 hours to an

OD600 of 0.2 at 30°C and shaking at 250 RPM. The culture was quickly brought to 4°C in an ice-slurry. Five 50 mL aliquots were pelleted. Four were snap-frozen in liquid nitrogen, while one was resuspended in 50mL ice-cold PBS. The library in PBS was directly subjected to FlowSeq. We conducted three consecutive flow sorts on a BD FACSAria IIu while keeping cells at 4°C. Each run sorted four non-adjacent log-spaced bins based on GFP:RFP Ratio. We sorted one million cells for the first bin (lowest ratio) because it had the most cells in it. For all other bins, we sorted 250,000 cells, except for the last two bins where we sorted 100,000 cells each. Cells were grown overnight with shaking at room temperature to minimize growth rate differences, and plasmids were isolated using a Qiagen miniprep kit. Each bin was separately amplified for 5 cycles by RT-PCR to prevent over-amplification using Kapa SybrFast RT-PCR master mix. The reverse primer was an equimolar mixture of five separate sequences to allow frame-shifting to give better sequence distributions during read 2 of sequencing:

```
FlowSeq-F: AATGATACGGCGACCACCGAGATCTACACTGAAGCACAGCAGCTCTTCGCCTTTACGCATATG
FlowSeq-R0: GTGACTGGAGTTCAGACGTGTGCTCTTCCGATCTGACAATGAAAAGCTTAGTCATGGCG
FlowSeq-R1: GTGACTGGAGTTCAGACGTGTGCTCTTCCGATCTTGACAATGAAAAGCTTAGTCATGGCG
FlowSeq-R2: GTGACTGGAGTTCAGACGTGTGCTCTTCCGATCTATGACAATGAAAAGCTTAGTCATGGCG
FlowSeq-R3: GTGACTGGAGTTCAGACGTGTGCTCTTCCGATCTCATGACAATGAAAAGCTTAGTCATGGCG
FlowSeq-R4: GTGACTGGAGTTCAGACGTGTGCTCTTCCGATCTGCATGACAATGAAAAGCTTAGTCATGGCG
FlowSeq-R5: GTGACTGGAGTTCAGACGTGTGCTCTTCCGATCTGCATTGACAATGAAAAGCTTAGTCATGGCG
```

A final RT-PCR step added barcodes to each binned construct using the following primers:

```
FlowSeq-F: AATGATACGGCGACCACCGAGATCTACACTGAAGCACAGCAGCTCTTCGCCTTTACGCATATG
Bin 1 FlowSeq-R-index_6nt_1
    CAAGCAGAAGACGGCATACGAGATtcaggtGTGACTGGAGTTCAGACGTGT
Bin 2 FlowSeq-R-index_6nt_2
    CAAGCAGAAGACGGCATACGAGATaagcgtGTGACTGGAGTTCAGACGTGT
Bin 3 FlowSeq-R-index_6nt_3
    CAAGCAGAAGACGGCATACGAGATgtcgatGTGACTGGAGTTCAGACGTGT
Bin 4 FlowSeq-R-index_6nt_4
    CAAGCAGAAGACGGCATACGAGATgccttgGTGACTGGAGTTCAGACGTGT
Bin 5 FlowSeq-R-index_6nt_7
    CAAGCAGAAGACGGCATACGAGATggtaagGTGACTGGAGTTCAGACGTGT
Bin 6 FlowSeq-R-index_6nt_9
    CAAGCAGAAGACGGCATACGAGATgattgcGTGACTGGAGTTCAGACGTGT
Bin 7 FlowSeq-R-index_6nt_11
    CAAGCAGAAGACGGCATACGAGATcggtccGTGACTGGAGTTCAGACGTGT
Bin 8 FlowSeq-R-index_6nt_13
    CAAGCAGAAGACGGCATACGAGATgcaaccGTGACTGGAGTTCAGACGTGT
Bin 9 FlowSeq-R-index_6nt_15
    CAAGCAGAAGACGGCATACGAGATatgaacGTGACTGGAGTTCAGACGTGT
Bin 10 FlowSeq-R-index_6nt_16
    CAAGCAGAAGACGGCATACGAGATcttataGTGACTGGAGTTCAGACGTGT
Bin 11 FlowSeq-R-index_6nt_17
    CAAGCAGAAGACGGCATACGAGATagcagaGTGACTGGAGTTCAGACGTGT
Bin 12 FlowSeq-R-index_6nt_20
    CAAGCAGAAGACGGCATACGAGATcaataaGTGACTGGAGTTCAGACGTGT
```

The amplified bins were quantitated using the Kapa Library Quantification Kit, and mixed in equimolar ratios before sequencing all twelve on a single HiSeq 2000 paired-end 100 bp lane with the following sequencing primers:

```
Custom Read 1: 5' GAAGCACAGCAGCTCTTCGCCTTTACGCATATG
Illumina Multiplexing Read 2: GTGACTGGAGTTCAGACGTGTGCTCTTCCGATCT
Illumina Multiplexing Index Read: GATCGGAAGAGCACACGTCTGAACTCCAGTCAC
```

A.1.5 *Spike-in controls*

A separate library underwent the same procedure. Prior to back-dilution, we spiked in a subset of 42 of the perfect sequences and performed all procedures including DNASeq, RNASeq, and FlowSeq identically.

A.1.6 *DNASeq & RNASeq*

For DNASeq, we isolated plasmids using a Qiagen midiprep kit from two frozen cell pellets from the 50 mL library growth culture. We amplified the library as we did in the FlowSeq experiment, using only primers FlowSeq-R-index_6nt_1 and FlowSeq-R-index_6nt_4. We also processed the spike-in libraries similarly, but using FlowSeq-R-index_6nt_15 and FlowSeq-R-index_6nt_16. All four DNASeq libraries were run on a single lane in the same HiSeq run as the FlowSeq data.

For RNASeq, we used the remaining two cell pellets and first isolated total RNA using a Qiagen RNEasy Midi Kit, and removed ribosomal RNA using Epicentre's Ribo-Zero rRNA Magnetic Removal Kit for Meta-Bacteria as per manufacturer's instructions. Then we used 250 ng of mRNA and removed the 5′ triphosphate group with RNA 5′ Polyphosphatase (Epicentre) as follows:

- 50µL RNA (250ng)
- 6µL RNA Polyphosphatase
- 10x Reaction Buffer
- 1.5µL RiboGuard RNase Inhibitor (Epicentre)
- 3µL RNA 5′ Polyphosphatase (60 Units)
- 37°C for 30 minutes

The resulting reaction was cleaned up using a Qiagen RNAEasy MinElute Kit. Then we ligated the following RNA adaptor to the processed mRNA:

```
RNA ligation primer GACAAUGAAAAGCUUAGUCAUGGCGNN
```

The two trailing Ns indicate degenerate bases that are used to greatly reduce biases in found in RNA ligation efficiency across different templates [36]. We used the following procedure for ligation using T4 RNA Ligase (Epicentre):

- 10µL RNA from previous step
- 2µL 250µM RNA oligo
- 2µL 10X Ligase buffer
- 2µL 10U T4 RNA Ligase (Epicenter)
- 2µL 10mM ATP
- 1µL RiboGuard RNase Inhibitor (Epicentre)
- 1µL DMSO
- 25°C for 3 hours

The resulting reaction was cleaned up again using Qiagen RNAEasy Minelute Kit. To make cDNA, we used Invitrogen's SuperScript III with the following procedure:

1. We added the following components to a nuclease-free microcentrifuge tube:
   0.2µL of 10µM RT-Primer
   2 pmol of RT primer: ACCGTTGACATCACCATCCAGTTCC
   12µL RNA from RNA ligation reaction
   1 µl 10 mM dNTP Mix
2. We heated mixture to 65°C for 5 minutes and incubated on ice for 1 minute.
3. We collected the contents of the tube by brief centrifugation and added:
   4 µl 5X First-Strand Buffer
   1 µl 0.1 M DTT
   1 µl RNaseOUT Recombinant RNase Inhibitor (Invitrogen 40 units/µl).
   1 µl of SuperScript^{TM} III RT (200 units/µl)
4. We mixed by pipetting gently up and down.
5. We incubated at 55°C for 60 min.
6. We inactivated the reaction by heating at 70°C for 15 min
7. We added 1 µl (2 units) of E. coli RNase H and incubated at 37°C for 20 min.

The resulting cDNA was amplified using the same procedures as DNASeq and FlowSeq, and using the same barcodes for technical and spike-in replicates on a separate lane of the same HiSeq 2000 run.

A.2 DATA ANALYSIS

A.2.1 *Contig formation and trimming*

We used a modified version of SeqPrep [87] and custom Python scripts to pair and trim reads into contigs with increased sequencing fidelity for regions of paired-end coverage. Each set of two paired-end 100 bp reads were aligned and merged into a contig based on their overlapping sequence. The adapter and constant primer sequences were trimmed from both ends of the contig. If only a portion of the adapter sequence was identifiable then the cloning restriction sites were used to identify the region for trimming. Reads that did not pair were discarded, as all sequences are under 200 bp and thus contigs should be created where the two paired reads overlap. Additionally, the first two bases of RNA contigs were trimmed, corresponding to the two degenerate ligated bases used in the experimental protocols.

Deduplication and sorting of unique contigs to library After trimming, occurrences of each unique contig were counted per bin and merged to generate a vector of 12 numbers corresponding to the occurrences per bin per contig. These unique contigs were then aligned to the Promoter + RBS sequence library. In the case of the protein data, grep and USEARCH 5.2.32 were used. We aligned all unique contigs, but used the intersection of three criteria to filter for downstream analysis. Contigs were required to: (1) be perfect end-to-end matches to the library, (2) consist of at least 100 occurrences, and (3) occur in multiple bins, excepting the final bin. In the case of DNASeq and RNASeq data, Bowtie[50] was used. We filtered matching contigs on three criteria: contigs (1) were allowed no more than 3 mismatches, (2) were required to match best to only one library combination, and (3) to remove DNA contamination, contigs were required to begin at least two bases into a library combination and match up until the very end of the RBS (corresponding to the start codon).

A.2.2 *Protein Level Calculation*

To calculate protein expression levels for each construct, we first normalized the counts from each bin to one another using the total fraction of cells in the library that fell into each particular bin. We defined the fraction of cells sorted in each bin as f_j, so that $\sum_j a_{ij} = 1$, and the number of occurrences of sequence i in each bin j as c_{ij}. Then normalized fractional contribution of each bin j per sequence i, a_{ij} is calculated as:

$$a_{ij} = \frac{f_j \cdot c_{ij}}{\sum_i c_{ij}} / \sum_j \frac{f_j \cdot c_{ij}}{\sum_i c_{ij}}_j$$

so that $\sum_j a_{ij} = 1$ for all i.

Once the compensated bin distributions were calculated, we used the median fluorescence level in each bin as the value for all observations in that bin. We defined the center of the measurement range for each sorted bin j as m_j. The protein level, p_i was then calculated as:

$$p_i = \exp\left[\sum_j \left(a_{ij} \cdot \log^{\cdot}(m_j)\right)\right]$$

A.2.3 *FlowSeq Minimum and maximum cutoff*

Due to the placement of the bin cutoffs during sorting, there were upper and lower boundaries on the linear measurement range for protein level. These thresholds were empirically determined to be two times the minimum protein level and 99% of the maximum protein level (noted with a dotted line in Fig. A.10). 14.3% of constructs were below this range and 6.5% were above. These out-of-range data were not used to calculate ordering or average strength of promoters and RBSs, though we do display them as measured in Figs. 2.2 & 2.3.

A.2.4 *Calculation of Transcription Start Sites*

Using the RNA contigs aligned to the library, we determined the transcription start sites (TSS) for each promoter. After filtering RNA contigs as described above, the transcription start site for each unique sequence was determined, relative to the RBS + Promoter junction. RNA contigs could in most cases be assigned uniquely to an RBS + Promoter pair because of the unique barcode appended to the end of every promoter sequence. To calculate a single transcription start site per promoter, the alignment offset of each RNA contig against its DNA sequence was recorded. 87% of all promoters had one dominant start position (>60% of all mapped contigs). The most prevalent start site was used to calculate the RNA secondary structure as described below. Two promoters (marked with a * in Fig. A.7) had very few uniquely mapping contigs, did not show a strong start site, and showed unrealistic translation efficiency calculations. These observations indicated that we were missing most of the RNA data (but not protein data) from these promoters because of transcription starting after the end of the barcode sequence. The 222 constructs (1.7%) containing these promoters were removed from all analyses.

A.2.5 *RNA Level Calculation*

RNA Levels were calculated separately for each technical replicate, using a ratio of normalized RNA to normalized DNA:

$$RNA_i = \frac{c_{i,RNA}}{\sum c_{RNA}} / \frac{c_{i,DNA}}{\sum c_{DNA}}$$

where i is each individual construct, c_i is the number of DNA or RNA contigs for construct i, and $\sum c_{RNA}$ and $\sum c_{DNA}$are the total number of sequenced and merged RNA and DNA contigs prior to filtering. The RNA levels between the replicates showed a high level of correlation (R^2=0.992) and were averaged.

A.2.6 *Filtering of RNASeq and DNASeq Data*

RNA and DNA data was adjusted or discarded from some constructs based on low contig counts. 184 constructs (1.4%) did not have at least 10 DNA contig counts in both replicates, and were discarded. Seven additional constructs (0.7%) had fewer than 20 RNA contig counts and also had fewer than 50 DNA contig counts and were also discarded. 275 constructs (2.2%) had sufficient DNA but insufficient

RNA contig counts , and so their RNA contig counts were set to 10 (separately for each technical replicate) for purposes of RNA level calculation described above.

A.2.7 *Calculation of Average Transcription/Translation Levels*

Average transcription and translation levels were calculated for all Promoters and RBSs respectively. To calculate the average promoter transcription level, the geometric mean of the RNA level was calculated across each promoter, excluding constructs with insufficient RNASeq/DNASeq contig counts as described above. To calculate the average RBS translation level, the translation efficiency was first calculated per construct as the ratio of protein level to RNA level. The average translation level for each RBS was then calculated as the geometric mean of this translation efficiency. Constructs with protein levels above and below the aforementioned minimum and maximum thresholds were excluded from this calculation, as were constructs with insufficient RNASeq/DNASeq contig counts.

A.2.8 *Element ordering*

Because we did not want missing constructs with strongly expressing Promoter and RBS elements to influence the element ordering, we used the average deviation from mean values across all elements for ranking purposes.

The naming and ordering of each Promoter was determined as:

$$o_p = \frac{1}{n_r}\sum_r \left[\ln\left(RNA_{p,r}\right) - \frac{1}{n_p}\sum_p \ln\left(RNA_{p,r}\right)\right]$$

where n_r and n_p are the number of RBS and Promoter elements respectively, and $RNA_{p,r}$ is the RNA level for a Promoter/RBS combination. This ranks the promoters by how much each Promoter/RBS construct deviates from the average RNA level across all RBSs. Promoters were sorted and named Ec-TTL-P### (E. coli Transcription/Translation Library Promoter #), from 001 to n_p based on their rank-ordered o_p value.

RBSs were ordered similarly, with the equation:

$$o_r = \frac{1}{n_p}\sum_p \left[\ln\left(PROT_{p,r}\right) - \frac{1}{n_r}\sum_r \ln\left(PROT_{p,r}\right)\right]$$

where $PROT_{p,r}$ is the Protein level for a Promoter/RBS combination. This ranks the RBSs by how much each Promoter/RBS construct deviates from the average protein level across all promoters. Individual RBS were ordered from 001 to n_r

based on o_r, as Ec-TTL-R### (E. coli Transcription/Translation Library Ribosome Binding Site #).

A.2.9 *Calculation of Secondary Structure*

The 5′ UTRs used for secondary structure free energy determination were taken from the start of the dominant transcription start site to 30 bases into the coding sequence of sfGFP. Free energy of 5′ UTR regions were calculated using UNAFOLD's(56) "hybrid-ss-min -NA=RNA" command line program with default parameterizations.

A.2.10 *Simple model of transcription and translation*

To create a simple prediction for protein level, we took the product of the mean transcription per promoter and the mean normalized translation (i.e. translation efficiency) per RBS:

$$ln\left(\widehat{TRX_p}\right) = \frac{1}{n_r}\sum_r \ln\left(RNA_{p,r}\right)$$

$$ln\left(\widehat{TLX_r}\right) = \frac{1}{n_r}\sum_r \ln\left(\frac{PROT_{p,r}}{RNA_{p,r}}\right)$$

$$\widehat{PROT_{p,r}} = exp\left[ln\left(\widehat{TLX_r}\right) + ln\left(\widehat{TRX_p}\right)\right]$$

A.2.11 *Linear Modeling (ANOVA)*

We also constructed a linear model to determine the contribution of Promoter and RBS to both protein level and RNA expression level:

$$\log\left(PROT_{p,r}\right) = \alpha + P_p + R_r$$

$$\log\left(PROT_{p,r}\right) = \alpha + P_p + R_r$$

where α is the average signal, P_p is the p^{th} promoter and R_r is the r^{th} RBS.

A.2.12 *Statistical Analysis Software*

All statistics and tables described above were generated using custom software written in Python and R. Graphs were generated using the ggplot2 package in R.

A.3 SUPPLEMENTAL FIGURES

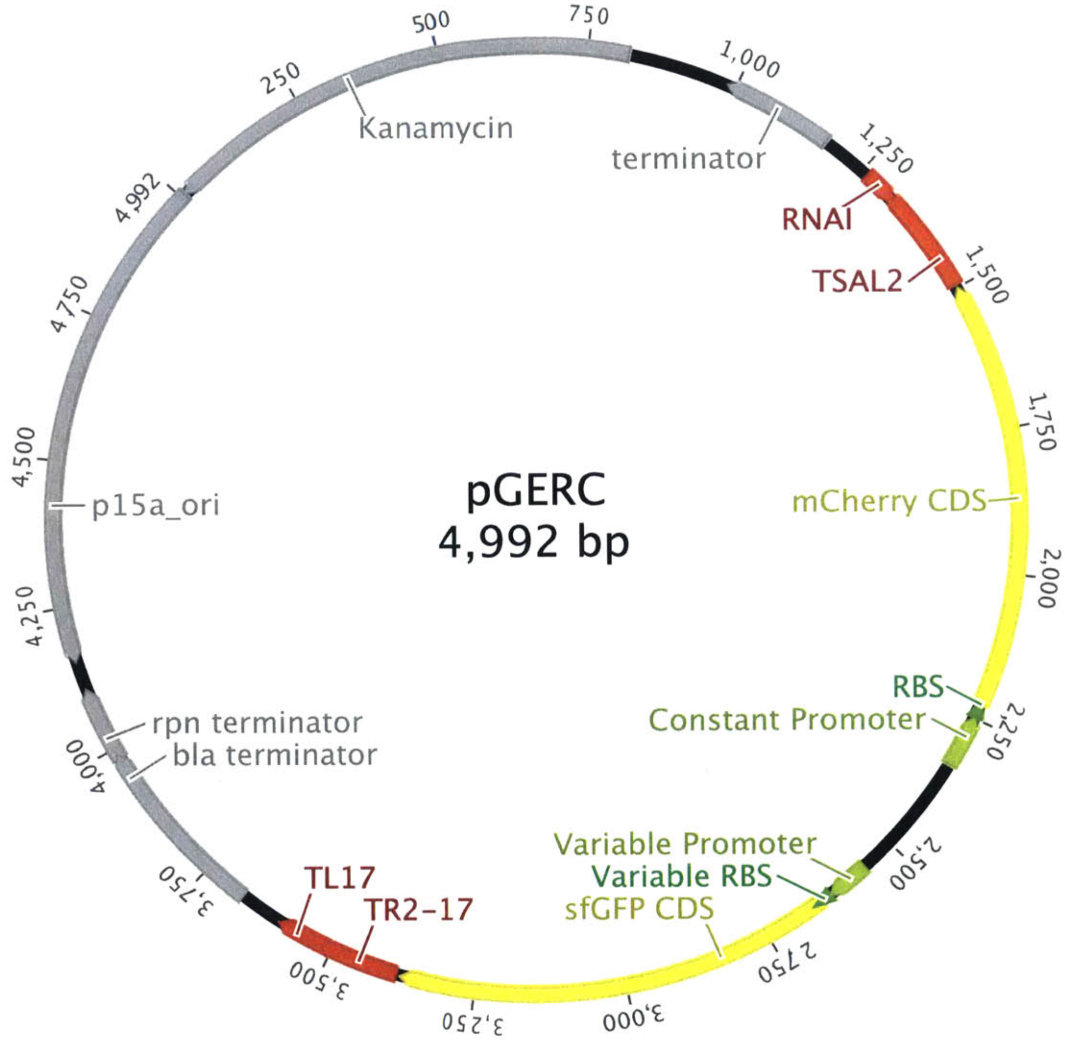


Figure A.1: *Plasmid Map of pGERC.* A plasmid map showing the sequence of pGERC (based on pZS-123) including the plasmid backbone in gray with Kanamycin resistance cassette, origin of replication, and terminators. The two fluorescent protein CDS regions are shown in yellow, while promoter and RBS regions are shown in green. Terminators for the fluorescent protein coding regions are shown in red.

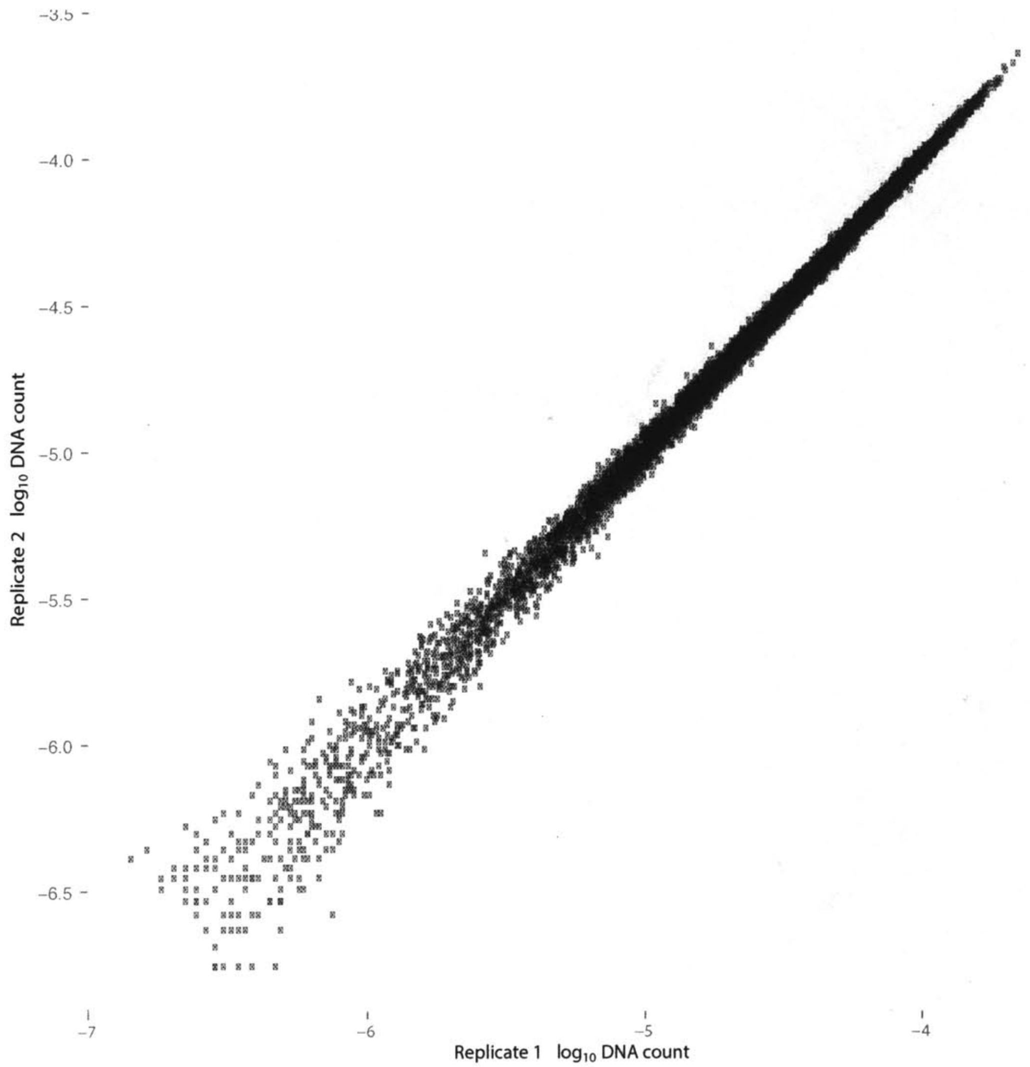


Figure A.2: *DNA technical replicate 1 & 2.* Observation frequency of library members across two technical replicates of DNA isolation, amplification, and sequencing are plotted against one another. The R2 of the linear model is 0.997. (F-test, p-value: < 2.2e-16)

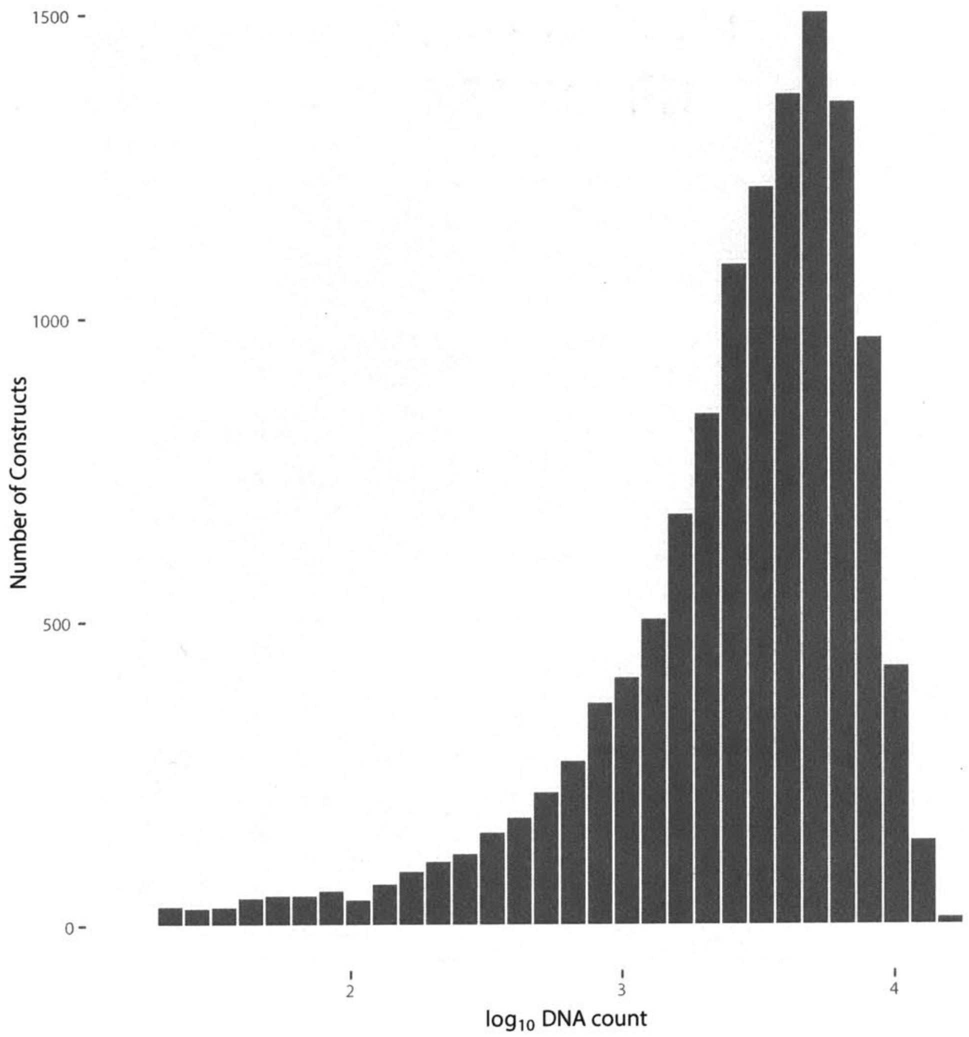


Figure A.3: *Distribution of contig counts for observed members of the library.* Library members with 5 or more counts across both replicates are binned and plotted on the histogram. 183 constructs were below the threshold and not plotted.

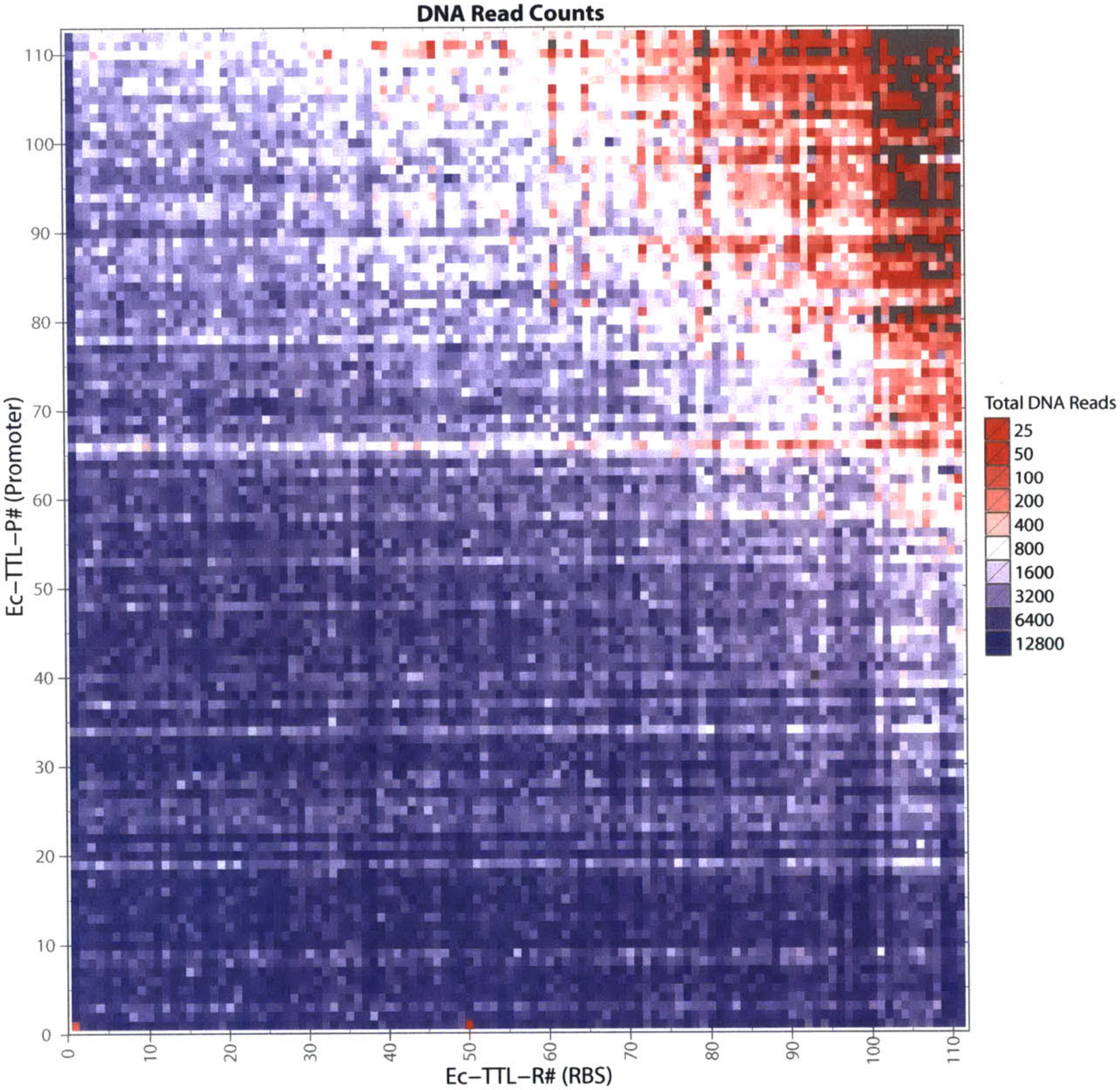


Figure A.4: *Distribution of DNA Contigs by construct.* Contig counts are displayed by color for each construct. Constructs are labeled by promoter (y-axis) and RBS (x-axis) and ordered as in Figs. 2.2 and 2.3. Dark grey boxes are unobserved contigs as well as one combination (040P-093R) that was not synthesized due to restriction site incompatibility. Most constructs with few contigs contain combinations of strong promoters and RBSs, potentially indicating that the high level of gene expression from these constructs affects growth and viability.

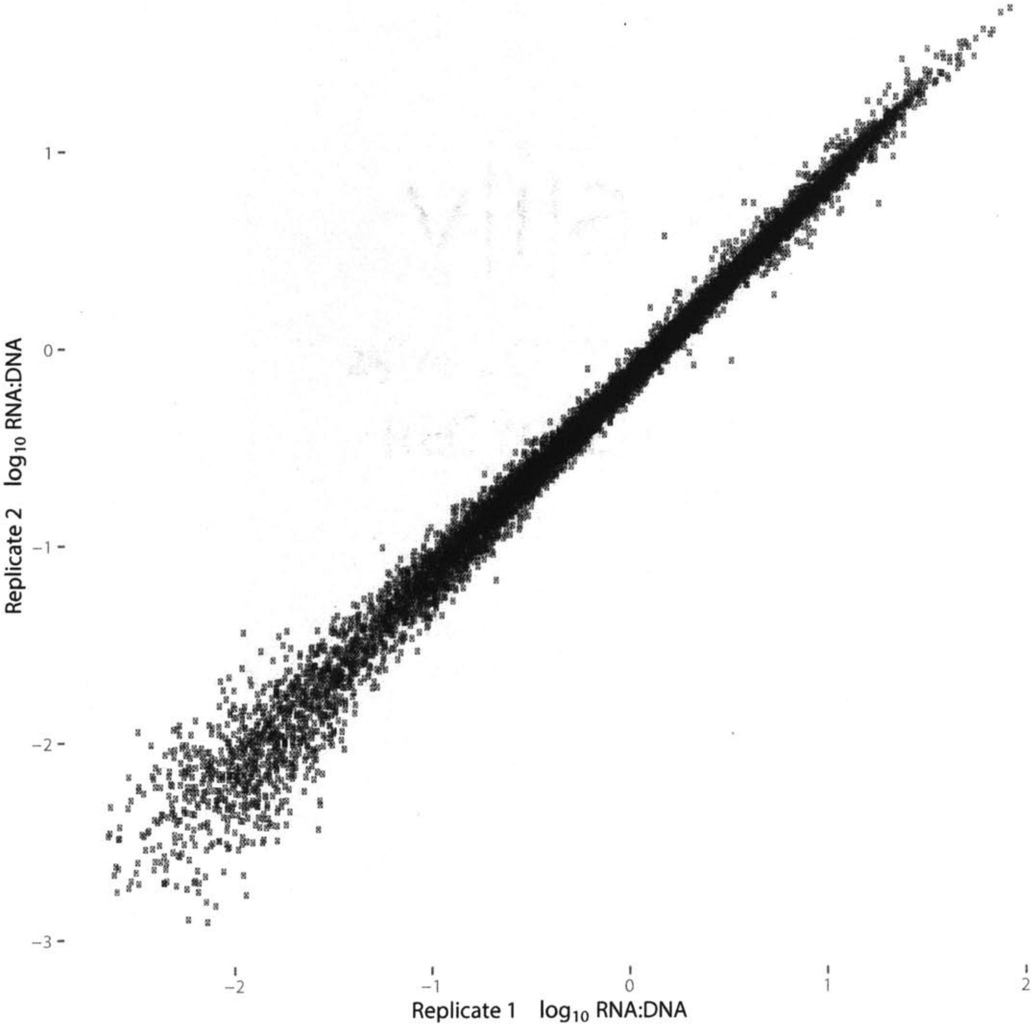


Figure A.5: *DNASeq ratio calculated separately for each technical replicate.* The RNA levels, as measured by the RNASeq:DNASeq ratios are plotted for two technical replicates and showed a high degree of concordance ($R^2 = 0.99$. F-test, p-value: < 2.2e-16).

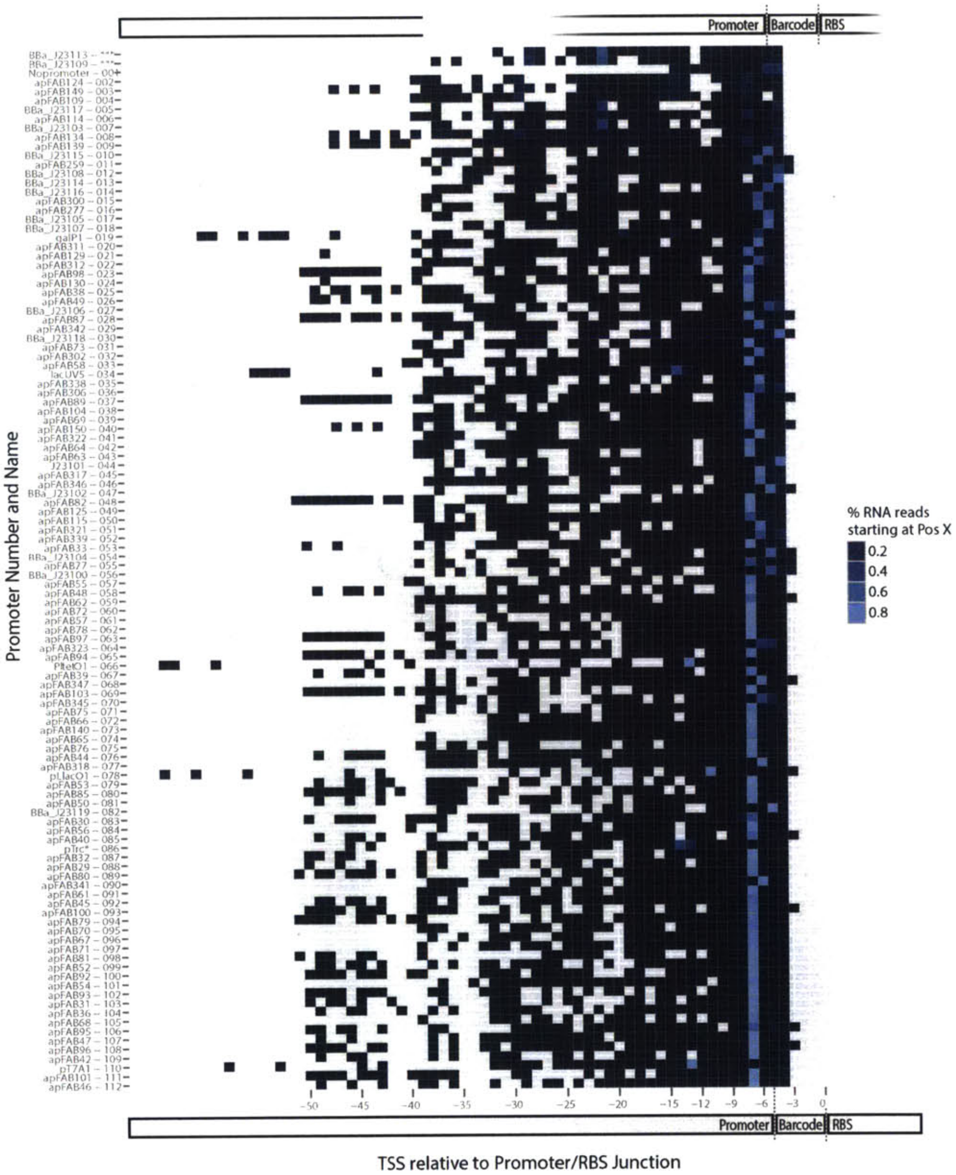


Figure A.6: *Transcription start site analysis.* The measured start positions from RNA contigs for each promoter are plotted, with more brightly colored squares indicating more common start sites. All positions are relative to the junction between the promoter-specific barcode and the RBS (see schematic at bottom). The 5-base promoter-specific barcode sequence allows promoter identification for RNA contigs that begin after the end of the functional promoter region. If an RNA contig begins more than 2 bases into the barcode, it cannot be mapped uniquely; those contigs are discarded. The first two constructs (top) were removed from further analysis due to start sites that presumably started after the barcode (see Chapter A.2.4); these are marked by three asterisks.

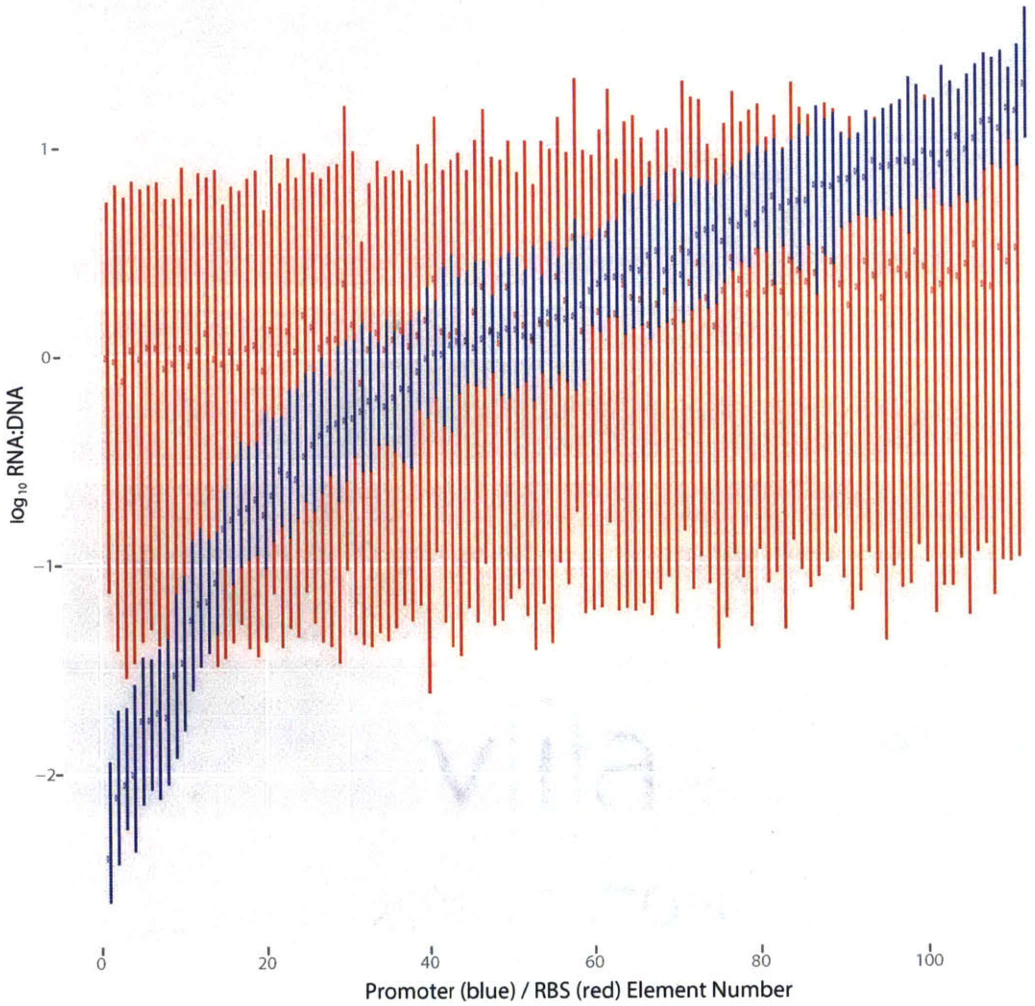


Figure A.7: *RNA levels across each promoter and RBS.* Mean RNA levels across all promoters (blue circles) and RBSs (red cirles) are plotted with lines corresponding to 10th and 90th percentile values. Promoter identity is tightly correlated to RNA level, while RBS identity has a slight effect positive effect, albeit with large variation.

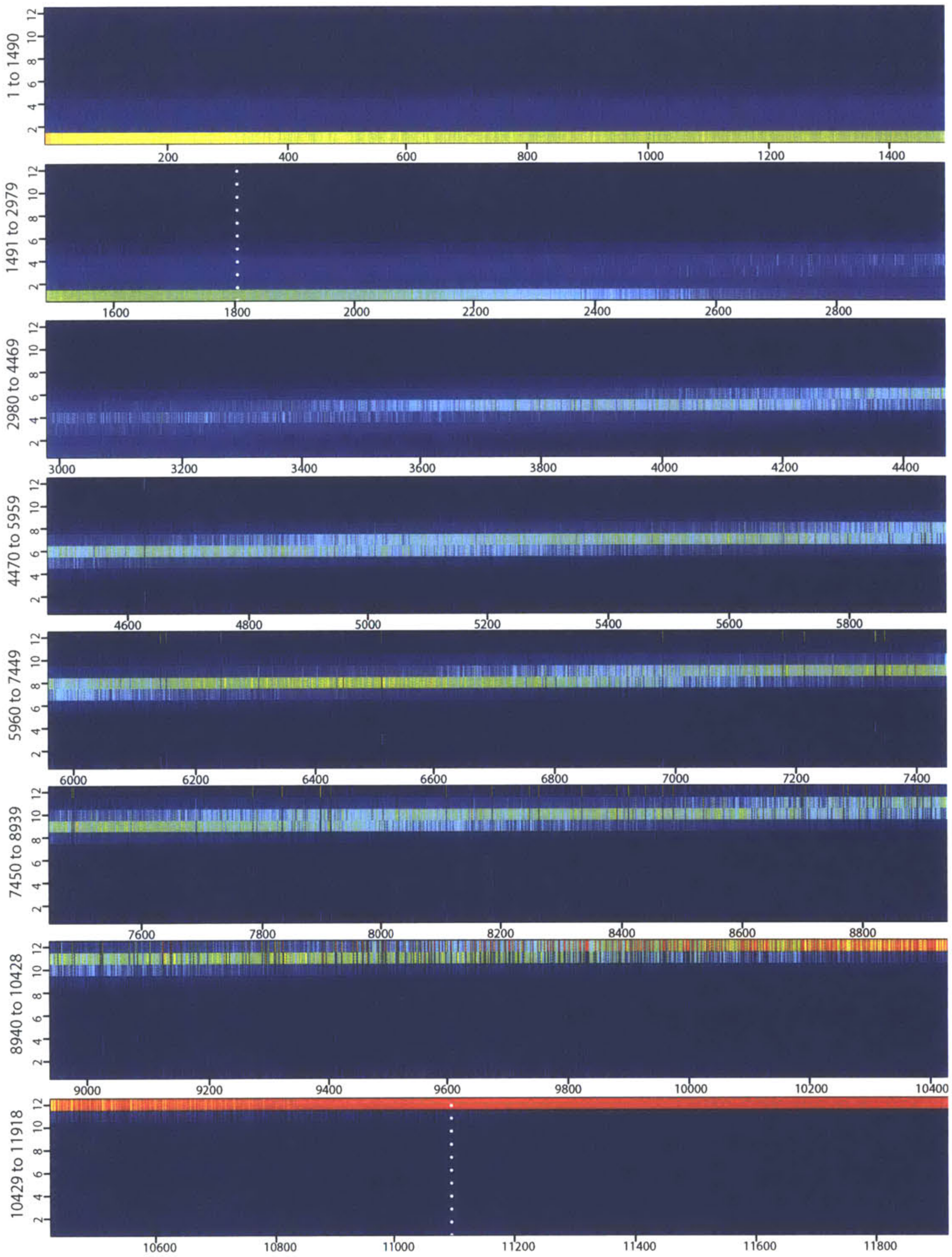


Figure A.8: *Percentages of contigs falling into each of the 12 bins across all constructs.* 11,981 constructs are shown on the x-axis, ordered by increasing protein level as estimated by FlowSeq. White dotted lines show the high and low protein level cutoffs, beyond which constructs cannot be accurately measured. Contigs for most constructs fall into a few contiguous bins, suggesting a continuous distribution of gene expression level among cells harboring the same construct. 735 Constructs with fewer than 100 counts or constructs whose contigs fell entirely into one bin (save the final bin) were discarded from analysis and are not shown here.

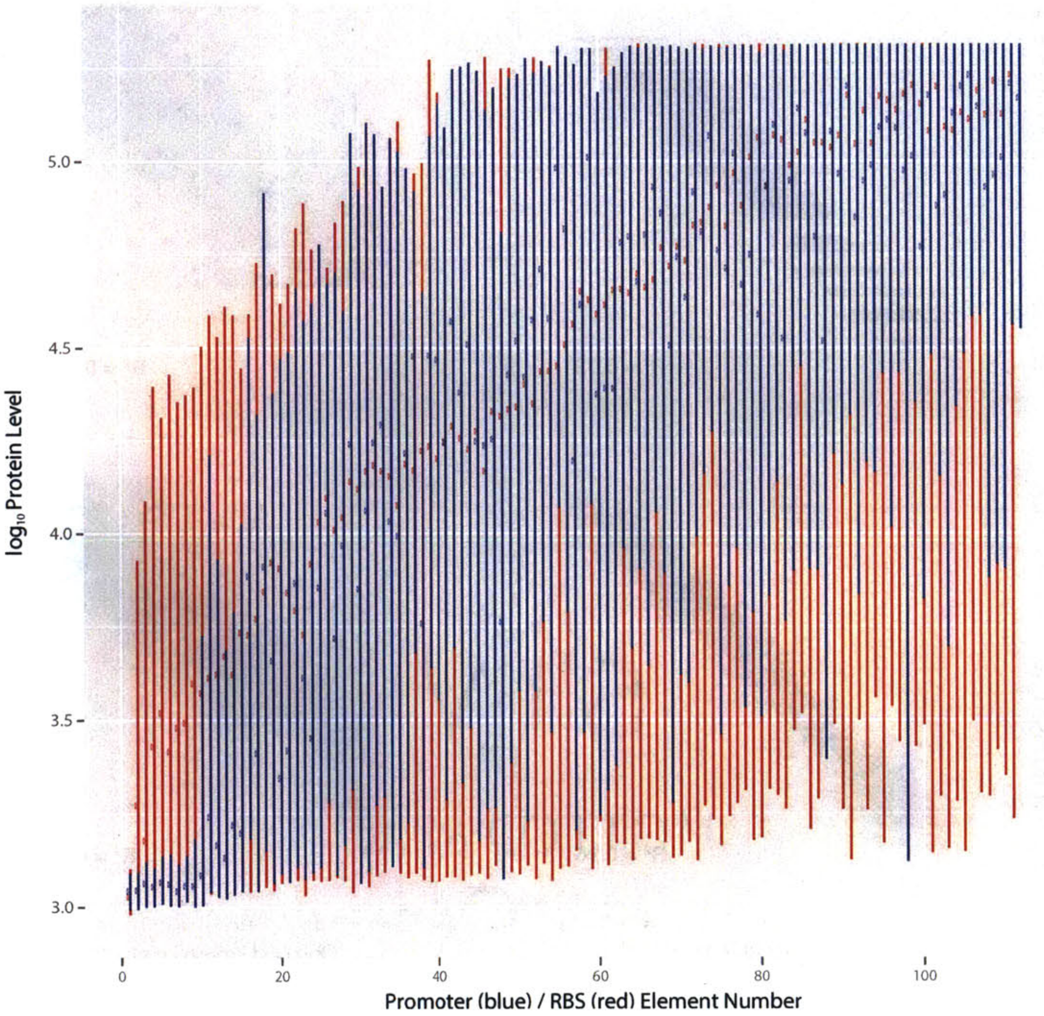


Figure A.9: *Protein levels across each Promoter and RBS.* Mean Protein levels across all promoters (blue circles) and RBSs (red cirles) are plotted with lines corresponding to 10th and 90th percentile values.

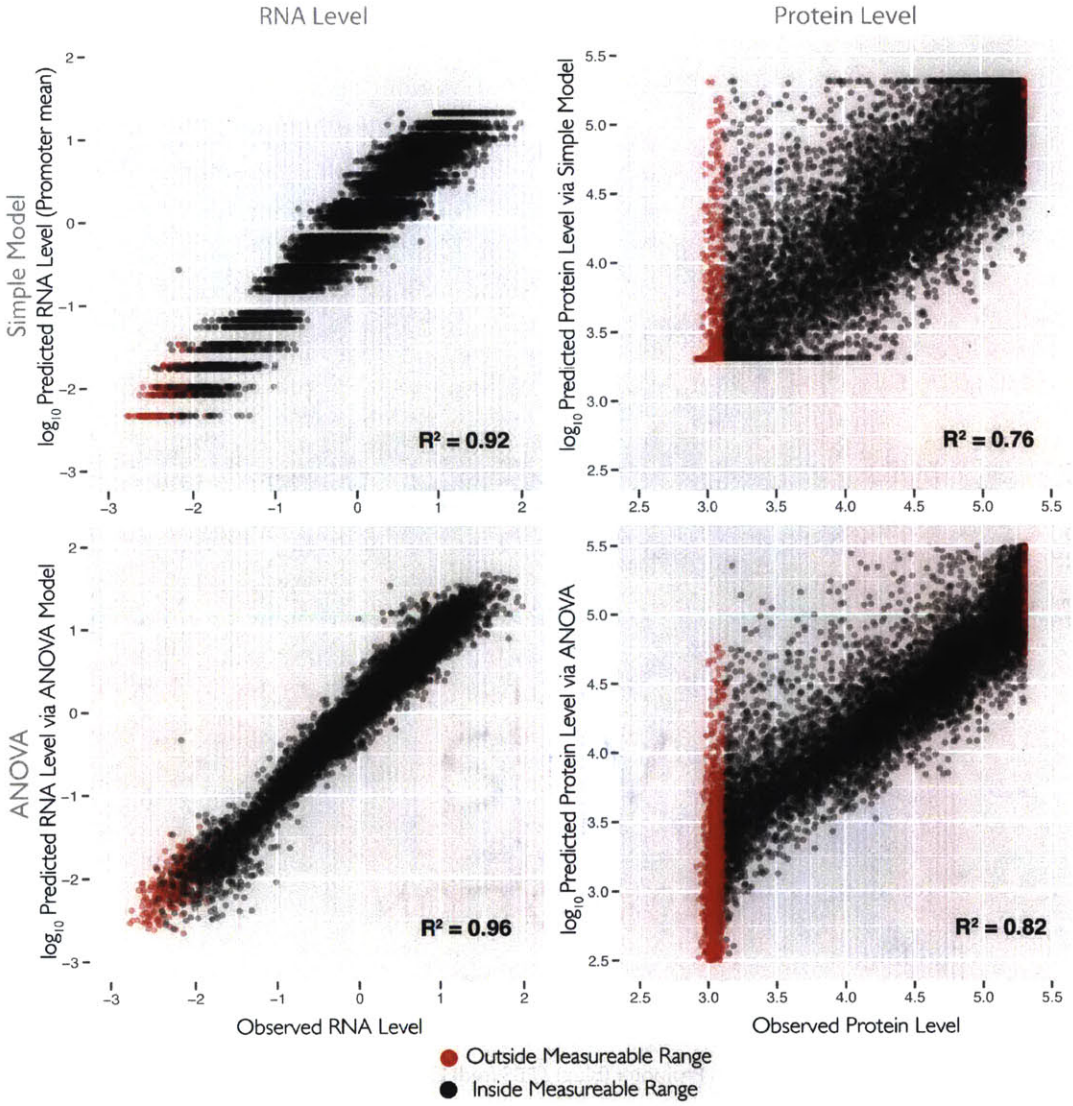


Figure A.10: *Comparison of simple and ANOVA models.* For each construct, we plotted predicted versus observed protein and RNA levels for the simple Promoter + RBS model (top) and the ANOVA model (bottom). Red points are those outside of the linear range of our FlowSeq measurement.

B

SUPPLEMENTAL INFORMATION FOR CHAPTER 3

This appendix contains the extended experimental methods, extended data analytical methods, and additional figures for Chapter 3. It is adapted from the supplemental information included with Goodman et al.

B.1 EXTENDED EXPERIMENTAL METHODS

B.1.1 *Reporter Construct*

We used the same reporter strain, pGERCA. Briefly, pGERC allows for independent expression of mCherry from the $P_{LtetO-1}$ promoter and the divergent expression of sfGFP with a replaceable promoter-RBS-peptide sequence driving expression. The design of this plasmid is based on pZS2-123 from Cox et al. [22], and uses a p15A origin of replication with a kanamycin resistance cassette.

B.1.2 *Library Design, Construction and Cloning*

We chose two promoters (BBaJ23100 and BBaJ23108) and three RBSs (BBa_B0032, BBa_B0030, BBa_B0034) from the Registry of Biological Parts that we previously characterized [43]. We used the first 11 amino acids including the initiating methionine from 137 essential genes in E. coli [101]. In addition, we added a fourth RBS that represented the natural RBS for each of the 137 genes by taking the 20bp sequence upstream of the start codon from the E. coli genome sequence [12]. For each promoter and RBS pair, we generated 13 variants where we changed codons used to encode the peptide, though always keeping the start codon as ATG. We refer to these 13 variants as a codon variant set, and in each set we included the natural sequence encoded on the genome (wt), a variant that only used the most common codon on the E. coli genome for each amino acid (C), and a variant that used the least common codon (R). In addition, we computationally generated 500 codon variants where each codon was chosen by a biased random pick based on natural genomic frequencies of codons encoding a particular amino acid. We used UNAFOLD [56] to predict free energy of folding for each RBS-codon variant, sorted the 500 based on this free energy, and took every 50th variant to give a total of 10 variants spanning a range of free energies. Finally, all sequences were flanked by AscI and NdeI, as well as sequences to allow for PCR amplification (skpp-203-F; TGT CGT GCC TCT TTA TCT GT & skpp-203-R; GCT TCG GTG TAT CGG AAA TG).

We synthesized and cloned the oligo library as previously described in Kosuri et al. [43]. Briefly, the oligo library was synthesized on DNA microarrays, cleaved, and lyophilized as a pool by Agilent Technologies (USA). The library was amplified by limited-cycle PCR using real-time PCR, cloned into pGERC at ~50 fold clonal coverage, and transformed into E. coli MG1655 (Yale CGSC No. 6300). We kept several frozen aliquots of the initial library for all subsequent experiments.

B.1.3 *Control Colonies and Flow Cytometry*

We plated the transformed MG1655 library and Sanger sequenced 282 clones, and found that 131 (46%) were error-free, and one sequence had two perfect duplicate clones. These sequence-perfect clones were inoculated from glycerol stocks into 200 μL of LB with kanamycin (50 μg/mL) and grown overnight at 30°C with shaking in 96-well culture plates. The cells were then backdiluted 1:1000 in 200 μL LB with kanamycin and grown for 3.5 hours, until the cells reached an of ~0.15-0.25. The cells were then immediately put on ice, pelleted by centrifugation, and diluted 1000-fold in ice-cold PBS. We measured mCherry and sfGFP fluorescence levels using a BD FACS LSRFortessa flow cytometer with a high-throughput sampling attachment (at least 30,000 events per observation). Events were gated on forward and side scatter to exclude debris, dead cells, and doublets. The overnight growth, backdilution, and flow cytometry procedure was performed on four separate days from two fresh back-dilutions per day, for a total of 8 replicates per clone. 7 clones were highly variable between replicates and were removed.

B.1.4 *Library Growth and FlowSeq*

The library was grown, sorted, and sequenced exactly as in Kosuri et al. [43]. Briefly, a frozen aliquot of the library was grown overnight at 30°C. 0.5mL of the overnight culture was backdiluted in 300mL of pre-warmed LB with kanamycin (50 μg/mL), and grown shaking at 250 RPM and 30°C to an OD_{600} of 0.2. The culture was immediately cooled in an ice-water slurry. Five 50 mL aliquots were pelleted by centrifugation, and four were snap frozen in liquid nitrogen. One aliquot was resuspended in 50 mL of ice-cold PBS, which was then directly put into the FlowSeq protocol. For FlowSeq, cells were sorted using three consecutive runs at 4°C on a BD FACSAria IIu. Each run sorted four non-adjacent log-spaced bins of varying sfGFP:mCherry ratio. One million cells were sorted in bin 1, 250,000 in bins 2-10, and 100,000 cells in bins 11 and 12. Cells in all bins were grown overnight while shaking in room temperature LB to minimize growth rate differences. Plasmids were isolated, and constructs were PCR amplified using limited-cycle RT-PCR to add barcodes and primer sequences for Illumina sequencing. All 12 bins were sequenced on a single lane of a HiSeq 2000 paired-end 100nt lane.

B.1.5 *DNASeq and RNASeq*

Both DNA and RNASeq experiments were done as previously described [101]. Briefly, for DNASeq, we isolated plasmids from two frozen cell pellets from the 50 mL library growth culture using a Qiagen midiprep kit. We amplified the library as we did in the FlowSeq experiment, using two separate barcodes. Both DNASeq libraries were run on a separate lane in the same HiSeq run as the FlowSeq data. RNAseq was done by isolating RNA independently from two 50mL frozen pellets and removing rRNA, removing 5′ triphosphate using RNA 5′ Polyphosphatase. Then we ligated a RNA primer with two bases of 3′ degeneracy to control for ligation biases using T4 RNA Ligase [87]. Then we used SuperScript III (Invitrogen) to make cDNA, amplified, barcoded, and sequenced RNA again using a separate lane in the same HiSeq run as the FlowSeq data.

B.2 DATA ANALYSIS

B.2.1 *Contig building and trimming*

Contig building and trimming was done exactly as reported in Kosuri et al. [43]. Briefly, we used a custom modified version of SeqPrep [43] and custom scripts to pair and trim reads into contigs to increase sequencing fidelity over regions of paired-end coverage.

B.2.2 *Deduplication and sorting of unique reads*

After trimming, occurrences of each unique contig were counted per bin and merged to generate a vector of 12 numbers corresponding to the count per bin per merged contig. These unique de-duplicated contigs were then aligned to the Promoter + RBS reference sequence library using Bowtie [50]. In the case of FlowSeq contigs: first, contigs were required to be perfect end-to-end matches to a library member, second, consist of at least 100 read pairs, and third, occur in multiple bins, excepting the final bin. For DNA and RNA reads: first, no more than three single base mismatches were allowed (no indels), second, were required to match to only one reference sequence, and third, to remove potential DNA contamination, contigs were required to begin at least two bases into a library combination and match up to the beginning of the sfGFP sequence.

B.2.3 *Calculation of Transcription Start Sites*

Using the RNA contigs aligned to the library, we determined the transcription start sites (TSS) for each promoter. After filtering RNA reads as described above, the transcription start site for each unique sequence was determined, relative to the RBS/Promoter junction. For every library member, we ensured that RNA reads could be mapped uniquely due to a 5 base unique sequence appended after each of the promoters. To calculate a single transcription start site per library member, the alignment offset for each RNA read against its DNA sequence was recorded, and the most frequent transcription start location was used for RNA secondary structure prediction.

B.2.4 *FlowSeq Measurement Reconstruction*

Protein levels were reconstructed using the FlowSeq measurements exactly as in Kosuri et al. [101]. Briefly, we calculated protein expression levels for each construct by the normalized frequency of constructs in each sorted bin, multiplied by the log-mean fluorescence level per bin.

B.2.5 *FlowSeq Measurement Boundaries*

Due to the bin-based sorting procedure, constructs falling entirely into the lowest or highest bin cannot be accurately measured. Any constructs whose reconstructed fluorescence level fell below the middle of the second bin (< 1584 RFU) were considered below measurement range. Any constructs where > 95% of reads fell into the final bin ($RFU \geqq 255,000$) were considered above measurement range. Values for these constructs were still factored into normalization measurements if at least half of the codon variants in a set were within measurement range.

B.2.6 *Expression Normalization*

We normalized our expression measurements across each 13-member codon variant set to control for potential differences in sfGFP fluorescence that might result from changing the amino acid sequence of the N-terminal peptide. We first took the log-mean of all 13 FlowSeq measurements as the average, and subtracted the log fluorescence measurement for each from that average, to get the logarithmic fold-change from average. We only considered codon variant sets where at least 50 percent of constructs were within measurement range.

B.2.7 *Individual Codon Correlations*

For every codon we conducted a linear regression where we first took all constructs containing at least one instance of the amino acid encoded by the codon and where at least 5 of the 13 set members were within expression range. We regressed the frequency of each codon against the logarithmic fold-change from average expression. The slope of the regression (labeled in the figures and text as the "Mean Fold-Change in Expression Due to Codon Substitution") thus represents the average fold-change in expression after using one additional instance of the codon in lieu of another synonymous codon. A Bonferroni correction ($p/58$, for 58 codons w/ synonymous substitutions possible) was applied to each of the p-values.

B.2.8 *Shine Dalgarno motifs*

We used an approach similar to the one employed by Li et al. [52]. We began by finding the free energy of hybridization between the 8-base pair anti-Shine Dalgarno sequence (CCTCCTTA) and each of the nucleotide hexamers. For each construct we summed these hybridization scores for every hexamer starting at position -2 relative to ATG up to +32 (each variable region extends from +3 to +33). We then normalized the summed hybridization scores among each of the codon variants per set and performed a linear regression between the difference in hybridization score and the logarithmic fold change in protein expression.

B.2.9 *Normalized Translation Efficiency (nTE)*

Pechmann et al. recently described a new metric, the normalized translation efficiency or nTE, which is the tAI score adjusted for codon demand (i.e. expression level) as well as tRNA supply [65]. Using the mRNA measurements from Shiroguchi et al. [83] and protein and mRNA measurements from Lu et al. [55], we calculated multiple different nTE statistics as they described. We removed pseudogenes, genes with known frameshifts, and for calculations involving the expression measurements, only included genes where measurements were provided.

Upon correlation with our per-codon expression metric (Fig. 3.2A), nTEs derived from all expression measures performed similarly to the tAI, and the protein measurements from Lu et al. [55] generated nTE scores with the best, albeit still insignificant p-value ($R^2 = 0.0241$, $p = 0.124$). This nTE score was also weakly yet significantly correlated with RSCU ($R^2 = 0.206$, $p = 1.76 \times 10^{-4}$) and genomic N-terminal codon enrichment ($R^2 = 0.107$, $p = 6.54 \times 10^{-3}$). However, the unadjusted tAI score performed better in these correlations than any of the nTE metrics, including per-codon expression with our data ($R^2 = 0.0438$, $p = 0.111$) and codon frequency enrichment at the N-terminus ($R^2 = 0.114$, $p = 5.10 \times 10^{-3}$).

B.2.10 *Transcript Secondary Structure Prediction*

First, we used our mRNA sequencing data to find the most frequent transcription start site for each construct. We then used the NUPACK software [103] to find the minimum free energy of folding for the mRNA transcript of every construct, using the sequence starting from the computed transcription start site (TSS) to 96 bases into the sfGFP protein, for a total of at least 149 bases per construct. The mfe program within NUPACK was used, at a temperature of 37°C, with contributions from dangling ends (-dangles all) included. We used 37°C because the most commonly used RNA model can only calculate structure at that temperature. Secondary structure was also calculated starting at -20 before the ATG for every construct, instead of computing individual transcription start sites. These measurements were extremely similar ($R^2 = 0.9385$) and did not change the analyses, and secondary structure predicted from the TSS was used.

B.2.11 *Window Structure Prediction*

For each construct, we first computed the pair probabilities using the NUPACK pairs command. This generates the probability of hybridization between every pair of bases i and j for the entire RNA sequence of length l, from the transcription start site to 96 bases into the sfGFP protein. From this we calculated the pairing probability for each base, the probability of base pairing with any other base in the transcript:

$$p_i = \sum_{j=1}^{l} p_j$$

We then took a windowed average of the pairing probabilities for each base over a sliding window of size w, to generate wp_i, where each the window pairing probability is centered, rounding down, at base i. We calculated these measurements for a w of 5, 10, 20, and 40 bases.

Next, we calculated the fold change in window relative hybridization probability rhp_i by comparing the pairing probability of each window with the same window position for other constructs in the codon variant set, and log-adjusting the measurements to get logarithmic fold change:

$$rhp_{i,c} = \log_{10}\left(wp_{i,c}\right) - \frac{\sum_{j=1}^{C} \log_{10}\left(wp_{i,j}\right)}{length(C)}$$

Here c is a single construct, and$C$ is the set of all 13 constructs in the codon variant set (same promoter, same RBS, different synonymous codon usage).

This window relative hybridization probability represents the relative probability that bases within window are paired, compared to other synonymous codon variants with the same UTR. This value is lower than zero if the region is less likely to be folded, and higher than zero if the region is more likely to be folded. It does not explicitly consider the strength of folding energy (due to GC content differences, for instance).

We chose this metric rather than local window folding because it represents a more global picture of RNA folding, as pairings between bases inside and outside of the window are captured, whereas with local folding in the window, any interactions outside of the window are not considered. This allows smaller windows to be used with more accuracy, and allowed us to assay how codons within the N-terminus might hybridize to nucleotides both in the constant sfGFP sequence and in the upstream RBS.

B.2.12 *Multiple Linear Model*

We used multiple linear regression to explain the log-adjusted protein expression levels for codon variant sets within the range of our assay. We used promoter identity, RBS identity, codon GC content, and N-terminal folding energy as regressors, and included second-order interaction terms. We then applied Type II ANOVA to find the explained sum of squares, which was 0.541. We excluded wild-type RBSs from our regression as each wild-type RBS is only ever used with its cognate peptide sequence, which would unbalance our data. Promoter choice explained 28.6% of variation, followed by folding energy (11.7%), RBS choice (10.9%), and GC content (1.94%). While interaction terms were statistically significant, no individual interaction explained more than 0.385% of variation. 45.8% of variation was unexplained (residual sum of squares). We did not include tAI or CAI in our multivariate regressions because after comparing models with and without these regressors using ANOVA, we found that they did not significantly improve the fit.

B.3 SUPPLEMENTAL FIGURES

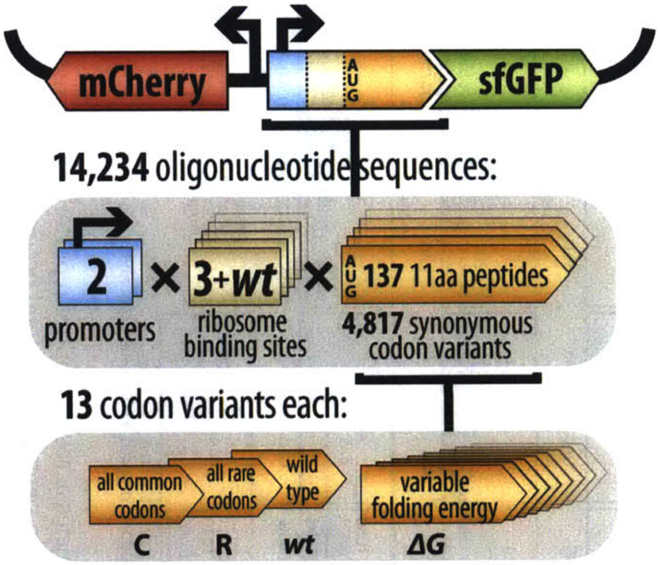


Figure B.1: *Diagram of Expression System and Library Design.* Each library construct contained a promoter, RBS, and 11 codon N-terminal peptide (including the initiating 'ATG'). The peptide sequences correspond to the N-terminus of 137 natural E. coli genes. For each promoter/RBS/peptide combination, we encoded 13 codon variants including the most common codons (C), most rare codons (R), wild-type sequence (wt), and codon variants with variable secondary structures (ΔG). The library was cloned in-frame with superfolder GFP (sfGFP). The GFP expression level is compared via relative fluorescence to a constitutively co-expressed mCherry protein.

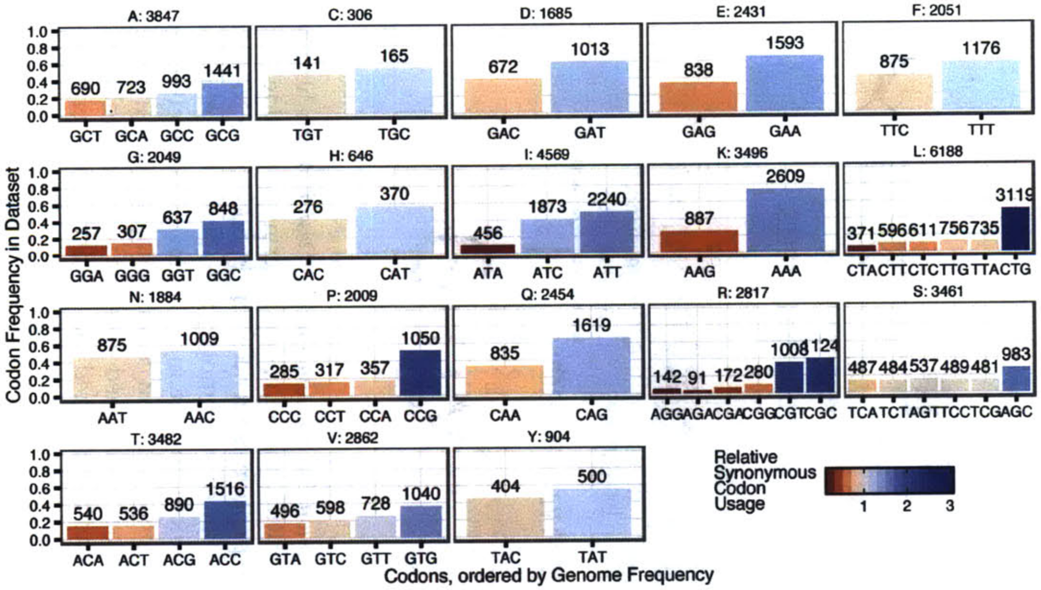


Figure B.2: *Codon Frequencies in Library.* The codon frequencies across the entire reporter library are representative of natural E. coli codon frequency distributions. Bar heights represent the percentage of total occurrences of the amino acid, and numbers above each bar is the absolute number of occurrences. Bars for each codon are colored by their relative synonymous codon usage (RSCU).

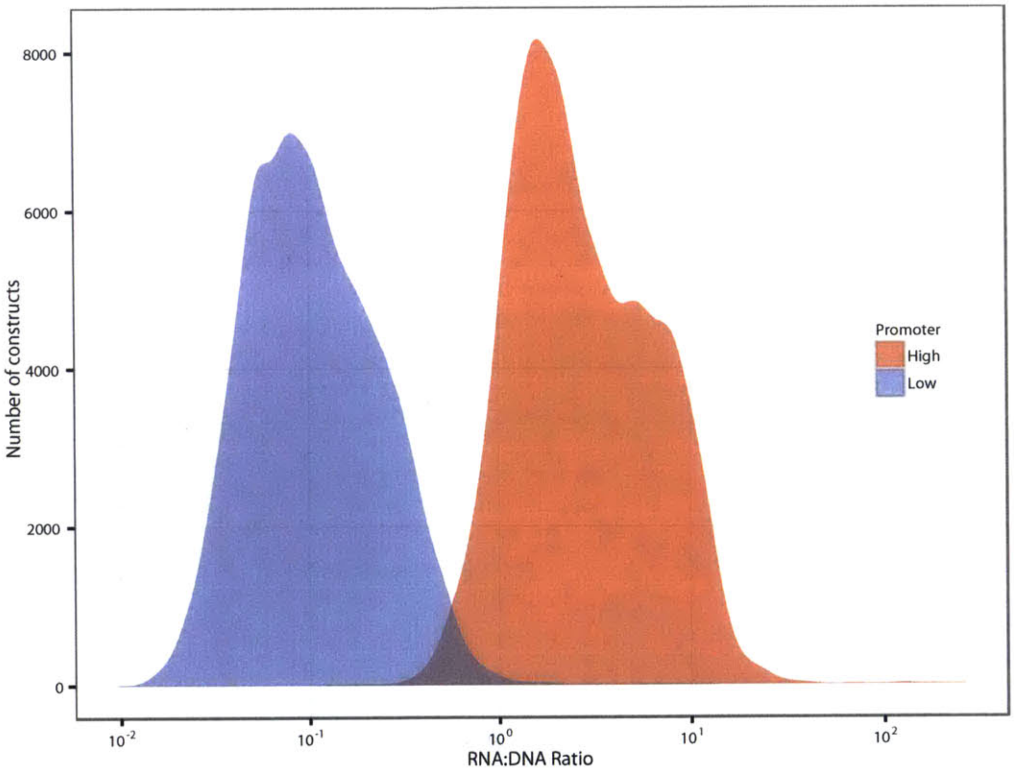


Figure B.3: *RNA Abundance by Promoter.* The relative RNA abundance for all constructs expressed as a ratio of RNA to DNA contig count show that the promoters strengths are different.

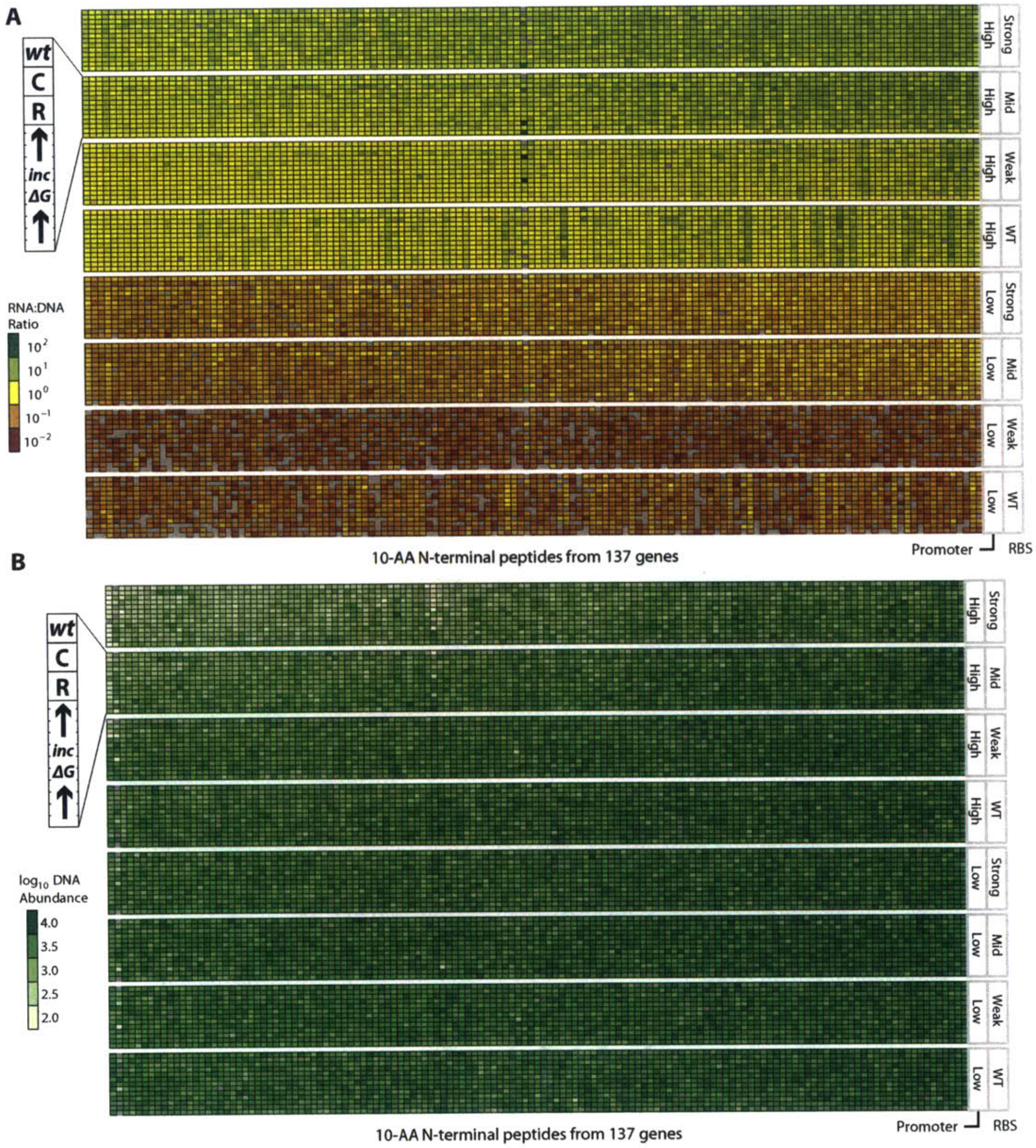


Figure B.4: *RNA and DNA Abundance.* (A) Relative RNA abundances (ratio of RNA to DNA contig counts) for each construct are displayed grouped by promoter and RBS identities. (B) Same plot as in (A) but displaying DNA abundances for each construct. Labels at the right are the same as in Figure 1E, where promoter identity is the first column and RBS identity is the second column. DNA abundances varied due to differences in DNA synthesis efficiencies as well as lower growth rate for very highly expressed genes.

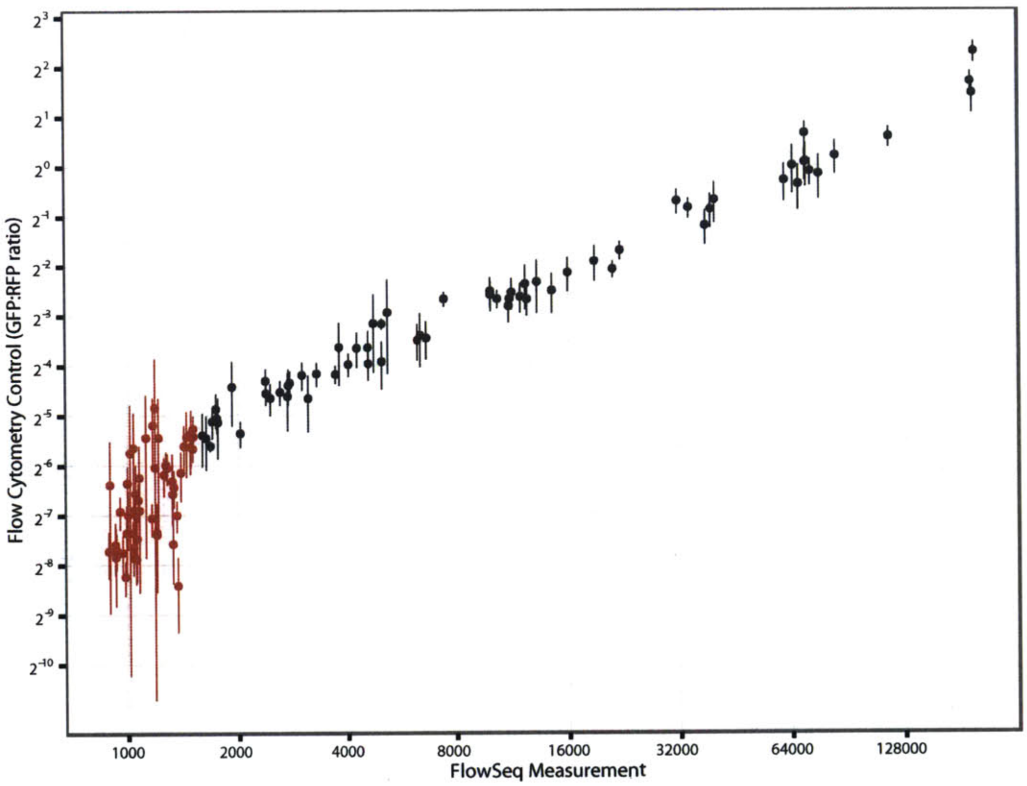


Figure B.5: *All Flow Cytometry Controls.* FlowSeq estimates of fluorescence ratio correlate well (R2 = 0.955, p < 2x10-16) with individually measured fluorescence ratios from sequence-verified clones. 51 constructs that were outside of the quantitative FlowSeq range are shown in red.

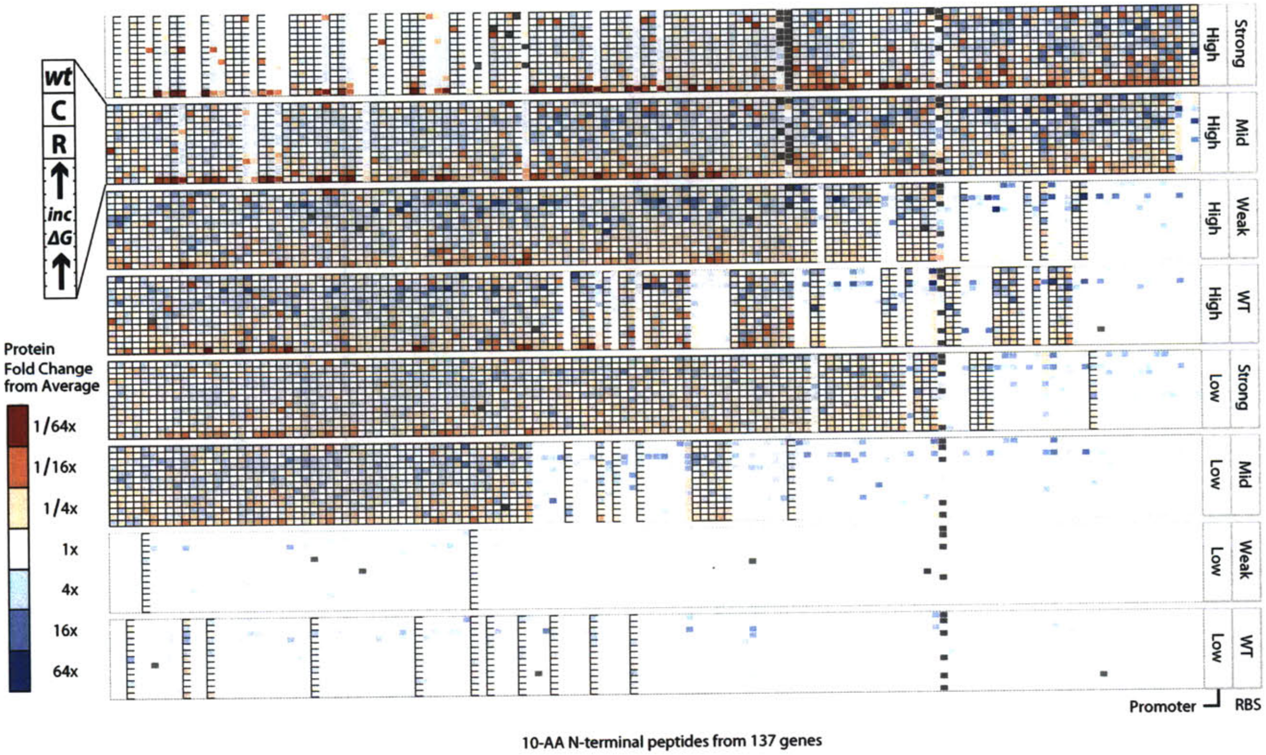


Figure B.6: *Codon Variant Sets and Protein Fold Change.* Each 13-member codon variant set is shown as a column, and each member is colored according to its fold-change from the set log-mean. Labels at the right are the same as in Figure 1E, where promoter identity is the first column and RBS identity is the second column. All variant sets are shown here, including those on the Low Promoter / WT RBS and Low Promoter / Weak RBS combinations that are removed from Figure 1E. The sets where at least half of the variants were below or above quantitative range are outlined in white, while sets within range are outlined in black.

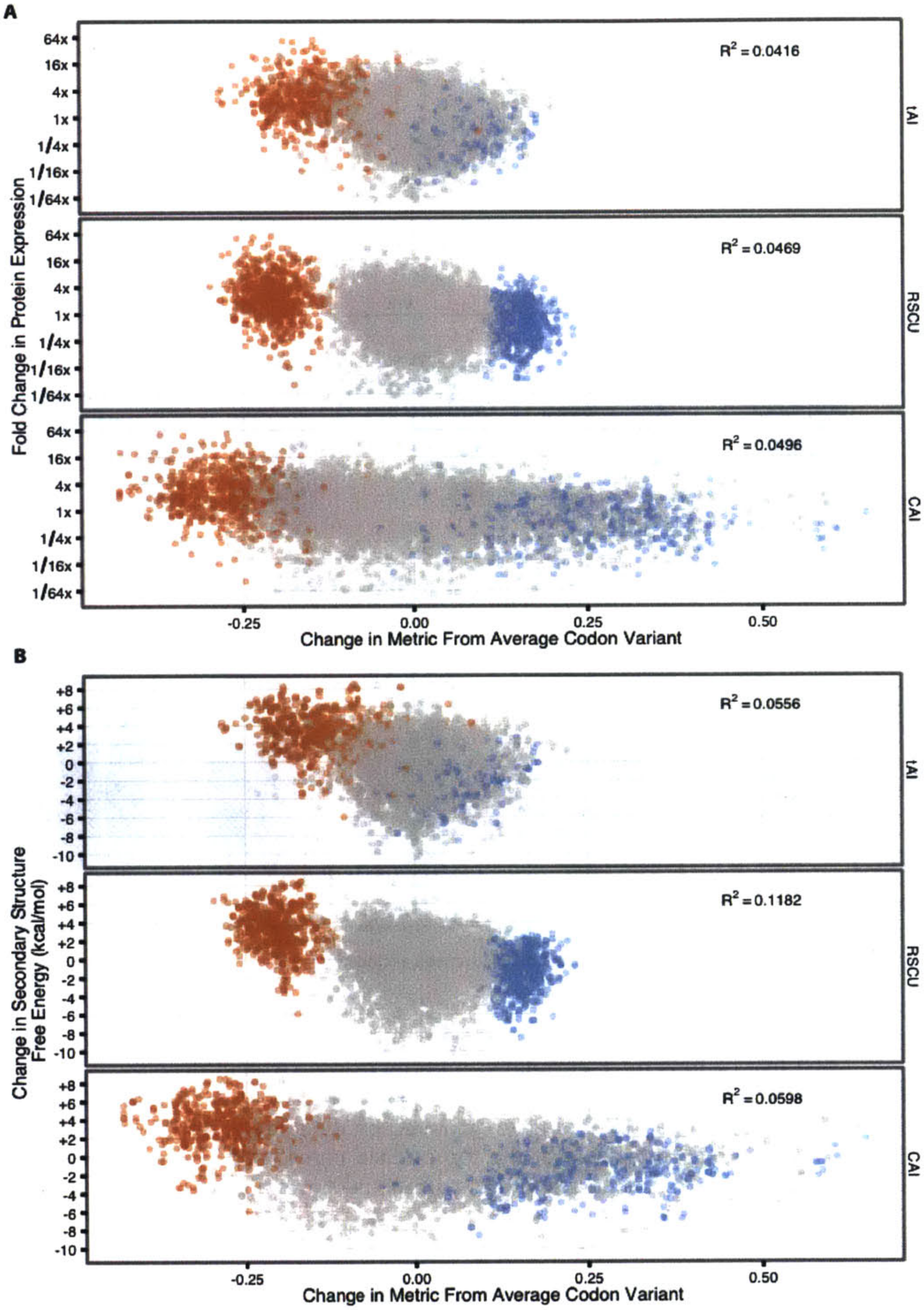


Figure B.7: *Codon Metrics versus Expression and Secondary Structure.* Codon Metrics versus Expression and Secondary Structure. (A) The relationship between three commonly used codon metrics and fold change in protein expression within a codon variant set is fairly weak. (B) The relationship between these codon metrics and change in the free energy of folding at the N-terminus within a codon variant set is fairly similar. In both plots, codon variants that use rarest codons only are red, and variants using only the most common variant are blue, and all other constructs are in gray.

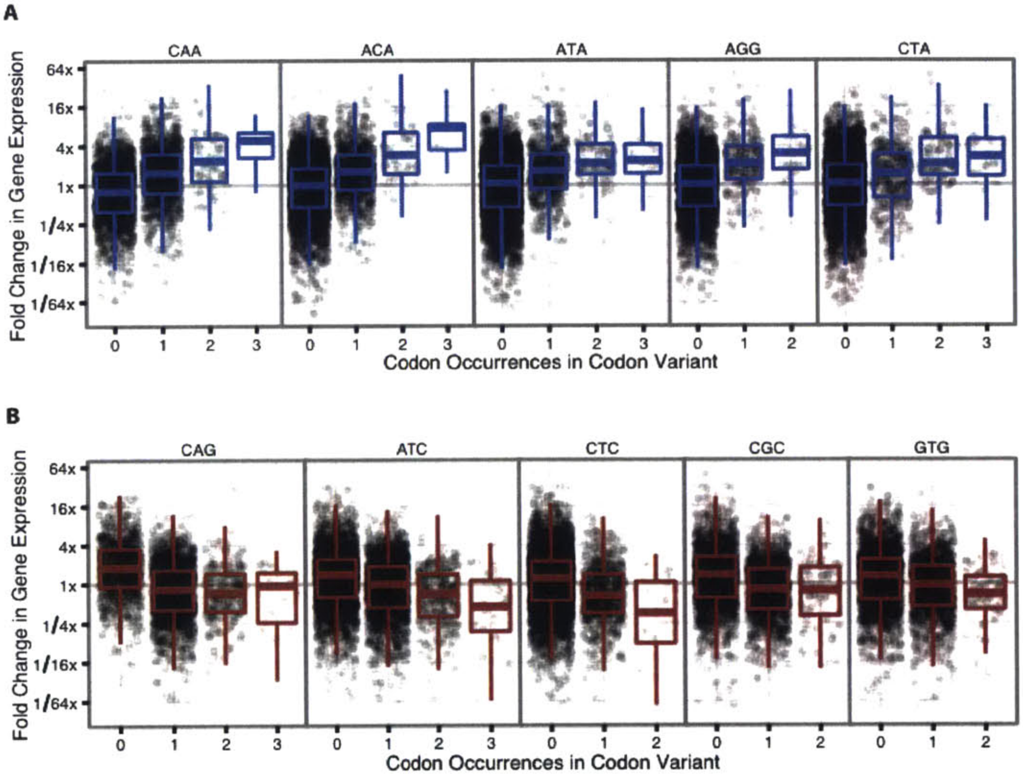


Figure B.8: *Individual Codon Correlations.* Codon variants containing increasing counts of the five codons most correlated with increased (A) and decreased (B) expression are shown as red and blue boxplots, respectively. Each point represents a single variant, and variants are grouped by the number of times they use each codon, which is labeled at the top of each panel. Only variants with at least one instance of the corresponding amino acid are plotted.

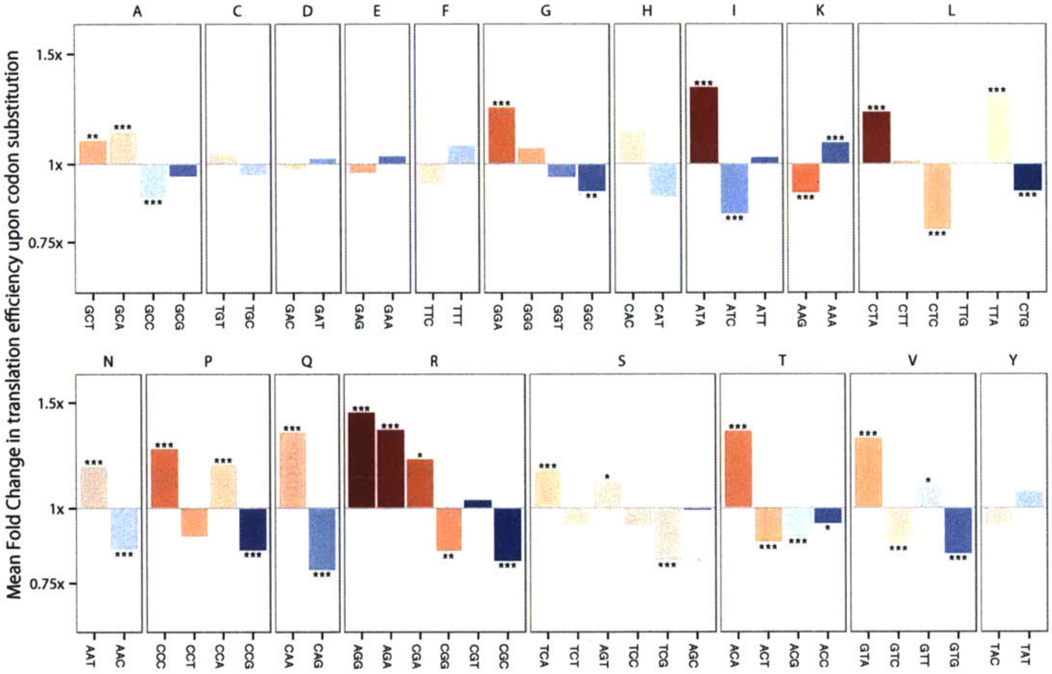


Figure B.9: *Rare codons increase translation efficiency.* The average fold change in translation efficiency (sfGFP fluorescence divided by mRNA level) is correlated with the choice of codon for an amino acid. The y-axis is the slope of a linear model linking codon use to relative translation efficiency. Codons are sorted left to right by increasing genomic frequency, and colored according to their relative synonymous codon usage (RSCU) in E. coli. These data can be compared with Fig. 2A, which used protein levels instead of translation efficiency. All other analyses using translational efficiency instead of protein levels display similar correspondence.

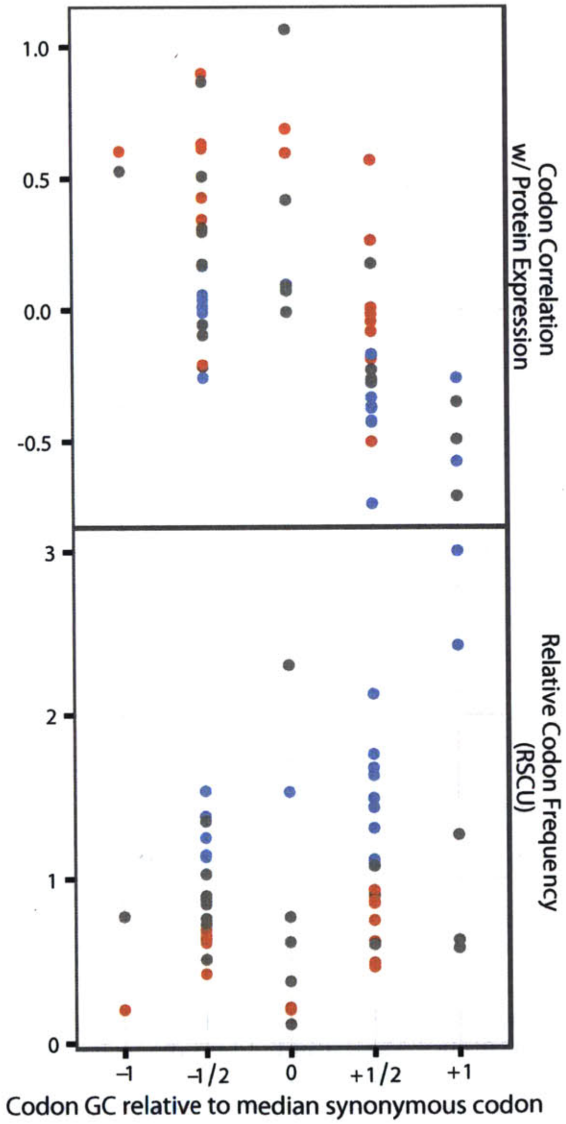


Figure B.10: *Comparison between rarity and GC content for synonymous codons.* The relationships between codon GC, genomic codon frequency, and codon correlation with expression in our dataset are shown. Codon GC is plotted on the x-axis relative to the median codon GC. For instance, glycine has four codons, GGN, and the median number of GC nucleotides is 2.5. GGA and GGT would each have a relative codon GC of -1/2, while GGG and GGC would each have a relative GC of +1/2. Isoleucine has three codons, AAH, and has a median of 2 GC nucleotides. AAC has a relative GC of +1, and AAT and AAA have relative GCs of 0. The top panel shows each codon's expression correlation (as in Figure 2A) compared with relative GC, while the bottom panel shows each codon's relative genomic frequency compared with relative GC. The most rare synonymous codons are in red, the most common are blue, and all other codons are gray.

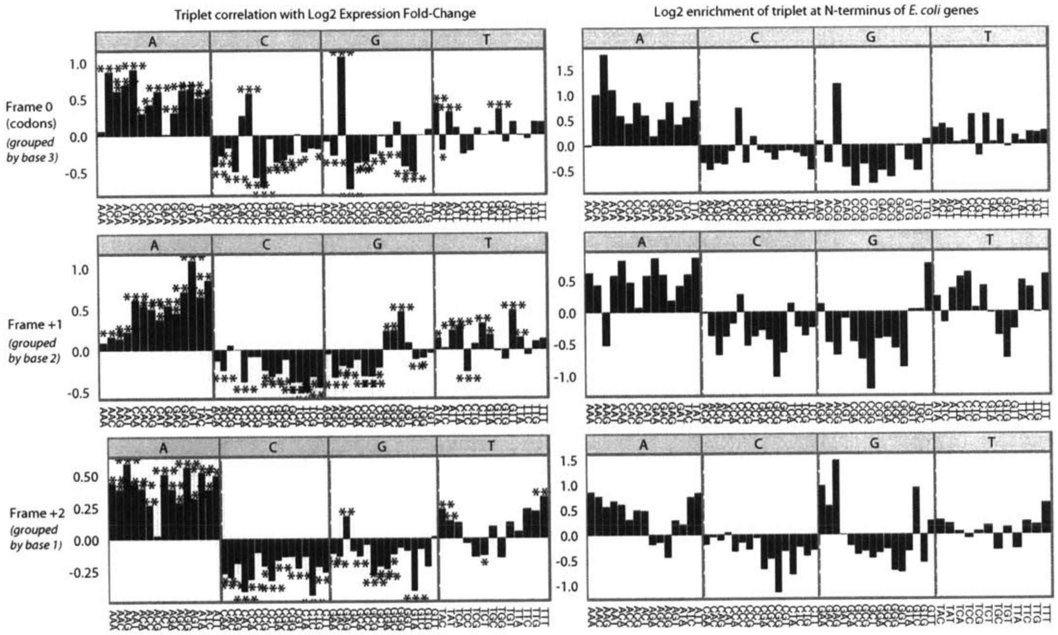


Figure B.11: *Triplet Enrichment.* A comparison of the effects of nucleotide triplets in all three frames on expression in our dataset (left) and those same triplets' enrichment at the N-terminus of all genes in E. coli (right). Triplets that are in frame (i.e. codons, frame 0) are in the top row, triplets in frame +1 are in the middle row, and in frame +2 are at the bottom. Triplets are grouped (panels with gray labels) by their wobble base in each position – base 3 in frame 0, base 2 in frame +1, and base 3 in frame +2. On the left, the y-axis represents the linear model slope as in Figure 2 and described in the text and supplemental information. On the right, the y-axis represents the log2-fold enrichment of triplets in the first 30 bases of all genes in E. coli. Triplets with A at the wobble base position in all three frames are generally correlated with increased expression in our data set, and are also enriched at the N-terminus of natural genes.

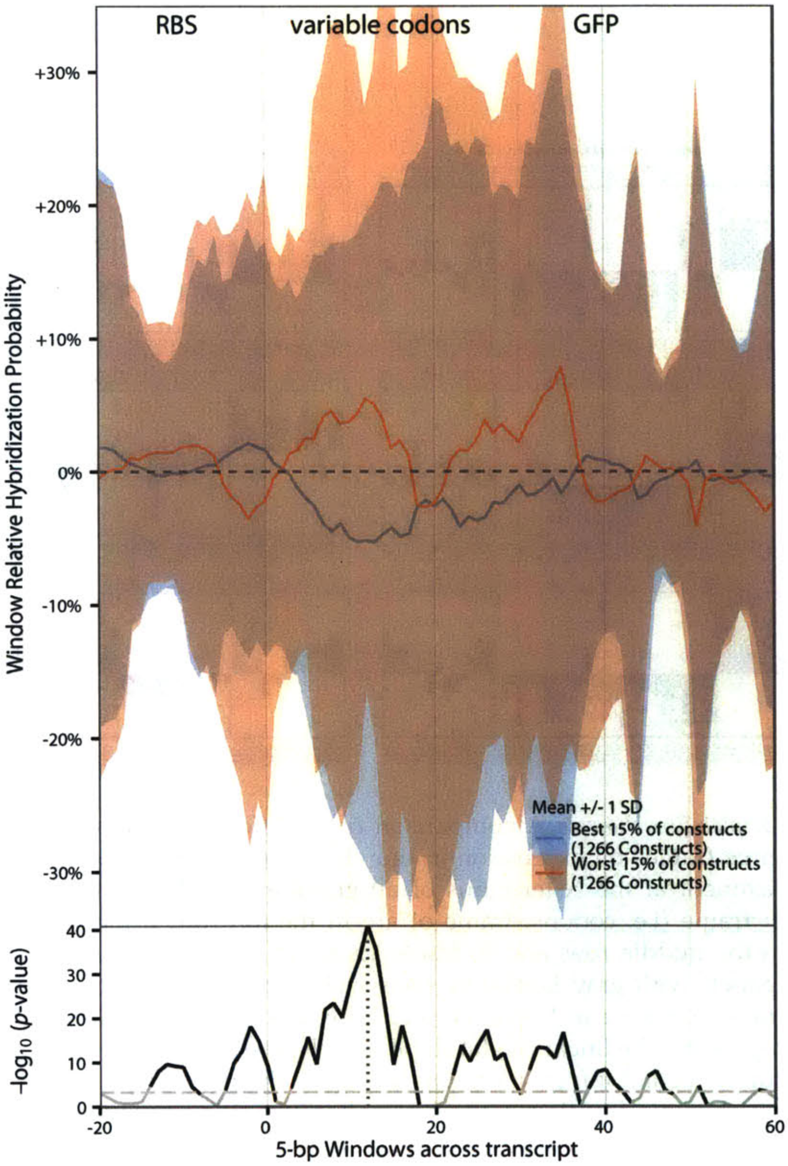


Figure B.12: *Relative Hybridization for 5 bp Windows.* The plots are the same as Figure 4, but with different window sizes, where the x-axis position is the window center. Across the x-axis are sliding window centers 5 bases in length. The window most correlated with expression change is centered at least 10 bases into the transcript, and not at the RBS or start codon.

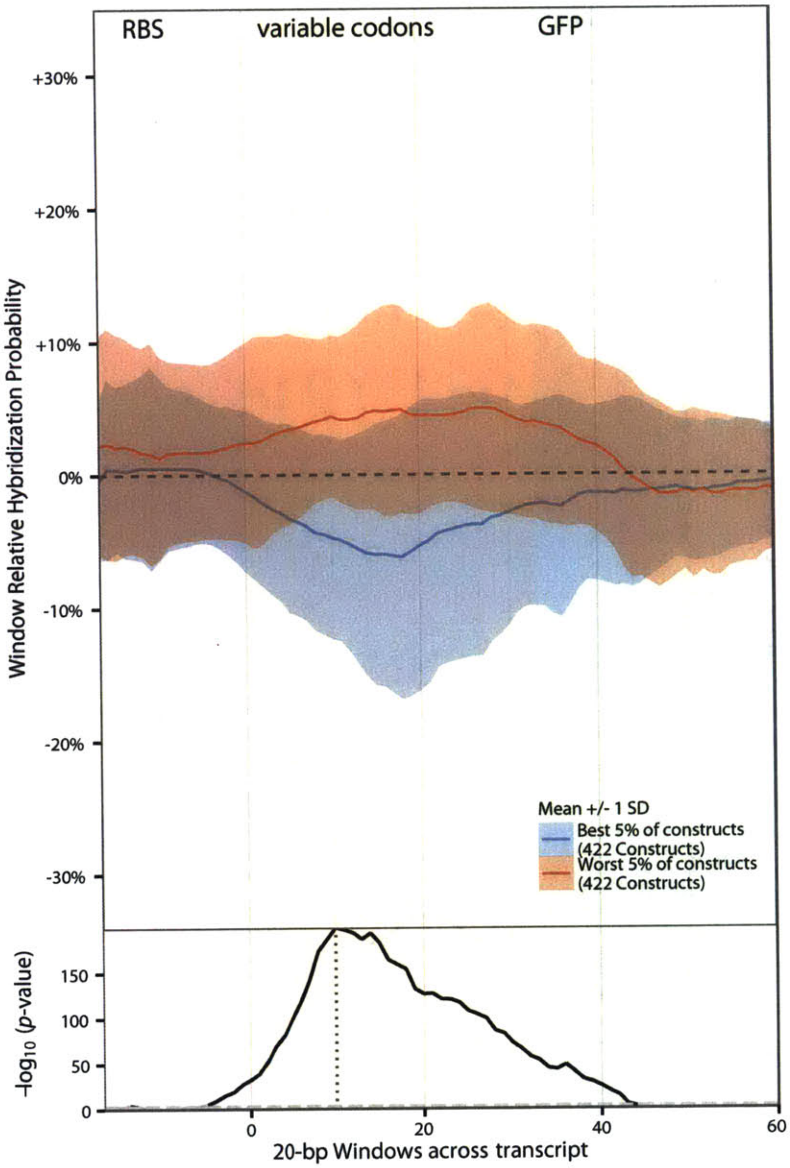


Figure B.13: *Relative Hybridization for 20 bp Windows.* The plots are the same as Figure 4, but with different window sizes, where the x-axis position is the window center. Across the x-axis are sliding window centers 20 bases in length. The window most correlated with expression change is centered at least 10 bases into the transcript, and not at the RBS or start codon.

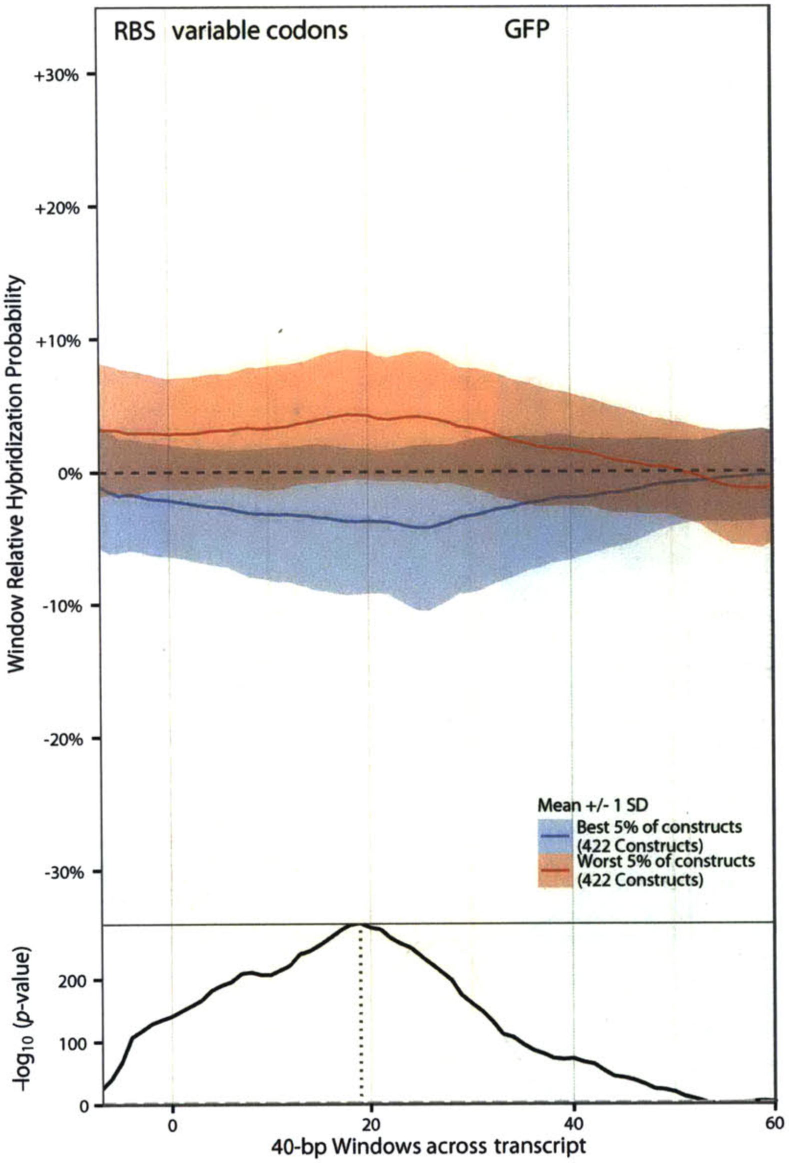


Figure B.14: *Relative Hybridization for 40 bp Windows.* The plots are the same as Figure 4, but with different window sizes, where the x-axis position is the window center. Across the x-axis are sliding window centers 40 bases in length. The window most correlated with expression change is centered at least 10 bases into the transcript, and not at the RBS or start codon.

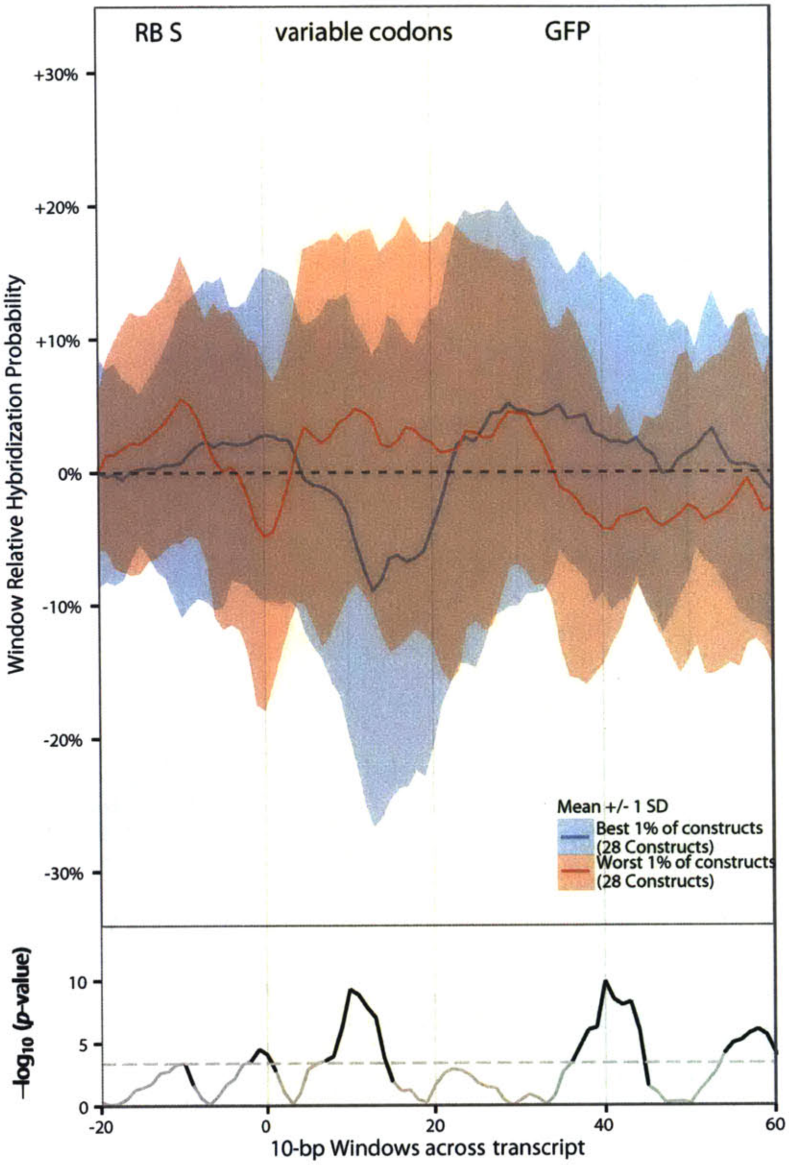


Figure B.15: *Relative Hybridization for Constructs with no ΔG Change in 10 bp Windows.* Similar to Figure 4 and Fig S10, this figure shows considers only constructs with no global ΔG change (i.e. within the green shaded region in Figure 3E). The relative hybridization probability is defined as the change in probability of bases within a 10 bp window being paired with any other base in the transcript. Across the x-axis are sliding window centers bases in length. In the top panel, the best and worst 1% (28 measurements) of constructs ranked by relative expression within a codon variant set are grouped and plotted as blue and red ribbons, respectively. The ribbon tops and bottoms are one standard deviation out from the mean, which is shown as a solid line. The bottom panel shows the p-value for a linear model correlating hybridization probability within that window to expression fold change for all constructs with no global ΔG change.

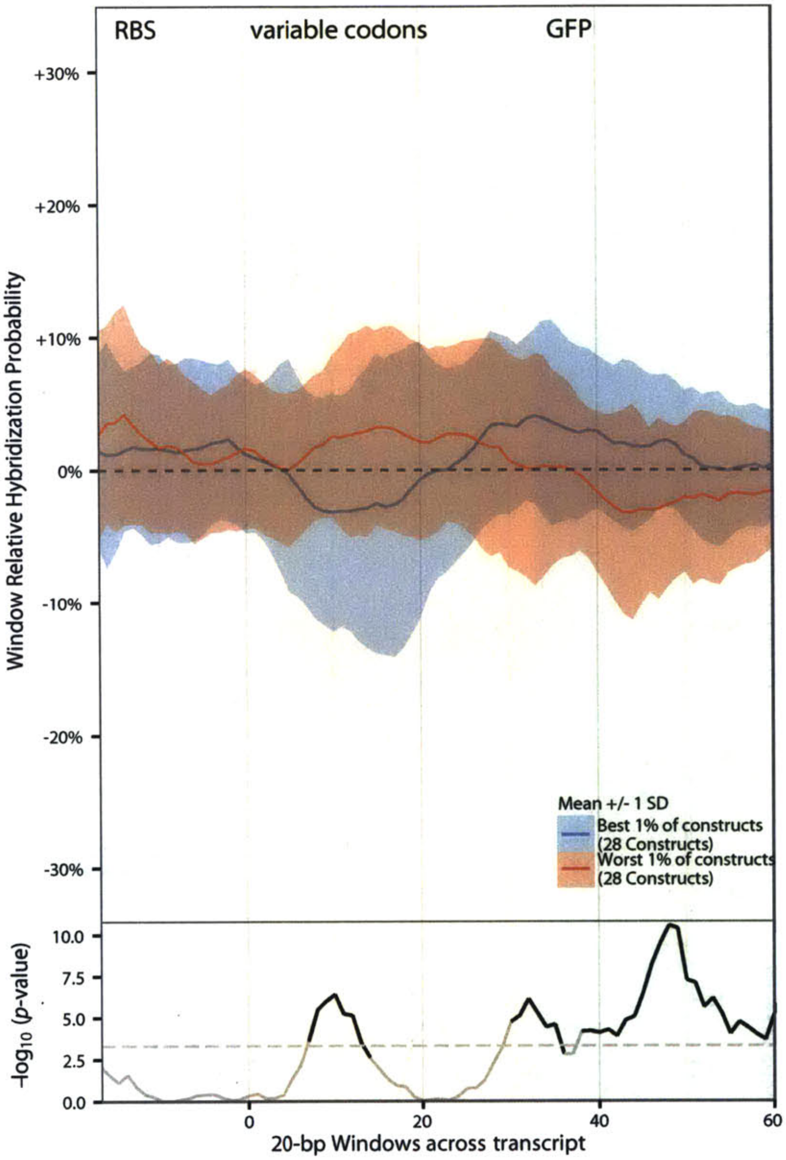


Figure B.16: *Relative Hybridization for Constructs with no ΔG Change in 20 bp Windows.* Similar to Figure 4 and Fig S10, this figure shows considers only constructs with no global ΔG change (i.e. within the green shaded region in Figure 3E). The relative hybridization probability is defined as the change in probability of bases within a 20 bp window being paired with any other base in the transcript. Across the x-axis are sliding window centers bases in length. In the top panel, the best and worst 1% (28 measurements) of constructs ranked by relative expression within a codon variant set are grouped and plotted as blue and red ribbons, respectively. The ribbon tops and bottoms are one standard deviation out from the mean, which is shown as a solid line. The bottom panel shows the p-value for a linear model correlating hybridization probability within that window to expression fold change for all constructs with no global ΔG change.

C

SUPPLEMENTAL INFORMATION FOR CHAPTER 4

This appendix contains the extended supplemental information and additional figures for Chapter 4.

C.1 SUPPLEMENTAL METHODS

C.1.1 *Generation and Scoring of Promoters and DNA Regulatory Elements with Promuter*

Promuter uses strength and spacing of multiple position weight matrices to identify and score promoters and transcription factor binding sites. We use data from our previous high-throughput studies[43] to train a model of transcriptional rate based on sequence alone. To create new genetic elements, we then build a mutational landscape in which certain mutations are favored and others are disfavored, and iteratively search that landscape for combinations of mutations that satisfy constraints, such as the creation or removal of regulatory elements, or the maintenance of the current transcriptional or translational rates. We used Promuter to generate a wide range of circuit designs by modifying number, location, and strength of binding sites.

Promoter begins with a starting sequence which is annotated with transcription factor binding motifs and core promoter motifs from Promuter's database (FigureC.1, step 1). Once the sequence is annotated, sequence constraints are applied. In this example, the constraints include the removal of a tetR binding site upstream of the -35, the addition of a lacI binding site in the central region of the core promoter, and to maintain the motif scores of the -35 and -10 binding sites within some acceptable range, so as not to ablate basal transcription. Next, the costraints are combined into a mutational landscape (FigureC.1, step 2). Red grid squares are bases that are unfavored in the landscape - they either represent currently chosen bases that are essential to the tetR motif, mutations that would be deleterious to the -35 promoter, or bases that currently unfavored in the lacI binding site motif. (3) Promuter samples combinations of mutations from this landscape, starting with only a few, using a prioity queue of scored mutations combinations. It applies each combination of mutations and re-scores the new sequence as in step 1 and continues to retry new mutation sets of increasing size from the queue until the constraints are met and an acceptable new sequence is generated.

Inducer	aTC (tetR)	IPTG (lacI)	Acrylate (acuR)
Conc. 1	20 nM	1 µM	40 µM
Conc. 2	150 nM	10 µM	200 µM
Conc. 3	300 nM	100 µM	1000 µM
Conc. 4	450 nM	1 M	5000 µM

Table C.1: Inducer concentrations for each of the small-molecule inducers used in this study. Each inducer concentration (1-4) corresponds to a separate FlowSeq sorting experiment.

C.2 SUPPLEMENTAL FIGURES

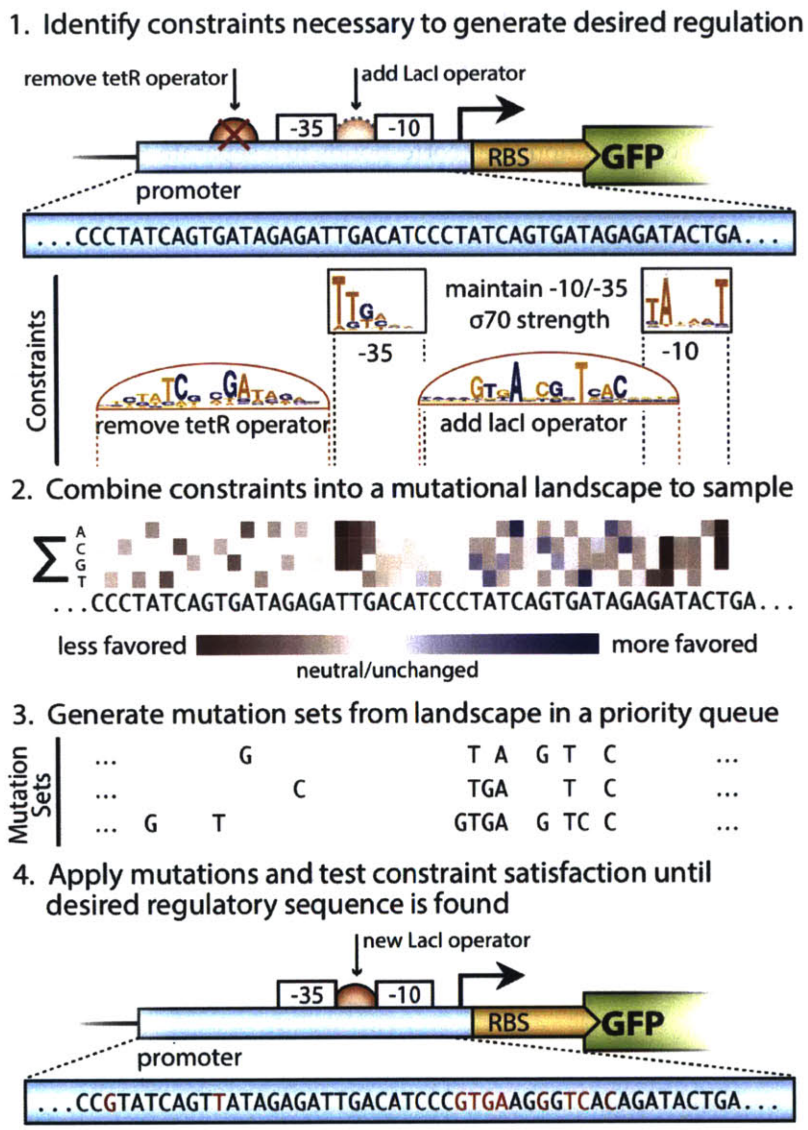

1. Identify constraints necessary to generate desired regulation
remove tetR operator
add LacI operator
-35
-10
RBS
GFP
promoter
...CCCTATCAGTGATAGAGATTGACATCCCTATCAGTGATAGAGATACTGA...
Constraints
maintain -10/-35
σ70 strength
-35
-10
remove tetR operator
add lacI operator
2. Combine constraints into a mutational landscape to sample
Σ
A
C
G
T
...CCCTATCAGTGATAGAGATTGACATCCCTATCAGTGATAGAGATACTGA...
less favored
neutral/unchanged
more favored
3. Generate mutation sets from landscape in a priority queue
Mutation Sets
... G T A G T C ...
... C TGA T C ...
... G T GTGA G TC C ...
4. Apply mutations and test constraint satisfaction until desired regulatory sequence is found
new LacI operator
-35
-10
RBS
GFP
promoter
...CCGTATCAGTTATAGAGATTGACATCCCGTGAAGGGTCACAGATACTGA...

Figure C.1: *Design of Genetic Regulatory Elements with Promuter.* See description in the text (Section C.1.1).

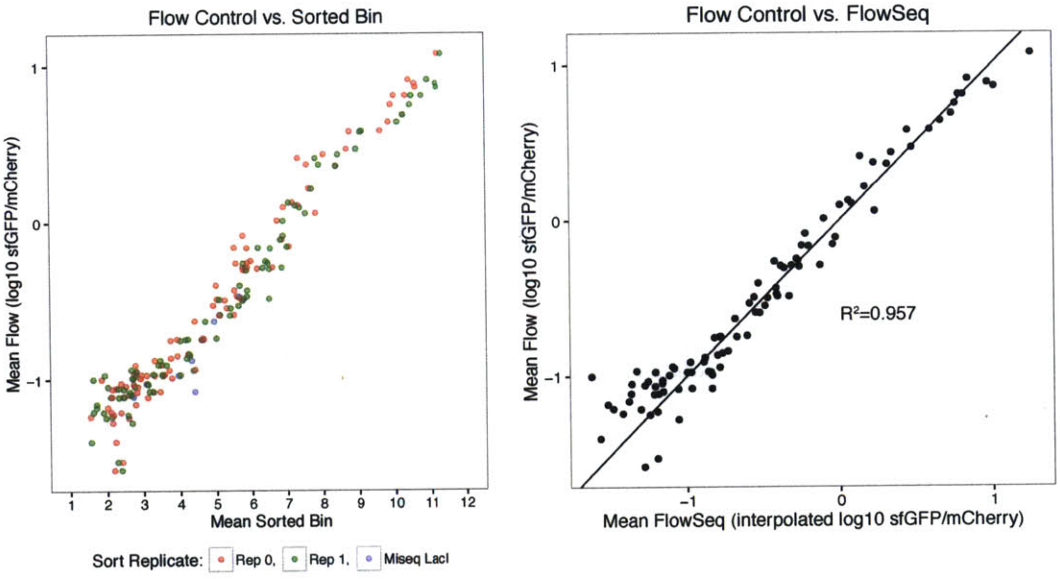


Figure C.2: Individual Flow Cytometry Controls. Individual constructs chosen at random from the library were measured independently using Flow Cytometry and were then compared to the mean flow cytometry bin measured by FlowSeq (left). From these measurements, the maximum and minimum of each of the 12 bins was calculated using expectation-maximization over Dirichlet distribution and an estimated measurement was interpolated (right).

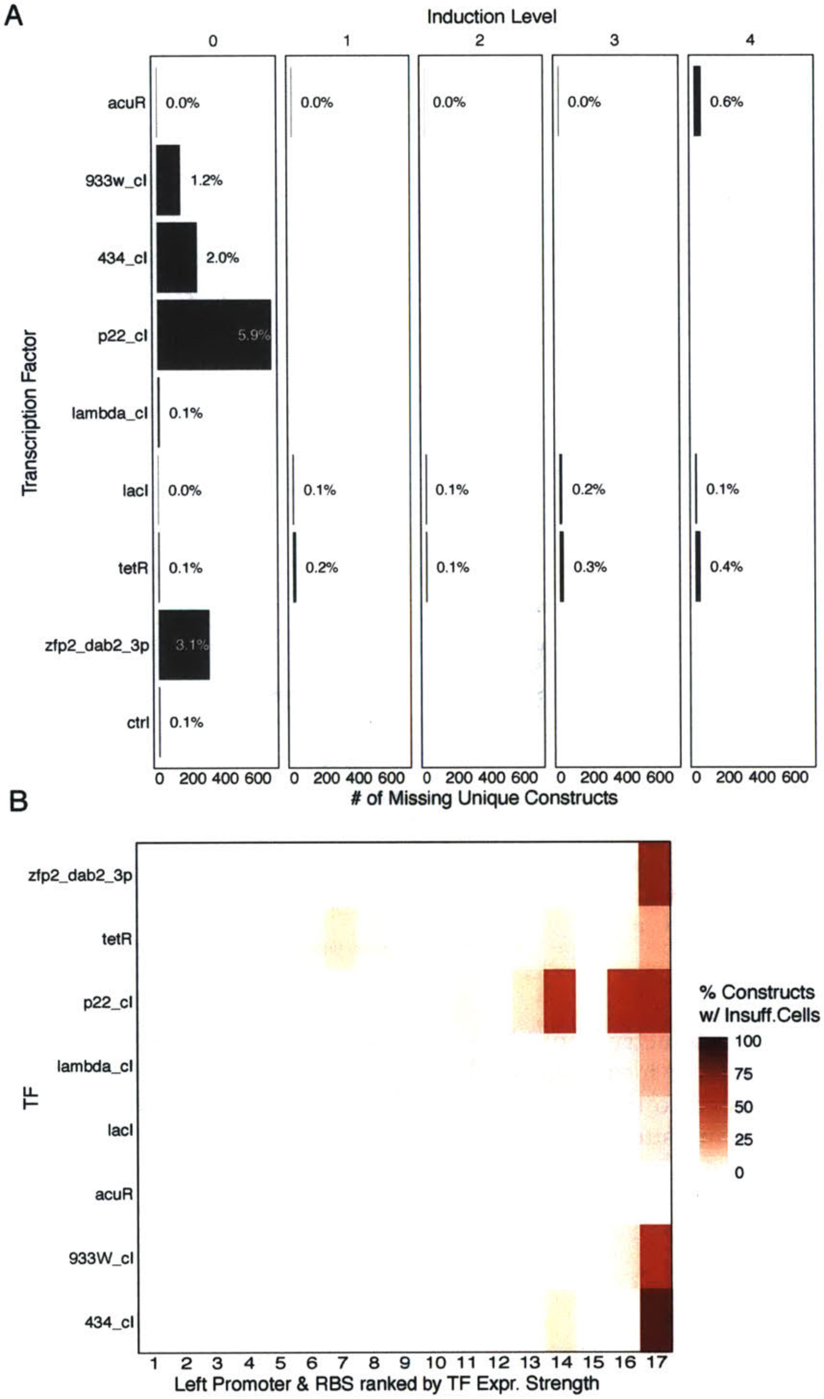


Figure C.3: *Library Sequencing Coverage and Fitness.* (A) An overview of which sequences were missing from library measurements across different TFs and induction levels, suggesting toxicity effects of TF expression and inducer concentrations. (B) A comparison of TF expression and number of sorted cells (approximated by read count). Some TFs were toxic at high expression and were not seen in the library.

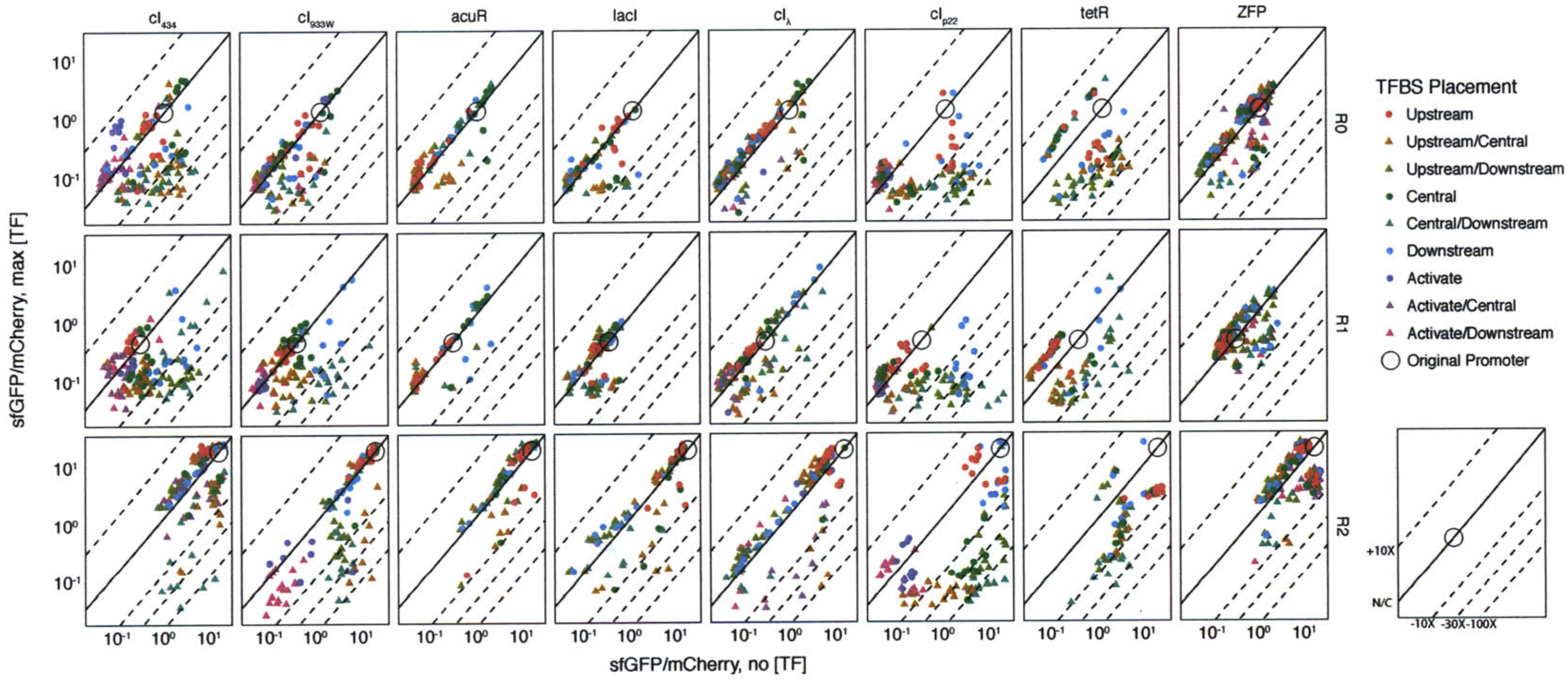


Figure C.4: Dynamic Ranges of Repression and Activation for 4,221 P_{RX} variants. Each point represents one promoter variant based on the P_{R0-R2}promoter set, generated using Promuter. The shape of each point corresponds to a single or double motif, and the color corresponds to the location(s) of the motif(s). The X axis measures sfGFP in the absence of TF expression, and the Y axis measures sfGFP under maximal TF expression. Points along the diagonal are unaffected by TF expression, while points below are repress and points above are activated. An open circle corresponds to the unmodified promoter. Dotted diagonal lines correspond to 10x activation, and 10, 30, and 100x repression.

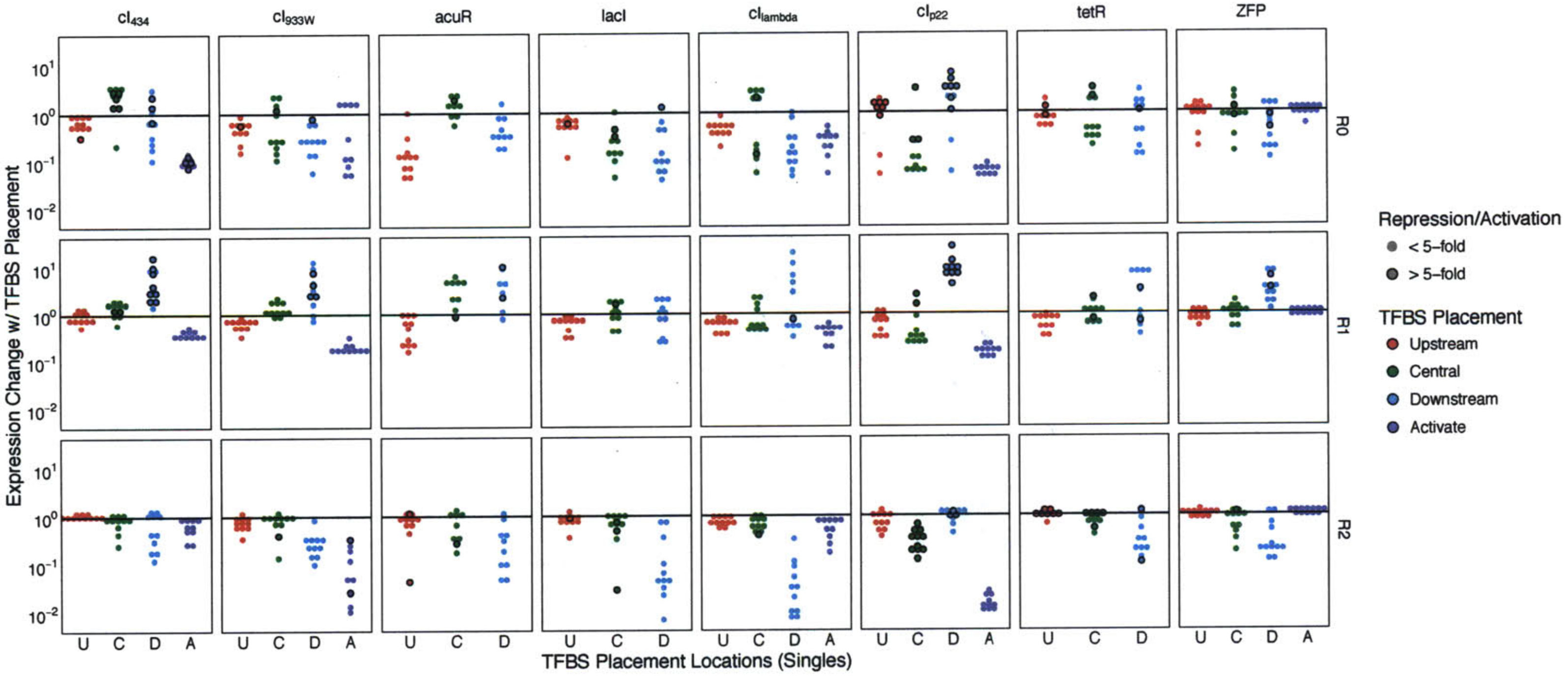


Figure C.5: *Change in Expression with TF Binding Site Placement.* Placing individual TF binding sites has differing effects on each promoter's basal expression. Change in basal expression for each promoter (P_{R0-R2}) is plotted on the Y axis for each TF binding site motif and categorial location, either upstream of the -35, centrally positioned between the -35 and -10, or downstream of the -10. Some TF binding sites have no effect on basal expression, others strongly modify the promoter's ability to initiate transcription. These effects are more consistent within promoters than within TFs.

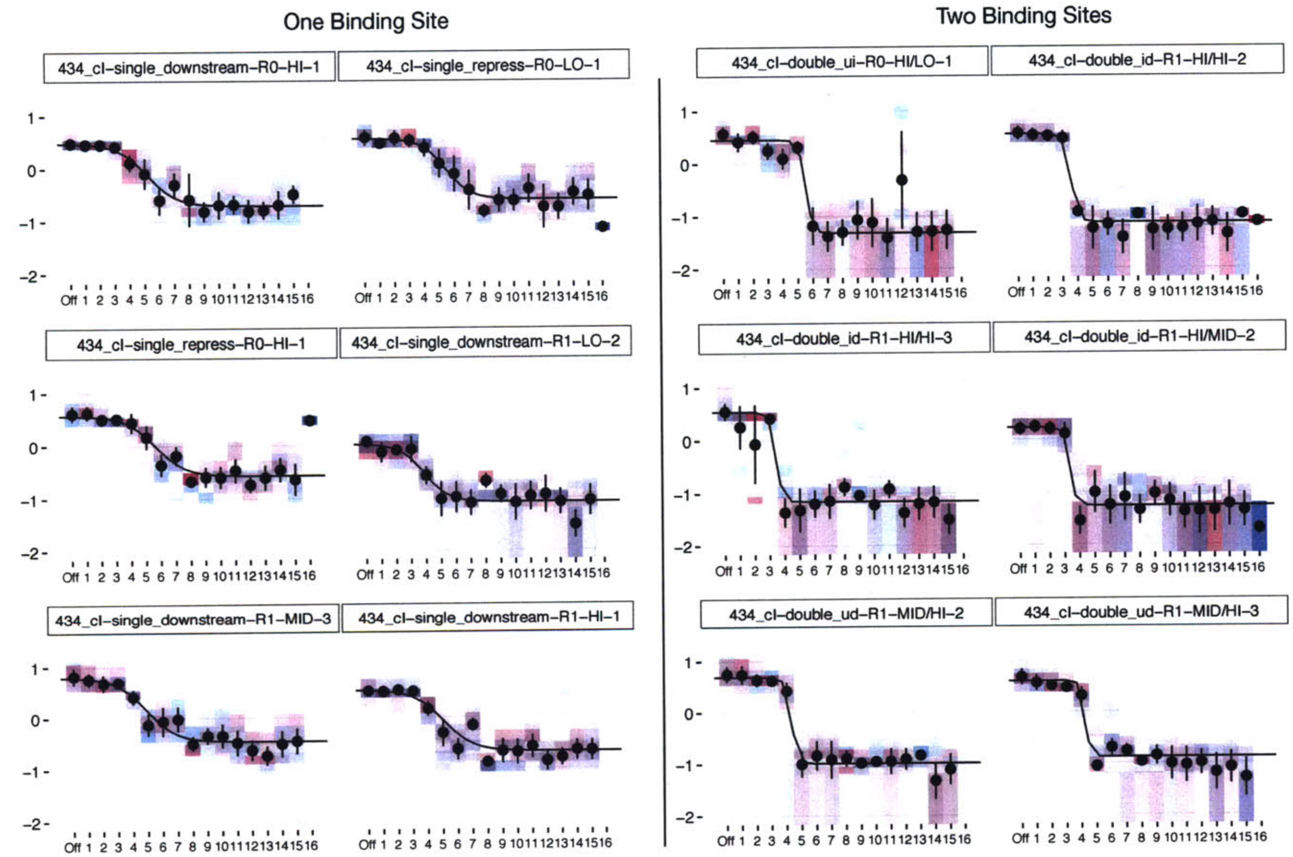


Figure C.6: *Multiple TF binding sites confer increased TF sensitivity and repression cooperativity.* (Left) Promoters with a single binding site that are repressed >10-fold by the cI_{434} transcription factor. They repress transcription slowly over 4-5 successive TF expression levels. (Right) Promoters with combinations of binding sites from the the set of promoters on the left. They respond to increased TF expression much sooner, and rapidly turn off transcription from the promoter.

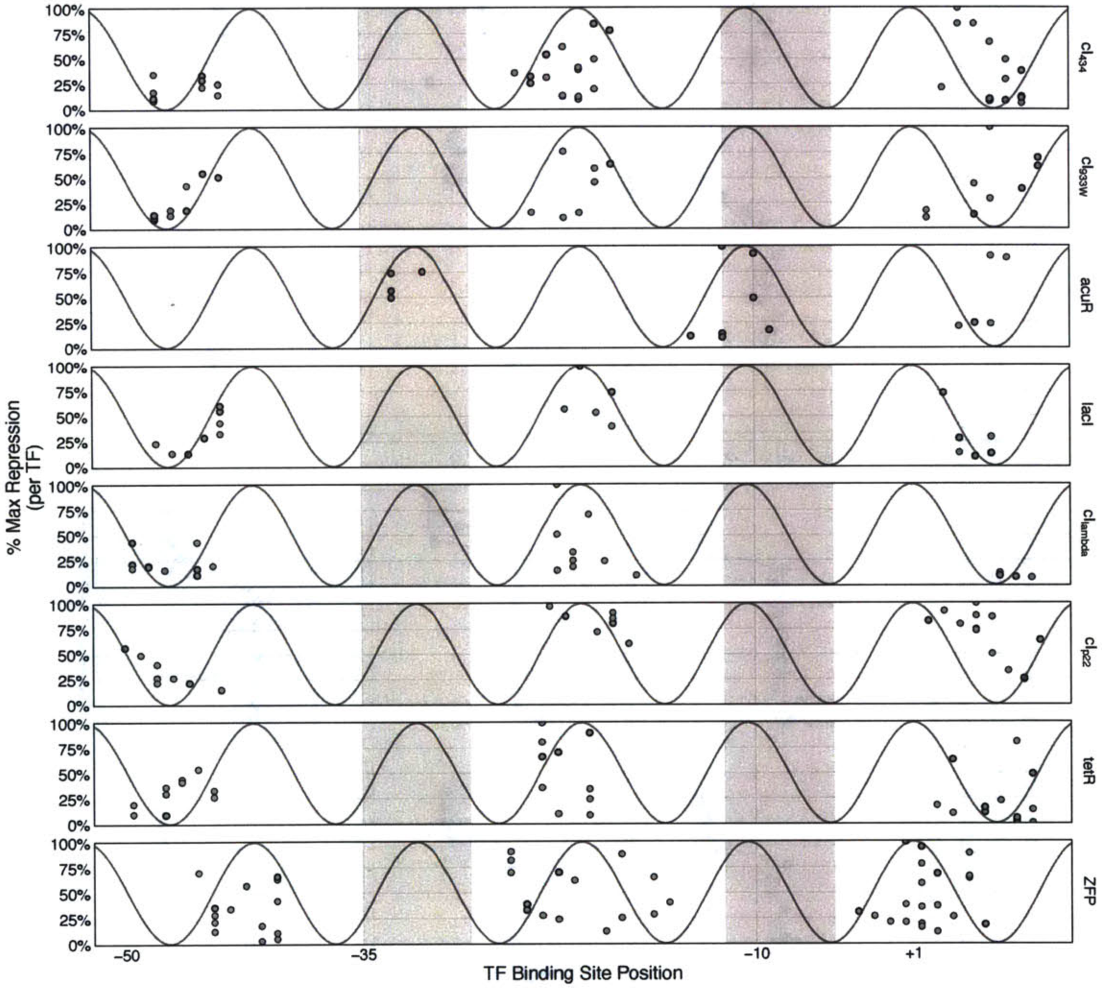


Figure C.7: *TF binding site position correlation with helical position.* Center of TF binding site versus normalized repression, as a percentage of the maximal fold-repression observed for that TF. Only promoters that have the potential to be repressed by at least 10-fold through our dynamic range are plotted.

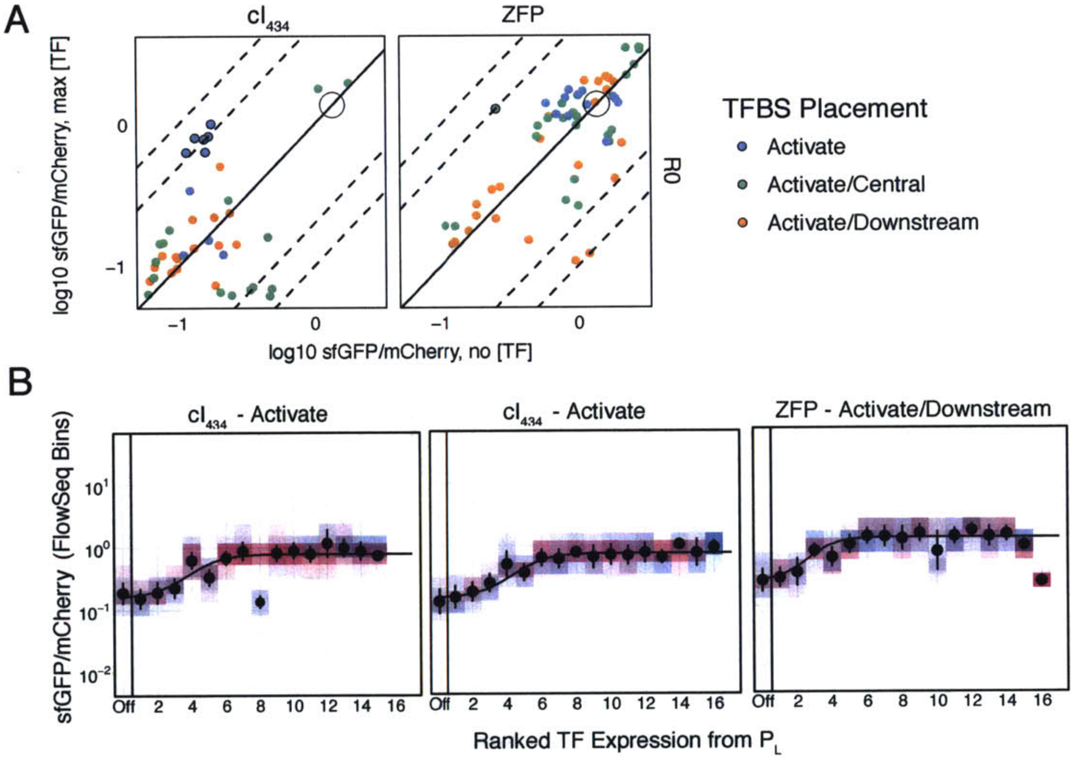


Figure C.8: *Activation from $cI_{434}$and the ZFP-ω fusion TFs.* (A) Fully repressed and fully expressed expression comparison (a subset of points from Figure C.4) for constructs that include TFBSs placed as activators. For the *cI* repressors a single activating position (overlapping the -35) is thought to allow for recruitment of the polymerase. For the ZFP fused to theω subunit of the RNA polymerase, the activating position is farther upstream of the -35 site. Dotted lines correspond to 5x and 10x activation (above the solid line) and repression (below the solid line). (B) Examples of individual Flowseq response curves for 3 constructs highlighted in black from (A).

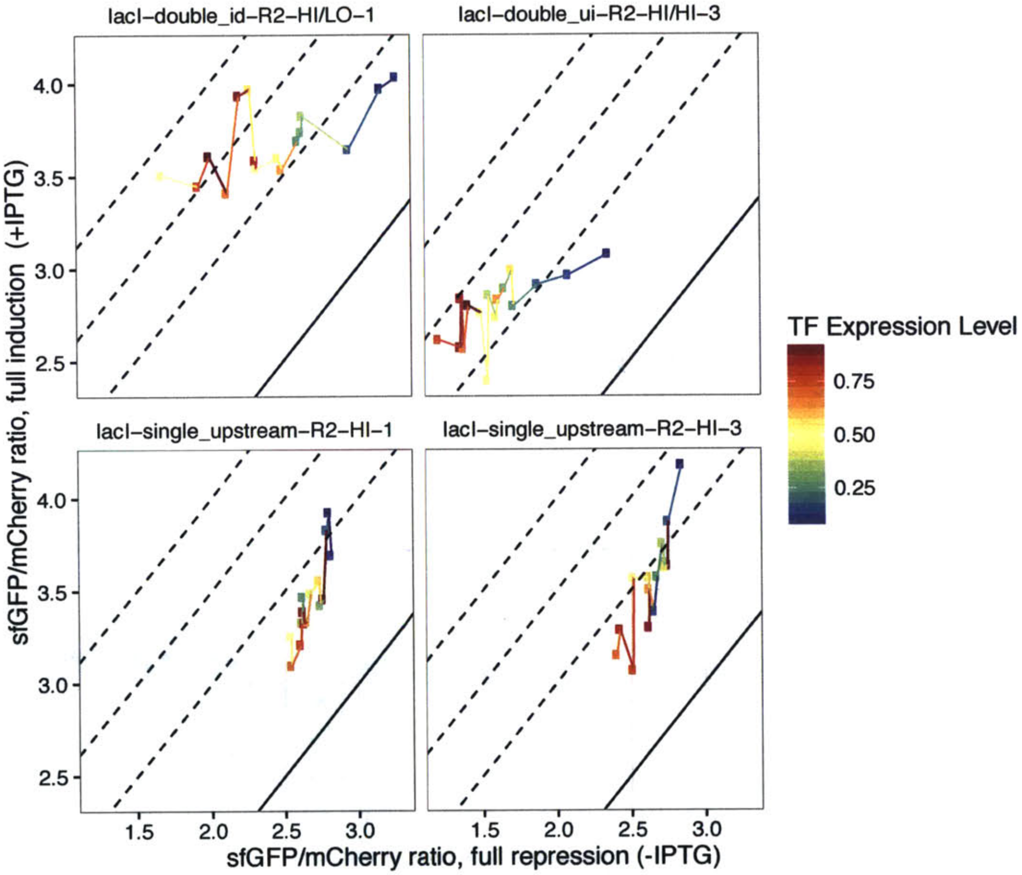


Figure C.9: *Some TF binding sites show reduced induction capacity with increased TF concentration.* In these figures, each panel consists of a set of divergent P_L/P_Rcircuits with the same $P_R$promoter and TF binding site. All 4 are sensitive to lacI and IPTG. (Top Left & Top Right) In the top two panels, two TF binding sites located centrally and downstream (top left) or upstream and downstream (top right), fully induced expression (+IPTG) remains constant regardless of the expression level of lacI, but the 'floor' of repression drops as more TF is expressed. (Bottom Left & Bottom Right) In the bottom two panels, single TF binding sites located upstream of the core promoter repress expression 10-20 fold in the absence of IPTG and at the lowest expression level of lacI. However, as the concentration of LacI is increased, the expression of sfGFP under full IPTG induction drops, suggesting that lacI is still binding to the promoter in the presence of IPTG.

D SUPPLEMENTAL INFORMATION FOR CHAPTER 5

This appendix contains the extended supplemental information and additional figures for Chapter 5. It is adapted from the supplemental information for a manuscript which is in preparation.

D.1 TOOL COMPARISON

While other packages exist to solve the integration and automation of whole genome resequencing and annotation, most of these tools are built for large diploid genomes, usually Homo sapiens. Some tools, like Galaxy, allow users to create their own custom pipelines without bash scripting, and do support the creation of pipelines for microbial genomes. However, Galaxy requires that the user to understand and optimize settings for each individual tool. Galaxy also does not allow visualization or interactive querying of the output, and cannot generate new reference genomes or use the output of one round of sequencing to inform the next round. Finally, because of Galaxy's one-size-fits-all nature, optimizing pipeline performance (via inline compression and piping of input and output streams) is not possible.

Another recent tool, SPANDx, can also perform genome resequencing for multiple strains simultaneously, but its widespread use is limited to the fact that it can only be run on UNIX computing clusters running the venerable and closed-source commercial PBS job scheduling system. SPANDx has no user interface or interactive components, and so users are required to gather the data manually and run the pipeline using a command line interface. Because we could not readily locate a PBS system to test the pipeline on, we were not able to compare the output between SPANDx and Millstone.

Breseq is purpose-built to perform haploid genome resequencing, but also has some shortcomings. Most notably, it only works on single genomic samples, making it challenging to compare alleles between samples. It also does not make use of paired-end sequencing reads, limiting its usefulness for detecting structural variation, and it cannot identify or place de-novo contigs or generate new reference genomes. And while BreSeq can be made to function on remote servers like Amazon Web Services or university compute clusters, this requires the user to be proficient in UNIX and capable of installing various dependencies. Millstone also automates the process of copying data to the remove server via its web interface.

Feature	Millstone	BreSeq	SPANDx	Galaxy
Variant Visualization	X	X		
Multiple Sample Comparison	X		X	
Optimized for Large Datasets	X		X	X
Interactive Querying	X			
Structural Variant Detection	X	X		X
Genome Versioning	X			
Effect Prediction	X	X	X	X
Easy Deployment / Install	X	X	X	X
Scalability	X		X	X
Data Sharing via Web	X	X		X
Genome Editing	X			
Optimized for Microbial NGS	X	X	X	
Free and Open Source	X	X		X
Sharing / Collaboration	X	X		

Table D.1: Comparison of features between Millstone and other microbial genome sequencing tools.

D.2 DATA MODEL

The data model at the core of the Millstone software was designed in response to requirements including project organization, data storage, and user activities such as uploading data, running analysis pipelines, exploring the resulting data, and generating actionable outputs. Figure D.1 presents the major models and how they are related to each other. Below, we focus on a few models at a time to explain the role of each model and which requirements they were created in response to. For clarity, we have omitted some models here. The full data model can be studied from the declaration in the source code: `https://github.com/churchlab/millstone/blob/master/genome_designer/main/models.py`.

D.2.1 *Basic Scaffold Models*

The parent model for all related data is a Project. A Project can have multiple ReferenceGenomes and ExperimentSamples. Variants are described relative to a single ReferenceGenome, in particular in the fields of position and ref_value.

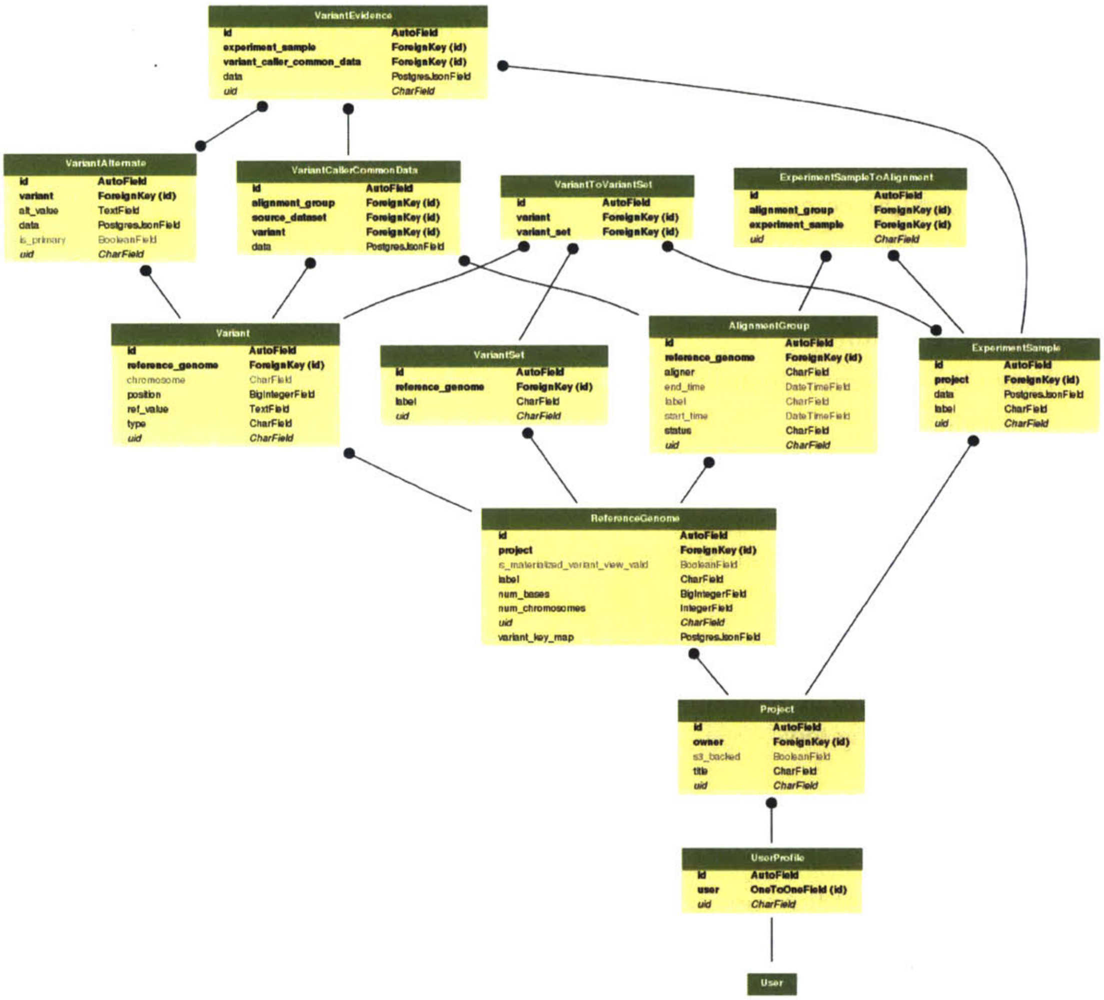


Figure D.1: Full data model.

D.2.2 *Alignment and Variant-calling*

Millstone allows a user to run an alignment of an ExperimentSample [fastq] data against a specific ReferenceGenome [genbank or fasta]. Then, Variants are called, describing the differences between the ExperimentSample and ReferenceGenome. The AlignmentGroup model stores the data from an alignment. Since a Variant can be called in multiple alignments, or uploaded by a user as a designed variant, we include an intermediate data model VariantCallerCommonData which describes which AlignmentGroups a Variant was called in and stores metadata provided by the variant calling tool (e.g. Freebayes).

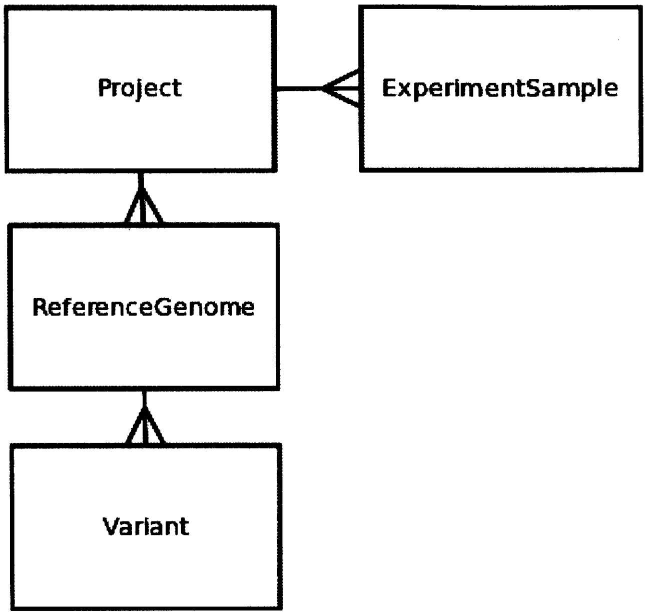


Figure D.2: Basic scaffold models.

D.2.3 *Variant Data / per-sample Variant Data*

Variants are effectively primitives in Millstone. A Variant, in combination with a VariantAlternate, describes a specific genomic event. A Variant can be uploaded by a user, or be called during alignment. The latter results in additional metadata provided by different variant callers (multiple variant callers can call the same Variant). This variant caller-specific metadata is stored in the VariantCallerCommonData model. Additionally, since we use variant callers on multiple samples at the same time, we store the data about the presence/absence and additional data describing confidence of the Variant's occurrence in a particular ExperimentSample in a model called VariantEvidence.

D.2.4 *VariantSet*

VariantSets allow the user to organize Variants into groups and take actions on groups of variants. A Variant can belong to more than one VariantSet. The VariantSet concept is very similar to tags in other software contexts. The user can take actions with these sets, including search by set, export set, print MAGE oligos, etc.

The VariantToVariantSet model represents each association between a Variant and a VariantSet. Additionally, a VariantToVariantSet has a many-to-many relation to ExperimentSample, which enables representing information like *Variant x is in VariantSet y for ExperimentSamples a, b, c.*

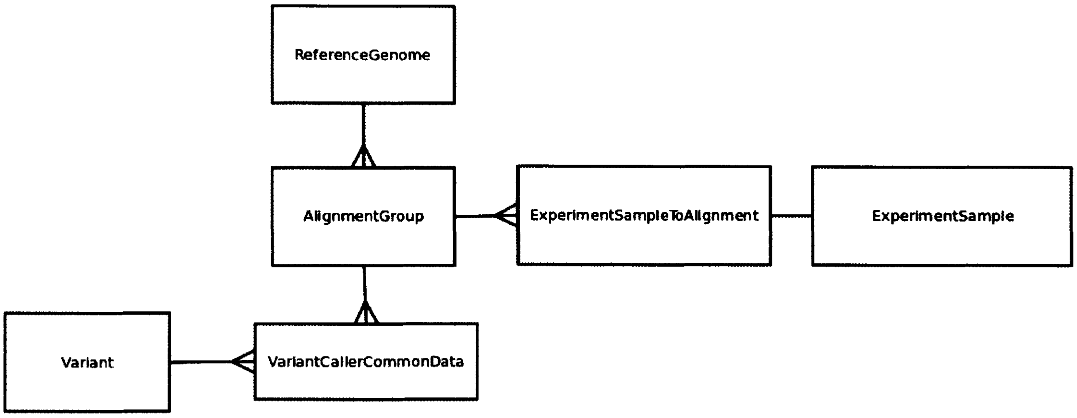


Figure D.3: An AlignmentGroup connects scaffold models to called variants.

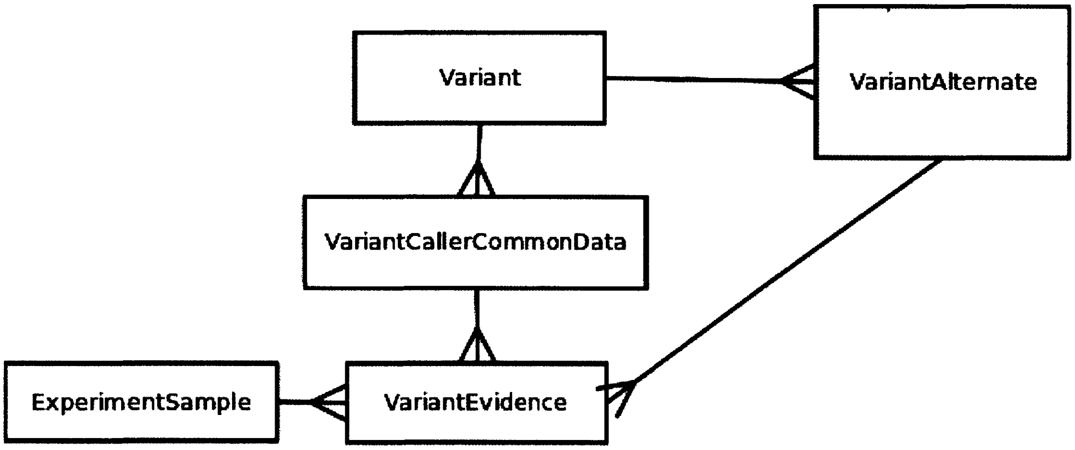


Figure D.4: A Variant describes a specific genomic event.

D.3 ANALYSIS PIPELINE

Millstone's analysis pipeline can be broken into initial read alignment and alignment processing steps, followed by single nucleotide variant calling, structural variant calling, and annotation of variants. In addition to annotating variants, Millstone also annotates genomic regions with poor mapping quality, non-unique alignments, and that have low or no coverage.

Initial read alignment is performed by BWA [53]. We use the package's BWA-MEM algorithm to place reads from a paired-end or single-end FASTQ file onto a provided reference genome, which can either be in Genbank or FASTA format. The SAMTOOLS package is used to sort and index the BAM alignment and remove PCR duplicates (via rmdup). The MD flag in the BAM file is also filled in by SAMTOOLS to allow for mutation visualization via Jbrowse. Multithreading at the alignment and BAM processing steps is accomplished at a per-sample level.

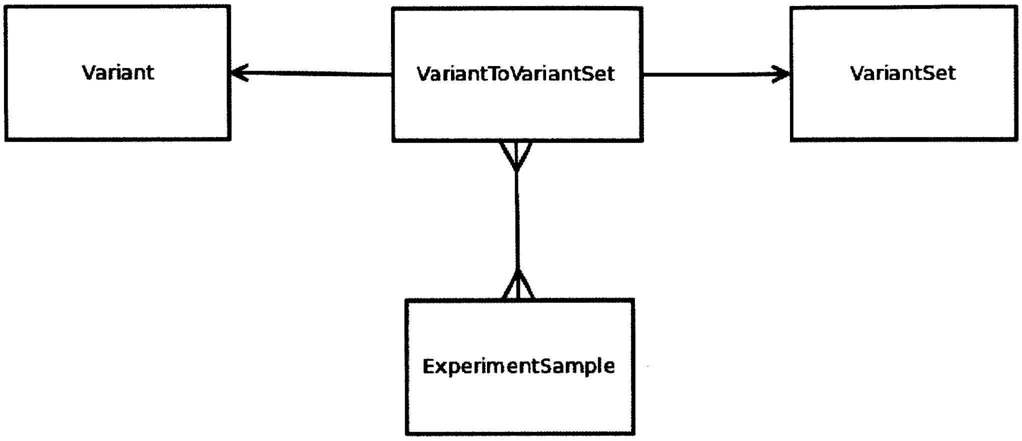


Figure D.5: A VariantSet allows grouping Variants.

Read group flags added by BWA allow the sample BAMs to be subsequently merged for SNV calling.

For single nucleotide variant calling, we use Freebayes. In addition to being completely free and open source, unlike the GATK toolkit, Freebayes performs indel realignment internally, avoiding a time-consuming indel realignment step usually required in GATK-based WGS pipelines. All sample BAMs are merged via sample read-groups and snps are separately called by region to allow multithreading.

For structural variant detection we use a combination of the Lumpy package and a custom-built de-novo contig assembly and placement pipeline (see Contig Assembly and Placement below). Lumpy structural variants are detected separately for individual samples, which allows for multithreading. After all samples are complete the single-sample VCFs are subsequently merged.

Variant annotation is performed by SnpEff. We generate a custom SnpEff genome config file from a user-provided Genbank file, and run the merged VCF files (one for SNVs across all samples, and another for SVs across all samples) through the SnpEff software. The SnpEff effect prediction string is then parsed with regular expressions to populate additional variant annotation INFO fields in the VCF and in the variant database.

Custom software walks along each individual aligned BAM sample and identifies regions that might be challenging to call variants within because they are low coverage or non-unique. For this purpose we use the pysam SAMTOOLS API. For low-quality mapping regions, we walk along each chromosome and flag contiguous regions where more than 50% of the reads have less than a MAPQ score of 20. If less than 4 reads cover a genomic position, we mark it as low coverage, and if 50% of reads map nonuniquely, then the region is marked as non-unique. In addition to generating a BED file for visualization in Jbrowse, flags are assigned to SNVs and SVs within these regions so that the user has the proper context in which to assess the validity of the variant.

D.4 QUERY LANGUAGE

A powerful, unique feature of Millstone is being able to filter and compare called variants across all aligned samples. When analyzing engineered and evolved genomes, questions of interest might include:

What are commonly mutated genes?

What fraction of my genomes successfully received target mutations?

What are the most common occurring SNVs?

Researchers can use Millstone's *Analyze* view to filter through variant call data and identify answers to these questions. The *Analyze* view provides an input box that accepts queries in an intuitive syntax. Variants that match the filter are displayed in a table, along with additional metadata requested by the user. The simplified syntax allows users to specify boolean combinations of `key op value` statements which are then converted on the backend into the correct SQL query. The syntax is simpler than SQL and does not require knowledge of the underlying data table structure.

The most basic query comes in the form:
`key op value` (e.g. `POSITION = 10000`)
Multiple conditions can be combined using logical operators:
`cond1 logical_op cond2` (e.g. `POSITION = 10000 & GT_TYPE = 2`)
Additional example queries can be found by clicking on the arrow dropdown in the filter box. All filter keys can be found by clicking on the *Fields* button in the *Analyze* view. When uploading samples, users can provide custom keys that will be recognized by the analysis engine. See the section on benchmarking search functionality for a discussion of performance.

D.4.1 *Implementation*

There are two major aspects of the query language implementation: First is generating a materialized view representation of the underlying data offline, which obviates the need to perform expensive JOINs. The second is parsing a user query and converting into the equivalent SQL query to be executed against the materialized view.

The underlying data is stored in a normalized representation across multiple tables in a PostgreSQL database as described in the D.2 section of the supplement. Executing the a typical SQL query to filter across Variants requires performing JOINs across multiple tables. This becomes unwieldy with increasing dataset size. For example, running Millstone on whole genome sequencing data from 68 strains derived during the construction the genomically recoded organism [49] results in 5053 unique SNPs. While this is $68 * 5053 = 343604$ combinations of Variant and

ExperimentSample, the SQL query must typically JOIN across 10 different tables in order to handle all parts of the filter. To mitigate this expensive JOIN requirement, we leverage the *materialized views* feature in PostgreSQL, which allows us to perform the expensive JOIN offline, that is, not in response to a user query. The resulting denormalized representation is then the target of converted user queries.

To parse a user query, we convert it into disjunctive normal form which can then be converted into the corresponding SQL query against the materialized view.

Once the results are fetched from the database, they are cleaned up and prepared for display by the frontend. Results are paginated so that only the number of results that can be displayed in a single view the user interface are actually queried and sent to the user interface and also giving a performance boost.

The user query is validated and then converted into disjunctive normal form (DNF), an AND of OR statements. This is done by converting the query into a symbolic representation and then using SymPy a python library for symbolic mathematics to convert the query to DNF. The DNF can then be converted to a SQL query.

D.4.2 *Example Queries and Benchmarking*

Configuration A: EC2 m4.2xlarge, 115 *E. coli* genomes, 4698 total events

---

Blank (get all results)
A: 3.63 sec

Get all SNPs that are homozygous alt (diploid calling mode).
`GT_TYPE = 2`
A: 3.62 sec

Get all SNPs with coverage at least 10 reads.
`DP >= 10`
A: 6.24 sec

Get SNPS that are called homozygous alt with at least read depth of 10
`GT_TYPE = 2 & DP >= 10`
A: 6.08 sec

Get SNPs in gene thrA
`INFO_EFF_GENE = thrA`
A: 3.69 sec

D.5 CONTIG ASSEMBLY AND GENOME FINISHING

D.5.1 *Overview*

Available tools for SNP and SV often do not identify complex events despite sufficient information for these events being available in individual next-generation sequencing reads. In particular, unmapped, clipped, and split reads may indicate the presence of complex events. The bigger picture of contig assembly is to fit into what we call genome finishing, or creating a new updated reference genome from an existing reference and mutation data.

De-novo contigs are assembled from reads identified by their alignment to the reference as being internal or proximal to structural variation. Each contig is then decomposed into the reads that contributed to its assembly, which are used to identify a left and right breakend in the reference. The reference sequences leading up to each of these two breakends is then located in the contig to identify the position of the reference-contig junction in the assembled contig. Once junctions between the contig and the reference are identified, the ends of the contig homologous to the reference are excised and the novel sequence is called as an insertion-type variant.

D.5.2 *Structural Variant Indicating Reads*

The identification of structural variation indicating reads resultant from an alignment is performed by selecting reads on the basis of belonging to one of five improper-alignment classes: unmapped, clipped, discordant, split, and piled. Unmapped reads are those not mapped to the reference in the alignment. Clipped reads are reads that are aligned up to a point in the reference after which the remaining bases in the read are unmapped. Because read quality tends to fall off towards the end of a read, most aligned reads end in a couple bases of clipping, and so clipped reads are only identified as useful if they fall above thresholds for the number of clipped bases and the average Phred score of the clipped bases. Discordant reads have paired-end mates which map to the outside the range of normal template lengths. Piled reads are groups of n reads that all stop mapping to the reference at the same point, where n is above the piling threshold. The piling threshold is set as three standard deviations above the mean read endpoint pileup. Split reads are reads in which segments of the read are mapped to different regions of the reference genome.

D.5.3 *Algorithm*

The structural variant indicating reads are aggregated into a single bam file with their mate-pairs added. This bam is then used as input to Velvet[106], a de novo assembly program that constructs contigs using De Bruijn graphs. The contigs are then evaluated for placement in the reference. First, the IDs of the reads that went into the assembly of each contig are extracted. These IDs then queried in the alignment to determine the number of left and right read endpoints piled up at each position in the reference. A breakend in the reference is called if the highest pileup is significantly greater than the second highest pileup. If a left and right breakend can be found, the mapped portion of the reads piled up at the reference breakends are aligned to the contig, and left and right contig-reference junctions are called at the positions of highest left and right read endpoint pileup in the contig. These contig-reference junctions represent the ends of regions of reference homology in the contig that flank the novel inserted sequence, and are removed. The remaining sequence is called as an insertion-type variant, potentially compounded with a deletion if the left and right breakends are not directly adjacent in the reference.

D.5.4 *Genome Finishing*

In whole genome assemblies, genome finishing typically refers to the process of figuring out the remaining portions of the genome that may be more difficult to sequence due to the context of the DNA. Issues such as GC-extremes and repeats may complicate this. In the context of engineering microbial genomes, genome finishing is the process of identifying the actual genome of a modified strain.

To implement genome finishing in Millstone, we combine results from SNV and SV callers with the results from contig assembly to yield a new genome that represents the best ground truth representation of the actual state of the genome of the microbe of interest.

D.6 SYSTEM ARCHITECTURE

The design and implementation of Millstone was driven by two main goals: The first was to enable execution of more sophisticated endeavors in iterative engineering of microbial genomes through technologies like MAGE. The second was to enable non-computational researchers and researchers outside of our lab to use the types of methods we have developed and use extensively in our own lab. Both of these goals required an integrated solution that automated boilerplate analysis steps like genome alignment and variant calling and facilitated exploration of results and visualization of evidence. Additionally, we wanted to make Millstone scale well with increasing project requirements, as well as facilitate collaboration. All of these needs pointed to implementing Millstone as a web wapplication. The backend could reside in the cloud, an in-house cluster, or a local machine. A rich interface could be provided through a web browser. Ultimately, Millstone could be transitioned to deploying via Amazon Web Services, which enables scaling up computational resources limited only by availability of funds allocated for computational spending, which can be significantly if not entirely covered by AWS research grant credits available to academic institutions.

D.6.1 *Software*

Millstone is built using a stack of open source software tools. The back end is primarily written in python and the front end is written in JavaScript and HTML. Scaffolding for the application is provided by Django, a popular python web framework with an extensive community of users and developers which remains under active development. The software is open source and available at `https://github.com/churchlab/millstone`

For data storage, we chose the PostgresSQL database management system, a popular, well-supported and actively developed relational database. In addition, PostgresSQL includes several non-relational features including support for JSON columns and materialized views. JSON is important for storing the large amounts of key-value data generated by different bioinformatic tools. Materialized views allow us to solve the problem of expensive queries across many tables by precomputing a denormalized representation of variant data and storing them in a single table. Specifics of how exploring variant data stored in a materialized view are discussed in section D.4.

Several of the bioinformatic analysis steps in Millstone, including alignment and variant calling, require several minutes to complete. We use Celery, a distributed task queue, for queuing, executing, and tracking the status of jobs processed asynchronously. Further, Celery integrates well with Django. Additionally, Celery can be configured to have multiple workers than can execute in parallel, as limited by the number of machine processor cores. In particular, we're able to take advantage of this when Millstone is deployed to an Amazon EC2 machine having dozens of cores.

D.6.2 *Deployment on Amazon Web Services*

The recommended method for deploying Millstone is through the Amazon cloud. Millstone can also be deployed to a local server or to a laptop. We discuss these options below.

The recommended method for deploying Millstone is for a new user to create a new AWS instance using the publicly available Millstone Amazon Machine Image (AMI). This requires very little configuration beyond choosing an Elastic Cloud Compute (EC2) instance with adequate resources. Within a few minutes of launching an instance of Millstone via Amazon, the user can navigate to the URL of the newly instantiated virtual machine and observe the Millstone landing page.

Users can find the latest guide to deploying Millstone to AWS at `http://millstone.readthedocs.org`

D.6.2.1 *Preparing an Amazon Machine Image*

Millstone developers prepare the Amazon Machine Image using Cloudbiolinux [46]. Cloudbiolinux makes deployment reproducible by allowing the developer to create a versioned snapshot of configuration files that will download and install software dependencies. Cloudbiolinux offers custom configurations through *flavors*. We have implemented a Millstone flavor and it is now available as part of the Cloudbiolinux source. At time of writing, Millstone configuration files and a README are located at `https://github.com/chapmanb/cloudbiolinux/tree/master/contrib/flavor/millstone`.

D.6.2.2 *Costs*

The price of running Millstone on Amazon Web Services depends on the configuration. A typical setup using a *m3.medium* EC2 instance and 100 GB of storage costs **$58.24 per month**. This price can be significantly reduced by shutting down the EC2 machine when it is not in use and other simple cost-saving measures discussed below. Additionally, Academic researchers may be able to entirely cover the cost of using AWS resources by applying for an AWS Research Grant at `https://aws.amazon.com/grants/`. In our own case, the development and use of Amazon resources with Millstone over the past two years have entirely been covered by Amazon research grants.

There are two main decisions with regard to cost: compute and storage. Compute is charged per hour while the machine is turned on and depends on the Elastic Cloud Compute (EC2) instance configuration chosen. EC2 prices currently range from $0.013 per hour for a *t2.micro* instance up to $5.52 per hour for a *d2.8xlarge* instance with 48 TB of storage with 36 vCPUs (approximately equivalent to cores) and 244 GB of memory. Storage, provided by Amazon Elastic Block Store (EBS) effectively serves as virtual external disks that can be provisioned and moved between EC2 instances. SSD-backed storage is charged at $0.10 per GB-month. Slower, magnetic drive backed storage costs $0.05 per GB-month. Intermediate users of AWS can modify possible these configurations on a Millstone instance at different stages of analysis without losing data, as discussed below.

For compute, we recommend at least an *m3.medium* instance, although Millstone is designed to leverage multiple cores to parallelize certain tasks such as alignment and variant calling. At time of writing, the price of running a *m3.medium* machine is $0.067 per hour. Users should refer to the EC2 pricing guide (`https://aws.amazon.com/ec2/pricing/`) for more information on pricing.

For storage, we have found it effective to allocate approximately three times the size of the input genome sequencing data files via an SSD-backed EBS. This is done at time of instance creation. A large portion of this requirement is allocated for storing generated alignment files. A typical instance of Millstone requires between 100 GB and 1 TB of storage.

Users who are more familiar with AWS can change the EC2 instance type and storage requirements on an existing Millstone instance without losing data. This is particularly advantageous to do once alignment and variant calling is complete and only analysis and exploration is being performed. For example, in our own use of Millstone, we will typically provision a highly compute-optimized machine such as a *c4.8xlarge* to run alignment and variant calling. Then we will switch back to an *m3.medium* machine for analysis. Running a *c4.8xlarge* machine is currently priced at $1.763 per hour, and we are able to align and call variants for 96 *E. coli* genomes in under 3 hours. We then switch the instance to *m3.medium* for follow-up analysis. Before repeating an analysis, we will switch to a bigger instance. Instructions on changing instance types are available in the online documentation at `http://millstone.readthedocs.org/`.

D.6.3 *Local Deployment*

Millstone can be installed on a local server or laptop. The Millstone repository can be cloned from our public Github repository at `https://github.com/churchlab/millstone`. This requires additional steps for installing prerequisites as detailed in the README.

BIBLIOGRAPHY

[1] Malin Allert, J Colin Cox, and Homme W Hellinga. "Multifactorial Determinants of Protein Expression in Prokaryotic Open Reading Frames." In: *Journal of Molecular Biology* 402.5 (Oct. 2010), pp. 905–918 (cit. on pp. 34, 40, 46).

[2] H Alper, C Fischer, E Nevoigt, and G Stephanopoulos. "Tuning genetic control through promoter engineering." In: *Proceedings of the National Academy of Sciences of the United States of America* 102.36 (Sept. 2005), pp. 12678–12683 (cit. on p. 26).

[3] J Christopher Anderson. *Anderson Promoter Library, Registry of Standard Biological Parts.* Tech. rep. (cit. on p. 77).

[4] J Christopher Anderson. *Anderson RBS Library, Registry of Standard Biological Parts.* Tech. rep. (cit. on p. 77).

[5] Ernesto Andrianantoandro, Subhayu Basu, David K Karig, and Ron Weiss. "Synthetic biology: new engineering rules for an emerging discipline." In: *Molecular Systems Biology* 2 (May 2006) (cit. on p. 26).

[6] Adam Arkin. "Setting the standard in synthetic biology." In: *Nature Biotechnology* 26.7 (July 2008), pp. 771–774 (cit. on p. 26).

[7] T E Arnold, J Yu, and J G Belasco. "mRNA stabilization by the ompA 5′ untranslated region: two protective elements hinder distinct pathways for mRNA degradation." In: *RNA* 4.3 (1998), pp. 319–330 (cit. on p. 34).

[8] Doug Barrick, Keith Villanueba, John Childs, Rhonda Kalil, Thomas D Schneider, Charles E Lawrence, Larry Gold, and Gary D Stormo. "Quantitative analysis of ribosome binding sites in E.coli." In: *Nucleic Acids Research* 22.7 (1994), pp. 1287–1295 (cit. on p. 26).

[9] Michael Baym, Sergey Kryazhimskiy, Tami D Lieberman, Hattie Chung, Michael M Desai, and Roy Kishony. *Inexpensive Multiplexed Library Preparation for Megabase-Sized Genomes.* Tech. rep. Jan. 2015 (cit. on p. 70).

[10] N Benson, P Sugiono, and P Youderian. "DNA sequence determinants of lambda repressor binding in vivo." In: *Genetics* 118.1 (Jan. 1988), pp. 21–29 (cit. on p. 53).

[11] K Bentele, P Saffert, R Rauscher, Z Ignatova, and N Bluthgen. "Efficient translation initiation dictates codon usage at gene start." In: *Molecular Systems Biology* 9.1 (Jan. 2013), pp. 675–675 (cit. on pp. 40, 46).

[12] F R Blattner et al. "The complete genome sequence of Escherichia coli K-12." In: *Science (New York, NY)* 277.5331 (Sept. 1997), pp. 1453–1462 (cit. on p. 99).

[13] J Bonnet, P Subsoontorn, and D Endy. "Rewritable digital data storage in live cells via engineered control of recombination directionality." In: *Proceedings of the National Academy of Sciences of the United States of America* 109.23 (June 2012), pp. 8884–8889 (cit. on p. 26).

[14] Douglas F Browning and Stephen J W Busby. "The regulation of bacterial transcription initiation." In: *Nature Reviews Microbiology* 2.1 (Jan. 2004), pp. 57–65 (cit. on p. 52).

[15] Tammy J Bullwinkle, Daniel Samorodnitsky, Rayna C Rosati, and Gerald B Koudelka. "Determinants of Bacteriophage 933W Repressor DNA Binding Specificity." In: *PLoS ONE* 7.4 (Apr. 2012), e34563 (cit. on p. 53).

[16] F D Bushman and M Ptashne. "Activation of transcription by the bacteriophage 434 repressor." In: *Proceedings of the National Academy of Sciences of the United States of America* 83.24 (Dec. 1986), pp. 9353–9357 (cit. on pp. 53, 58).

[17] Barry Canton, Anna Labno, and Drew Endy. "Refinement and standardization of synthetic biological parts and devices." In: *Nature Biotechnology* 26.7 (July 2008), pp. 787–793 (cit. on p. 26).

[18] Robert Carlson. "Laying the foundations for a bio-economy." In: *Systems and Synthetic Biology* 1.3 (2007), pp. 109–117 (cit. on p. 26).

[19] Peter A Carr and George M Church. "Genome engineering." In: *Nature Biotechnology* 27.12 (Dec. 2009), pp. 1151–1162 (cit. on p. 26).

[20] J Robert Coleman, Dimitris Papamichail, Steven Skiena, Bruce Futcher, Eckard Wimmer, and Steffen Mueller. "Virus attenuation by genome-scale changes in codon pair bias." In: *Science (New York, NY)* 320.5884 (June 2008), pp. 1784–1787 (cit. on p. 46).

[21] Tim F Cooper, Susanna K Remold, Richard E Lenski, and Dominique Schneider. "Expression profiles reveal parallel evolution of epistatic interactions involving the CRP regulon in Escherichia coli." In: *PLoS Genetics* 4.2 (Feb. 2008), e35 (cit. on p. 58).

[22] Robert Sidney Cox, Mary J Dunlop, and Michael B Elowitz. "A synthetic three-color scaffold for monitoring genetic regulation and noise." In: *Journal of Biological Engineering* 4.1 (2010), p. 10 (cit. on pp. 27, 36, 77, 99).

[23] Richard Dawkins. *The Blind Watchmaker*. Why the Evidence of Evolution Reveals a Universe Without Design. W. W. Norton & Company, 1986 (cit. on p. 7).

[24] S L Dove and A HOCHSCHILD. "Conversion of the omega subunit of Escherichia coli RNA polymerase into a transcriptional activator or an activation target." In: *Genes and Development* 12.5 (Mar. 1998), pp. 745–754 (cit. on p. 58).

[25] Drew Endy, Adam P Arkin, and Jay D Keasling. *BIOFAB*. Tech. rep. (cit. on pp. 26, 77).

[26] M Gouy and C Gautier. "Codon usage in bacteria: correlation with gene expressivity." In: *Nucleic Acids Research* 10.22 (Nov. 1982), pp. 7055–7074 (cit. on p. 41).

[27] Christopher J Gregg, Marc J Lajoie, Michael G Napolitano, Joshua A Mosberg, Daniel B Goodman, John Aach, Farren J Isaacs, and George M Church. "Rational optimization of tolC as a powerful dual selectable marker for genome engineering." In: *Nucleic Acids Research* 42.7 (2014), pp. 4779–4790 (cit. on p. 19).

[28] Wanjun Gu, Tong Zhou, and Claus O Wilke. "A universal trend of reduced mRNA stability near the translation-initiation site in prokaryotes and eukaryotes." In: *PLoS Computational Biology* 6.2 (Feb. 2010), e1000664 (cit. on pp. 34, 40, 46).

[29] Călin C Guet, Michael B Elowitz, Weihong Hsing, and Stanislas Leibler. "Combinatorial synthesis of genetic networks." In: *Science (New York, NY)* 296.5572 (May 2002), pp. 1466–1470 (cit. on p. 52).

[30] Adrian D Haimovich, Paul Muir, and Farren J Isaacs. "Genomes by design." In: *Nature Reviews Genetics* 16.9 (2015), pp. 501–516 (cit. on p. 73).

[31] D L Hartl, E N Moriyama, and S A Sawyer. "Selection intensity for codon bias." In: *Genetics* 138.1 (Sept. 1994), pp. 227–234 (cit. on p. 40).

[32] Matthias Heinemann and Sven Panke. "Synthetic biology-putting engineering into biology." In: *Bioinformatics (Oxford, England)* 22.22 (2006), pp. 2790–2799 (cit. on p. 26).

[33] Farren J Isaacs et al. "Precise manipulation of chromosomes in vivo enables genome-wide codon replacement." In: 333.6040 (2011), pp. 348–353 (cit. on p. 70).

[34] F JACOB, D PERRIN, C SANCHEZ, and J MONOD. "[Operon: a group of genes with the expression coordinated by an operator]." In: *Comptes rendus hebdomadaires des séances de l'Académie des sciences* 250 (Feb. 1960), pp. 1727–1729 (cit. on p. 52).

[35] F JACOB, A ULLMAN, and J MONOD. "[THE PROMOTOR, A GENETIC ELEMENT NECESSARY TO THE EXPRESSION OF AN OPERON]." In: *Comptes rendus hebdomadaires des séances de l'Académie des sciences* 258 (Mar. 1964), pp. 3125–3128 (cit. on p. 52).

[36] Anitha D Jayaprakash, Omar Jabado, Brian D Brown, and Ravi Sachidanandam. "Identification and remediation of biases in the activity of RNA ligases in small-RNA deep sequencing." In: *Nucleic Acids Research* 39.21 (Nov. 2011), e141–e141 (cit. on p. 81).

[37] Wenyan Jiang, David Bikard, David Cox, Feng Zhang, and Luciano A Marraffini. "RNA-guided editing of bacterial genomes using CRISPR-Cas systems." In: *Nature Biotechnology* 31.3 (2013), pp. 233–239 (cit. on p. 70).

[38] Jay D Keasling. "Manufacturing molecules through metabolic engineering." In: *Science (New York, NY)* 330.6009 (2010), pp. 1355–1358 (cit. on p. 26).

[39] Justin B Kinney, Anand Murugan, Curtis G Callan, and Edward C Cox. "Using deep sequencing to characterize the biophysical mechanism of a transcriptional regulatory sequence." In: *Proceedings of the National Academy of Sciences of the United States of America* 107.20 (2010), pp. 9158–9163 (cit. on p. 52).

[40] Joshua T Kittleson, Gabriel C Wu, and J Christopher Anderson. "Successes and failures in modular genetic engineering." In: *Current opinion in chemical biology* (2012) (cit. on p. 26).

[41] Anton A Komar. "A pause for thought along the co-translational folding pathway." In: *Trends in biochemical sciences* 34.1 (Jan. 2009), pp. 16–24 (cit. on p. 46).

[42] Sriram Kosuri and George M Church. "Large-scale de novo DNA synthesis: technologies and applications." In: *Nature Methods* 11.5 (May 2014), pp. 499–507 (cit. on p. 17).

[43] Sriram Kosuri, Daniel B Goodman, Guillaume Cambray, Vivek K Mutalik, Yuan Gao, Adam P Arkin, Drew Endy, and George M Church. "Composability of regulatory sequences controlling transcription and translation in Escherichia coli." In: *Proceedings of the National Academy of Sciences of the United States of America* (Aug. 2013) (cit. on pp. 40, 52, 53, 99–101, 123).

[44] Jonathan W Kotula, S Jordan Kerns, Lev A Shaket, Layla Siraj, James J Collins, Jeffrey C Way, and Pamela A Silver. "Programmable bacteria detect and record an environmental signal in the mammalian gut." In: *Proceedings of the National Academy of Sciences of the United States of America* 111.13 (Apr. 2014), pp. 4838–4843 (cit. on p. 61).

[45] Astrid P Koudelka, Lisa A Hufnagel, and Gerald B Koudelka. "Purification and characterization of the repressor of the shiga toxin-encoding bacteriophage 933W: DNA binding, gene regulation, and autocleavage." In: *Journal of Bacteriology* 186.22 (Nov. 2004), pp. 7659–7669 (cit. on p. 53).

[46] Konstantinos Krampis, Tim Booth, Brad Chapman, Bela Tiwari, Mesude Bicak, Dawn Field, and Karen E Nelson. "Cloud BioLinux: pre-configured and on-demand bioinformatics computing for the genomics community." In: *BMC Bioinformatics* 13.1 (2012), p. 42 (cit. on p. 146).

[47] Grzegorz Kudla, Andrew W Murray, David Tollervey, and Joshua B Plotkin. "Coding-sequence determinants of gene expression in Escherichia coli." In: *Science (New York, NY)* 324.5924 (Apr. 2009), pp. 255–258 (cit. on pp. 34, 40, 43, 46).

[48] M J Lajoie, D Söll, and G M Church. "Overcoming challenges in engineering the genetic code." In: *Journal of Molecular Biology* (2015) (cit. on p. 19).

[49] Marc J Lajoie, Alexis J Rovner, Daniel B Goodman, Hans-Rudolf Aerni, Adrian D Haimovich, Gleb Kuznetsov, Jaron A Mercer, Harris H Wang, Peter A Carr, Joshua A Mosberg, et al. "Genomically recoded organisms expand biological functions." In: *Science (New York, NY)* 342.6156 (2013), pp. 357–360 (cit. on pp. 19, 72–74, 141).

[50] Ben Langmead, Cole Trapnell, Mihai Pop, and Steven L Salzberg. "Ultrafast and memory-efficient alignment of short DNA sequences to the human genome." In: *Genome Biology* 10.3 (2009), R25 (cit. on pp. 37, 83, 101).

[51] Emily M LeProust, Bill J Peck, Konstantin Spirin, Heather Brummel McCuen, Bridget Moore, Eugeni Namsaraev, and Marvin H Caruthers. "Synthesis of high-quality libraries of long (150mer) oligonucleotides by a novel depurination controlled process." In: *Nucleic Acids Research* 38.8 (May 2010), pp. 2522–2540 (cit. on pp. 17, 27, 36, 40, 53, 78).

[52] Gene-Wei Li, Eugene Oh, and Jonathan S Weissman. "The anti-Shine-Dalgarno sequence drives translational pausing and codon choice in bacteria." In: *Nature* 484.7395 (Apr. 2012), pp. 538–541 (cit. on pp. 40, 41, 103).

[53] Heng Li and Richard Durbin. "Fast and accurate short read alignment with Burrows–Wheeler transform." In: *Bioinformatics (Oxford, England)* 25.14 (2009), pp. 1754–1760 (cit. on p. 139).

[54] Kevin D Litcofsky, Raffi B Afeyan, Russell J Krom, Ahmad S Khalil, and James J Collins. "Iterative plug-and-play methodology for constructing and modifying synthetic gene networks." In: *Nature Methods* 9.11 (Nov. 2012), pp. 1077–1080 (cit. on p. 52).

[55] Peng Lu, Christine Vogel, Rong Wang, Xin Yao, and Edward M Marcotte. "Absolute protein expression profiling estimates the relative contributions of transcriptional and translational regulation." In: *Nature Biotechnology* 25.1 (Jan. 2007), pp. 117–124 (cit. on p. 103).

[56] Nicholas R Markham and Michael Zuker. "UNAFold: software for nucleic acid folding and hybridization." In: *Methods in molecular biology (Clifton, NJ)* 453 (2008), pp. 3–31 (cit. on p. 99).

[57] Vincent J J Martin, Douglas J Pitera, Sydnor T Withers, Jack D Newman, and Jay D Keasling. "Engineering a mevalonate pathway in Escherichia coli for production of terpenoids." In: *Nature Biotechnology* 21.7 (2003), pp. 796–802 (cit. on p. 26).

[58] Steven A Mauro, David Pawlowski, and Gerald B Koudelka. "The role of the minor groove substituents in indirect readout of DNA sequence by 434 repressor." In: *The Journal of biological chemistry* 278.15 (Apr. 2003), pp. 12955–12960 (cit. on p. 53).

[59] Vivek K Mutalik et al. "Precise and reliable gene expression via standard transcription and translation initiation elements." In: *Nature Methods* 10.4 (Apr. 2013), pp. 354–360 (cit. on p. 26).

[60] Vivek K Mutalik et al. "Quantitative estimation of activity and quality for collections of functional genetic elements." In: *Nature Methods* 10.4 (Apr. 2013), pp. 347–353 (cit. on pp. 26, 32, 34, 36).

[61] Dokyun Na, Sunjae Lee, and Doheon Lee. "Mathematical modeling of translation initiation for the estimation of its efficiency to computationally design mRNA sequences with desired expression levels in prokaryotes." In: *BMC systems biology* 4 (2010), p. 71 (cit. on p. 26).

[62] Sivan Navon and Yitzhak Pilpel. "The role of codon selection in regulation of translation efficiency deduced from synthetic libraries." In: *Genome Biology* 12.2 (Feb. 2011), R12 (cit. on pp. 40, 41).

[63] OECD. *The Bioeconomy to 2030: Designing a Policy Agenda.* OECD Publishing, 2009 (cit. on p. 26).

[64] Rupali P Patwardhan, Choli Lee, Oren Litvin, David L Young, Dana Pe'er, and Jay Shendure. "High-resolution analysis of DNA regulatory elements by synthetic saturation mutagenesis." In: *Nature Biotechnology* 27.12 (Jan. 2009), pp. 1173–1175 (cit. on pp. 26, 27, 52).

[65] Sebastian Pechmann and Judith Frydman. "Evolutionary conservation of codon optimality reveals hidden signatures of cotranslational folding." In: *Nat Struct Mol Biol* 20.2 (Feb. 2013), pp. 237–243 (cit. on pp. 40, 41, 46, 103).

[66] Jean-Denis Pédelacq, St e phanie Cabantous, Timothy Tran, Thomas C Terwilliger, and Geoffrey S Waldo. "Engineering and characterization of a superfolder green fluorescent protein." In: *Nature Biotechnology* 24.1 (Jan. 2006), pp. 79–88 (cit. on pp. 27, 40, 77).

[67] Joshua B Plotkin and Grzegorz Kudla. "Synonymous but not the same: the causes and consequences of codon bias." In: *Nat Rev Genet* 12.1 (Jan. 2011), pp. 32–42 (cit. on pp. 40, 41, 46).

[68] Mark Ptashne. *A Genetic Switch.* Phage Lambda Revisited. CSHL Press, 1986 (cit. on p. 52).

[69] Tali Raveh-Sadka, Michal Levo, Uri Shabi, Boaz Shany, Leeat Keren, Maya Lotan-Pompan, Danny Zeevi, Eilon Sharon, Adina Weinberger, and Eran Segal. "Manipulating nucleosome disfavoring sequences allows fine-tune regulation of gene expression in yeast." In: *Nature Genetics* 44.7 (2012), pp. 743–750 (cit. on p. 27).

[70] Mario dos Reis, Renos Savva, and Lorenz Wernisch. "Solving the riddle of codon usage preferences: a test for translational selection." In: *Nucleic Acids Research* 32.17 (2004), pp. 5036–5044 (cit. on pp. 40, 41).

[71] Dae-Kyun Ro et al. "Production of the antimalarial drug precursor artemisinic acid in engineered yeast." In: *Nature* 440.7086 (2006), pp. 940–943 (cit. on p. 26).

[72] Jameson K Rogers and George M Church. "Genetically encoded sensors enable real-time observation of metabolite production." In: *Proceedings of the National Academy of Sciences of the United States of America* 113.9 (Mar. 2016), pp. 2388–2393 (cit. on p. 53).

[73] Jameson K Rogers, Christopher D Guzman, Noah D Taylor, Srivatsan Raman, Kelley Anderson, and George M Church. "Synthetic biosensors for precise gene control and real-time monitoring of metabolites." In: *Nucleic Acids Research* 43.15 (Sept. 2015), pp. 7648–7660 (cit. on pp. 53, 61).

[74] Nitzan Rosenfeld, Jonathan W Young, Uri Alon, Peter S Swain, and Michael B Elowitz. "Accurate prediction of gene feedback circuit behavior from component properties." In: *Molecular Systems Biology* 3 (Nov. 2007), p. 143 (cit. on p. 26).

[75] Emily F Ruff, M Thomas Record, and Irina Artsimovitch. "Initial events in bacterial transcription initiation." In: *Biomolecules* 5.2 (2015), pp. 1035–1062 (cit. on p. 52).

[76] Howard M Salis, Ethan A Mirsky, and Christopher A Voigt. "Automated design of synthetic ribosome binding sites to control protein expression." In: *Nature Biotechnology* 27.10 (2009), pp. 946–950 (cit. on pp. 26, 34, 77).

[77] Luis Serrano. "Synthetic biology: promises and challenges." In: *Molecular Systems Biology* 3 (2007), p. 158 (cit. on p. 26).

[78] Premal Shah, Yang Ding, Malwina Niemczyk, Grzegorz Kudla, and Joshua B Plotkin. "Rate-Limiting Steps in Yeast Protein Translation." In: *Cell* 153.7 (June 2013), pp. 1589–1601 (cit. on pp. 40, 46).

[79] Nathan C Shaner, Michael Z Lin, Michael R McKeown, Paul A Steinbach, Kristin L Hazelwood, Michael W Davidson, and Roger Y Tsien. "Improving the photostability of bright monomeric orange and red fluorescent proteins." In: *Nature Methods* 5.6 (2008), pp. 545–551 (cit. on p. 40).

[80] Elaine B Shapland et al. "Low-Cost, High-Throughput Sequencing of DNA Assemblies Using a Highly Multiplexed Nextera Process." In: *ACS Synthetic Biology* 4.7 (July 2015), pp. 860–866 (cit. on p. 70).

[81] Eilon Sharon, Yael Kalma, Ayala Sharp, Tali Raveh-Sadka, Michal Levo, Danny Zeevi, Leeat Keren, Zohar Yakhini, Adina Weinberger, and Eran Segal. "Inferring gene regulatory logic from high-throughput measurements of thousands of systematically designed promoters." In: *Nature Biotechnology* 30.6 (2012), pp. 521–530 (cit. on p. 27).

[82] P M Sharp, T M Tuohy, and K R Mosurski. "Codon usage in yeast: cluster analysis clearly differentiates highly and lowly expressed genes." In: *Nucleic Acids Research* 14.13 (July 1986), pp. 5125–5143 (cit. on p. 41).

[83] Katsuyuki Shiroguchi, Tony Z Jia, Peter A Sims, and X Sunney Xie. "Digital RNA sequencing minimizes sequence-dependent bias and amplification noise with optimized single-molecule barcodes." In: *Proceedings of the National Academy of Sciences of the United States of America* 109.4 (Jan. 2012), pp. 1347–1352 (cit. on p. 103).

[84] Xiaokun Shu, Nathan C Shaner, Corinne A Yarbrough, Roger Y Tsien, and S James Remington. "Novel chromophores and buried charges control color in mFruits." In: *Biochemistry* 45.32 (2006), pp. 9639–9647 (cit. on pp. 27, 77).

[85] Ryan K Shultzaberger, Daniel S Malashock, Jack F Kirsch, and Michael B Eisen. "The fitness landscapes of cis-acting binding sites in different promoter and environmental contexts." In: *PLoS Genetics* 6.7 (July 2010), e1001042 (cit. on p. 52).

[86] M H de Smit and J van Duin. "Secondary structure of the ribosome binding site determines translational efficiency: a quantitative analysis." In: *Proceedings of the National Academy of Sciences of the United States of America* 87.19 (Oct. 1990), pp. 7668–7672 (cit. on p. 46).

[87] John St. John. *SeqPrep*. Tech. rep. (cit. on pp. 37, 82, 101).

[88] Eric J Steen, Yisheng Kang, Gregory Bokinsky, Zhihao Hu, Andreas Schirmer, Amy McClure, Stephen B Del Cardayre, and Jay D Keasling. "Microbial production of fatty-acid-derived fuels and chemicals from plant biomass." In: *Nature* 463.7280 (2010), pp. 559–562 (cit. on p. 26).

[89] Arvind R Subramaniam, Tao Pan, and Philippe Cluzel. "Environmental perturbations lift the degeneracy of the genetic code to regulate protein levels in bacteria." In: *Proceedings of the National Academy of Sciences of the United States of America* 110.6 (Feb. 2013), pp. 2419–2424 (cit. on pp. 40, 46).

[90] Jeffrey J Tabor, Howard M Salis, Zachary Booth Simpson, Aaron A Chevalier, Anselm Levskaya, Edward M Marcotte, Christopher A Voigt, and Andrew D Ellington. "A synthetic genetic edge detection program." In: *Cell* 137.7 (2009), pp. 1272–1281 (cit. on p. 26).

[91] Noah D Taylor et al. "Engineering an allosteric transcription factor to respond to new ligands." In: *Nature Methods* 13.2 (Feb. 2016), pp. 177–183 (cit. on p. 61).

[92] Karsten Temme, Dehua Zhao, and Christopher A Voigt. "Refactoring the nitrogen fixation gene cluster from Klebsiella oxytoca." In: *Proceedings of the National Academy of Sciences of the United States of America* 109.18 (2012), pp. 7085–7090 (cit. on p. 26).

[93] O Tenaillon, A Rodriguez-Verdugo, R L Gaut, P Mcdonald, A F Bennett, A D Long, and B S Gaut. "The Molecular Diversity of Adaptive Convergence." In: *Science (New York, NY)* 335.6067 (Jan. 2012), pp. 457–461 (cit. on pp. 19, 72, 74).

[94] Tamir Tuller, Asaf Carmi, Kalin Vestsigian, Sivan Navon, Yuval Dorfan, John Zaborske, Tao Pan, Orna Dahan, Itay Furman, and Yitzhak Pilpel. "An evolutionarily conserved mechanism for controlling the efficiency of protein translation." In: *Cell* 141.2 (Apr. 2010), pp. 344–354 (cit. on pp. 40, 41).

[95] Tamir Tuller, Yedael Y Waldman, Martin Kupiec, and Eytan Ruppin. "Translation efficiency is determined by both codon bias and folding energy." In: *Proceedings of the National Academy of Sciences of the United States of America* 107.8 (Feb. 2010), pp. 3645–3650 (cit. on pp. 40, 41, 46).

[96] Dieter Voges, Manfred Watzele, Cordula Nemetz, Sabine Wizemann, and Bernd Buchberger. "Analyzing and enhancing mRNA translational efficiency in an Escherichia coli in vitro expression system." In: *Biochem Biophys Res Commun* 318.2 (May 2004), pp. 601–614 (cit. on p. 43).

[97] Irina O Vvedenskaya, Yuanchao Zhang, Seth R Goldman, Anna Valenti, Valeria Visone, Deanne M Taylor, Richard H Ebright, and Bryce E Nickels. "Massively Systematic Transcript End Readout, "MASTER": Transcription Start Site Selection, Transcriptional Slippage, and Transcript Yields." In: *Molecular Cell* 60.6 (Dec. 2015), pp. 953–965 (cit. on pp. 52, 59).

[98] Harris H Wang, Farren J Isaacs, Peter A Carr, Zachary Z Sun, George Xu, Craig R Forest, and George M Church. "Programming cells by multiplex genome engineering and accelerated evolution." In: *Nature* 460.7257 (Aug. 2009), pp. 894–898 (cit. on p. 26).

[99] Harris H Wang, Farren J Isaacs, Peter A Carr, Zachary Z Sun, George Xu, Craig R Forest, and George M Church. "Programming cells by multiplex genome engineering and accelerated evolution." In: *Nature* 460.7257 (2009), pp. 894–898 (cit. on p. 70).

[100] Mark Welch, Sridhar Govindarajan, Jon E Ness, Alan Villalobos, Austin Gurney, Jeremy Minshull, and Claes Gustafsson. "Design parameters to control synthetic gene expression in Escherichia coli." In: *PLoS ONE* 4.9 (Sept. 2009), e7002 (cit. on pp. 34, 40, 46).

[101] Yukiko Yamazaki, Hironori Niki, and Jun-ichi Kato. "Profiling of Escherichia coli Chromosome database." In: *Methods in molecular biology (Clifton, NJ)* 416 (2008), pp. 385–389 (cit. on pp. 40, 99, 101, 102).

[102] O Yarchuk, N Jacques, J Guillerez, and M Dreyfus. "Interdependence of translation, transcription and mRNA degradation in the lacZ gene." In: *Journal of Molecular Biology* 226.3 (1992), pp. 581–596 (cit. on p. 34).

[103] Joseph N Zadeh, Conrad D Steenberg, Justin S Bois, Brian R Wolfe, Marshall B Pierce, Asif R Khan, Robert M Dirks, and Niles A Pierce. "NUPACK: Analysis and design of nucleic acid systems." In: *Journal of computational chemistry* 32.1 (Jan. 2011), pp. 170–173 (cit. on pp. 43, 103).

[104] Gabriel E Zentner and Steven Henikoff. "High-resolution digital profiling of the epigenome." In: *Nature Reviews Genetics* 15.12 (Dec. 2014), pp. 814–827 (cit. on p. 17).

[105] Gabriel E Zentner and Steven Henikoff. "High-resolution digital profiling of the epigenome." In: *Nature Reviews Genetics* 15.12 (Dec. 2014), pp. 814–827 (cit. on p. 26).

[106] Daniel R Zerbino and Ewan Birney. "Velvet: algorithms for de novo short read assembly using de Bruijn graphs." In: *Genome Research* 18.5 (May 2008), pp. 821–829 (cit. on p. 143).

[107] Mian Zhou, Jinhu Guo, Joonseok Cha, Michael Chae, She Chen, Jose M Barral, Matthew S Sachs, and Yi Liu. "Non-optimal codon usage affects expression, structure and function of clock protein FRQ." In: *Nature* 495.7439 (Mar. 2013), pp. 111–115 (cit. on pp. 40, 46).