

UC Berkeley

UC Berkeley Electronic Theses and Dissertations

Title

Regression Modeling and Bias Correction of Ribosome Profiling Data

Permalink

<https://escholarship.org/uc/item/5js332t1>

Author

Tunney, Robert

Publication Date

2018

Peer reviewed|Thesis/dissertation

Regression Modeling and Bias Correction of Ribosome Profiling Data

by

Robert Tunney

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computational Biology

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Dr. Liana F. Lareau, Co-chair

Dr. Lior Pachter, Co-chair

Dr. Michael I. Jordan

Dr. Jamie H. D. Cate

Spring 2018

Regression Modeling and Bias Correction of Ribosome Profiling Data

Copyright 2018
by
Robert Tunney

Abstract

Regression Modeling and Bias Correction of Ribosome Profiling Data

by

Robert Tunney

Doctor of Philosophy in Computational Biology

University of California, Berkeley

Dr. Liana F. Lareau, Co-chair

Dr. Lior Pachter, Co-chair

Translational regulation is an important control point for gene expression, modulating the quantity and isoforms of proteins produced in a cell. The ribosome profiling method measures translation dynamics and output directly, by sampling the distribution of ribosomes across all mRNA transcripts in a sample. This method has demonstrated that ribosomes do not move at a uniform rate across transcripts, and that synonymous codon choice can have large effects on ribosome speed, RNA stability, and protein expression. We present a regression model using a feedforward neural network to predict the ribosome density at each codon in a transcriptome as a function of the local sequence neighborhood around that codon. This approach demonstrated a collection of sequence features that contain substantial predictive information about translation elongation rates. We apply this model to characterize the translation rates of naturally occurring genes, and also to design translation optimized coding sequences for a given protein. We present a novel and efficient algorithm that finds the fastest and slowest predicted coding sequences for a given protein. We validated our regression model and optimization procedure by designing synonymous variants of eCitrine, a yellow fluorescent protein, across a range of predicted translation rates. Our results showed that the levels of expressed protein closely tracked the predicted overall translation rates of the synonymous coding sequences. This demonstrated that our model captures information determining translation dynamics *in vivo*, that we can harness this information to design coding sequences, and that control of translation elongation alone is sufficient to produce large, quantitative differences in protein output. Analysis of our regression model also demonstrated that the terminal regions of ribosome footprints are important predictors of footprint density at a given codon. This suggests that ligation events in the experimental protocol are differentially recovering footprints based on their terminal sequences. We characterized this recovery bias both computationally and experimentally, and demonstrated that it can have a large impact on the count of footprints recovered at a given codon. To correct for this error, we developed a generative model of ribosome footprint experiments that

incorporates both the biological distribution of footprints across transcripts and the experimental steps that introduce recovery bias. We developed a software tool to estimate the parameters of this model, and present a statistical method to correct recovery bias and estimate the biological distribution of ribosomes across transcripts. This method enables improved estimates of translation for the many applications in which ribosome profiling data is used.

To my teachers,
at the end of my formal education.

Contents

Contents	ii
1 Introduction	1
2 Modeling and Optimization of Translation Rates	7
2.1 Background	7
2.2 Regression Model	8
2.3 Model Performance	11
2.4 Model Interpretation	13
2.5 Application - Coding Sequence Optimization	17
2.6 Supplementary Materials	20
2.7 Methods	29
3 Bias Correction of Ribosome Profiling Data	36
3.1 Background	36
3.2 Assumptions	38
3.3 Model Definitions	40
3.4 Handling Missing Data	42
3.5 Model Training	44
3.6 Simulation Results	48
3.7 Relative Ligation Bias	49
3.8 Ligation Bias Correction	53
Bibliography	61

Acknowledgments

Foremost, I would like to thank my thesis advisers, Drs. Liana Lareau and Lior Pachter. Without their guidance, this work would be impossible. I would also like to thank the faculty of the Computational Biology Graduate Group at UC Berkeley, for creating the community and educational environment that supported me as I pursued this work. Within this group, special thanks are due to Dr. Nicholas Ingolia, for his guidance and space in his lab community, and to the professors on my academic committees, including Drs. Haiyan Huang, Jamie Cate, Michael Jordan, and Nir Yosef. I would also like to our program officers, Kate Chase, Brian McClendon, and Xuan Quach, who make the program run. I couldn't forget my classmates and peers in the graduate school, who have been a beautiful and inspiring community to exist within. Much love to my peers past and present in the Ingolia, Lareau, and Pachter labs, and especially to my office buddy Shannon McCurdy for her advice and intellectual discussions. Finally, I'd like to express my gratitude for the opportunity to learn and grow in this environment. It's been a beautiful experience.

Chapter 1

Introduction

Background

The estimation of global gene expression profiles is a central goal in genomics research. Genome-wide expression measurements have increased our understanding of gene functions, interactions, regulatory networks, and conditions resulting from dysregulation. Many techniques estimate gene expression levels by quantifying the amount of mRNA for each gene in a sample [6, 35]. This approach captures transcriptional regulation of gene expression, but does not account for variability in translation rates, errors, and decay of protein products.

The ribosome profiling method has enabled genome-wide measurement of translation rates, revealing an additional layer in our emerging understanding of gene expression. This method samples the distribution of ribosomes across transcripts by isolating the fragments of mRNA protected within ribosome samples and recovering them in a sequencing library. Ultimately this procedure yields histograms of ribosome counts along positions in each transcript[24]. A key feature of this data is that high footprint counts indicate high ribosomal occupancy, thus yielding information about ribosome flow and translation dynamics. Ribosome profiling data have yielded more accurate measurements of gene expression levels, and have also revealed many novel features of translational regulation, including abundant 5' upstream open reading frames (uORFs) [24, 23, 5], alternative start codons [23, 50], and variable decoding speeds by codon [52, 2, 15, 39, 38]. This procedure has grown in popularity, and is broadly used to understand the mechanics and regulation of translation.

Nuclease Digestion and A site assignment

The ribosome profiling protocol has a number of distinctive features that we must understand in order to interpret the resulting data appropriately. The first of these features

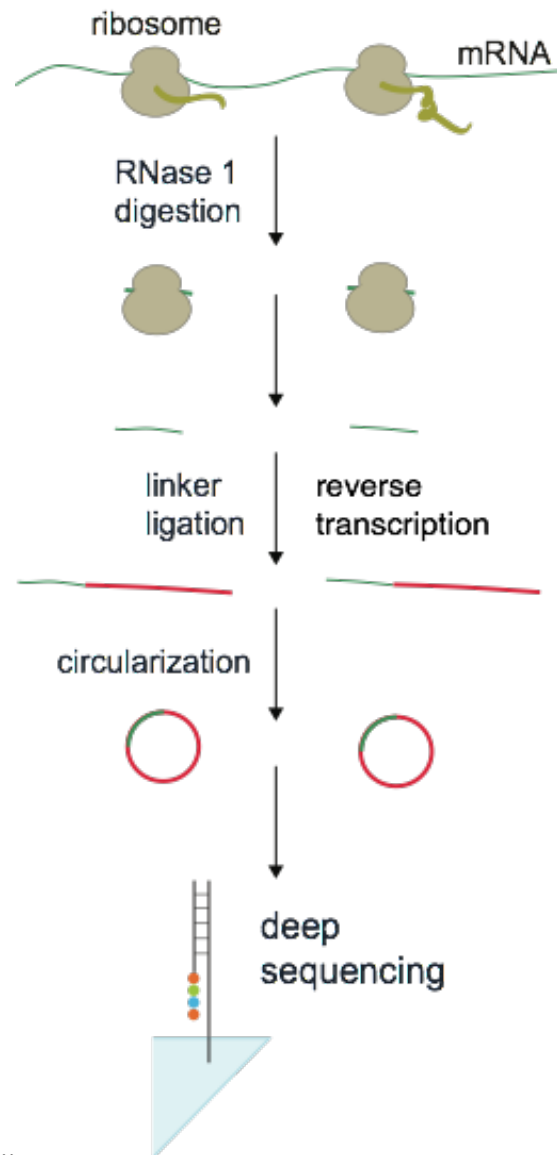


Figure 1.1: The ribosome profiling method. Total RNA is collected from a sample, and subjected to digestion with a nuclease. The fragments of mRNA protected within ribosomes, called ribosome footprints, are isolated to prepare a sequencing library. Footprints are ligated to adapters on each end, reverse transcribed, amplified via PCR, and then sequenced.

is the digestion process that defines the boundaries of a ribosome footprint. During the experimental protocol, total RNA is harvested and treated with a nuclease to digest solvent accessible RNA (Figure 1.1). The fragments of mRNA that survive this digestion are physically protected by some binding complex, typically the ribosome. When we isolate resulting fragments in the appropriate size range for the ribosome (around 28-30 nucleotides), we discover that this digestion is generally quite efficient. Footprints are observed in a narrow size window. Mapping these footprints back to a transcriptome, we observe strong three-nucleotide periodicity in the fragment ends, indicating that the digestion is accurate to below the resolution of a codon[24]. However, there is some variability in the lengths of footprints. This reflects the fact that nucleases can digest footprints to a small range of lengths relative to the margin of a ribosome. As a result, it is not appropriate to define the location of a footprint relative to its end. It is more natural to define its location relative to an internal site of functional significance to the translation process. The ribosome contains three internal sites for storing tRNAs during translation elongation. The A site is the site of mRNA decoding, where a charged tRNA enters the ribosome with an anti-codon complementary to the current A site codon. The P (peptidyl) site contains a tRNA with a 3' ester linkage to the nascent peptide chain. This peptide is elongated and transferred to the charged A site tRNA during peptide bond formation. The E (exit) site contains discharged tRNAs ready to be ejected from the ribosome (Figure 1.2). It is well established that the A site is the most important functional determinant of translation elongation rates[33, 37, 30]. This site serves as a natural reference position to define the location of ribosome footprints (Figure 1.3).

A range of methods have been used to assign A sites in ribosome footprints. At one end, there are simple heuristic rules that define a distance from the 5' end of a footprint to its A site, or restrict analyses to a set of footprint sizes where this assignment is unambiguous. At the other end there are methods that perform probabilistic A site assignment to handle uncertainty in the case of highly underdigested or overdigested footprints[53]. We find in practice that variation in digestion is small, and simple heuristic rules are effective for assigning A sites in the vast majority of footprints. To perform this task, we first define the 0th, 1st, and 2nd frames of a coding sequence as the set of 0th, 1st, and 2nd nucleotides in the codons of that coding sequence. We divide our footprints up into classes based on their length in nucleotides and the frame to which their 5' end maps. For each class, we can assign an offset between the 5' end and the A site by examining footprints near the start codon of their transcript and determining how far their 5' ends lie relative to the second codon in the transcript, which is the first A site codon to generate ribosome footprints. We can use this approach to determine a range of 5' and 3' digest lengths that comprise most of our observed data, and assign offset rules to each footprint class accordingly.

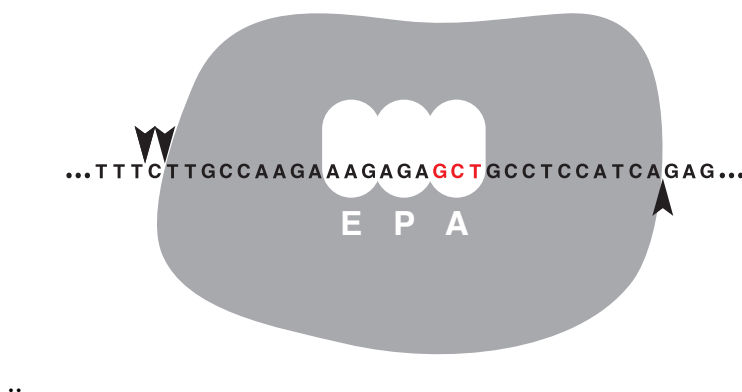


Figure 1.2: Cartoon diagram of a ribosome. In white, functional sites in the ribosome that hold tRNAs. in red, the A site, where tRNAs are acquired to decode codons into amino acids. P, the site of peptide bond formation in the nascent protein. E, the ejection site for discharged tRNAs. Arrows indicate common 5' and 3' ends for ribosome footprints after digestion.

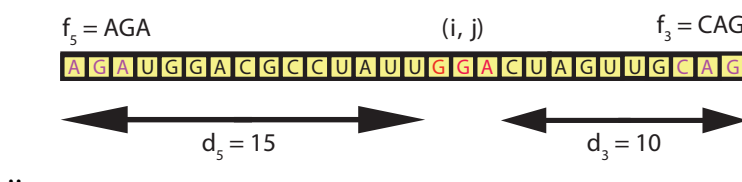


Figure 1.3: Diagram of a ribosome footprint. A 28 nucleotide footprint is depicted, with the A site codon highlighted in red. d_5 , d_3 indicate the distances from the A site to the 5' and 3' ends of footprints. These lengths can vary with digestion. f_5 , f_3 indicate the fragment ends that interact with ligases in sequencing library preparation. i indexes the transcript that generated the footprint, and j indexes the A site codon of the footprint within that transcript.

Linker Ligation and Recovery Bias

The ribosome profiling protocol requires addition of sequence onto the 5' and 3' ends of footprints in order to prepare a sequencing library (Figure 1.1). Several strategies exist to perform these modifications, but the most common strategy is a ligation of adapter sequence onto each end of the footprint sequence. These ligations can be performed by a number of RNA and ssDNA ligases, commonly T4 RNA Ligase I and II, and Circligase I and II. These enzymes are known to have sequence preferences for the fragments that they ligate[28, 19], which can introduce selective bias for the footprints that are recovered in a ribosome profiling library[37]. It is particularly important to be mindful of these biases when interpreting the counts of footprints at individual codons. Due to efficient digestion, the footprints from a given A site codon will exhibit only a few possible 5' and 3' end sequences. If these ends are highly preferred or disfavored by ligases, the observed count of footprints at that codon can be highly skewed. In addition, the average count of footprints at each codon is low for most experiments. This introduces additional uncertainty when estimating the local rate of translation elongation at each codon.

Determinants of Translation Elongation Rate

There are a number of outstanding challenges regarding interpretation and processing of ribosome profiling data. One problem that has generated substantial interest is using this data to interpret the sequence features that drive translation speed and regulation. Since ribosome profiling gives us measurements of the local rate of translation elongation along transcripts, it is natural to develop models that connect these rates with sequence features. We can use these models to understand interactions between the components of the translational system, and to study how the parameters of this system may drive sequence evolution[33, 37]. This analysis has identified a large suite of sequence features and characteristics of nascent peptides that affect translation rates and regulation, including wobble pairing, charged residues, polyproline tracts, and tRNA structure and abundance[49, 12, 2, 31, 39, 30]. While we have a good idea of the set of features that may impose selective constraints on coding sequences, this knowledge has not been applied extensively to engineer coding sequences with desirable properties. In Chapter 1, we develop an improved regression model to relate the sequence context of a ribosome to its elongation rate. We present a method to interpret the contributions of individual features to local elongation rate, and we develop a new algorithm to find the coding sequence of a gene that is optimized for elongation time. Finally, we present experimental evidence that this optimization is strongly related to the amount of protein produced per unit of mRNA.

Recovery Bias Correction

Several analyses have observed that biased sequence composition exists at the ends of ribosome footprints, and have suggested that these biases may arise from the ligation events we describe above[37, 33]. To our knowledge, there have been no attempts to correct these biases in ribosome profiling data. Well developed methods exist for bias correction in general mRNA sequencing experiments, but the assumptions of these models are not suited to analysis of ribosome profiling data. In particular, we expect nonuniform distribution of ribosome footprints across transcripts, because variation in translation elongation rates is an interesting property that is revealed from this data. Consequently, there is need for a comprehensive probabilistic model that incorporates both the underlying biological distribution of ribosome footprints along with the components of the experimental procedure that bias their recovery. In Chapter 2, we develop this model, and we present a procedure to correct ligation biases and recover the biological distribution of ribosome density across transcripts. This correction is particularly important for measuring the effects of codon level sequence features.

Chapter 2

Modeling and Optimization of Translation Rates

2.1 Background

Synonymous codon choice can have dramatic effects on ribosome speed, RNA stability, and protein expression. Ribosome profiling experiments have underscored that ribosomes do not move uniformly along mRNAs, exposing a need for models of coding sequences that capture the full range of empirically observed variation. We present a method, Inos, that models this variation in translation elongation using a feedforward neural network to predict the ribosome density at each codon as a function of its sequence neighborhood. Our approach revealed sequence features affecting translation elongation and quantified the impact of large technical biases in ribosome profiling. We applied our model to design synonymous variants of a fluorescent protein spanning the range of possible translation speeds predicted with our model. We found that levels of the fluorescent protein in yeast closely tracked the predicted translation speeds across their full range. We therefore demonstrate that our model captures information determining translation dynamics *in vivo*, that we can harness this information to design coding sequences, and that control of translation elongation alone is sufficient to produce large, quantitative differences in protein output.

As the ribosome moves along a transcript, it encounters diverse codons, tRNAs, and amino acids. This diversity affects translation elongation and, ultimately, gene expression. For instance, exogenous gene expression can be seriously hampered by a mismatch between the choice of synonymous codons and the availability of tRNAs. The consequences of endogenous variation in codon use have been more elusive, but new methods have revealed that synonymous coding mutations, upregulation of tRNAs, and mutations within tRNAs can have dramatic effects on protein expression, folding, and stability[25, 17, 27]. Codon usage can directly affect the speed of translation elongation[56].

However, translation initiation has been considered the rate-limiting step in translation, implying that changes in elongation speed should have limited effects[46]. Recent work has suggested a relationship between codon use and RNA stability; slower translation may destabilize mRNAs and thus decrease protein expression[40, 4]. These opposing viewpoints have yet to be fully reconciled, leaving us with an incomplete understanding of what defines a favorable sequence for translation. With the advent of high-throughput methods to measure translation elongation *in vivo*, we can understand the functional implications of codon usage. Ribosome profiling measures translation transcriptome-wide by capturing and sequencing the regions of mRNA protected within ribosomes, called ribosome footprints[24]. Each footprint reflects the position of an individual ribosome on a transcript, and we can reliably infer the A site codon the site of tRNA decoding in each footprint (Fig. 1a). This codon-level resolution yields the distribution of ribosomes along mRNAs from each gene. We can use the counts of footprints on each codon to infer translation elongation rates: slowly translated codons yield more footprints, and quickly translated codons yield fewer (Fig. 1b). Analyses of ribosome profiling data have shown a relationship between translation elongation rate and biochemical features like tRNA abundance, wobble base pairing, amino acid polarity, and mRNA structure[49, 10, 30, 55, 13, 33, 12, 39, 15, 8]. Expanded probabilistic and machine learning models have shown that the sequence context of a ribosome contributes to its elongation rate, both directly and through higher order features such as nascent protein sequence[37, 55, 33, 12]. Computational modeling has also indicated that technical artifacts and biases contribute to the distribution of ribosome footprints[2, 37, 13, 22]. However, it remains a challenge to distinguish experimental artifacts from the biological determinants of elongation rate. Here, we have used neural networks to model ribosome distribution along transcripts. The model captured both biological variation in translation elongation speed and technical biases affecting footprint count, which we confirmed experimentally. We have implemented a tool, *Ixnos*, that applies our model to design coding sequences, and used this to design sequences spanning a range of predicted translation elongation speeds. We found that the predicted elongation speeds accurately tracked protein expression, supporting a role for the elongation phase of translation in modulating gene expression.

2.2 Regression Model

First we developed a regression framework to model the distribution of ribosomes along transcripts as a function of local sequence features. As our measure of ribosome density on individual codon positions, we calculated scaled footprint counts by dividing the raw footprint count at each codon position by the average footprint count on its transcript (Fig. 1b). This normalization controls for variable mRNA abundances and translation initiation rates across transcripts. The scaled count thus reflects the relative speed of translation elongation at each position. We used a sequence neighborhood around the A

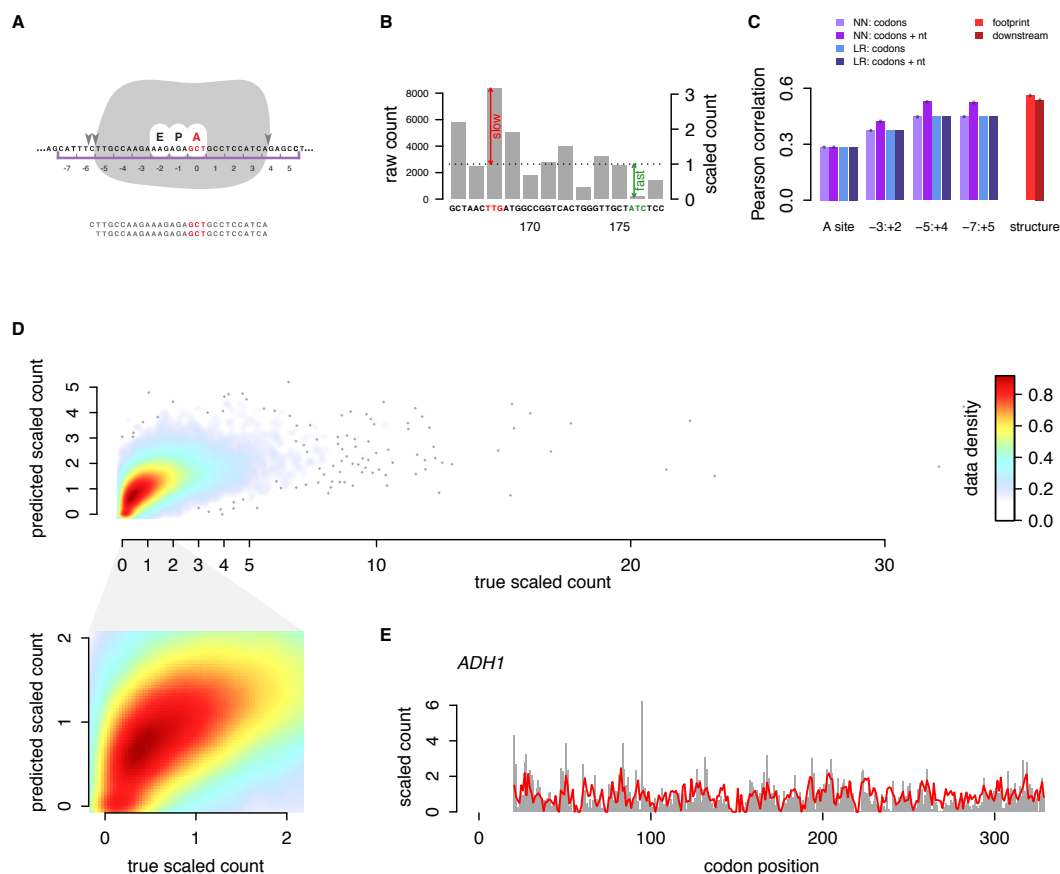


Figure 2.1: Design and performance of a neural network model of translation elongation rates **a** Each ribosome protects an mRNA footprint of approximately 28-29 nt. Sequence coordinates in a neighborhood around a ribosome are indexed relative to the codon in the A site of the ribosome. **b** Read count rescaling. For each gene, the counts of footprints assigned to each A site codon are divided by the average counts per codon over that gene. The resulting scaled footprint counts are used for model training and prediction. **c** Model performances (Pearson correlations between predicted and true scaled counts over the test set) for neural network and linear regression models over a range of sequence neighborhoods, with and without nucleotide features, as well as correlations for models that also incorporate structure scores of the three 30-nt windows overlapping the footprint region, or the maximum structure score within 59 nt downstream of the ribosome. Bars show the mean of 10 runs of each model; the 10 individual runs for each model are overlaid as gray points. **d** True vs. predicted scaled counts for the test set, under a model with codon and nucleotide features spanning codon positions -5 to +4. Color scale shows density of data points. **e** True scaled counts (gray bars) and predicted scaled counts (red line) for a highly translated gene.

site as the predictive region for scaled counts, and encoded this neighborhood as input to a regression model via one-hot encoding of the codons and nucleotides in this region (Supplementary Fig. 1). Then we learned a regression function with a feedforward neural network, trained on a large, high quality ribosome profiling data set from *Saccharomyces cerevisiae*[54]. We chose the top 500 genes by footprint density and coverage criteria, and sorted these into training and test sets of 333 and 167 genes, respectively.

We determined the sequence neighborhood that best predicted ribosome density by comparing a series of models ranging from an A-site-only model to a model spanning codon positions -7 to +5 (Fig. 1c). The identity of the A site codon was an important, but limited, predictor of the distribution of ribosome footprints (Pearsons $r = 0.28$). Expanding the sequence context around the A site steadily improved the predictive performance, up to the full span of a ribosome footprint (codons -5 to +4). Additional sequence context beyond the boundaries of the ribosome did not improve performance. We also observed a large boost in predictive performance by including redundant nucleotide features in addition to codon features over the same sequence neighborhood, especially near the ends of the ribosome footprint (Fig. 1c, $r = 0.53$ for -5 to +4 model including nucleotide features, $\Delta r = 0.08$ relative to no-nucleotide model). Linear regression models that only included codon features performed similarly to the neural networks we tested, but they did not improve with the inclusion of nucleotide features. This suggests that the neural network models learn a meaningful and nonlinear predictive relationship in nucleotide features, particularly toward the flanking ends of footprints, that makes them more successful than linear models.

Next we assessed the contribution of local mRNA structure to footprint distributions. We computed mRNA folding energies in sliding 30 nt windows over all transcripts, and trained a series of models that each included one window from nucleotide positions -45 to +72 relative to the A site. Performance improved upon including structure scores at nucleotide positions -17, -16, and -15, i.e., the windows that span the actual ribosome footprint ($\Delta r = 0.03$; Fig. 1c and Supplementary Fig. 2). No individual windows downstream of the footprint improved our predictions, and the maximum structure score over 30 sliding windows downstream of the ribosome had only a slight effect ($\Delta r < 0.01$) (Fig. 1c). Thus, our approach does not capture a conclusive effect of downstream mRNA structure on elongation rate. We were surprised to see an effect of structure within the ribosome, so we tested the direction of the effect and found that more structure in these windows led to lower predicted footprint counts. This suggests that stable mRNA structure in the footprint fragments themselves is inhibiting their in vitro recovery in ribosome profiling experiments, and our model is capturing the bias that this introduces to the data.

2.3 Model Performance

Our best model incorporated a sequence window from codons -5 to +4 represented as both codons and nucleotides, as well as structure features of the three windows spanning the footprint. It captured sufficient information to accurately predict footprint distributions on individual genes (Fig. 1e), and yielded a correlation of 0.57 (Pearson's r) between predicted and true scaled counts over all positions in the test set (Fig. 1d). Although our model performed well across a range of scaled counts, it had difficulty predicting very high scaled footprint counts at a small number of sites. These sites may represent ribosome stalling that is determined by biological factors encoded outside of this local sequence neighborhood[55].

Our model was trained on highly expressed genes because abundant ribosome footprints enable more accurate sampling of ribosome positions. However, highly expressed genes can have biased codon usage[47]. To ensure that our model was accurately predicting translation on genes across the full range of expression and codon usage, we computed the correlation between the observed and predicted scaled counts for all yeast genes. Performance decreased with lower expression (Fig. 2a), but we hypothesized that the decreased performance reflected noisier observed footprint counts arising from less-abundant mRNAs, rather than differences in their codon composition. To test this, we downsampled the footprints for each of the 1000 highest-expression genes to match the average counts per codon of the 1000th gene, and repeated this procedure for the top 2000, 3000, and 4000 genes. We then compared the predictions of our model, which had been trained on the full data from highly expressed genes, against the downsampled data. At each coverage level, our method performed equally well on high-expression genes and low-expression genes. Thus, our model had no decrease in performance on genes that tend to have less favored codon content, after controlling for data density.

We also compared the performance of our model against two earlier approaches that incorporate information from the sequence neighborhood of each codon to predict ribosome distributions: RUST, which computes the expected ribosome density at each codon based on its sequence window[37], and riboshape, which uses wavelet decomposition to denoise the observed counts by projecting them into different subspaces at different levels of resolution (smoothness), and then predicts ribosome density after transformation into these subspaces[33]. To compare riboshape to our own method and to RUST, we evaluated how well its predictions in the highest resolution subspace (i.e., closest to the raw data) correlated with the observed footprint counts. Our model out-performed both models, with an average Pearson correlation per gene of 0.56 versus 0.48 (RUST) and 0.41 (riboshape) across all genes that were included in all three analyses (Fig. 2b). We also found that our predictions of the raw data were better than riboshapes predictions of the transformed data at each resolution (Supplementary Fig. 3).

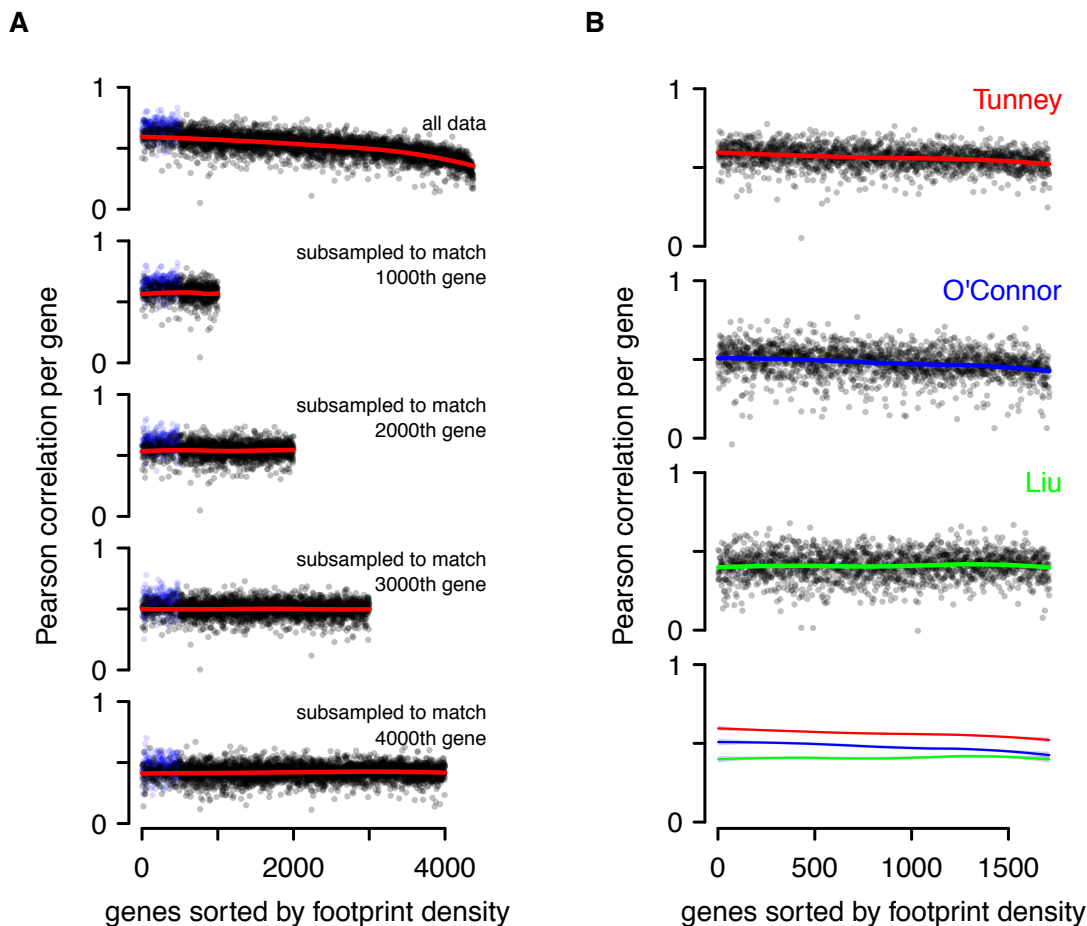


Figure 2.2: Performance comparisons on low coverage genes and with competing models **a** Top, per-gene correlations between true and predicted scaled counts, for all 4375 genes in our transcriptome that passed filtering criteria. Training set genes in blue (333/top 500 genes by footprint density). Loess curve on test set genes shown in red. Below, as above, with footprint counts on the top 1000, 2000, 3000, and 4000 genes subsampled to the density of footprint counts on the 1000th, 2000th, 3000th, and 4000th gene, respectively, and true scaled counts recomputed. Controlling for the density of footprint counts before rescaling, a model trained on high expression genes performs similarly across the full range of high and low expression genes. **b** Comparison of $I\chi$ nos with similar models, RUST[37] and riboshape[33]. Shown are per-gene correlations between true and predicted scaled counts, on 1711 genes passing the filtering criteria from all three methods. Training set genes from $I\chi$ nos are excluded. Colored lines are loess curves, which are also compared in the bottom panel.

2.4 Model Interpretation

Biological Determinants of Translation Elongation Rate

To quantify the influence of distinct positions in the sequence neighborhood on elongation rate, we trained a series of leave-one-out models that excluded individual codon positions from the input sequence neighborhood, and compared their performance to a reference model that included all positions. We found that the A site codon contributed the most to predictive performance ($r = 0.13$), but we also saw contributions from the surrounding sequence context, including the P and E sites ($r = 0.03$ and 0.03) (Fig. 3a). Each codon position from -5 to +4, the span of a typical 28 nt ribosome footprint, improved performance of the full model, whereas positions outside the span of a footprint decreased performance. Contributions from the E and P sites suggest that the continued presence of tRNAs at these positions modulates elongation rate. In contrast, the large contribution of the +3 codon ($r = 0.06$), at the 3' end of the footprint, likely reflects artifactual biases arising from the ribosome profiling process, corroborating previous reports of fragment end biases[2, 37].

We were also interested in understanding the relative influence of the A site codon and its immediate environment. Overall, the A site codon and its immediate environment predict ribosome density similarly well (Pearsons $r = 0.28$ for the A site only, $r = 0.26$ for the codons from -3 to +2 excluding the A site). To identify A-site codons that tend to dominate the prediction, contributing relatively more than their context, we compared the performance of a -3:+2 model and a model with codons -3 to +2 but excluding the A site (Supplementary Fig. 4a). We found that the presence of lysine codons AAA and AAG in the A site led to the strongest predictions, in agreement with a major effect of charged lysine residues on translation[8]. Conversely, we also identified a number of sequence contexts that tended to dominate the prediction, by looking at the sequence contexts of the positions with more error arising from the A-site-only prediction than the no-A-site context (Supplementary Fig. 4b).

Next, we examined what our model had learned about the relationship between sequence and ribosome density. The raw parameters of a neural network can be difficult to interpret, so we determined a score for each codon at each position by computing the mean increase in predicted scaled counts due to that codon (Fig. 3b and Supplementary Table 1). Time spent finding the correct tRNA is considered to be a main driver of elongation speed, and consequently footprint counts[38]. Indeed, the A site codon scores exhibited the widest range of codon scores, and scores at this position but not other positions correlated with tRNA Adaptation Index (tAI), a measure of tRNA availability[42], as has been widely observed (Pearsons $r = 0.50$; $p = 0.0005$ after Bonferroni correction). Our results highlighted the well-characterized slow translation of CCG (Pro), CGA (Arg), and CGG (Arg) codons at the A site[31]. Our data also underscore that sequences in the P

site contribute to elongation speed. The CGA codon showed a particularly strong inhibitory effect in the P site, in keeping with recent results[31, 14]. We noted that this codon forms a disfavored I:A wobble pair with its cognate tRNA, distorting the anticodon loop[36], while the four fastest P site codons all form I:C wobble pairs (Fig. 3c). Overall, I:C base pairs in the P site contributed to faster translation (Mann-Whitney $p = 0.014$ after Bonferroni correction, Fig. 3c). From this, we concluded that the conformation of the tRNA:mRNA duplex can influence its passage through the ribosome, not just initial recognition in the A site.

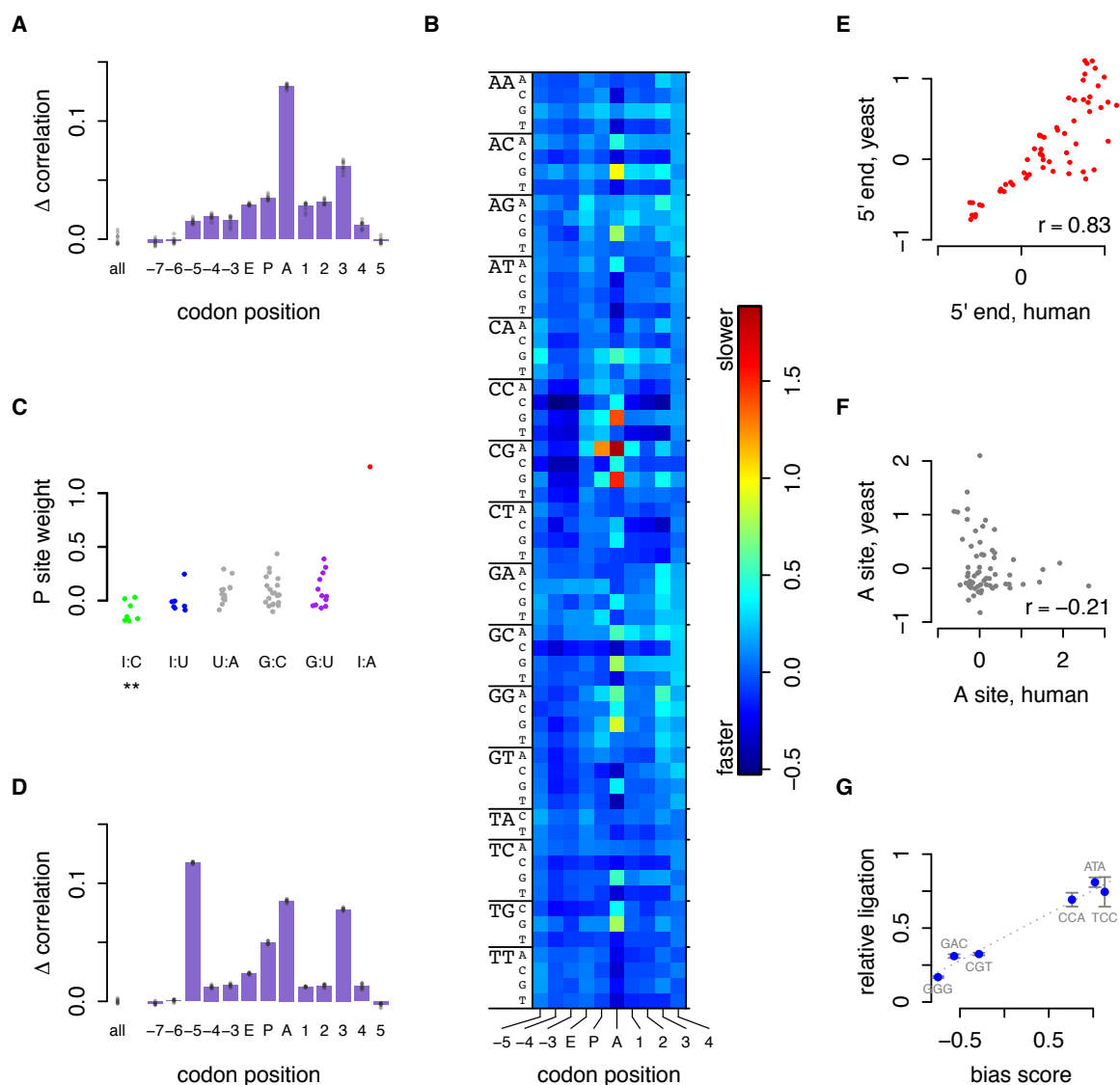


Figure 2.3: Biological interpretation of models of translation elongation rates **a** Predictive value of codon positions in a yeast ribosome profiling dataset from Weinberg et al.[54] We computed the Pearson correlations between true and predicted scaled counts on the test set, for a reference model including codon features from positions -7 to +5, along with the associated nucleotides. We also computed Pearson correlations for a series of leave-one-out models, each excluding one codon position and the associated nucleotide positions. Shown are differences between the reference and leave-one-out correlations. Gray points show Pearsons r for 10 runs of each model. Bars represent the mean correlation of each model compared to the mean correlation of the full model. **b** Mean contributions

to scaled counts by codon identity and position. **c** P site codon contributions grouped by the codon:anticodon base pair formed by the third nucleotide of each codon. Asterisks indicate $p < 0.05$ after Bonferroni correction, unpaired two-sided Mann-Whitney U test between each group and all other codons. I:C, $p = 0.014$. **d** Predictive value of codon positions as in A, from a yeast ribosome profiling library we constructed using CircLigase II as described by McGlincy and Ingolia[34]. **e, f** Contributions from **(e)** codon position -5, at the 5' ends of footprints, and **(f)** the A site, in human ribosome profiling from Iwasaki et al.[26] versus our yeast ribosome profiling, both using CircLigaseII. Analysis was limited to 28-nt footprints to avoid frame biases. Fragment end codons that contribute to recovery bias are highly correlated, whereas A site codons that contribute strongly to translation elongation rate are not correlated between species. **g** Ligation efficiency of CircLigase II. Oligonucleotide substrates resembling ribosome footprints at the circularization step of the protocol, with different three-nucleotide end sequences, were ligated by both enzymes. Circularization was assayed by qPCR using primers spanning the ligation as compared to primers in a contiguous region of the oligo. Ligation was calculated relative to CircLigase I ligation of the best-ligated substrate. Each point represents the ratio of the means of three qPCR replicates; error bars represent the standard error of that ratio.

Technical Recovery Biases

We also observed strong sequence preferences at the 3' end of ribosome footprints. Sequence bias has previously been noted in the 5' and 3' ends of ribosome footprints, and this bias has been suggested to arise from ligase preferences during library preparation[2, 37]. To compare features of ribosome profiling data generated in different experiments, we applied our model to a large ribosome profiling dataset that we generated from yeast using a standard ribosome profiling protocol[34]. Models trained on these data learned disconcertingly high weights for both the -5 and +3 codon positions (Fig. 3d). The -5 codon, i.e., the 5' end of a footprint, was the single strongest predictor of footprint counts, exceeding even the A site. We found similarly large 5' end contributions in published yeast and human datasets generated using similar protocols[26, 45] (Supplementary Fig. 5). These experiments, like our own, made use of CircLigase enzymes to circularize ribosome footprints after reverse transcription. In contrast, the experiment we first modeled used T4 RNA ligase to attach 5' linkers directly onto ribosome footprint fragments[54]. To compare end sequence preferences between experiments, we trained models on only 28-nt footprints so that the ends of the footprints corresponded to the -5 codon position. Comparing the T4 ligase yeast data with CircLigase yeast data[26, 45], we observed no relationship between the scores learned at 5' footprint ends ($r = 0.05$), but a high correlation between scores at the A site, where we would expect biological similarity ($r = 0.86$). In contrast, we observed a high correlation at the -5 position between our CircLigase yeast data and the CircLigase-generated human data set[26] ($r = 0.83$, Fig. 3e), but no relation-

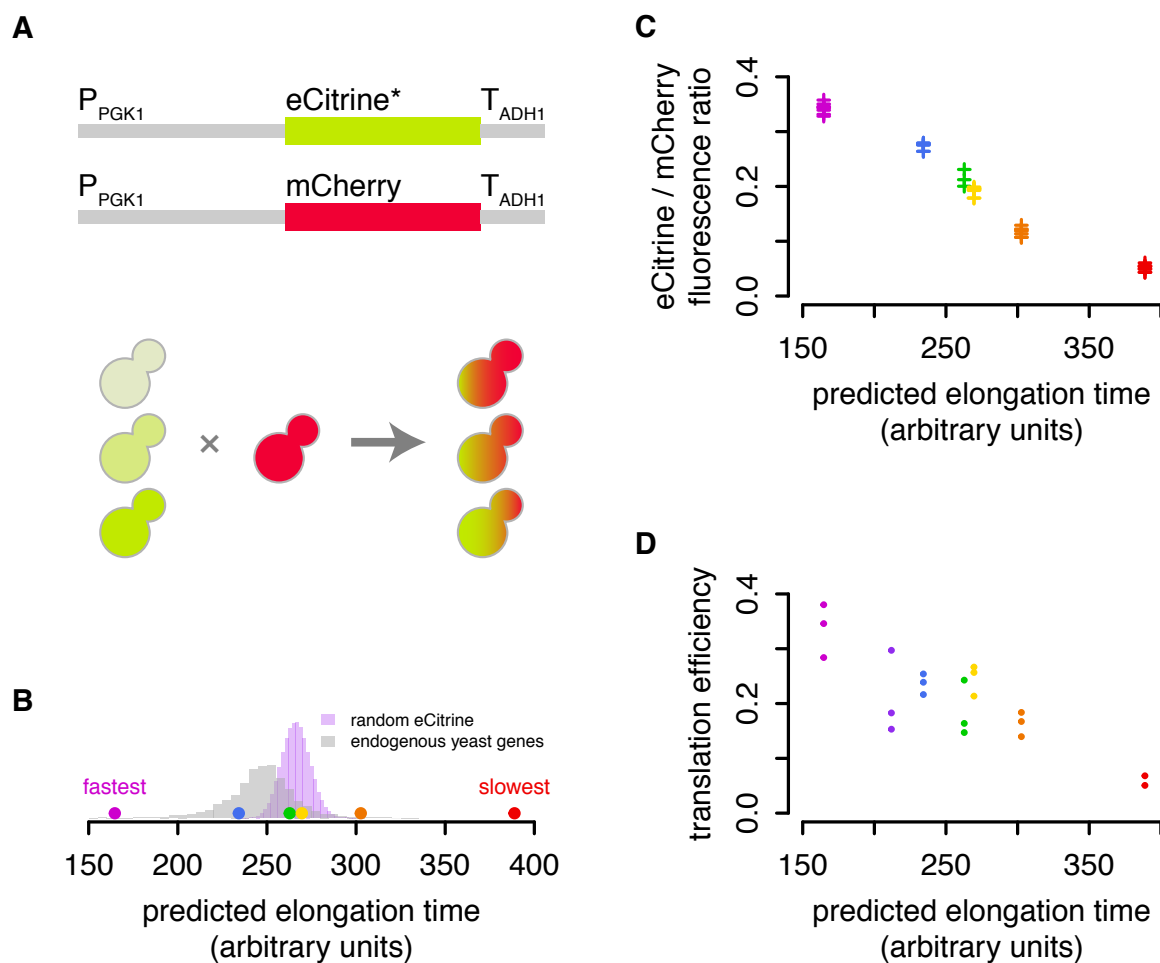
ship at the A site, where we would expect species-specific codon bias ($r = -0.21$, Fig. 3f). This suggested that the fragment end scores reflected experimental artifacts rather than in vivo biology.

To directly test the impact of enzyme biases on recovery of ribosome-protected fragments, we experimentally measured the ligation of synthetic oligonucleotides with end sequences shown to be favored or disfavored in our model. The relative ligation efficiency of each substrate closely mirrored the end sequence scores learned by our model for both CircLigase I and CircLigase II (Fig. 3g and Supplementary Fig. 6). The least-favored sequences were ligated by CircLigase II with only 20% the efficiency of the most-favored sequences, meaning that some ribosome footprints would be represented at five times the frequency of other footprints for purely technical reasons. This biased recovery of fragments could skew the results of ribosome profiling experiments, affecting estimates of elongation and overall per-gene translation.

Our model captured the quantitative preferences of ligases for footprint end sequences and established that a substantial portion of the predictive information of these end regions is due to technical artifacts. However, the biologically sensible weights learned for codons in the A site showed that the model captured substantial biology as well. We reasoned that, if our model were capturing biological aspects of translation elongation, we could use the parameters learned by the model to design sequences that would be translated at different rates. We relied on the information found in the codons closer to the A site, to focus on the biological contributions and reduce the influence of biases from the ends of footprints.

2.5 Application - Coding Sequence Optimization

To test our models ability to predict translation, we expressed synonymous variants of the yellow fluorescent protein eCitrine in yeast (Fig. 4a). First, using the yeast ribosome profiling data from Weinberg et al., we trained a neural network model with a sequence neighborhood extending from codon positions -3 to +2. Next, we designed a dynamic programming algorithm to compute the maximum- and minimum-translation-time synonymous versions of eCitrine based on our model. We defined the overall translation time (in arbitrary units) of a gene as the sum of predicted scaled counts over all codons in the gene. We also generated and scored a set of 100,000 random synonymous eCitrine CDSs and selected the sequences at the 0th, 33rd, 67th, and 99th percentiles of predicted translation time within that set (Fig. 4b). We used flow cytometry to measure the fluorescence of diploid yeast, each containing an eCitrine variant along with the red fluorescent protein mCherry as a control, and calculated relative fluorescence of each variant (Fig. 4c, Supplementary Fig. 7). The expression of eCitrine in each yeast strain closely



tracked its predicted elongation rate, with the predicted fastest sequence producing six-fold higher fluorescence than the predicted slowest sequence (Fig. 4c). However, the existing yeast-optimized yECitrine sequence[48] produced three-fold higher fluorescence than our predicted fastest sequence (Supplementary Fig. 8). To understand the source of this discrepancy, we measured eCitrine mRNA from all strains and found that sequences

Figure 2.3: Design of synonymous sequences shows elongation rate affects translation output **a** Six reporter constructs with distinct synonymous eCitrine coding sequences were inserted into the *his3 Δ 1* locus of BY4742 (-type) yeast, and an equivalent construct with a constant mCherry coding sequence was inserted into the *his3 Δ 1* locus of BY4741 (a-type) yeast. The haploids were mated to produce diploid yeast with eCitrine and mCherry reporters, whose fluorescence was then measured with flow cytometry. **b** The synonymous eCitrine sequences included the fastest and slowest predicted sequences un-

der our model (magenta and red), as well as sequences with predicted translation speed scores at the 0th, 33rd, 67th, and 100th percentiles of a randomly generated set of 100,000 synonymous eCitrine sequences (blue, green, yellow, and orange, respectively). The score distribution of 100,000 random eCitrine sequences is shown in lavender. The scores of endogenous yeast genes are shown in gray, and have been rescaled by length to compare with eCitrine. **c** eCitrine:mCherry fluorescence ratio, as measured by flow cytometry, versus score of each sequence in our model. Each + symbol represents the median ratio of yellow and red fluorescence from one biological replicate of the given eCitrine strain, as measured by flow cytometry of 11,000-18,000 yeast. Eight biological replicates, each an independent integration of the reporter construct, are included for each strain, except for the strains shown in blue and orange, which have seven, and the strain shown in green, which has three. Colors as in **(b)**. **d** Translation efficiency, or median eCitrine:mCherry fluorescence ratio divided by relative eCitrine:mCherry mRNA ratio (ratio of medians of three qPCR replicates), for each eCitrine variant, versus the score of each sequence in our model. Purple, yECitrine sequence; other colors as in **(b)**. Each point represents one biological replicate of the given eCitrine strain; three biological replicates were measured for each strain except two for the strain shown in red.

designed by our method had approximately equivalent mRNA levels, while yECitrine had five-fold more mRNA (Supplementary Fig. 8). Calculating translation efficiencies, or protein produced per mRNA, reconciled this disagreement. We observed a clear linear relationship between predicted elongation rate and translation efficiency (Fig. 4d).

These experiments demonstrate that our model is able to predict large, quantitative differences in protein production, based only on information about translation elongation. The sequences we designed and tested have predicted translation speeds that span the range of natural yeast genes (Fig. 4b). This supports an effect of elongation rate on the translation efficiency and protein output of endogenous genes. Initiation rather than elongation is usually thought to be rate limiting for protein production of most endogenous genes[46, 38]. Models have suggested that highly expressed trans-genes might deplete the effective supply of ribosomes, lowering initiation and thus causing elongation to be rate-limiting, but our reporter is expressed at the level of many endogenous genes and should represent well under 1% of mRNA. It remains to determine how translation speed can control translation efficiency. One contribution could come from pileups behind stalled or slow-moving ribosomes, diminishing the maximum throughput of protein production[12]. In particular, codon choice near the beginning of a gene, affecting elongation speed, can interfere with translation initiation and therefore control protein output[9]. Although codon choice can also affect mRNA stability and thus total protein output[40, 4], our fast and slow predicted sequences have equivalent steady-state mRNA. Further, an effect arising purely from mRNA stability would affect protein output but not translation efficiency, counter to our observations. Instead, our results indicate that optimized

elongation rates do result in more protein per mRNA, and this does not depend entirely on mRNA stability. The landscape of factors affecting codon optimality is complex[41], and codon preferences vary across species, tissues, and conditions. Our approach can capture empirical information about codon preferences in any system where translation can be measured by ribosome profiling, and apply it to design sequences for quantitative expression in that system.

2.6 Supplementary Materials

Feature importance codon scores

$s_{x,i}$ = codon score for codon x in position i
 average increase in predicted scaled counts when codon x is observed
 in position i

T_{te} = number of transcripts in test set

C_t = length of CDS t in codons

t = index over transcripts

c = index over codons

$\kappa(t, c)$ = function that returns the codon at position (t, c)

$v(t, c)$ = function that returns the sequence neighborhood around position (t, c)

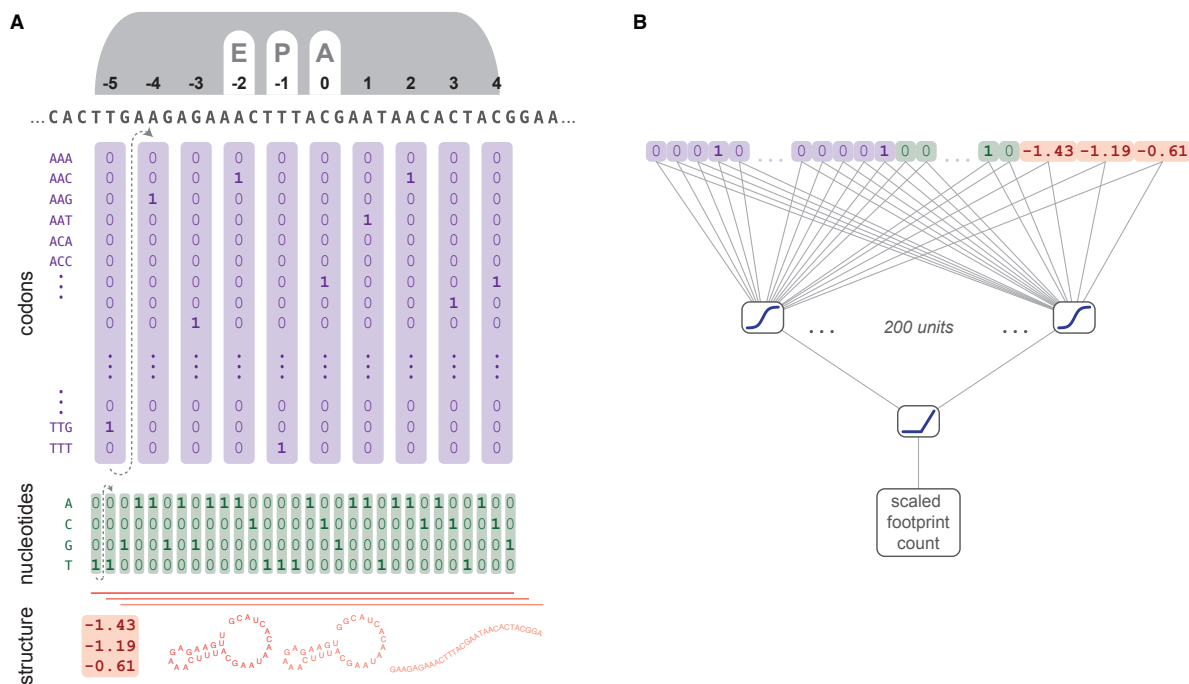
$v_{d,i}(t, c)$ = function that returns the sequence neighborhood around position (t, c) ,
 replaces the codon at position i with codon d

$$p_i(d) = \frac{\sum_{t=1}^{T_{te}} \sum_{c=1}^{C_t} \mathbf{1}(\kappa(t, c+i)=d)}{\sum_{s=1}^{T_{te}} C_s}$$

f = prediction function of the neural network model

$$s_{x,i} = \frac{1}{\sum_{t=1}^{T_{te}} \sum_{c=1}^{C_t} \mathbf{1}(\kappa(t, c+i)=x)} \sum_{t=1}^{T_{te}} \sum_{c=1}^{C_t} \mathbf{1}(\kappa(t, c+i)=x) \left[f(v(t, c)) - \sum_{d \in \{ACGT\}^3} p_i(d) f(v_{d,i}(t, c)) \right]$$

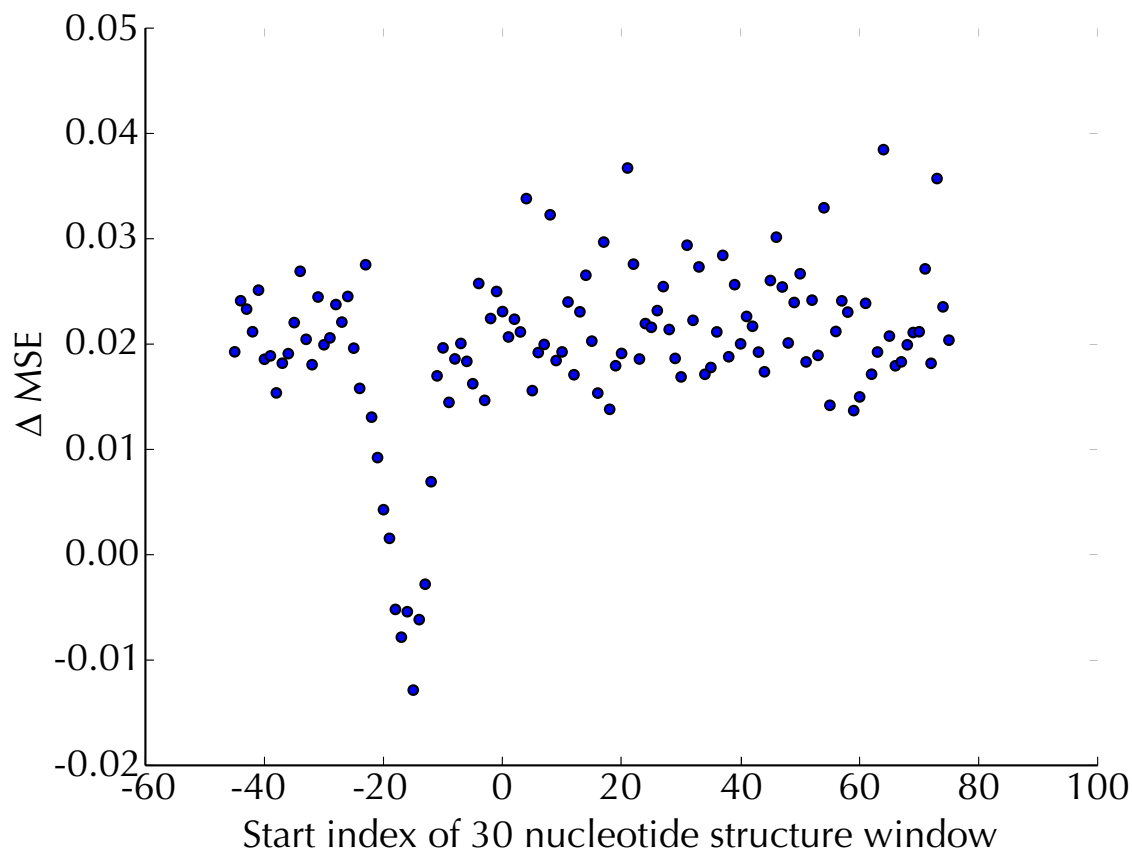
Supplementary figures



Supplementary Figure 1 Diagram of neural network model to predict scaled ribosome footprint counts.

(A) Counts are predicted at the A site codon. In the model shown, a sequence neighborhood spanning from 5 codons upstream of the A site (codon -5) to 4 codons downstream of the A site (codon +4) is used as the predictive region. This neighborhood is divided into codons and encoded via one-hot encoding (purple) for input into a regression model. We also encode the same region as nucleotide features (green) and include these features in the model. Encoding at two scales improves predictive performance. Finally, we compute RNA structure scores on three 30 nt sliding structure windows that span the width of a typical 28 nt footprint. These windows start 17, 16, and 15 nucleotides before the start of the A site.

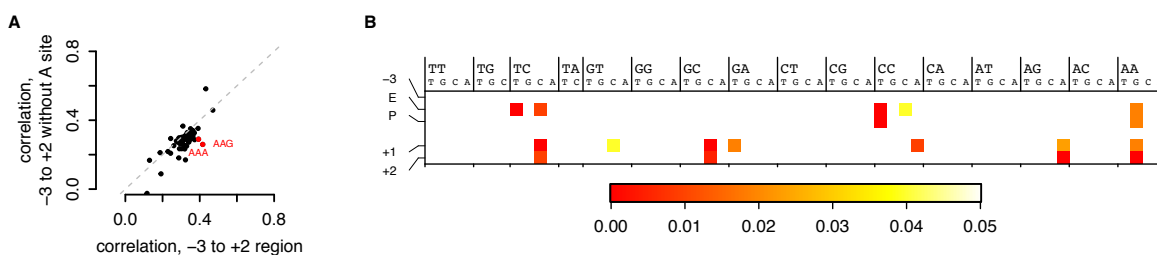
(B) These features are concatenated in a vector, which is used as the input to a fully connected feedforward neural network model. Each model in this paper contains one hidden layer with 200 hidden units, and a tanh activation function on the hidden units. The output layer contains one unit with a ReLU activation function to enforce nonnegativity of predicted scaled counts.



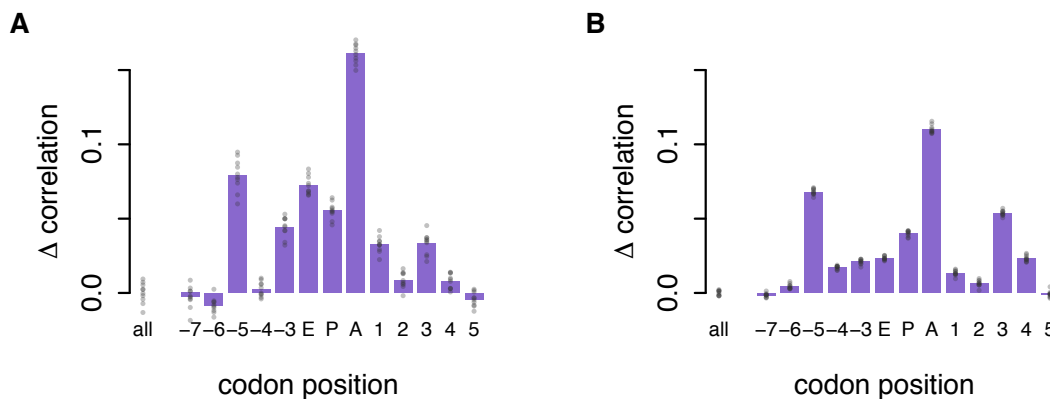
Supplementary Figure 2 Change in MSE upon including an individual mRNA structure feature computed from a 30 nucleotide window. The base model uses a sequence neighborhood from codons -7 to +5 as input features. The greatest improvement in MSE is achieved for a window starting at nucleotide position -17 and ending at position 12. This is roughly coterminal with a typical 28 nt. ribosome footprint (nucleotide positions -15 to +12).

Liu V_0	Liu V_1	Liu V_2	Liu V_3	Liu V_4	Liu V_5	Liu V_6	Liu V_7	Tunney
0.39	0.48	0.51	0.50	0.50	0.50	0.50	0.47	0.56

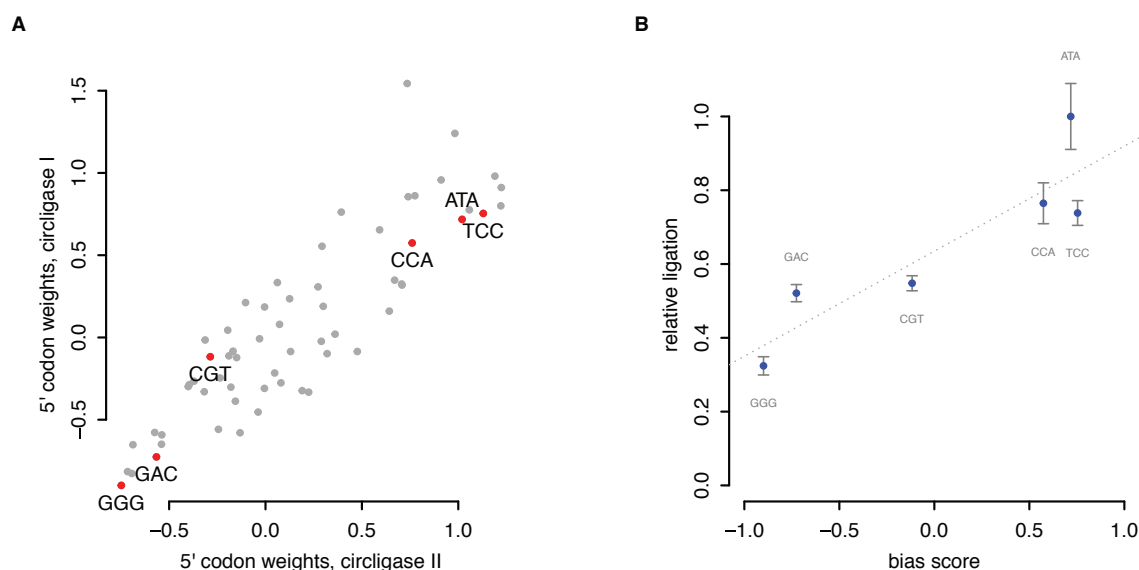
Supplementary Figure 3 Average per gene correlations between ribosome footprint counts (Weinberg et al. 2016) and predictions of these counts. Liu V_0 - V_7 , performance of Riboshape (Liu and Song, 2016), shown as average correlations per gene between denoised ribosome footprint data and predictions of that denoised data. Riboshape projects data into 8 subspaces of a Debauchies-8 basis for wavelet analysis. V_0 is the lowest resolution projection (most smoothed), and V_7 is the closest approximation of the raw data. Tunney, performance of $I\chi$ nos, shown as the average correlation per gene between true ribosome footprint data and predictions of that data. All results are reported on 1711 yeast genes, taking an intersection between genes used in the 'chxdata.mat' file published with riboshape, genes that passed filtering in our RUST analysis, and genes in our yeast transcriptome, excluding the $I\chi$ nos training set.



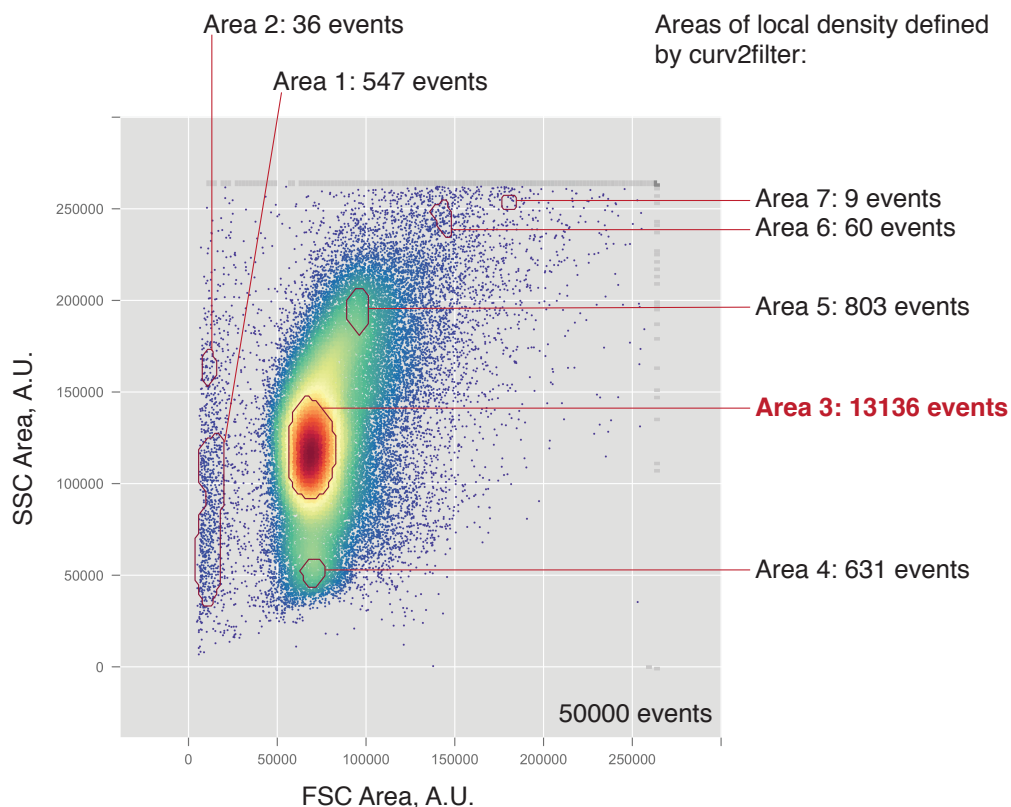
Supplementary Figure 4 Relative contributions of A site codons and context. **(A)** Pearson correlation of observed vs. predicted scaled counts per codon, for a model using codons -3 to +2 and associated nucleotides (x -axis) and a model using the same region but without the A site (y -axis). Codons whose inclusion in the model leads to significantly better prediction (higher correlation between the observed and predicted scaled counts), per a t -test of the Fisher transformation of correlations with an FDR of 5%, are shown in red. **(B)** We extracted the sequence context (codons -3 to +2) for all positions where the squared error was higher for an A-site-only model than for a model with codons -3 to +2 but no A site. The proportion of codons at each position in this set was compared to the overall distribution of codons in the test set with a two-sided proportion test, using an FDR of 5%.



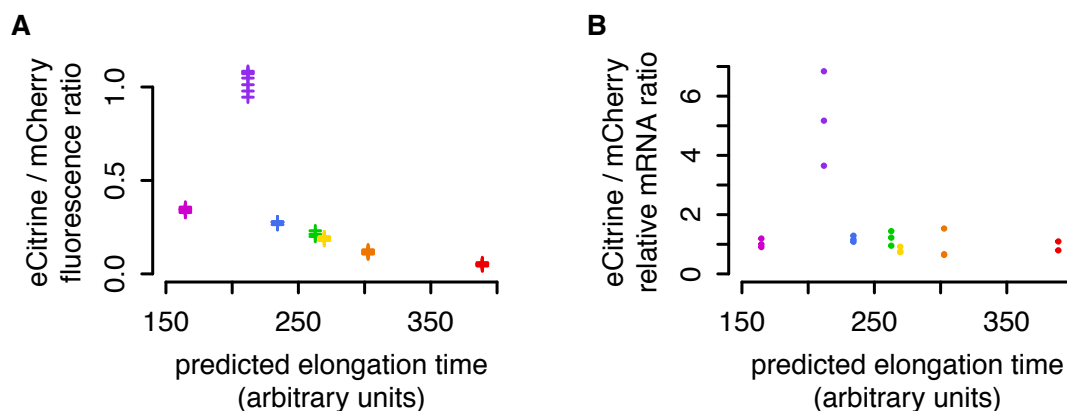
Supplementary Figure 5 Predictive value of codon positions in **(A)** a human ribosome profiling data set using Circligase II (Iwasaki *et al.*, 2016), and **(B)** a yeast ribosome profiling dataset using Circligase I (Schuller *et al.*, 2017). We train a reference model on codons -7 to +5 (with nucleotide features over the same neighborhood), and then a series of leave-one-out models each excluding exactly one codon in the sequence neighborhood, along with the corresponding nucleotides. For each model, we compute Pearson correlations between the true and predicted scaled counts over all codons in the test set. Shown is the difference in Pearson correlations between the reference model and the leave-one-out models. Higher Δr indicates increased importance of that codon position to model predictions.



Supplementary Figure 6 (A) Mean contributions to scaled counts at the 5' end of a ribosome footprint, for yeast data sets generated with CircLigase II (our data) and CircLigase I (Schuller *et al.*, 2017). Scores are from the -5 codon position. To generate these scores, we trained models only on 28 nt. footprints with their 5' end aligning with the beginning of the -5 codon. **(B)** Ligation efficiency of CircLigase I enzyme, as in Fig. 3G.



Supplementary Figure 7 Flow cytometry gating strategy. Scatter plot of forward scatter area (FSC, arbitrary units) against side scatter area (SSC, arbitrary units) for each of the 50000 events collected for a representative flow cytometry sample of diploid yeast expressing mCherry and a differentially optimized eCitrine. Events are coloured by their density on the plot, low density points being coloured blue moving to high density events being coloured dark red. Events outside the plotted area are denoted by grey lines at the edge of the plot. Annotated by red regions are seven areas of high local density defined by the curv2filter method, each with the number of events they contain. For each sample, events within the most populous region were taken forward for further analysis; in this representative sample, these would be the events within Area 3.



Supplementary Figure 8 (A) eCitricine:mCherry fluorescence ratio as in Fig. 3C, including this ratio for the yECitrine sequence (magenta). **(B)** eCitricine:mCherry mRNA ratio measured by qPCR in biological replicates of four strains (colors as in Fig. 3). Each data point represents the ratio of medians of three technical replicates, normalized to the median ratio of the highest expression strain.

2.7 Methods

Code availability

All *iXnos* software and analysis scripts, including a complete workflow of analyses in this paper, can be found at <https://github.com/lareaulab/iXnos>.

Ribosome profiling

Yeast ribosome profiling was performed exactly according to McGlincy and Ingolia[34] with the following modifications:

250 mL of YEPD media was inoculated from an overnight culture of BY474 to an OD600 of 0.1. Yeast were grown to mid-log phase and harvested at an OD600 of 0.565. Lysis proceeded according to McGlincy and Ingolia[34] except with no cycloheximide in the lysis buffer (20 mM Tris pH 7.4, 150 mM NaCl, 5 mM MgCl₂, 1 mM DTT, 1% v/v Triton X-1000, 25 U/ml Turbo DNase I). To quantify RNA content of the lysate, total RNA was purified from 200 L of lysate using the Direct-zol RNA MiniPrep kit (Zymo Research) and the concentration of RNA was measured with a NanoDrop 2000 spectrophotometer (ThermoFisher).

Lysate containing 30 g of total RNA was thawed on ice and diluted to 200 L with polysome buffer with no cycloheximide (20 mM Tris pH 7.4, 150 mM NaCl, 5 mM MgCl₂, 1 mM DTT). 0.1 l (1 U) of RNase I (Epicentre) was added to the diluted cell lysate and then incubated at room temperature for 45 minutes. Digestion and monosome isolation proceeded according to McGlincy and Ingolia[34], except with no cycloheximide in the sucrose cushion.

Purified RNA was separated on a 15% TBE/Urea gel, and fragments of 18-34 nt were gel extracted. Size was determined relative to RNA size markers NI-NI-800 and NI-NI-801[34] and NEB microRNA size marker (New England Biolabs). Library preparation proceeded according to McGlincy and Ingolia[34]. The library was made with downstream linker NI-NI-811 (/5Phos/NNNNNAGCTAAGATCGGAAGAGCACACGTCTGAA/3ddC/) and a modified RT primer with a preferred CircLigase II substrate (AG) at the 5' end (oLFL075, 5'-/5Phos/AGATCGGAAGAGCGTCGTGTAGGGAAAGAG/iSp18/GTGACTGGAGTTCAGACGTGTGCTC). Library amplification PCR used primers NI-NI-798 and NI-NI-825 (Illumina index ACAGTG). The resulting library was sequenced as single-end 51 nt reads on an Illumina HiSeq4000 according to the manufacturers protocol by the Vincent J. Coates Genomics Sequencing Laboratory at the University of California, Berkeley.

Sequencing data processing and mapping

A custom yeast transcriptome file was generated based on all chromosomal ORF coding sequences in `orf_coding.fasta` from the *Saccharomyces Genome Database* genome annotation R64-2-1 for reference genome version R64-1-1

(UCSC `sacCer3`) for *Saccharomyces cerevisiae* strain S288C. A human transcriptome file was generated from `GRCh38.p2`, `Gencode v. 22`, to include one transcript per gene based on the ENSEMBL canonical transcript tag. For both human and yeast, the transcriptome file included 13 nt of 5' UTR sequence and 10 nt of 3' UTR sequence to accommodate footprint reads from ribosomes at the first and last codons. For yeast transcripts with no annotated UTR, the flanking genomic sequence was included. For human transcripts with no annotated UTR, or UTRs shorter than 13 or 10 nt, the sequence was padded with N. Yeast ribosome profiling reads from Weinberg et al.[54] (SRR1049521) were trimmed to remove the ligated 3' linker (TCGTATGCCGTCTTCTGCTTG) off of any read that ended with any prefix of that string, and to remove 8 random nucleotides at the 5' end (added as part of the 5' linker). Yeast ribosome profiling reads generated in our own experiments (GEO accession GSE106572) were trimmed to remove the ligated 3' linker, which included 5 random nucleotides and a 5-nt index of AGCTA (NNNNNIIIIAGATCGGAAGAGCA-CACGTCTGAAC). Human ribosome profiling reads from Iwasaki et al.[26] (SRR2075925, SRR2075926) were trimmed to remove the ligated 3' linker (CTGTAGGCACCATCAAT). Yeast ribosome profiling reads from Schuller et al.[45] (SRR5008134, SRR5008135) were trimmed to remove the ligated 3' linker (CTGTAGGCACCATCAAT).

Trimmed fastq sequences of longer than 10 nt were aligned to yeast or human ribosomal and noncoding RNA sequences using `bowtie v. 1.2.1.1`[29], with options `bowtie -v 2 -S`. Reads that did not match rRNA or ncRNA were mapped to the transcriptome with options `bowtie -a -norc -v 2 -S`. Mapping weights for multimapping reads were computed using `RSEM v. 1.2.31`[32].

Assignment of A sites

A site codons were identified in each footprint using simple rules for the offset of the A site from the 5' end of the footprint. These rules were based on the length of the footprint and the frame of the 5' terminal nucleotide. For each data set, the set of lengths that included appreciable footprint counts was determined (e.g. Weinberg 27-31 nt.). For each length, the counts of footprints mapping to each frame were computed. The canonical 28 nucleotide footprint starts coherently in frame 0, with the 5' end 15 nt upstream of the A site (citation). For all other lengths, rules were defined if footprints mapped primarily to 1 or 2 frames, and offsets were chosen to be consistent with over digestion or under digestion relative to a 28 nucleotide footprint. Footprints mapping to other frames were discarded.

Scaled counts

For each codon, the raw footprint counts were computed by summing the RSEM mapping weights of each footprint with its A site at that codon. Scaled footprint counts were computed by dividing the raw counts at each codon by the average raw counts over all codons in its transcript. This controlled for variable initiation rates and copy numbers across transcripts. The resulting scaled counts are mean centered at 1, with scaled counts higher than 1 indicating slower than average translation. The first 20 and last 20 codons in each gene were excluded from all computations and data sets, to avoid the atypical footprint counts observed at the beginning and end of genes.

Genes were excluded from analysis if they had fewer than 200 raw footprint counts in the truncated CDS, or fewer than 100 codons with mapped footprints in this region. Then the top 500 genes were selected by footprint density (average footprint counts/codon). 2/3 of these genes were selected at random as the training set, and the remaining 1/3 of genes were used as the test set.

Input features

The model accepts user defined sets of codon and nucleotide positions around the A site to encode as input features for predicting ribosome density. The A site is indexed as the 0th codon, and its first nucleotide is indexed as the 0th nucleotide, with negative indices in the 5' direction, and positive indices in the 3' direction. Each codon and nucleotide feature is converted to a binary vector via one-hot encoding, and these vectors are concatenated as input into the regression models. The model also accepts RNA folding energies from the RNAfold package, and allows the user to define window sizes and positions to score RNA structure and include as inputs into the regression models. In our final model, codons -5 to +4 and nucleotides -15 to +14 were chosen, as well as folding energies from three 30-nt windows starting at nucleotides -17, -16, and -15.

Model construction

All models were constructed as feedforward artificial neural networks, using the Python packages Lasagne v. 0.2.dev1[3] and Theano v. 0.9.0[51]. Each network contained one fully connected hidden layer of 200 units with a tanh activation function, and an output layer of one unit with a ReLU activation function. Models were trained using mini-batch stochastic gradient descent with Nesterov momentum (batch size 500).

Comparisons to other models

RUST[37] was run via <https://ribogalaxy.ucc.ie/> according to the authors instructions. First, we computed a codon metafootprint on the Weinberg dataset, aligned to the transcriptome as described above. We used an A-site offset of 15 and limited the analysis to 28-nt footprints (the most abundant), in keeping with the authors analysis. Then, we ran the similarity of observed and expected profiles analysis using that codon metafootprint and retrieved the correlation of the observed and expected footprint distribution for each individual gene.

Riboshape[33] was downloaded from <https://sourceforge.net/projects/riboshape/> on 2/1/2018. We generated the riboshape data structure according to the README file, with custom scripts (`process_data.py` and `make_data_structure.m`, available on GitHub), on our processed footprint counts data from the Weinberg dataset. We restricted the analysis to the 2170 genes present in both our transcriptome and the `chx-data.mat` data structure that is shipped with riboshape. We binned our genes by truncated lengths 100-210, 211-460, 461-710, 711-960, and 961-4871, which matched the bins in Liu and Song after accounting for our 20 codon truncation regions at either end of genes. Then we trained riboshape models on these bins, using parameters of 1, 3, 5, 12.5, 25, 37.5, 50, and 75. We report the per gene correlations between the true footprint data and their regression fits (waveforms) in their wavelet decomposition subspace with the least amount of denoising. The values in this subspace are closest to the observed footprint data, and their model trained for this subspace performs the best at predicting observed footprint density. We also report for each subspace the correlation between their denoised footprint data and the regression fits in that subspace. The prior is more directly comparable to our work.

Feature importance measurements

A series of leave-one-out models was trained, excluding one codon position at a time from the sequence neighborhood. The importance of each codon position to predictive performance was computed as the difference in MSE between the reduced and full models. The contribution of codon c at position i to predicted scaled counts was calculated as the average increase in predicted scaled counts due to having that codon at that position, over all instances where codon c was observed at position i in the test set. This increase was computed relative to the expected predicted scaled counts when the codon at position i was varied according to its empirical frequency in the test set (Supplementary Materials).

Sequence optimization

The overall translation time of a coding sequence was computed as the sum of the predicted scaled counts over all codons in that coding sequence. This quantity corresponds to total translation time in arbitrary units. A dynamic programming algorithm was developed to find the fastest and slowest translated coding sequences in the set of synonymous coding sequences for a given protein, under a predictive model of scaled counts.

This algorithm was used to determine the fastest and slowest translating coding sequences for eCitrine, under a predictive model using a sequence window from codons -3 to +2, and using no structure features. Then 100,000 synonymous coding sequences for eCitrine were generated by selecting a synonymous codon uniformly at random for each amino acid. These coding sequences were scored, and the sequences at the 0th, 33rd, 67th, and 100th percentiles were selected for expression experiments.

Measuring circularization efficiency

We designed oligonucleotides that mimic the structure of the single-stranded cDNA molecule that is circularized by CircLigase during the McGlincy and Ingolia (2017) ribosome profiling protocol. These oligonucleotides have the structure: /5Phos/AGATCGGAAG-AGCGTCGTGTAGGGAAAGAG/iSp18GTGACTGGAGTTCAGACGTGTGCTCTTCC-GATCACAGTCATCGTTCGCATTACCCTGTTATCCCTAAJJJ, where /5Phos/ indicates a 5' phosphorylation; /iSp18/ indicates an 18-atom hexa-ethyleneglycol spacer; and JJJ indicates the reverse complement of the nucleotides at the 5' of the footprint favored or disfavored under the model (oligos defined in Supp. Table 2). Circularization reactions were performed using CircLigase I or II (Epicentre) as described in the manufacturers instructions, using 1 pmol oligonucleotide in each reaction. Circularization reactions were diluted 1/20 before being subjected to qPCR using DyNAmo HS SYBR Green qPCR Kit (Thermo Scientific) on a CFX96 Touch Real Time PCR Detection System (Biorad). For each circularization reaction, two qPCR reactions were performed: one where the formation of a product was dependent upon oligo circularization, and one where it was not (oligos defined in Supp. Table 2). qPCR data were analyzed using custom R scripts whose core functionality is based on the packages `qpcR`[43] and `dpcR`[7] (`qpcr_functions.R`, available on github). The signal from the circularization dependent amplicon was normalized to that from the circularization independent amplicon, and then expressed as a fold-change compared to the mean of the oligonucleotide with the most favored sequence under the model.

Plasmid and yeast strain construction

Yeast integrating plasmids expressing either mCherry or a differentially optimized version of eCitrine were constructed. The differentially optimized versions of eCitrine were synthesized as gBlocks by Integrated DNA Technologies inserted into the plasmid backbone by Gibson assembly[16]. Transcription of both mCherry and eCitrine is directed by a PGK1 promoter and an ADH1 terminator. To enable yeast transformants to grow in the absence of leucine, the plasmids contain the LEU2 expression cassette from *Kluyveromyces lactis* taken from pUG73[18], which was obtained from EUROSCARF. To enable integration into the yeast genome, the plasmids contain two 300 bp sequences from the his31 locus of BY4742. Genbank files describing the plasmids are provided in Supp. File 2. To construct yeast strains expressing these plasmids, the plasmids were linearized at the SbfI site and 1 g linearized plasmid was used to transform yeast by the high efficiency lithium acetate/single-stranded carrier DNA/PEG method, as described[11]. Transformants were selected by growth on SCD -LEU plates, and plasmid integration into the genome was confirmed by yeast colony PCR with primers flanking both the upstream and downstream junctions between the plasmid sequence and the genome (oligos defined in Supp. Table 2). PCR was performed using GoTaq DNA polymerase (Promega M8295). Haploid BY4742 and BY4741 strains expressing the eCitrine variants and mCherry, respectively, were then mated. For each eCitrine variant, eight transformants were mated to a single mCherry transformant. Diploids were isolated by their ability to grow on SCD -MET-LYS plates. Strains with sequence-confirmed mutations or copy number variation were excluded from further analysis.

Assessing fluorescent protein expression by flow cytometry

Overnight cultures of diploid yeast in YEPD were diluted in YEPD so that their optical density at 600 nm (OD600) was equal to 0.1 in a 1 mL culture, and then grown for six hours in a 2 mL deep-well plate supplemented with a sterile glass bead, at 30 C with shaking at 250 rpm. This culture was pelleted by five minutes centrifugation at 3000 x g and fixed by resuspension in 16% paraformaldehyde followed by 30 minutes incubation in the dark at room temperature. Cells were washed twice in DPBS (Gibco 14190-44) and stored in DPBS at 4 C until analysis. Upon analysis, cells were diluted ca. 1:4 in DPBS and subject to flow cytometry measurements on a BD Biosciences (San Jose, CA) LSR Fortessa X20 analyzer. Forward Light Scatter measurements (FSC) for relative size, and Side-Scatter measurements (SSC) for intracellular refractive index were made using the 488nm laser. eCitrine fluorescence was measured using the 488 nm (Blue) laser excitation and detected using a 505 nm Long Pass optical filter, followed by 530/30 nm optical filter with a bandwidth of 30nm (530/30, or 515 nm-545 nm). mCherry fluorescence was measured using a 561 nm (yellow-green) laser, for excitation and a 595 nm long-pass optical filter, followed by 610/20 nm band-pass optical filter with a bandwidth of 20 nm

(or 600 nm 620 nm). PMT values for each color channel were adjusted such that the mean of a sample of BY4743 yeast was 100. 50000 events were collected for each sample. Flow cytometry data were analyzed using a custom R script (`gateFlowData.R`, available on github) whose core functionality is based on the Bioconductor packages `flowCore`[20], `flowStats`[21], and `flowViz`[44]. In summary, for each sample, events that had values for red or yellow fluorescence that were less than one had those values set to one. Then, in order to select events that represented normal cells, we used the `curv2filter` method to extract events that had FSC and side-scatter SSC values within the values of the region of highest local density of all events as considered by their FSC and SSC values. For these events the red fluorescence intensity was considered a measure of mCherry protein expression and yellow fluorescence intensity a measure of eCitrine protein expression.

Measuring eCitrine and mCherry mRNA expression by qRT-PCR

Overnight cultures of diploid yeast in YEPD were diluted in YEPD so that their OD600 was equal to 0.1 in a 20 mL culture, and then grown at 30 C with shaking at 250 rpm until their OD600 reached 0.4 - 0.6. 10 mL of culture was then pelleted by centrifugation for 5 minutes at 3000 x g and snap frozen in liquid nitrogen. Total RNA was extracted from pelleted yeast cultures according to the method of Ares[1]. Thereafter, 10 g of this RNA was treated with Turbo DNase I (ambion) according to the manufacturer's instructions, then 1 g DNase treated RNA was reverse transcribed using anchored oligo dT and Protoscript II (NEB) according to the manufacturer's instructions. 1/20th of this reaction was then subjected to qPCR using the DyNAmo HS SYBR Green qPCR Kit (Thermo Scientific) on a CFX96 Touch Real Time PCR Detection System (Biorad). For each reverse transcription reaction, two qPCR reactions were performed: one with primers specific to the mCherry ORF, and one with primers specific to the eCitrine variant ORF in question (oligos defined in Supp. Table 2). qPCR data were analyzed using custom R scripts whose core functionality is based on the packages `qpcr`[1, 43] and `dpcr`[7] (`qpcr_functions.R`, available on github). The signal from each eCitrine variant ORF was normalized to that from the mCherry ORF in the same sample, and then expressed as a fold-change compared to the median of these values for the MIN (fastest predicted sequence) eCitrine variant.

Data availability

Ribosome profiling sequence data generated for this study have been deposited in NCBI GEO as accession GSE106572. All analyzed data used to create figures are available at <https://github.com/lareaulab/iXnos>. Source data for figure 2B, the scores of each codon, are also provided as a supplementary file.

Chapter 3

Bias Correction of Ribosome Profiling Data

3.1 Background

In the previous chapter, we observed that the sequences at the 5' and 3' termini of a ribosome footprint are important predictors of the density of footprints about a given codon. In some experiments these regions are on par with the A site codon as predictive features (Ch. 2, Figs. 2.3.a, 2.3.d, Supplementary Fig. 5). Several analyses have observed nonuniform sequence composition at the ends of ribosome footprints, with enrichment and depletion of certain terminal sequences around the expected background transcriptional levels[37, 54]. These observations are consistent with experimental work showing that the sequence preferences of ligase enzymes can bias the recovery of genetic material[28, 19]. Taken together, this suggests substantial bias in the recovery of ribosome footprints as a function of enzymatic steps in the experimental protocol. This bias acts as a filter between the true distribution of ribosome footprints across genes, which we would like to obtain, and the distribution of footprints that we actually observe in a sequencing library. Many well developed methods exist for correcting different types of biases arising in RNA-seq experiments, but these methods make assumptions about the biological processes generating RNA fragments that are not applicable to ribosome profiling experiments. In particular, in ribosome profiling experiments we should not assume that footprints are generated uniformly over transcripts - in many analyses of this data we are explicitly interested in variation in ribosome density across transcripts. While ligation bias in ribosome footprints has been observed a number of times, to our knowledge there is no existing bias correction method that is suitable for this experiment.

In this chapter, we develop a probabilistic model to describe the combined biological and experimental processes that generate ribosome profiling data. This allows us to infer the biological distribution of ribosomes across transcripts and also to quantify the contri-

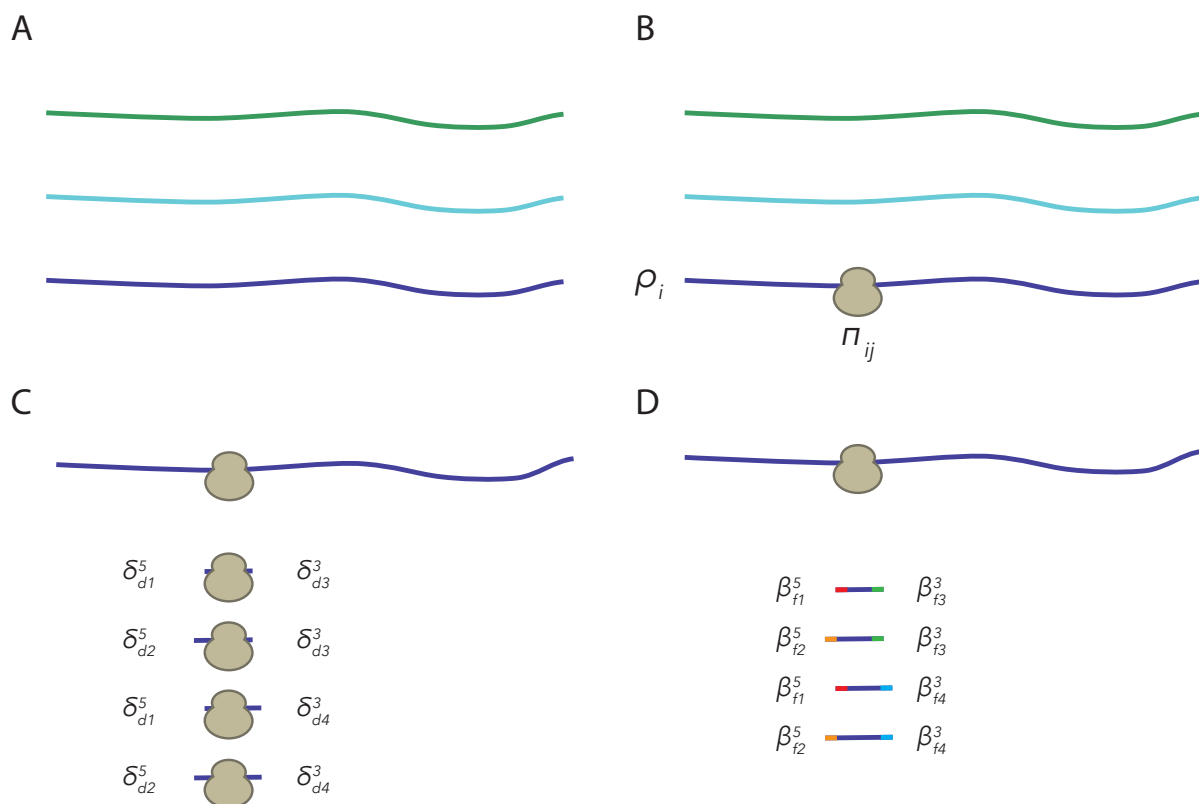


Figure 3.1: Generative model for ribosome profiling data **A** Representation of the transcriptome, containing transcripts of three types (green, blue, purple). **B** Biological distribution of ribosomes across transcripts. First, choose a gene i to generate a transcript according to probability vector ρ . Second chose an A site codon j to generate a footprint according to probability vector π_i . **C** RNase digestion can generate a set of digested footprints around an A site due to variability in the length of the 5' and 3' ends. Choose a 5' and 3' digestion length according to probability vectors δ^5 and δ^3 . **D** The efficiency of ligation steps in sequencing library preparation varies with the sequence observed at fragment ends. Bias weights β^5 and β^3 are proportional to the ligation probabilities of a footprint end, based on their terminal sequence.

butions of technical artifacts to the observed data. We demonstrate consistent recovery biases across experiments for a given ligation protocol. We also show how this method enables us to correct for ligation biases in ribosome profiling data.

3.2 Assumptions

Our first task is to specify a process that generates ribosome profiling data. This requires us to make assumptions about the important biological and experimental steps that influence the distributions of observed data. To start, we want to represent the biological distribution of ribosomes across transcripts. For the purposes of analyzing ribosome profiling data, we are typically interested in this distribution at two levels. The first level of interest is the overall count of ribosomes on mRNA transcripts of each gene, or the corresponding count corrected by gene length. This gives us a measurement of how much a given gene is being expressed. These counts differ between genes as a function of variable mRNA copy numbers and translation initiation rates. The second level of interest is the distribution of ribosomes within an individual gene. These distributions are observed to be nonuniform, and it is thought that this nonuniformity reflects variation in translation elongation rates that create regulatory opportunities and can induce selective pressure on the coding sequences of genes. In order to precisely define the position of a footprint on a transcript, we use the A site assigned in that footprint, as we did in Chapter 2. Taken together, our biological knowledge suggests that each gene and each codon within a gene can have a unique probability of generating a ribosome footprint. Consequently we choose to represent the biological distribution of footprints across transcripts with a multinomial distribution, where each codon in each gene has a unique parameter representing its probability of generating a footprint.

The biological component of our model is represented in Figures 3.1.a and 3.1.b. We imagine a set of transcripts (green, blue, purple) each with a distinct probability of generating a ribosome footprint. We call this vector of probabilities ρ . Conditional upon choosing gene i , we then choose a codon within that gene to serve as the A site for generating our footprint. We refer to the vector of probabilities for choosing each codon in gene i as π_i . Taken together, ρ and π give us the probability of generating a ribosome footprint at each A site codon in our transcriptome. We could equivalently conceive of all the codons in our transcriptome as generating footprints according to one large multinomial distribution, but it is convenient to factor this distribution into ρ and π for both analysis and numerical concerns.

The remainder of the model comprises the steps in the experimental protocol that can introduce technical bias into the observed distribution of ribosome footprints. It is well known that ligations interacting with the ends of footprints are important steps in this category. We also model the RNase digestion step in the ribosome profiling protocol, because this is an enzymatic step that interacts with fragment ends. The digestion step is represented in Figure 3.1.c. Once we have chosen an A site codon to generate a footprint, the next step is to choose 5' and 3' ends for the footprint around that codon. We define a digest length as the number of nucleotides between a fragment end and the A site, not including A site nucleotides. The RNase digestion is generally quite efficient and

accurate in digesting footprints to a consistent length, but there is some variability in the distance between the A site and each end of the footprint. This variability is at or below the width of one codon at either end for nearly all footprints. In our model, we choose ranges of legal 5' and 3' digest lengths comprising most of the observed footprint data in a given experiment. It is necessary that one of these ranges comprises no more than three consecutive digest lengths, in order to unambiguously assign an A site codon. Once we have chosen these ranges for an experiment, the parameter δ^5 is a vector representing the probabilities of digesting the 5' end to each length in the legal range. δ^3 is defined similarly.

Finally, in order to observe a read in our experiment, it must successfully pass through the sequencing library protocol and yield a sequenced read in our data file. There are a number of factors and steps in the experimental protocol that may introduce recovery bias into our data, including ligation bias, RNA structure, and PCR bias and drop off. In developing this model, we restrict ourselves to considering ligation bias, which is a well characterized source of bias in the data. There are several strategies to add sequence onto the 5' and 3' ends of ribosome footprints in order to prepare a sequencing library, with most including some combination of RNA and ssDNA ligation events. These extension events must succeed at both ends in order to successfully recover the footprint as data. We consider that ligation events may have a variable probability of success as a function of the sequence environment at the end of footprints, and we maintain generality in our model with respect to the length of the relevant bias region at the end of reads. We define a length in nucleotides for our bias region at each end, ν^5 and ν^3 . Then we define a parameter, γ^5 , containing the ligation probability at the 5' end for each permutation of length ν^5 over the genomic alphabet (A, C, G, T/U). We define a parameter γ^3 similarly. For the model training algorithm that follows, it is convenient to not find γ^5 and γ^3 outright, but rather β^5 and β^3 , which are rescalings of the elements of the gamma vectors by a constant multiple such that their elements sum to 1 (or another arbitrary constant). Thus the elements of β are proportional to the ligation probabilities at a given end for a given terminal sequence, with some common scaling factor for each end. We refer to these beta terms as recovery bias weights, because they are not proper probabilities. In Figure 3.1.d, we can see that for footprints situated at a given A site codon, the sequence environment encountered by a ligase varies as a function of the digest length at either end of the footprint. Thus the set of footprints digested about a given A site codon can have different probabilities of successful double ligation and passage into a sequencing library. Having specified the general assumptions we make about the process that generates ribosome profiling data, we now define a formal notation and vocabulary for this model.

3.3 Model Definitions

General definitions

T = # transcripts in transcriptome

i = index for transcripts

C_i = length of transcript i CDS in codons

j = index for codons in a transcript

N = # of reads sequenced in experiment

n = index for sequenced reads

d_{max}^5 = maximum legal 5' digest length for a footprint

d_{min}^5 = minimum legal 5' digest length for a footprint

k, p = index for 5' digest length

d_{max}^3 = maximum legal 3' digest length for a footprint

d_{min}^3 = minimum legal 3' digest length for a footprint

l, q = index for 3' digest length

Read mappings

t = index of transcript of origin for a footprint

c = index of A site codon for a footprint

d^5 = 5' digest length for a footprint

d^3 = 3' digest length for a footprint

s_n = sequence of footprint n

$s(t, c, d^5, d^3)$ = function that returns sequence of a mapping

$\simeq$, similarity function, compares footprints to mapping sequences in transcriptome

$$s_n \simeq s(t, c, d^5, d^3) = \begin{cases} \text{True,} & \text{if } s_n \text{ maps to } (t, c, d^5, d^3) \\ \text{False,} & \text{otherwise} \end{cases}$$

$$x_n = (t_n, c_n, d_n^5, d_n^3)$$

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}$$

Bias regions

$$A = \{A, C, G, T\}$$

ν^5 = length of 5' bias region in nucleotides

ν^3 = length of 3' bias region in nucleotides

$f^5(t, c, d^5)$ = function that returns the sequence of the 5' bias region of a footprint

$f^3(t, c, d^3)$ = function that returns the sequence of the 3' bias region of a footprint

f_n^5 = 5' bias region of nth footprint

f_n^3 = 3' bias region of nth footprint

Additional parameters

$$\gamma_f^5 = P(5' \text{ ligation} | f^5 = f)$$

$$\Gamma^5 = \sum_{f \in A^{\nu^5}} \gamma_f^5$$

$$\gamma_f^3 = P(3' \text{ ligation} | f^3 = f)$$

$$\Gamma^3 = \sum_{f \in A^{\nu^3}} \gamma_f^3$$

$$\tilde{\rho}_i = P(t = i)$$

$$\forall i \in 1 \dots T$$

$$\tilde{\pi}_{ij} = P(c = j | t = i)$$

$$\forall i \in 1 \dots T \quad \forall j \in 1 \dots C_i$$

Model parameters

$$\begin{aligned}
\rho_i &= P(t = i \mid \text{footprint observed}) & \forall i \in 1 \dots T \\
\pi_{ij} &= P(c = j \mid t = i, \text{footprint observed}) & \forall i \in 1 \dots T \quad \forall j \in 1 \dots C_i \\
\delta_k^5 &= P(d^5 = k) & \forall k \in d_{min}^5 \dots d_{max}^5 \\
\delta_k^3 &= P(d^3 = k) & \forall k \in d_{min}^3 \dots d_{max}^3 \\
\beta_f^5 &= \text{bias weight for sequence } f \text{ in } 5' \text{ bias region} & \forall f \in A^{\nu^5} \\
&= \frac{\gamma_f^5}{\Gamma^5} \propto P(5' \text{ ligation} \mid f^5 = f) \\
\beta_f^3 &= \text{bias weight for sequence } f \text{ in } 3' \text{ bias region} & \forall f \in A^{\nu^3} \\
&= \frac{\gamma_f^3}{\Gamma^3} \propto P(3' \text{ ligation} \mid f^3 = f) \\
\Theta &= (\rho, \pi, \delta^5, \delta^3, \beta^5, \beta^3) \\
\zeta_{ijkl}^n &= P(t_n = i, c_n = j, d^5 = k, d^3 = l \mid s_n, \Theta) & \forall n \in 1 \dots N \quad \forall i \in 1 \dots T \\
& & \forall j \in 1 \dots C_i \\
& & \forall k \in d_{min}^5 \dots d_{max}^5 \quad \forall l \in d_{min}^3 \dots d_{max}^3
\end{aligned}$$

3.4 Handling Missing Data

In order to estimate the parameters outlined above, we need to address a challenge of incomplete data. Our model assumes that some fraction of footprints do not pass through the sequencing library protocol due to unsuccessful ligation. Consequently, we believe that our recovered footprints represent only one component of the biological distribution of footprints, and that there is another component of this distribution that we cannot directly observe. This creates a challenge for estimating ligation probabilities (i.e. γ , β). If we define the simplest likelihood function under our model specification, we discover that we cannot compute ligation probabilities (γ), because we have no knowledge about the reads that we do not observe.

Let $o_n = \mathbb{1}(\text{fp. n observed})$

Let $N = \text{Number of ribosome footprints before ligation}$

$$\begin{aligned}\mathcal{L}(X; \Theta) &= \prod_{n=1}^N \tilde{\rho}_{t_n} \tilde{\pi}_{t_n c_n} \delta_{d_n^5}^5 \delta_{d_n^3}^3 \left[\gamma_{f^5(t_n, c_n, d_n^5)}^5 \gamma_{f^3(t_n, c_n, d_n^3)}^3 \right]^{o_n} \left[1 - \gamma_{f^5(t_n, c_n, d_n^5)}^5 \gamma_{f^3(t_n, c_n, d_n^3)}^3 \right]^{1-o_n} \\ \frac{\partial \log \mathcal{L}(X; \Theta)}{\partial \gamma_f^5} &= \sum_{n=1}^N \sum_{i=1}^T \sum_{j=1}^{C_i} \sum_{k=d_{min}^5}^{d_{max}^5} \mathbb{1}(f^5(i, j, k) = f) \left[o_n \frac{1}{\gamma_f^5} + (1 - o_n) \frac{-\gamma_{f^3(i, j, l)}^3}{1 - \gamma_f^5 \gamma_{f^3(i, j, l)}^3} \right] \\ \hat{\gamma}_f^5 &= \frac{\sum_{n=1}^N \sum_{i=1}^T \sum_{j=1}^{C_i} \sum_{k=d_{min}^5}^{d_{max}^5} \mathbb{1}(f^5(i, j, k) = f) o_n}{\sum_{n'=1}^N \sum_{i'=1}^T \sum_{j'=1}^{C_{i'}} \sum_{k'=d_{min}^5}^{d_{max}^5} \mathbb{1}(f^5(i', j', k') = f) \gamma_{f^3(i', j', l')}^3}\end{aligned}$$

The estimator for $\hat{\gamma}_f^5$ above can be interpreted as the count of footprints with a 5' bias region of f and successful ligations at both ends, divided by the count of footprints with a 5' bias region of f and successful ligations at the 3' end (irrespective of 5' ligation status). This estimator is intuitive, but it is not possible to compute, because we don't know the count of ribosome footprints before our ligation step. We could attempt to quantify the amount of footprint RNA contained in a sample at various steps in the experimental protocol. These types of experiments could provide useful validation in future work for some of the theory that we develop here.

In addition, there is an alternate computational approach to handle the problem of missing data. We can alter the model formulation presented in the likelihood function above to condition each step on the fact that each footprint in our data set has successfully been observed. This gives us our model outlined in Figure 3.1, and presented in the likelihood function below, where ρ and π , are the probabilities of generating a footprint from a gene and codon, respectively, conditional on the fact that this footprint is observed. The remainder of the model is represented by the last term in our likelihood function. This quotient is the probability of generating a specific digested footprint out of the set of possible digested footprints around an A site codon, conditional on the fact that some footprint is generated about that codon. When we formulate the problem in this way, we only need data from observed footprints. We will see that we can use our trained model parameters to compute most of our quantities of interest, including the unconditional analogs of ρ and π , $\tilde{\rho}$ and $\tilde{\pi}$.

3.5 Model Training

Likelihood model

$$\begin{aligned}
\mathcal{L}(X; \Theta) &= \prod_{n=1}^N \rho_{t_n} \pi_{t_n c_n} \frac{\beta_{f^5(t_n, c_n, d_n^5)}^5 \delta_{d_n^5}^5 \beta_{f^3(t_n, c_n, d_n^3)}^3 \delta_{d_n^3}^3}{\sum_{k=d_{min}^5}^{d_{max}^5} \sum_{l=d_{min}^3}^{d_{max}^3} \beta_{f^5(t_n, c_n, k)}^5 \delta_k^5 \beta_{f^3(t_n, c_n, l)}^3 \delta_l^3} \\
\log \mathcal{L}(X; \Theta) &= \sum_{n=1}^N \left[\log \rho_{t_n} + \log \pi_{t_n c_n} + \log \beta_{f^5(t_n, c_n, d_n^5)}^5 + \log \delta_{d_n^5}^5 \right. \\
&\quad \left. + \log \beta_{f^3(t_n, c_n, d_n^3)}^3 + \log \delta_{d_n^3}^3 - \log \sum_{k=d_{min}^5}^{d_{max}^5} \sum_{l=d_{min}^3}^{d_{max}^3} \beta_{f^5(t_n, c_n, k)}^5 \delta_k^5 \beta_{f^3(t_n, c_n, l)}^3 \delta_l^3 \right] \\
\mathbb{E}_{\Theta}[\log \mathcal{L}(X; \Theta)] &= \sum_{n=1}^N \sum_{i=1}^T \sum_{j=1}^{C_i} \sum_{k=d_{min}^5}^{d_{max}^5} \sum_{l=d_{min}^3}^{d_{max}^3} \zeta_{ijkl}^n \left[\log \rho_i + \log \pi_{ij} \right. \\
&\quad \left. + \log \beta_{f^5(i, j, k)}^5 + \log \delta_k^5 + \log \beta_{f^3(i, j, l)}^3 + \log \delta_{d_n^3}^3 \right. \\
&\quad \left. - \log \sum_{p=d_{min}^5}^{d_{max}^5} \sum_{q=d_{min}^3}^{d_{max}^3} \beta_{f^5(i, j, p)}^5 \delta_p^5 \beta_{f^3(i, j, q)}^3 \delta_q^3 \right]
\end{aligned}$$

Observe that we have replaced γ with β in our likelihood function. Under this formulation of the model, one consequence of the quotient term is that we could linearly rescale the elements of γ up or down by a constant with no effect on the likelihood function. We handle this by defining the β parameter vectors as a rescaling of the γ parameter vectors such that their elements sum to an arbitrary constant (1). This constraint allows us to identify maximum likelihood parameter estimates below.

Model constraints

$$\begin{aligned}
\sum_{i=1}^T \rho_i &= 1 \\
\sum_{j=1}^{C_i} \pi_{ij} &= 1 & \forall i \in 1 \dots T \\
\sum_{k=d_{min}^5}^{d_{max}^5} \delta_k^5 &= 1 & \sum_{l=d_{min}^3}^{d_{max}^3} \delta_l^3 = 1 \\
\sum_{f \in A^{v^5}} \beta_f^5 &= 1 & \sum_{f \in A^{v^3}} \beta_f^3 = 1
\end{aligned}$$

Model training

$$\begin{aligned}
\Lambda = \mathbb{E}_{\Theta}[\log \mathcal{L}(X; \Theta)] &+ \lambda_{\rho} \left(1 - \sum_{i=1}^T \rho_i\right) + \sum_{i=1}^T \lambda_{\pi_i} \left(1 - \sum_{j=1}^{C_i} \pi_{ij}\right) \\
&+ \lambda_{\delta^5} \left(1 - \sum_{k=d_{min}^5}^{d_{max}^5} \delta_k^5\right) + \lambda_{\delta^3} \left(1 - \sum_{l=d_{min}^3}^{d_{max}^3} \delta_l^3\right) \\
&+ \lambda_{\beta^5} \left(1 - \sum_{f \in A^{v^5}} \beta_f^5\right) + \lambda_{\beta^3} \left(1 - \sum_{f \in A^{v^3}} \beta_f^3\right)
\end{aligned}$$

Rho update

$$\begin{aligned}
\frac{\partial \Lambda}{\partial \rho_i} &= \sum_{n=1}^N \sum_{j=1}^{C_i} \sum_{k=d_{min}^5}^{d_{max}^5} \sum_{l=d_{min}^3}^{d_{max}^3} \frac{\zeta_{ijkl}^n}{\hat{\rho}_i} - \lambda_\rho = 0 & \forall i \in 1 \dots T \\
\sum_{n=1}^N \sum_{i=1}^T \sum_{j=1}^{C_i} \sum_{k=d_{min}^5}^{d_{max}^5} \sum_{l=d_{min}^3}^{d_{max}^3} \zeta_{ijkl}^n &= \sum_{i=1}^T \hat{\rho}_i \lambda_\rho \\
N &= \lambda_\rho \\
\sum_{n=1}^N \sum_{j=1}^{C_i} \sum_{k=d_{min}^5}^{d_{max}^5} \sum_{l=d_{min}^3}^{d_{max}^3} \frac{\zeta_{ijkl}^n}{N} &= \hat{\rho}_i & \forall i \in 1 \dots T
\end{aligned}$$

Pi update

$$\begin{aligned}
\frac{\partial \Lambda}{\partial \pi_{ij}} &= \sum_{n=1}^N \sum_{k=d_{min}^5}^{d_{max}^5} \sum_{l=d_{min}^3}^{d_{max}^3} \frac{\zeta_{ijkl}^n}{\hat{\pi}_{ij}} - \lambda_{\pi_i} = 0 & \forall i \in 1 \dots T \quad \forall j \in 1 \dots C_i \\
\sum_{n=1}^N \sum_{j=1}^{C_i} \sum_{k=d_{min}^5}^{d_{max}^5} \sum_{l=d_{min}^3}^{d_{max}^3} \zeta_{ijkl}^n &= \sum_{j=1}^{C_i} \hat{\pi}_{ij} \lambda_{\pi_i} & \forall i \in 1 \dots T \\
\sum_{n=1}^N \sum_{j=1}^{C_i} \sum_{k=d_{min}^5}^{d_{max}^5} \sum_{l=d_{min}^3}^{d_{max}^3} \zeta_{ijkl}^n &= \lambda_{\pi_i} & \forall i \in 1 \dots T \\
\frac{\sum_{n=1}^N \sum_{k=d_{min}^5}^{d_{max}^5} \sum_{l=d_{min}^3}^{d_{max}^3} \zeta_{ijkl}^n}{\sum_{n=1}^N \sum_{j=1}^{C_i} \sum_{p=d_{min}^5}^{d_{max}^5} \sum_{q=d_{min}^3}^{d_{max}^3} \zeta_{ijpq}^n} &= \hat{\pi}_{ij} & \forall i \in 1 \dots T \quad \forall j \in 1 \dots C_i
\end{aligned}$$

Delta update

$$\begin{aligned}
\frac{\partial \Lambda}{\partial \delta_k^5} &= \sum_{n=1}^N \sum_{i=1}^T \sum_{j=1}^{C_i} \sum_{l=d_{\min}^3}^{d_{\max}^3} \zeta_{ijkl}^n \left[\frac{1}{\delta_k^5} - \frac{\beta_{f^5(i,j,k)}^5}{\sum_{p=d_{\min}^5}^{d_{\max}^5} \beta_{f^5(i,j,p)}^5 \delta_p^5} \right] - \lambda_{\delta^5} = 0 \\
&= \sum_{n=1}^N \sum_{i=1}^T \sum_{j=1}^{C_i} \sum_{k=d_{\min}^5}^{d_{\max}^5} \sum_{l=d_{\min}^3}^{d_{\max}^3} \zeta_{ijkl}^n \left[1 - \frac{\beta_{f^5(i,j,k)}^5 \delta_k^5}{\sum_{p=d_{\min}^5}^{d_{\max}^5} \beta_{f^5(i,j,p)}^5 \delta_p^5} \right] = \sum_{p=d_{\min}^5}^{d_{\max}^5} \delta_k^5 \lambda_{\delta^5} \\
N - N &= 0 = \lambda_{\delta^5} \\
\frac{\partial \Lambda}{\partial \delta_k^5} &= \sum_{n=1}^N \sum_{i=1}^T \sum_{j=1}^{C_i} \sum_{l=d_{\min}^3}^{d_{\max}^3} \zeta_{ijkl}^n \left[\frac{1}{\delta_k^5} - \frac{\beta_{f^5(i,j,k)}^5}{\sum_{p=d_{\min}^5}^{d_{\max}^5} \beta_{f^5(i,j,p)}^5 \delta_p^5} \right] \quad \forall k \in d_{\min}^5 \dots d_{\max}^5
\end{aligned}$$

Beta update

$$\begin{aligned}
\frac{\partial \Lambda}{\partial \beta_f^5} &= \sum_{n=1}^N \sum_{i=1}^T \sum_{j=1}^{C_i} \sum_{k=d_{\min}^5}^{d_{\max}^5} \sum_{l=d_{\min}^3}^{d_{\max}^3} \zeta_{ijkl}^n \left[\frac{\mathbb{1}(f^5(i,j,k) = f)}{\hat{\beta}_f^5} - \frac{\mathbb{1}(f^5(i,j,k) = f) \delta_k^5}{\sum_{p=d_{\min}^5}^{d_{\max}^5} \hat{\beta}_{f^5(i,j,p)}^5 \delta_p^5} \right] - \lambda_{\beta^5} = 0 \\
&= \sum_{n=1}^N \sum_{i=1}^T \sum_{j=1}^{C_i} \sum_{k=d_{\min}^5}^{d_{\max}^5} \sum_{l=d_{\min}^3}^{d_{\max}^3} \sum_{f \in A^{v^5}} \zeta_{ijkl}^n \left[\mathbb{1}(f^5(i,j,k) = f) - \frac{\mathbb{1}(f^5(i,j,k) = f) \delta_k^5}{\sum_{p=d_{\min}^5}^{d_{\max}^5} \hat{\beta}_{f^5(i,j,p)}^5 \delta_p^5} \right] \\
&= \sum_{f \in A^{v^5}} \hat{\beta}_f^5 \lambda_{\beta^5} \\
N - N &= 0 = \lambda_{\beta^5} \\
\frac{\partial \Lambda}{\partial \beta_f^5} &= \sum_{n=1}^N \sum_{i=1}^T \sum_{j=1}^{C_i} \sum_{k=d_{\min}^5}^{d_{\max}^5} \sum_{l=d_{\min}^3}^{d_{\max}^3} \zeta_{ijkl}^n \left[\frac{\mathbb{1}(f^5(i,j,k) = f)}{\beta_f^5} - \frac{\mathbb{1}(f^5(i,j,k) = f) \delta_k^5}{\sum_{p=d_{\min}^5}^{d_{\max}^5} \beta_{f^5(i,j,p)}^5 \delta_p^5} \right] \quad \forall f \in A^{v^5}
\end{aligned}$$

As a result of the quotient term in our likelihood function, we cannot directly solve for the elements of $\hat{\delta}$ and $\hat{\beta}$. However, we can compute the gradient of our log likelihood function with respect to these parameter vectors. During model training, we update each β and δ parameter vector by computing the gradient of the log likelihood with respect to that vector, projecting the gradient onto a simplex to satisfy the constraint that the elements of the vector sum to 1, and performing a line search in the direction of that gradient.

Zeta update

$$\begin{aligned}
\zeta_{ijkl}^n &= P(t_n = i, c_n = j, \delta_n^5 = k, \delta_n^3 = l | s_n, \Theta, \text{fp obs.}) \\
&= \frac{P(t_n = i, c_n = j, \delta_n^5 = k, \delta_n^3 = l, s_n | \Theta, \text{fp obs.})}{P(s_n | \Theta, \text{fp obs.})} \\
&= \frac{P(t_n = i, c_n = j, \delta_n^5 = k, \delta_n^3 = l, s_n | \Theta, \text{fp obs.})}{\sum_{v=1}^T \sum_{w=1}^{C_v} \sum_{p=d_{\min}^5}^{d_{\max}^5} \sum_{q=d_{\min}^3}^{d_{\max}^3} P(t_n = v, c_n = w, \delta_n^5 = p, \delta_n^3 = q, s_n | \Theta, \text{fp obs.})} \\
&= \frac{\mathbb{1}(s_n \simeq s(i, j, k, l)) P(t_n = i, c_n = j, \delta_n^5 = k, \delta_n^3 = l | \Theta, \text{fp obs.})}{\sum_{v=1}^T \sum_{w=1}^{C_v} \sum_{p=d_{\min}^5}^{d_{\max}^5} \sum_{q=d_{\min}^3}^{d_{\max}^3} \mathbb{1}(s_n \simeq s(v, w, p, q)) P(t_n = v, c_n = w, \delta_n^5 = p, \delta_n^3 = q | \Theta, \text{fp obs.})} \\
&= \frac{\mathbb{1}(s_n \simeq s(i, j, k, l)) \rho_i \pi_{ij} \frac{\beta_{f^5(i,j,k)}^5 \delta_k^5 \beta_{f^3(i,j,l)}^3 \delta_l^3}{\sum_{k'=d_{\min}^5}^{d_{\max}^5} \sum_{l'=d_{\min}^3}^{d_{\max}^3} \beta_{f^5(i,j,k')}^5 \delta_{k'}^5 \beta_{f^3(i,j,l')}^3 \delta_{l'}^3}}{\sum_{v=1}^T \sum_{w=1}^{C_v} \sum_{p=d_{\min}^5}^{d_{\max}^5} \sum_{q=d_{\min}^3}^{d_{\max}^3} \mathbb{1}(s_n \simeq s(v, w, p, q)) \rho_v \pi_{vw} \frac{\beta_{f^5(v,w,p)}^5 \delta_p^5 \beta_{f^3(v,w,q)}^3 \delta_q^3}{\sum_{p'=d_{\min}^5}^{d_{\max}^5} \sum_{q'=d_{\min}^3}^{d_{\max}^3} \beta_{f^5(v,w,p')}^5 \delta_{p'}^5 \beta_{f^3(v,w,q')}^3 \delta_{q'}^3}}
\end{aligned}$$

3.6 Simulation Results

To test our generative model and training procedure, we first trained model parameters for the ribosome profiling data sets from Schuller *et al.*, Weinberg *et al.*, and our own experiment used in chapter 2. Each of these experiments was run in *S. cerevisiae*, with a distinct combination of ligation strategies (Table 1). For each experiment, we used the trained parameters of its model to generate 75 million reads via simulation. This quantity of data was comparable to the size of the experiments. Then we trained a new model on the simulated data, and compared the parameters of our original models to the parame-

ters recovered from simulated data.

Our model training procedure recovered accurate starting parameters from the simulation data (Figures 3.2 and 3.3). The only trained parameter set that was not nearly identical to the starting parameters in each data set was π . While we simulated large data sets, there are about 3 million codons in our transcriptome. This yields about 25 simulated footprints on average per π parameter estimated. However, as we can observe from the ranges of our parameters, the probabilities of generating footprints from A site codons can vary over many orders of magnitude. Consequently, the number of footprints generated at some codons is small, and the π parameters estimated at these positions are more highly variable.

To confirm that the bias and digestion parameters that we learned were not reflecting the background sequence composition of the transcriptome, we shuffled the β and δ parameter vectors learned for each experiment and repeated the simulation and model training. The recovery of shuffled parameters was similar to the recovery of unshuffled parameters (Figure 3.4).

3.7 Relative Ligation Bias

The experiments that we have studied to test our model each use distinct combinations of ligation strategies, outlined in Table 3.1. We compared the ligation bias weights learned for each of these experiments to determine if fragment biases were consistent under a given ligation strategy, across different laboratory and experimental conditions. We observed the highest correlation in the β^3 weights between our own data and that of Schuller *et al.* (Pearson's $r = 0.91$, Figure 3.5). These experiments each used T4 RNA Ligase II K227Q for their 3' ligation strategy. We also observed high correlations between the β^3 weights for these experiments and β^3 from Weinberg *et al.*, which used T4 RNA Ligase I at this end. At the 5' end, the β^5 weights learned for Schuller *et al.* and our own experiment were well correlated ($r = 0.79$). These experiments used Circligase I and II, which differ by their level of adenylation at the active site. However, we observed no relationship between the 5' bias weights learned with a Circligase strategy and the corresponding weights for Weinberg *et al.*, which used T4 RNA Ligase I for its 5' ligation.

Experiment	5' Ligase	3' Ligase
Weinberg <i>et al.</i>	T4 RNA Ligase I	T4 RNA Ligase I
Schuller <i>et al.</i>	Circligase I	T4 RNA Ligase II Truncated K227Q
Self <i>et al.</i>	Circligase II	T4 RNA Ligase II Truncated K227Q

Table 3.1 Ligases used in preparing ribosome profiling experiments.

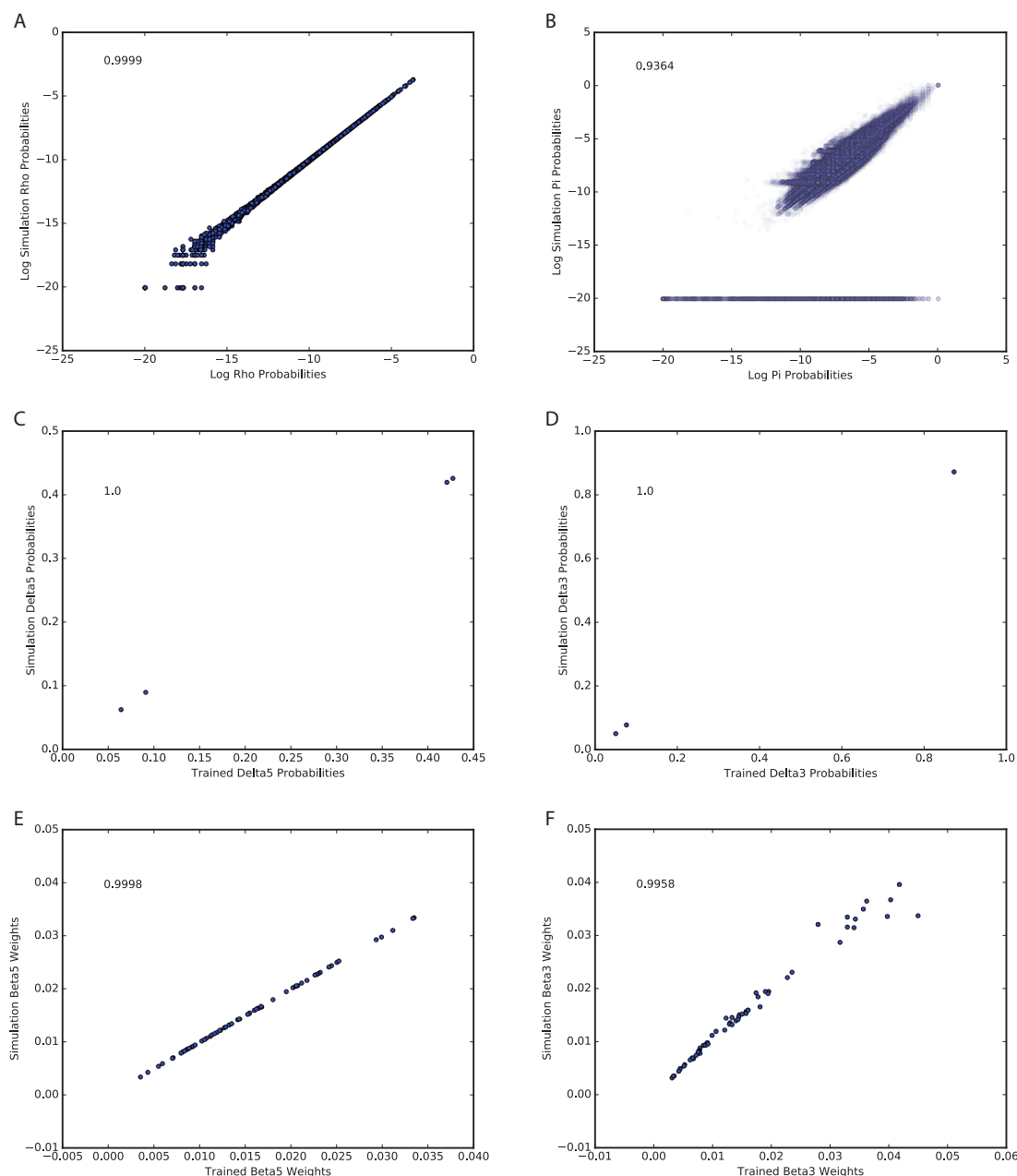


Figure 3.2: Schuller *et al.* simulation results. Parameters learned from simulations are nearly identical to the parameters used to generate data, with the exception of pi parameters in **b**. Pi parameters are learned from the least amount of data per parameter. This leads to greater variability in the learned parameters. Observe that a subset of nonzero pi parameters are learned as zero in the simulation model. These points represent A site codons where no data were generated in the simulation. Pearson's r indicated in top left of each panel.

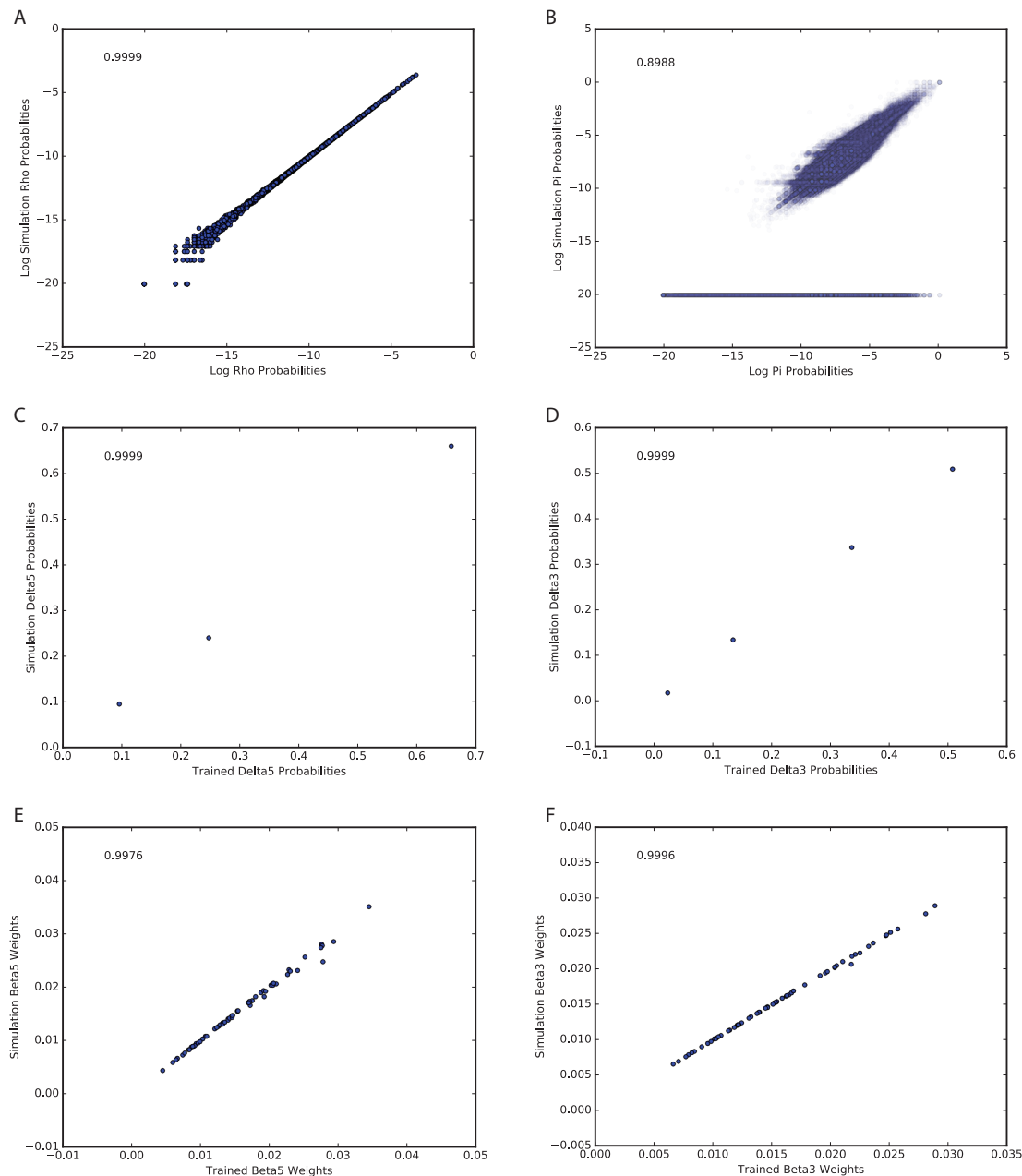


Figure 3.3: Weinberg *et al.* simulation results. Parameters learned from simulations are nearly identical to the parameters used to generate data, with the exception of pi parameters in **b**, as in Figure 3.2. Pearson's r indicated in top left of each panel.

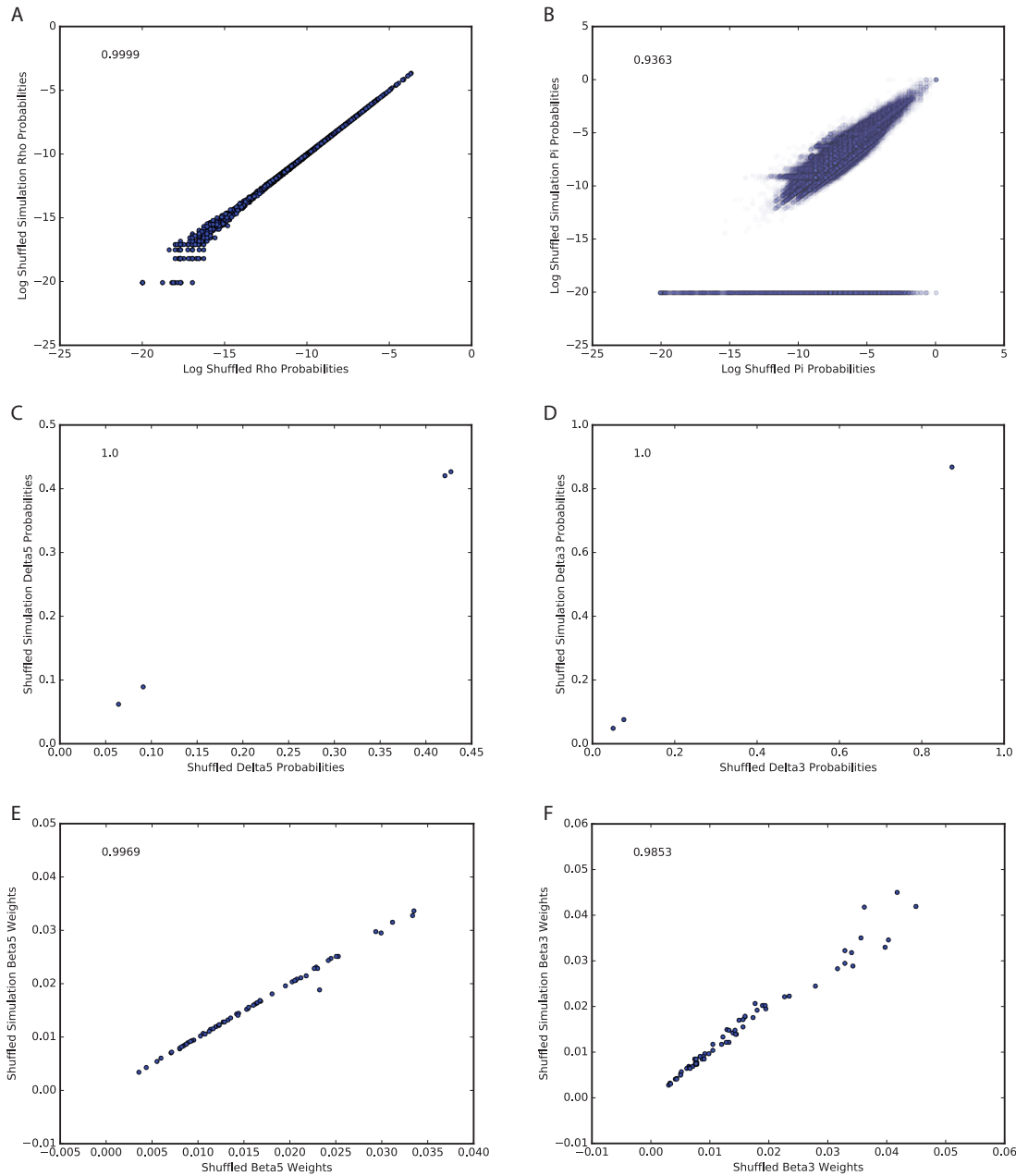


Figure 3.4: Schuller *et al.* simulation results with shuffled delta and beta parameters. Parameter recovery with shuffled parameters is similar to recovery in Figure 3.2, indicating that background sequence composition in the transcriptome is not influencing bias parameter training. Pearson's r indicated in top left of each panel.

3.8 Ligation Bias Correction

Our model formulation allows us to determine the probability of generating a footprint from a given A site codon, conditional upon that footprint being observed. That is, our model learns the distribution of ribosome footprints as observed after a ribosome profiling experiment. We would prefer to know the true biological distribution of ribosome footprints across transcripts, before it is modified by the experimental recovery procedure. We now show that under the assumptions of our model, we can recover this distribution from our trained model parameters.

$$\begin{aligned}
& P_{\Theta}(t = i, c = j, d^5 = k, d^3 = l) \\
&= \frac{P_{\Theta}(t = i, c = j, d^5 = k, d^3 = l) P_{\Theta}(t = i, c = j, d^5 = k, d^3 = l, \text{fp obs.})}{P_{\Theta}(t = i, c = j, d^5 = k, d^3 = l, \text{fp obs.})} \\
&= \frac{P_{\Theta}(t = i, c = j, d^5 = k, d^3 = l, \text{fp obs.})}{\frac{P_{\Theta}(t=i, c=j, d^5=k, d^3=l, \text{fp obs.})}{P_{\Theta}(t=i, c=j, d^5=k, d^3=l)}} \\
&= \frac{P_{\Theta}(t = i, c = j, d^5 = k, d^3 = l, \text{fp obs.})}{P_{\Theta}(\text{fp obs.} | t = i, c = j, d^5 = k, d^3 = l)} \\
&= \frac{P_{\Theta}(t = i, c = j, d^5 = k, d^3 = l | \text{fp obs.}) P_{\Theta}(\text{fp obs.})}{P_{\Theta}(\text{fp obs.} | t = i, c = j, d^5 = k, d^3 = l)} \\
&\propto \frac{P_{\Theta}(t = i, c = j, d^5 = k, d^3 = l | \text{fp obs.})}{P_{\Theta}(\text{fp obs.} | t = i, c = j, d^5 = k, d^3 = l)} \\
&= \frac{\rho_i \pi_{ij} \frac{\beta_{f^5(i,j,k)}^5 \delta_k^5 \beta_{f^3(i,j,l)}^3 \delta_l^3}{\sum_{p=d_{\min}^5}^{d_{\max}^5} \sum_{q=d_{\min}^3}^{d_{\max}^3} \beta_{f^5(i,j,p)}^5 \delta_p^5 \beta_{f^3(i,j,q)}^3 \delta_q^3}}{\gamma_{f^5(i,j,p)}^5 \gamma_{f^3(i,j,q)}^3} \\
&P_{\Theta}(t = i, c = j, d^5 = k, d^3 = l) \propto \frac{\rho_i \pi_{ij} \frac{\beta_{f^5(i,j,k)}^5 \delta_k^5 \beta_{f^3(i,j,l)}^3 \delta_l^3}{\sum_{p=d_{\min}^5}^{d_{\max}^5} \sum_{q=d_{\min}^3}^{d_{\max}^3} \beta_{f^5(i,j,p)}^5 \delta_p^5 \beta_{f^3(i,j,q)}^3 \delta_q^3}}{\beta_{f^5(i,j,p)}^5 \beta_{f^3(i,j,q)}^3}
\end{aligned}$$

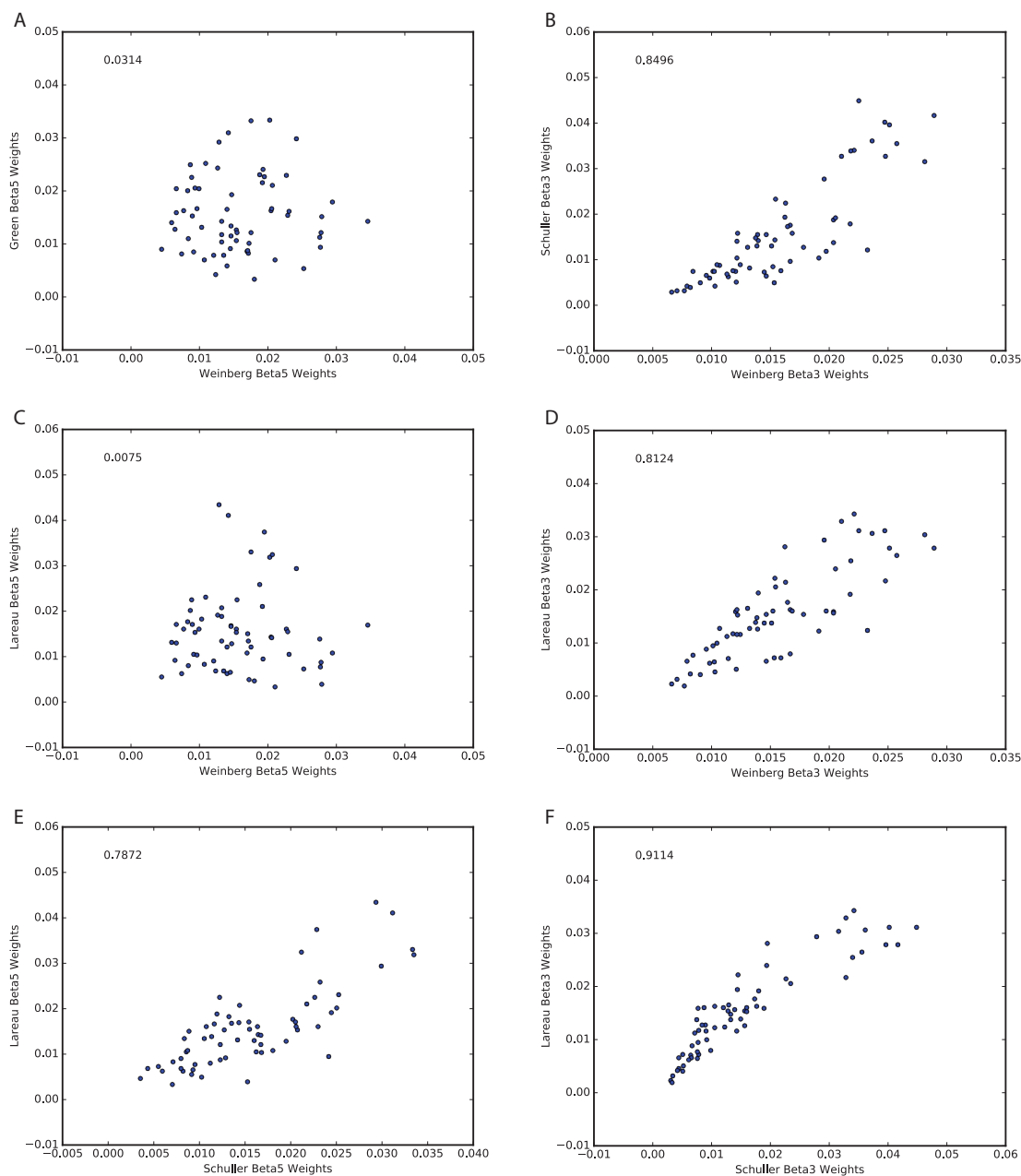


Figure 3.5: 5' and 3' bias weight comparisons for Weinberg *et al.*, Schuller *et al.*, and our own data. Bias weight parameters are highly correlated for **B**, **D** T4 RNA Ligase I vs. T4 RNA Ligase II Truncated K227Q, **E** Circligase I vs. Circligase II, and **F**, both T4 RNA Ligase II Truncated K227Q. Bias weights are uncorrelated for unrelated ligases, as in **A**, **C** T4 RNA Ligase I vs. Circligase I and II. Pearson's r indicated in top left of each panel.

We can compute quantities proportional to $P_{\Theta}(t = i, c = j, d^5 = k, d^3 = l)$ for all legal footprint digests over the transcriptome, and then rescale this set into a proper probability distribution. Finally we sum over k and l to compute $P_{\Theta}(t = i, c = j)$ for all codons in the transcriptome.

We applied this bias correction procedure to the Weinberg and Schuller experiments, and also to our own experiment. For each experiment we computed $\tilde{\rho}$ and $\tilde{\pi}$, the probabilities of generating a footprint from each gene, and from each codon given a gene, without conditioning on the footprint being observed. We then compared the ρ and π parameters between experiments, before and after bias correction. Each of these experiments was performed in *S. cerevisiae*, so we reasoned that they should have similar distributions of ribosome density across transcripts. However, these experiments each had different sources of ligation bias. We thought this might modify the observed footprints in different ways off of a common biological distribution, in ways that we could rectify with bias correction.

After bias correction, we did not observe a discernable change in the correlation of the ρ parameters (Figure 3.6). These parameters were highly correlated ($r > 0.96$) in all the pairs of experiments, but the correlations did not increase or decrease appreciably from ρ to $\tilde{\rho}$. This suggests that recovery biases tend to average out over the many footprint digests that a gene can generate, and that the overall count of footprints per gene is not affected much by recovery biases. In contrast, we do observe an increase in the correlation of π parameters after bias correction, in each pair of experiments (Figures 3.7, 3.8, 3.9). The improvement in correlation is small for the overall π parameters ($\Delta r = 0.01$ for Schuller vs. Weinberg, 0.03 for our data vs. Weinberg, 0.01 for our data vs. Schuller), but it increases as we restrict our gene set to more highly expressed genes (top 500 genes, $\Delta r = 0.04$ Schuller vs. Weinberg, 0.08 for our data vs. Weinberg, 0.06 for our data vs. Schuller). In more highly expressed genes, read counts per codon and per footprint digest are less variable, and we observe fewer zero counts. This suggests that our procedure is better able to correct for recovery bias and reconcile the underlying biological distributions of ribosomes in positions where those biases are more consistently reproduced. It also suggests that our bias correction procedure will work better for more deeply sampled experiments.

In addition, we measured the improvement in the correlation of the π parameters upon bias correction, excluding data points with a zero or near-zero ($< e^{-20}$) coordinate. These are codons to which either no footprints have mapped, or effectively all of the footprint density has been assigned to an alternate multimapping position. From one perspective, these positions contain meaningful information, because they may represent positions in which translation elongation occurs quickly. The ability of our model to remove bias even in the presence of these positions is reflected in the analysis above. From another perspective, it is not true that there's any codon where the ribosome spends no time during elongation. Consequently these are positions where we have undersampled

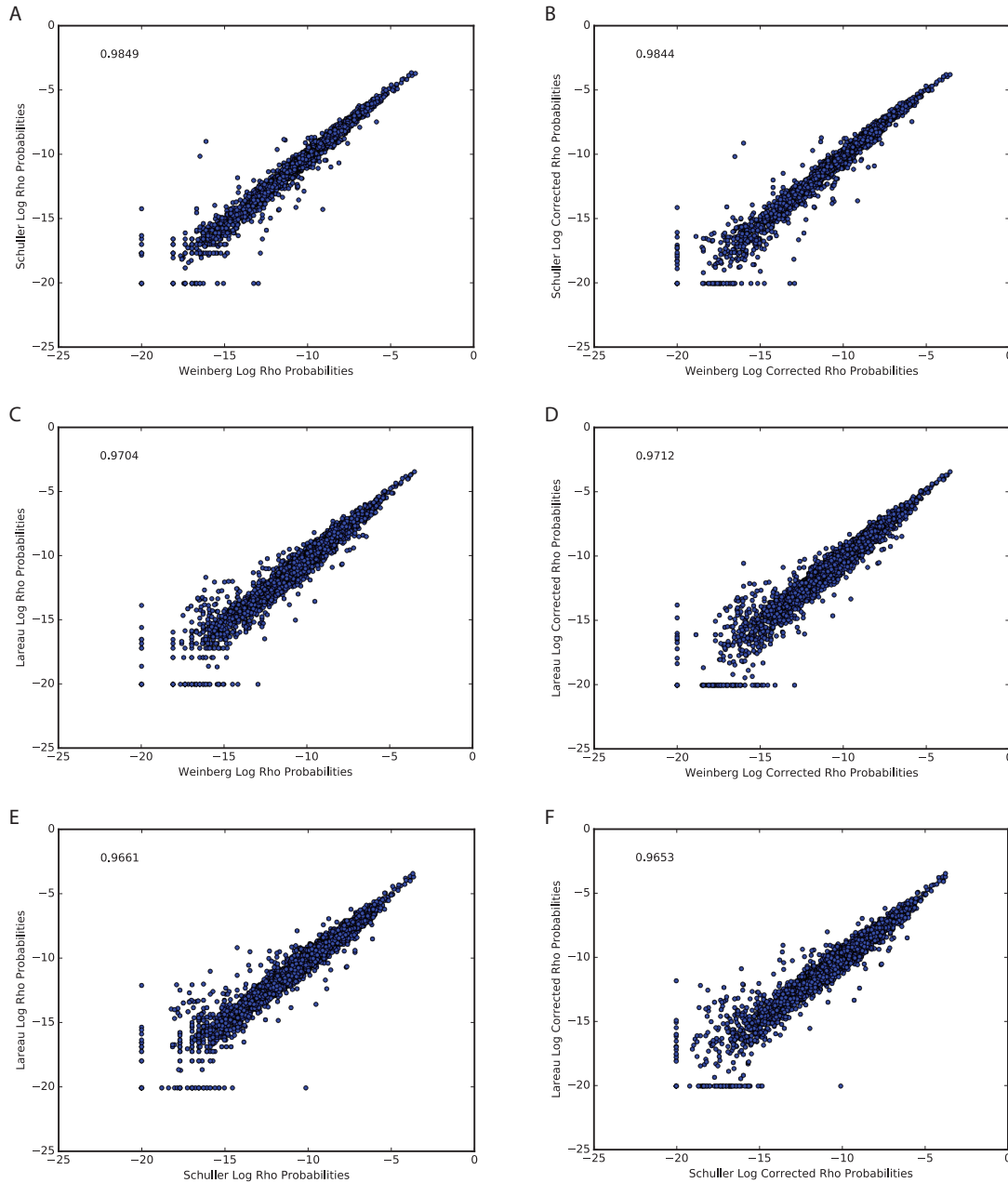


Figure 3.6: Comparison between per gene probabilities of generating ribosome footprints before bias correction ρ and after $\tilde{\rho}$. In each pair of experiments, the correlation between ρ parameters is high, and does not improve appreciably after bias correction. Log parameters below -20 are assigned -20. Pearson's r indicated in top left of each panel.

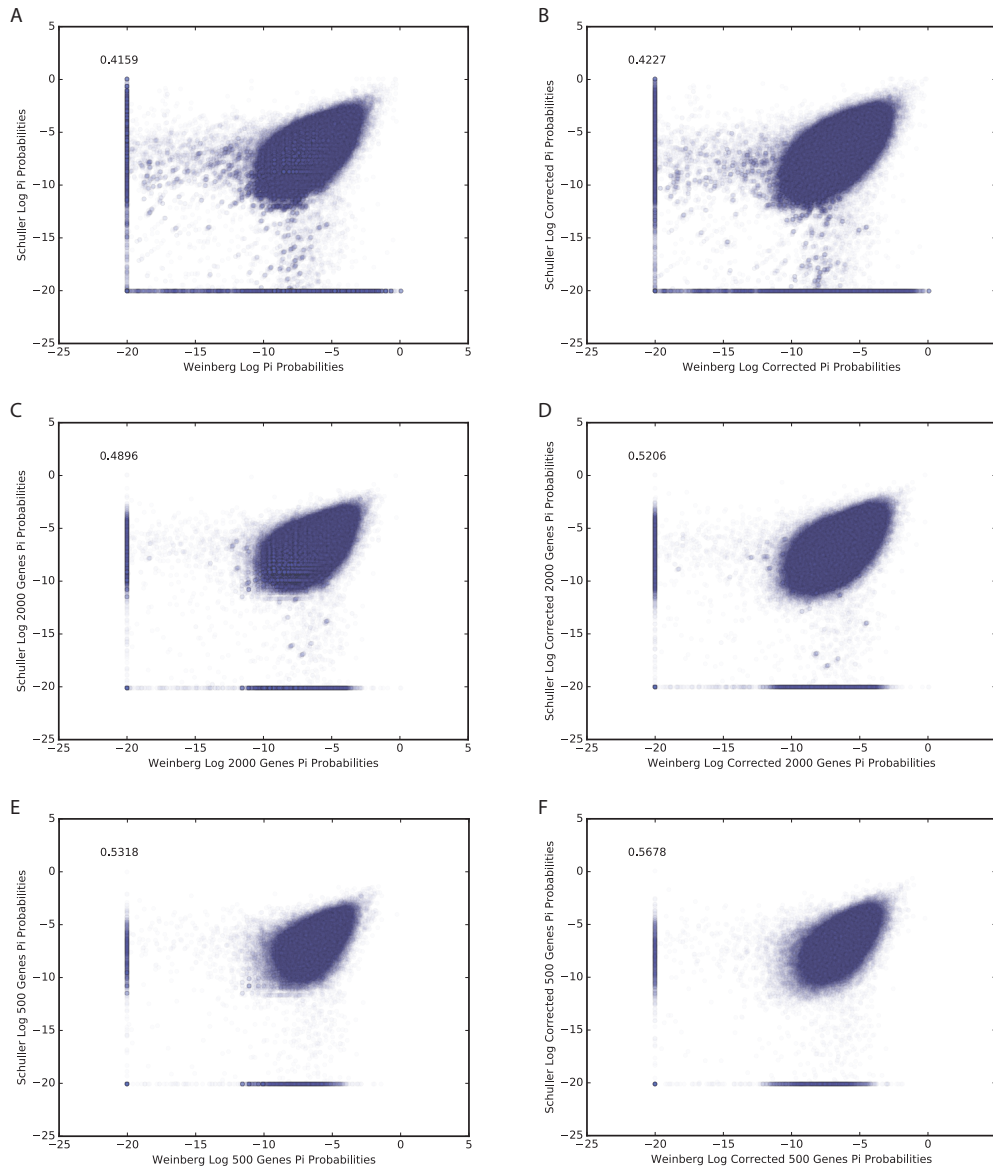


Figure 3.7: Improvement in correlation of per codon probabilities after bias correction, Schuller and Weinberg. Log parameters below -20 are assigned -20. **A, B** Correlation between π and $\tilde{\pi}$ parameters. Correlation improves slightly after bias correction, but not meaningfully. Excluding log parameters below -20, correlation improves by 0.04 after bias correction. **C, D** Correlation between π and $\tilde{\pi}$ parameters on the top 2000 genes by footprint density. Correlation improves more after bias correction on more well sampled genes ($\Delta r = 0.03$). **E, F** Correlation between π and $\tilde{\pi}$ parameters on the top 500 genes by footprint density. Correlation improves even more ($\Delta r = 0.04$). Pearson's r indicated in top left of each panel.

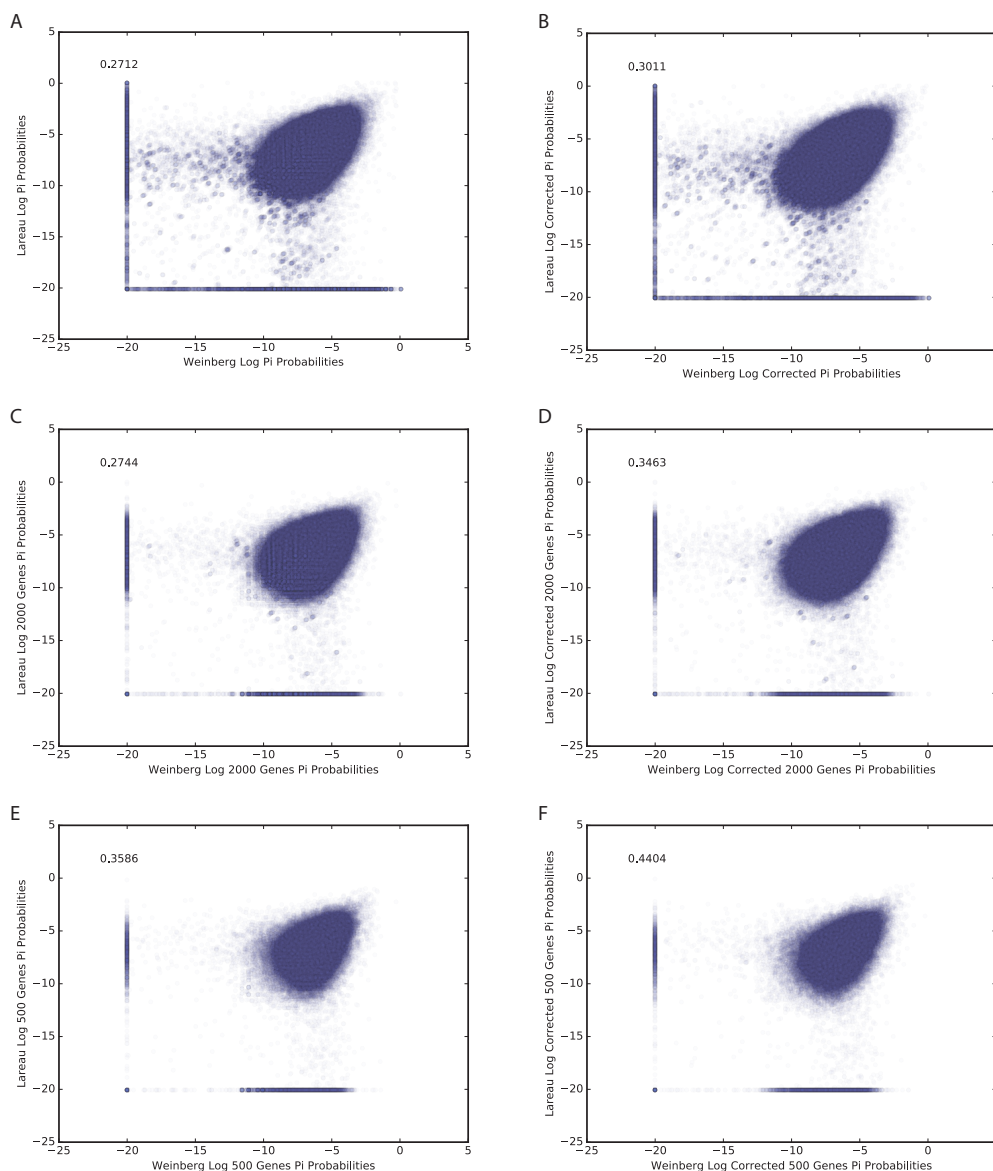


Figure 3.8: Improvement in correlation of per codon probabilities after bias correction, our data and Weinberg. Log parameters below -20 are assigned -20. **A, B** Correlation between π and $\tilde{\pi}$ parameters. Correlation improves after bias correction ($\Delta r = 0.03$). Excluding log parameters below -20, correlation improves by 0.06 after bias correction. **C, D** Correlation between π and $\tilde{\pi}$ parameters on the top 2000 genes by footprint density. Correlation improves more after bias correction on more well sampled genes ($\Delta r = 0.07$). **E, F** Correlation between π and $\tilde{\pi}$ parameters on the top 500 genes by footprint density. Correlation improves more ($\Delta r = 0.08$). Pearson's r indicated in top left of each panel.

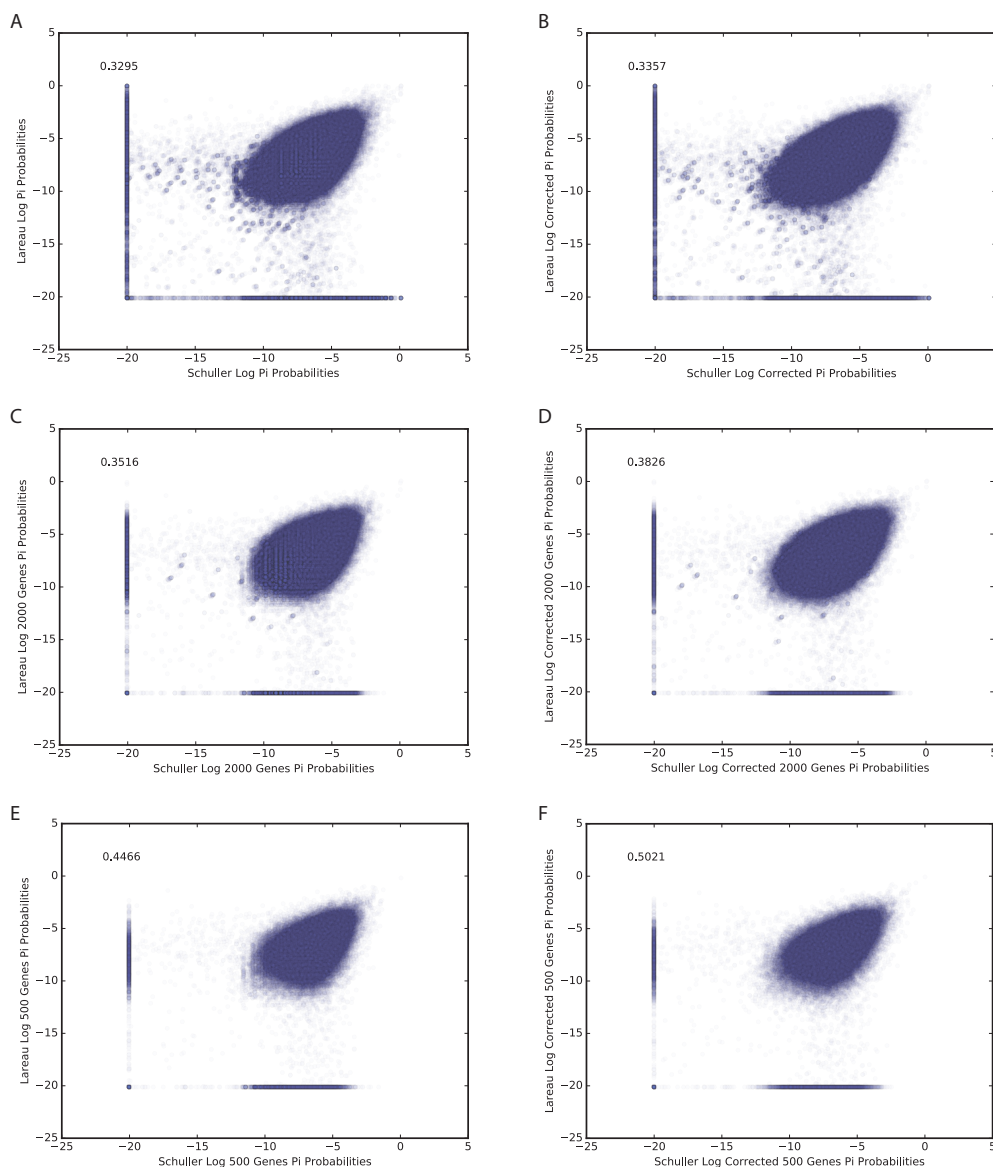


Figure 3.9: Improvement in correlation of per codon probabilities after bias correction, Schuller and Weinberg. Log parameters below -20 are assigned -20. **A, B** Correlation between π and $\tilde{\pi}$ parameters. Correlation improves slightly after bias correction, but not meaningfully. Excluding log parameters below -20, correlation improves by 0.08 after bias correction. **C, D** Correlation between π and $\tilde{\pi}$ parameters on the top 2000 genes by footprint density. Correlation improves more after bias correction on more well sampled genes ($\Delta r = 0.03$). **E, F** Correlation between π and $\tilde{\pi}$ parameters on the top 500 genes by footprint density. Correlation improves even more ($\Delta r = 0.06$). Pearson's r indicated in top left of each panel.

reads, and where the probability of generating reads is falling below representation given the sequencing depth of an experiment. In addition, our bias correction procedure can never shift an estimated zero π parameter away from zero. From this perspective, we may get a more accurate sense of the performance of our bias correction procedure by performing correction in positions where some footprints are observed. Using all genes, but excluding data points with an estimated π or $\tilde{\pi}$ coordinate below e^{-20} we again consistently observe improvement in the correlation between π parameters after bias correction. The correlation between $\tilde{\pi}$ for Weinberg and Schuller increases to 0.56 (from 0.52), for Weinberg and our own data to 0.48 (from 0.41), and for Schuller and our own data to 0.55 (from 0.47). These are substantial increases over the improvements that we see when including positions with a zero or near-zero coordinate.

We've observed that our bias correction procedure is improving agreement in ribosome density distributions in available data sets. In addition, we would like to outline an experimental validation that is more ideally suited for this purpose. While we have observed that the fraction of ribosome footprints on each gene is highly replicable across experiments, the distributions of footprints within genes is more variable. Experimental conditions like harvest time, temperature, and drugs used to arrest the ribosome can introduce variability in how effectively ribosomes are arrested. Ideally we would like to compare experiments where we have high confidence that the biological distribution of ribosomes within transcripts is exactly the same. To validate our bias correction procedure, we are pursuing an experiment where we harvest one large yeast sample, and applying different ligation strategies to fractions of the sample. Since these fractions come from the same biological source, and are subjected to the same experimental procedure with the exception of ligation steps, we will have higher confidence that the only differences in these two samples are the effects of the differing ligation strategies on footprint recovery. This should provide the clearest opportunity to evaluate how well our bias correction procedure corrects for ligation bias and recovers the biological distributions of ribosomes over transcripts.

Bibliography

- [1] Manuel Ares. "Isolation of total RNA from yeast cell cultures". en. In: *Cold Spring Harb. Protoc.* 2012.10 (Oct. 2012), pp. 1082–1086.
- [2] Carlo G Artieri and Hunter B Fraser. "Accounting for biases in riboprofiling data indicates a major role for proline in stalling translation". en. In: *Genome Res.* 24.12 (Dec. 2014), pp. 2011–2021.
- [3] E Battenberg et al. *Lasagne: First release*. Aug. 2015.
- [4] Ariel A Bazzini et al. "Codon identity regulates mRNA stability and translation efficiency during the maternal-to-zygotic transition". en. In: *EMBO J.* 35.19 (Oct. 2016), pp. 2087–2103.
- [5] Gloria A Brar et al. "High-resolution view of the yeast meiotic program revealed by ribosome profiling". In: *science* 335.6068 (2012), pp. 552–557.
- [6] Patrick O Brown and David Botstein. "Exploring the new world of the genome with DNA microarrays". In: *Nature genetics* 21.1s (1999), p. 33.
- [7] Michal Burdukiewicz et al. "Methods for comparing multiple digital PCR experiments". en. In: *Biomol Detect Quantif* 9 (Sept. 2016), pp. 14–19.
- [8] Catherine A Charneski and Laurence D Hurst. "Positively charged residues are the major determinants of ribosomal velocity". en. In: *PLoS Biol.* 11.3 (Mar. 2013), e1001508.
- [9] Dominique Chu et al. "Translation elongation can control translation initiation on eukaryotic mRNAs". In: *EMBO J.* 33.1 (2013), pp. 21–34.
- [10] Alexandra Dana and Tamir Tuller. "Determinants of translation elongation speed and ribosomal profiling biases in mouse embryonic stem cells". en. In: *PLoS Comput. Biol.* 8.11 (Nov. 2012), e1002755.
- [11] R Daniel Gietz and Robin A Woods. "Transformation of yeast by lithium acetate/single-stranded carrier DNA/polyethylene glycol method". In: *Methods in Enzymology*. 2002, pp. 87–96.
- [12] Khanh Dao Duc and Yun S Song. "The impact of ribosomal interference, codon usage, and exit tunnel interactions on translation elongation rate variation". en. In: *PLoS Genet.* 14.1 (Jan. 2018), e1007166.

- [13] Han Fang et al. *Scikit-ribo: Accurate estimation and robust modeling of translation dynamics at codon resolution*. 2017.
- [14] Caitlin E Gamble et al. "Adjacent Codons Act in Concert to Modulate Translation Efficiency in Yeast". en. In: *Cell* 166.3 (July 2016), pp. 679–690.
- [15] Justin Gardin et al. "Measurement of average decoding rates of the 61 sense codons in vivo". en. In: *Elife* 3 (Oct. 2014).
- [16] Daniel G Gibson et al. "Enzymatic assembly of DNA molecules up to several hundred kilobases". en. In: *Nat. Methods* 6.5 (May 2009), pp. 343–345.
- [17] Hani Goodarzi et al. "Modulated Expression of Specific tRNAs Drives Gene Expression and Cancer Progression". en. In: *Cell* 165.6 (June 2016), pp. 1416–1427.
- [18] U Gueldener et al. "A second set of loxP marker cassettes for Cre-mediated multiple gene knockouts in budding yeast". en. In: *Nucleic Acids Res.* 30.6 (Mar. 2002), e23.
- [19] Markus Hafner et al. "RNA-ligase-dependent biases in miRNA representation in deep-sequenced small RNA cDNA libraries". In: *Rna* 17.9 (2011), pp. 1697–1712.
- [20] Florian Hahne et al. "flowCore: a Bioconductor package for high throughput flow cytometry". en. In: *BMC Bioinformatics* 10 (Apr. 2009), p. 106.
- [21] Florian Hahne et al. *flowStats: Statistical methods for the analysis of flow cytometry data*. 2017.
- [22] Jeffrey A Hussmann et al. "Understanding Biases in Ribosome Profiling Experiments Reveals Signatures of Translation Dynamics in Yeast". en. In: *PLoS Genet.* 11.12 (Dec. 2015), e1005732.
- [23] Nicholas T Ingolia, Liana F Lareau, and Jonathan S Weissman. "Ribosome profiling of mouse embryonic stem cells reveals the complexity and dynamics of mammalian proteomes". In: *Cell* 147.4 (2011), pp. 789–802.
- [24] Nicholas T Ingolia et al. "Genome-wide analysis in vivo of translation with nucleotide resolution using ribosome profiling". en. In: *Science* 324.5924 (Apr. 2009), pp. 218–223.
- [25] R Ishimura et al. "Ribosome stalling induced by mutation of a CNS-specific tRNA causes neurodegeneration". In: *Science* 345.6195 (2014), pp. 455–459.
- [26] Shintaro Iwasaki, Stephen N Floor, and Nicholas T Ingolia. "Rocaglates convert DEAD-box protein eIF4A into a sequence-selective translational repressor". en. In: *Nature* 534.7608 (June 2016), pp. 558–561.
- [27] Sebastian Kirchner et al. "Alteration of protein function by a silent polymorphism linked to tRNA abundance". en. In: *PLoS Biol.* 15.5 (May 2017), e2000779.
- [28] Chun Kit Kwok et al. "A hybridization-based approach for quantitative and low-bias single-stranded DNA ligation". In: *Analytical biochemistry* 435.2 (2013), pp. 181–186.

- [29] Ben Langmead et al. "Ultrafast and memory-efficient alignment of short DNA sequences to the human genome". en. In: *Genome Biol.* 10.3 (Mar. 2009), R25.
- [30] Liana F Lareau et al. "Distinct stages of the translation elongation cycle revealed by sequencing ribosome-protected mRNA fragments". en. In: *Elife* 3 (May 2014), e01257.
- [31] Daniel P Letzring, Kimberly M Dean, and Elizabeth J Grayhack. "Control of translation efficiency in yeast by codon-anticodon interactions". en. In: *RNA* 16.12 (Dec. 2010), pp. 2516–2528.
- [32] Bo Li and Colin N Dewey. "RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome". en. In: *BMC Bioinformatics* 12 (Aug. 2011), p. 323.
- [33] Tzu-Yu Liu and Yun S Song. "Prediction of ribosome footprint profile shapes from transcript sequences". In: *Bioinformatics* 32.12 (2016), pp. i183–i191.
- [34] Nicholas J McGlincy and Nicholas T Ingolia. "Transcriptome-wide measurement of translation by ribosome profiling". en. In: *Methods* 126 (Aug. 2017), pp. 112–129.
- [35] Ali Mortazavi et al. "Mapping and quantifying mammalian transcriptomes by RNA-Seq". In: *Nature methods* 5.7 (2008), p. 621.
- [36] Frank V Murphy 4th and V Ramakrishnan. "Structure of a purine-purine wobble base pair in the decoding center of the ribosome". en. In: *Nat. Struct. Mol. Biol.* 11.12 (Dec. 2004), pp. 1251–1252.
- [37] Patrick B F O'Connor, Dmitry E Andreev, and Pavel V Baranov. "Comparative survey of the relative impact of mRNA features on local ribosome profiling read density". en. In: *Nat. Commun.* 7 (Oct. 2016), p. 12915.
- [38] Joshua B Plotkin and Grzegorz Kudla. "Synonymous but not the same: the causes and consequences of codon bias". en. In: *Nat. Rev. Genet.* 12.1 (Jan. 2011), pp. 32–42.
- [39] Cristina Pop et al. "Causal signals between codon bias, mRNA structure, and the efficiency of translation and elongation". en. In: *Mol. Syst. Biol.* 10 (Dec. 2014), p. 770.
- [40] Vladimir Presnyak et al. "Codon optimality is a major determinant of mRNA stability". en. In: *Cell* 160.6 (Mar. 2015), pp. 1111–1124.
- [41] Wenfeng Qian et al. "Balanced Codon Usage Optimizes Eukaryotic Translational Efficiency". In: *PLoS Genet.* 8.3 (2012), e1002603.
- [42] Mario dos Reis, Renos Savva, and Lorenz Wernisch. "Solving the riddle of codon usage preferences: a test for translational selection". en. In: *Nucleic Acids Res.* 32.17 (Sept. 2004), pp. 5036–5044.
- [43] C Ritz and A-N Spiess. "qpcR: an R package for sigmoidal model selection in quantitative real-time polymerase chain reaction analysis". In: *Bioinformatics* 24.13 (2008), pp. 1549–1551.

- [44] D Sarkar, N Le Meur, and R Gentleman. “Using flowViz to visualize flow cytometry data”. en. In: *Bioinformatics* 24.6 (Mar. 2008), pp. 878–879.
- [45] Anthony P Schuller et al. “eIF5A Functions Globally in Translation Elongation and Termination”. en. In: *Mol. Cell* 66.2 (Apr. 2017), 194–205.e5.
- [46] Premal Shah et al. “Rate-limiting steps in yeast protein translation”. en. In: *Cell* 153.7 (June 2013), pp. 1589–1601.
- [47] P M Sharp, T M Tuohy, and K R Mosurski. “Codon usage in yeast: cluster analysis clearly differentiates highly and lowly expressed genes”. en. In: *Nucleic Acids Res.* 14.13 (July 1986), pp. 5125–5143.
- [48] Mark A Sheff and Kurt S Thorn. “Optimized cassettes for fluorescent protein tagging in *Saccharomyces cerevisiae*”. en. In: *Yeast* 21.8 (June 2004), pp. 661–670.
- [49] Michael Stadler and Andrew Fire. “Wobble base-pairing slows in vivo translation elongation in metazoans”. en. In: *RNA* 17.12 (Dec. 2011), pp. 2063–2073.
- [50] Noam Stern-Ginossar et al. “Decoding human cytomegalovirus”. In: *Science* 338.6110 (2012), pp. 1088–1093.
- [51] The Theano Development Team et al. “Theano: A Python framework for fast computation of mathematical expressions”. In: (May 2016). eprint: 1605.02688.
- [52] Tamir Tuller et al. “Translation efficiency is determined by both codon bias and folding energy”. In: *Proceedings of the National Academy of Sciences* 107.8 (2010), pp. 3645–3650.
- [53] Hao Wang, Joel McManus, and Carl Kingsford. “Accurate recovery of ribosome positions reveals slow translation of wobble-pairing codons in yeast”. In: *International Conference on Research in Computational Molecular Biology*. Springer. 2016, pp. 37–52.
- [54] David E Weinberg et al. “Improved Ribosome-Footprint and mRNA Measurements Provide Insights into Dynamics and Regulation of Yeast Translation”. en. In: *Cell Rep.* 14.7 (Feb. 2016), pp. 1787–1799.
- [55] Sai Zhang et al. *ROSE: a deep learning based framework for predicting ribosome stalling*. 2016.
- [56] Fangzhou Zhao, Chien-Hung Yu, and Yi Liu. “Codon usage regulates protein structure and function by affecting translation elongation speed in *Drosophila* cells”. en. In: *Nucleic Acids Res.* 45.14 (Aug. 2017), pp. 8484–8492.