A Neural Network Model of Translation Elongation Rates

Robert Tunney

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2020-26
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2020/EECS-2020-26.html

May 1, 2020

A Neural Network Model of Translation Elongation Rates

by Robert Tunney

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Professor Lior Pachter
Research Advisor

5/8/2018

(Date)

* * * * * * *

Professor Satish Rao
Second Reader

5/8/2018

(Date)

A Neural Network Model of Translation Elongation Rates

Abstract

A Neural Network Model of Translation Elongation Rates

by

Robert Tunney

Master of Science in Computer Science

University of California, Berkeley

Lior Pachter, Chair

Quantification and regulation of gene expression are central areas of inquiry in genomic analysis. Ribosome profiling experiments allow us to directly measure the process of gene expression directly at the point of protein production. We present a feed forward neural network model to predict the local translation rate of a protein as a function of the sequence context undergoing translation, as well as RNA structure in this region. Our model predictions correlated well with observed translation rates as measured by ribosome profiling (Pearson's r = 0.58). We describe a procedure to process ribosome profiling data for this model, discuss underlying assumptions of our model formulation, and present a series of model selection analyses. Finally, we present an algorithm to optimize the coding sequence of a gene for fast protein production, as predicted by our neural network regression model.

Contents

Contents **i**

1 Introduction **1**
1.1 Gene Expression . . . 1
1.2 Decoding Rates . . . 2
1.3 Additional Terms . . . 4
1.4 Ribosome Profiling . . . 4
1.5 Predicting Local Translation Rates . . . 5

2 Methods **9**
2.1 Data Processing and Mapping . . . 9
2.2 A Site Assignment . . . 9
2.3 Ribosome Profiles . . . 10
2.4 Input Features . . . 11
2.5 Model Architecture and Training . . . 13

3 Results **15**
3.1 Model dimensions . . . 15
3.2 Model Performance . . . 17
3.3 Poisson Error Model . . . 18
3.4 Testing for Overfitting . . . 21
3.5 Coding Sequence Optimization . . . 22

Bibliography **27**

Acknowledgments

First, I would like to thank my advisors, Drs. Lior Pachter and Liana Lareau. 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 the professors on my academic committes, 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. Finally, I would like to thank Dr. Satish Rao, for very generously reviewing this technical report.

Chapter 1

Introduction

A central goal of the genomics era is to understand how living systems are regulated and controlled. With the advent of inexpensive, high throughput DNA sequencing technology, our ability to measure the internal state of biological systems has expanded greatly. Specifically, this technology allows us to measure the extent to which each gene in a system is expressed. First through microarrays and bulk mRNA sequencing, and more recently through sequencing at the level of individual cells, we can quantify the extent to which each gene is transcribed into RNA copies for later protein production[4, 26, 25]. The expression of genes is generally measured at the RNA level, largely because we can easily quantify RNA abundance[22, 34]. However, the ultimate goal of gene expression is to produce protein. A specialized RNA sequencing experiment called ribosome profiling allows us to more directly measure the amount of protein produced. In this experiment, we measure the abundance of ribosomes, the complexes that synthesize proteins, in order to understand how much protein is in production for each gene[18]. This experiment has generated substantial interest because it allows us to measure the expression of genes as proteins more directly, and also because it helps us to understand regulatory features of protein production[1, 3, 6, 8, 9, 11, 16, 29, 30, 32]. Consequently, it is important that we have tools to accurately quantify gene expression rates as measures by ribosome profiling, and also to interpret what this data tells us about the regulatory constraints in genomic sequences.

1.1 Gene Expression

The genome is a set of DNA instructions that contain the information required to make proteins. DNA stores this information in a set of long polymers that we can imagine as strings over a four letter alphabet (A, C, G, T). Each of these characters is referred to as a nucleotide. The genome contains a number of subsequences, called genes, each of which encodes the sequence of one or more proteins. These genes are activated when a given

protein is needed by the cell, in a process called gene expression. Most cells in an organism contain a complete set of genomic DNA, and therefore have the ability to express all proteins native to that organism.

In order to produce a protein, a cell will first amplify the DNA sequence for the corresponding gene. In a process called transcription, the DNA sequence of a gene is copied into a molecule called mRNA, which is also a string over the genomic alphabet (replacing T with U), and contains the same sequence information to produce the desired protein (Figure 1.1). Transcription allows the gene to produce many instruction sets for a given gene, which allows greater control over the amount of protein produced.

mRNA serves as the direct template to produce protein, via a process called translation (Figure 1.1). Translation begins when a large complex called the ribosome loads onto the beginning of an mRNA sequence. This ribosome scans along an mRNA until it encounters a signal sequence that indicates it should begin to translate mRNA and construct a new protein. The subsequent region is referred to as the coding sequence (CDS) of a gene, and it contains the direct information necessary to construct a protein. The CDS is subdivided into non-overlapping three character segments, called codons, each of which codes for a single subunit of the protein. These subunits are called amino acids. The ribosome reads along the coding sequence, and it translates each codon into one amino acid that is added to a growing protein sequence. Specifically, the ribosome translates codons when they are situated within a region of the ribosome called the A site (Figure 1.2). A codon in the A site is paired with a complementary molecule called a tRNA, that is attached to the amino acid encoded by that codon. This amino acid is then ligated to the nascent linear sequence of amino acids that constitute a protein. When a ribosome reaches a special stop codon that encodes no amino acids, the fully translated protein is released.

1.2 Decoding Rates

The process of translating a codon into an amino acid in the A site is referred to as decoding. Different A site codons can require variable amounts of time for decoding, due to a number of interacting features in the translational system. For example, the amount of tRNA available for a given codon affects the rate at which it can load into the A site for decoding. tRNA structure and modifications can affect the rate of passage through the ribosome, as can the size and charge of amino acids loaded on the tRNA. Taken together, these properties can induce variable local rates of translation at each A site codon. In addition, similar properties in the upstream environment have been shown to affect passage of the nascent peptide through the ribosome. Consequently a broader sequence neighborhood can also affect local rates of translation at a given A site codon.

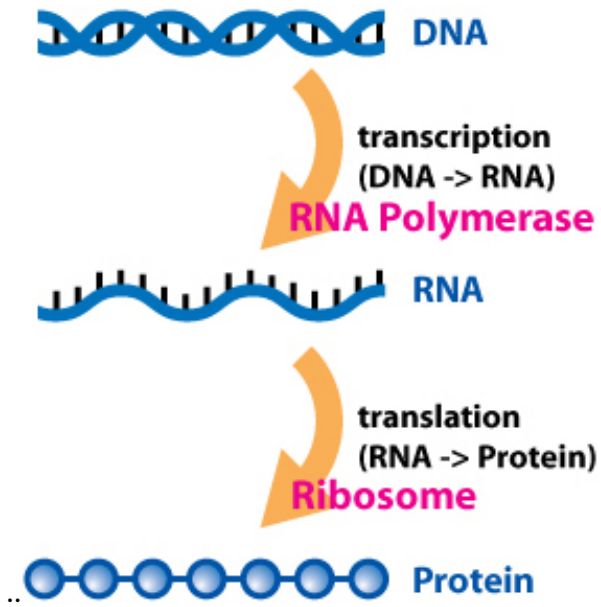

Figure 1.1: Overview of gene expression. DNA sequence contains subsequences called genes, which encode the sequence of proteins. In transcription, genes are amplified into mRNA copies. In translation, ribosomes read mRNAs in nonoverlapping three position subsequences called codons. Each codon is translated into an amino acid, which are added in a linear sequence to produce a protein. [15].

Figure 1.2: Cartoon diagram of the ribosome. White ovals are sites where the ribosome can hold a tRNA. In red, the A site, where tRNAs match up with codons and bring along a corresponding amino acid. The P site is the site where an amino acid is added to the growing protein chain. In the E site, a discharged tRNA is ejected. Arrows indicate common digestion termini for ribosome footprints in a ribosome profiling experiment.

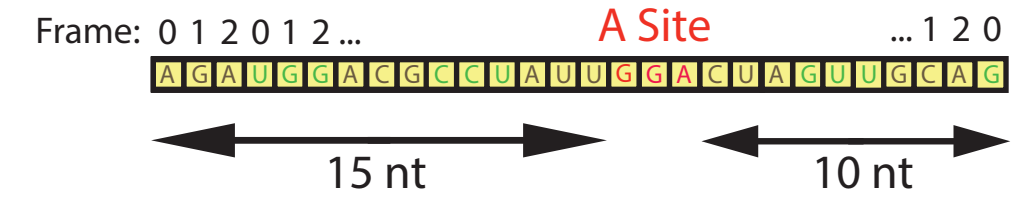

Figure 1.3: Diagram of a ribosome footprint. A ribosome footprint is approximately 28-30 nucleotides in length. Shown is a typical 28 nucleotide footprint, with codons highlighted in alternating colors. Above, frame annotations indicating the index of each nucleotide in its codon. The A site codon is highlighted in red. Below, distances between the A site and the 5′ and 3′ ends of the footprint. For all footprints of a given size, with their 5′ end mapping to a given frame, we define a distance between the 5′ end and the A site. This allows us to identify the A site in each footprint, and assign each footprint to an A site codon in the transcripts where it maps.

1.3 Additional Terms

Some additional terminology to describe RNAs will facilitate our discussion. The coding sequence of an RNA is divided into consecutive three character subsequences called codons. These codons induce a concept of frame on coding sequences. We refer to the set of nucleotides in the 0th, 1st, and 2nd positions in their codons as comprising the 0th, 1st, and 2nd frames of the coding sequence (Figure 1.3). In addition, nucleic acids have a polarity that indicates the direction in which their sequence should be read. The upstream direction is referred to as the 5 prime (5′) end of the molecule, and the downstream direction is referred to as the 3 prime (3′) end of the molecule.

1.4 Ribosome Profiling

Ribosome profiling is an RNA sequencing experiment that measures the distributions of ribosomes across all mRNA transcripts in a biological sample (Figure 1.4) [18]. First, a biological sample is frozen in liquid nitrogen to preserve the global distribution of ribosomes across mRNAs at a moment in time. Next bulk RNA is extracted from the sample and treated with a nuclease enzyme that digests all accessible RNA. As a result, the only fragments of mRNA that survive are the regions that are inaccessible, whether bound by some protein complex or sequestered within a ribosome. Fragments contained within a ribosome are referred to as ribosome footprints. This digestion is very efficient, such that ribosome footprints are observed at characteristic lengths between about 28-30 nucleotides (Figure 1.3). These fragments are size selected and prepared for a sequencing

library. After sequencing, we obtain a text file with one footprint sequence on each line. We use a string alignment algorithm to align these footprints back to an annotated set of mRNA transcripts. Within each footprint, we can assign an A site codon that is currently undergoing decoding. Finally, we can compute a histogram for each transcript indicating the counts of footprints generated from each codon in that transcript. We call this distribution a ribosome profile. Ribosome profiles yield two levels of information about translation. The overall count of footprints on each transcript indicates the relative level of protein production occurring that gene. Within an individual transcript, the count of footprints on each codon indicates the relative time of decoding. Many footprint counts at a codon indicate that the ribosome spends more time decoding this position, whereas fewer footprint counts indicate that the ribosome translates the codon more quickly. In general, we observe nonuniform distributions of ribosome footprint counts within transcripts. This indicates a range of per-codon translation rates across the codons in a transcript. There has been considerable interest in connecting these variable local translation rates with predictive features of the mRNA sequence undergoing translation, as well as the biochemical properties of the nascent peptide and the ribosome itself. This analysis helps us to understand the mechanics and regulation of the translational system, as well as evolutionary constraints on coding sequences[28, 21].

1.5 Predicting Local Translation Rates

Several analyses have attempted to connect mRNA sequence features and biochemical properties of components of the translational system with local translation rates as measured by ribosome profiling data. Most of this work has restricted itself to computing marginal effects of individual features on observed footprint counts. More recently, a few analyses have developed more comprehensive models that combine a set of predictive features to either reconstruct or predict observed distributions of ribosome density in ribosome profiles.

One tool, RUST, performed a binary transformation on per-codon footprint counts reflecting whether the codon was above or below average in counts for its gene [27]. After this binary transformation, they computed summary statistics on each codon in each position in a sequence neighborhood around the A site reflecting how often each codon was co-observed with above average footprint counts at the A site. RUST then combines these scores in the sequence neighborhood around the A site to reconstruct the observed data in an experiment. This approach performed well on reconstructing observed ribosome profiles, particularly for highly expressed genes with more dense and higher quality data. One weakness of this approach was that the authors did not hold out a test or validation set of genes for assessing their model performance, and they do not expose this functionality in their available tool.

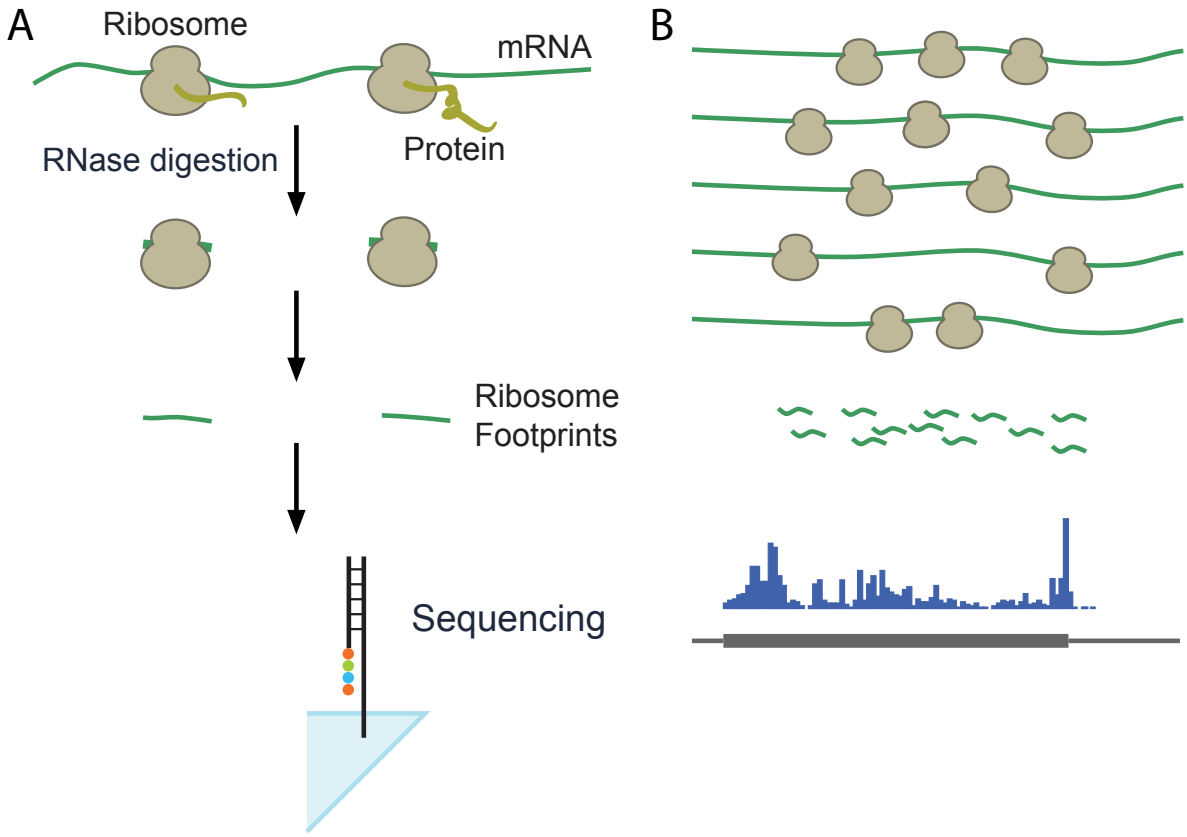

Figure 1.4: The ribosome profiling method. **A** Total RNA is collected from a sample, and subjected to digestion with a nuclease. The fragments of mRNA protected within ribosomes, called ribosome footprints, are isolated to prepare a sequencing library. Footprints are ligated to adapters on each end, reverse transcribed, amplified via PCR, and then sequenced. **B** As in **(A)**, emphasizing that ribosome profiling is a bulk assay on total mRNA in a sample (above). Below, after sequencing the population of ribosome footprints, these footprints are mapped back to a transcriptome and assigned to an A site codon undergoing translation in that footprint. For each gene, we can assemble a count of footprints per A site codon in a histogram. This represents the steady state distribution of ribosomes across a transcript.

Another project, riboshape, attempted to identify sequence features that affected the local shape of a ribosome profile at a number of different bandwidths, and then to use these features to predict the global shape of ribosome profiles with a sparse regression model [23]. This approach made an interesting and novel assumption drawn from signal processing, that a sequence feature could influence the local distribution of ribosome counts at a variety of scales. If this assumption has an underlying biological intuition, it could reflect that very slow positions in a transcript can create traffic jams with backed up ribosomes upstream of a given position. This model projected observed ribosome profiles down into subspaces of a Debauchies-8 basis to smooth out the profiles at various scales, and then performed sparse regression on the subspace representations of ribosome profiles. Finally, riboshape uses the trained weights of this regression along with kernel density estimation to construct predictions for new ribosome profiles. One limitation of the published tool is that it only performs predictions for ribosome profiles as represented in a subspace of the Debauchies-8 wavelet decomposition. Consequently, it cannot be used to predict observed data, and tends to perform poorly on real data. If we believe that ribosome profiles should be de-noised in this manner, then this regression model is appropriate. However, this projection blurs per-codon count data that is generally believed to reflect biologically meaningful variability per-codon translation rates. A strength of this model was that its sparse regression approach resulted in very good estimation of ribosome profiles for genes with low amounts of data, relative to other available tools.

A third project, ROSE, used a neural network model to perform a slightly different analysis task. A characteristic feature of ribosome profiling data is that we observe a subset of positions with very large footprint counts. These positions have been thought to correspond with translational stall sites, which have been characterized in a variety of contexts and are sometimes interesting regulatory positions in the process of translation. ROSE divides footprints into a small subset of stall sites and a much larger group of non-stall sites, and then performs a classification problem on this data using a neural network. The publication then connects the trained model to sequence and biochemical factors that contribute to ribosome stalling [38].

Our goal is to combine the more successful elements of these approaches to predict the distributions of ribosome footprints across genes. We define a regression problem where our predictive features are a sequence neighborhood around the A site codon, and our target of prediction is the density of ribosomes at a given A site. Specifically, we define this density as the raw footprint counts mapping to an A site codon, rescaled by the average counts over that codon's gene. This rescaling renders the set of counts on each gene as mean centered around one. This allows us to control for variable mRNA abundance and variable total footprint counts between genes, and build a single predictive model for all of our genes. The resulting scaled counts reflect whether a codon is translated quickly or slowly in the context of our gene. We then train a feedforward neural network

model to predict scaled counts at the A site codon as a function of that codon's sequence neighborhood.

Chapter 2

Methods

2.1 Data Processing and Mapping

We obtained the ribosome profiling data set published in Weinberg *et al.*[36]. This data set has been used in several analyses because it is relatively large and high quality[27, 23]. It contains about 70 million footprints after quality filtering and mapping to a transcriptome. It also uses a version of the experimental protocol that has comparatively low bias in the footprints it recovers by using a less biased ligase in the sequencing library preparation than is commonly used. The model organism is yeast (*S. cerevisiae*), which is convenient to work with because most of its genes produce only a single protein, compared with more complex eukaryotes that frequently produce a number of different mRNA transcripts and proteins from the same gene. Consequently, using yeast limits the challenge of footprints that map ambiguously to multiple places in the transcriptome.

We used custom scripts to trim sequencing adapters off of the ends of our sequenced footprints, as specified in Weinberg *et al*. Then we used the Bowtie package to map our reads to an annotated collection of transcripts for *S. cerevisiae*, which we refer to as a transcriptome [20]. This transcriptome was acquired from the Saccharomyces Genome Database and filtered to exclude mitochondrial genes[7]. When mapping with Bowtie, we required footprints to align to a transcript with no mismatches, and allowed for multiple mappings of a given footprint. Where a footprint mapped to multiple locations in the transcriptome, we allocated its mapping weight uniformly over all identified mapping sites.

2.2 A Site Assignment

In order to compute a ribosome profile for each gene, we first need to assign each footprint to an A site codon. The A site is a more natural reference position for the location of a footprint than the termini of the footprint, because the termini can vary with the ef-

ficiency of the digestion that created footprints out of full mRNAs. Overall this digestion is quite efficient, but it can vary by a small number of nucleotides (typically 1-2 nt.) at both the 5′ and 3′ ends of the footprint[18]. Consequently, for a set of ribosomes decoding the same A site codon, we can recover corresponding footprints that vary in the position of their 5′ and 3′ ends. If we can accurately assign an A site codon, this serves as a more consistent and biologically meaningful reference position.

Fortunately, nuclease digestion is generally quite efficient, and has resolution below the width of one codon. This allows us to assign A sites unambiguously for almost all footprints, with a simple assignment procedure. By examining footprints at the beginning of a coding sequence, we can unambiguously identify the distance between the 5′ end of the footprint and the location of the first A site codon that is generating footprints[21, 35]. For a canonical 28 nucleotide footprint with its 5′ end mapping in the 0th frame, the A site is found 15 nucleotides downstream of the 5′ end (Figure 1.4). We can apply this procedure similarly to other footprint sizes and mapping frames for the 5′ end. This generates a set of offset rules for each size and frame class. 3′ offset rules can be determined for each class by subtracting the lengths of the 5′ offset and A site. We can determine a set of legal 5′ and 3′ offsets that comprise most of our observed footprint data. Our only restriction is that one of these sets must contain no more than 3 consecutive lengths, otherwise A site assignment is ambiguous. In practice this is typically not a limitation, as digestion is fairly efficient. We filter out footprints in size and frame classes for which we do not have an A site assignment rule.

2.3 Ribosome Profiles

Once we have assigned an A site in each of our footprints, we are ready to compute ribosome profiles for each gene. The ribosome profile of a gene is our basic data type for the regression problem that we have posed. We first compute the number of footprints with their A site located at each codon in a given transcript. A histogram of these counts per codon is the ribosome profile for the transcript. We have to apply an additional data transformation to our ribosome profiles in order to build one predictive model that will apply across a range of genes. Each gene has a distinct mRNA expression level, and can have variable rates of loading ribosomes onto mRNA transcripts. Each of these factors has the result of scaling the absolute count of footprints per codon up or down. In order to compare data between genes, we need to control for these sources of variability. We accomplish this by computing the mean counts per codon over the coding sequence of the gene, and then dividing the counts at each codon by the mean counts for that gene. This has the effect of mean centering the counts on each gene around 1, and renders high and low counts more comparable to each other across genes. As a note, when we compute these profiles, we exclude the first and last 20 codons in each coding sequence from all

calculations. This is because of the observation that these regions often contain skewed counts due to artifacts in the experimental procedure or perhaps differences in the translation process in these regions[18, 30, 13].

After rescaling each ribosome profile to have equal mean counts, we refer to our counts per codon as scaled counts. We should interpret these scaled counts as measuring whether a codon is fast or slow in the context of its gene. A key assumption in approaching this problem is that a fast codon in the context of one gene is similar to a fast codon in another gene. Equivalently, we assume that there is a similar distribution of fast and slow codons in each gene. There is some evidence that this assumption is not true across the range of expression levels. Specifically, higher expression genes tend to be more optimized in our codon usage. However, we have shown that this assumption does not affect the performance of our model over the range of expression levels. The quality of our predictions is consistent over the full range of expression levels, when controlling for abundance of data. This assumption is necessary in order to aggregate data from multiple genes, and consequently it is made by all of the available models approaching this data.

2.4 Input Features

Early work connecting distributions of ribosome density with sequence features mostly focused on the A site as a predictive feature[1, 21, 32]. The A site codon identity is generally the most informative feature related to observed footprint counts at that codon[27, 23, 35]. However, this is not the only important predictive feature. More recently, models have used an expanded sequence neighborhood and achieved greater success at predicting ribosome density. In some experiments, proximal sites like the P site can provide comparable predictive information. In fact, where an expanded sequence neighborhood is used, all of the sites within the margins of a ribosome have been shown to improve predictive performance[35]. In many experiments, the termini of the ribosome footprints are also important features, sometimes on par with the A site itself. These observations reflect a combination of the biological significance of the sequence neighborhood, and also features of the experimental protocol. For example, sites proximal to the A site contain information about the nascent peptide and its biochemical properties[12, 9]. These affect translocation of the peptide through the ribosome, and consequently translation elongation rates. Neighboring sequence may also contain important information when ribosomes are poorly arrested during mRNA recovery and digestion. In this case, the ribosomes may continue to translocate or shift along the RNA during the experimental procedure. Fragment ends contain predictive information due to a technical bias where ligases used in library preparation preferentially react with certain end sequences[14, 19, 35]. Consequently footprints are recovered at variable rates as a function of their fragment ends. Taken together, the current state of the field is to use a sequence neighborhood

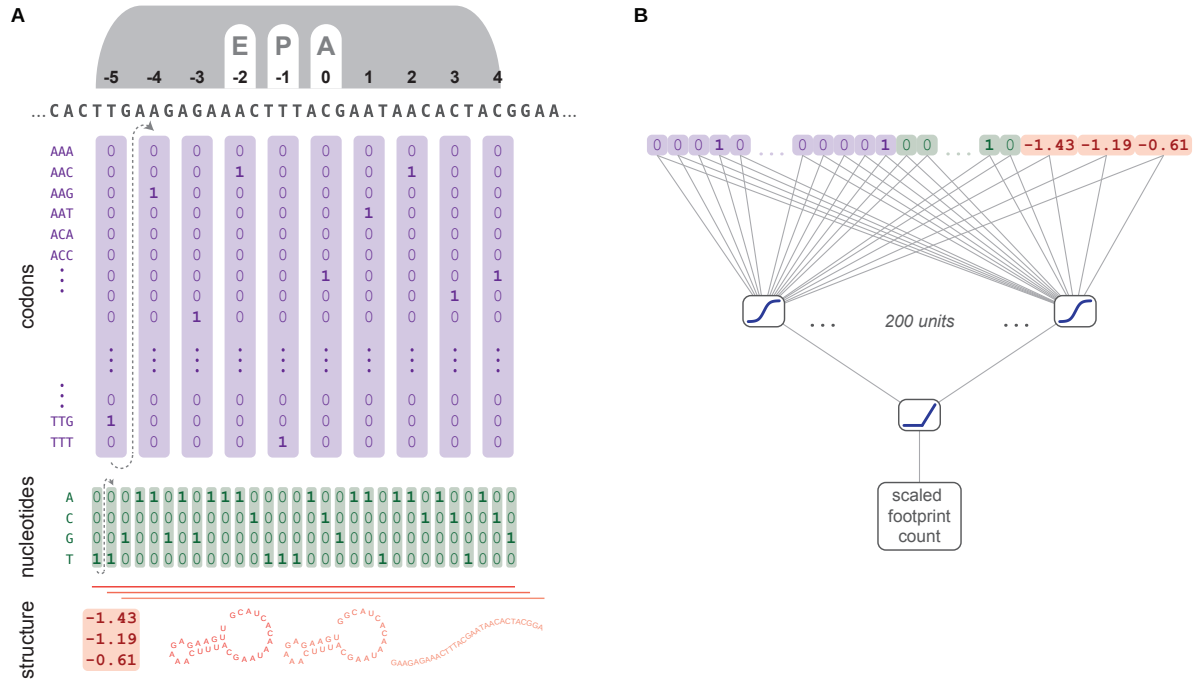

Figure 2.1: Diagram of a neural network model for ribosome density. **A** Conversion of a sequence neighborhood into input features. In gray, the span of a ribosome, annotated with E/P/A sites and codon indexes. Purple, one-hot encoding of a codon neighborhood from positions -5 to +4. Green, dual encoding of the same sequence neighborhood at the nucleotide level. Red, predicted structure scores for three sliding windows over this sequence neighborhood. **B** Sample neural network model. The input vectors (green, purple, red) are concatenated and fed into the input layer. Model depth and width can vary. We use tanh activation functions at hidden layers, and a ReLU activation function on the output to constrain predicted scaled counts as nonnegative.

roughly coterminal with a typical 28-30 nt. footprint as the predictive region for counts at the A site. We choose a 30 nt. neighborhood from 5 codons upstream of the A site to 4 codons downstream of the A site, which includes all codons contained within the ribosome.

We encode this sequence for input in a neural network via one-hot encoding. We divide the sequence neighborhood into codons, and perform one-hot encoding on each of these codons. We also observe improved performance in the model when we perform a second encoding of the individual nucleotides, and feed these features into our model as well (Figure 1.5). The advantage of this dual encoding reflects the fact that sequence can encode predictive information at each of these levels. Biological features are more

precisely represented by codons, which correspond to tRNAs and amino acids interacting within the ribosome. In contrast, nucleotides are a more relevant semantic unit for ligation biases. Ligases interact with substrate sequences without any sense of frame or codon interactions. While these two encodings are formally equivalent, we find in practice that this featurization strategy improves predictive performance.

Finally as a matter of notation, we index our codon and nucleotide features relative to the A site. The A site codon is indexed as the 0th codon, and its first nucleotide is indexed as the 0th nucleotide. Codons and nucleotides upstream (5′) of these positions are indexed with negative numbers, and downstream (3′) features are indexed with positive numbers.

In addition, we include a set of RNA structure scores computed on the same sequence neighborhood. It has been shown that there is predictive information in the structure of the ribosome footprint itself. We hypothesize that this information influences footprint counts through recovery biases, by affecting whether the footprint can successfully undergo steps in the recovery protocol. To incorporate structure scores in our model, we first take three sliding 30 nucleotide windows starting 17, 16, and 15 nucleotides upstream of the A site, and compute RNA structure scores on them with the RNAfold package (Figure 1.5)[24]. We choose these regions for structure prediction because they surround the span of a canonical 28 nucleotide footprint, and they allow for some variability in the digestion of the fragment ends. We then input these structure scores as regular features along with our one-hot sequence vectors.

2.5 Model Architecture and Training

Our general model architecture fell into the class of feed forward artificial neural networks. Each model took 763 codon, nucleotide, and structure features as input. The depths and widths of the models were left as free parameters for model selection. The hidden layers in each model used a tanh activation function. Output layers, which were all a single unit, used a ReLU activation function. We applied this function to the output to constrain our predicted scaled counts to be nonnegative.

We restricted our data set to the top 600 genes by data density for model training and evaluation purposes. Under our formulation of the regression problem, each training, test, and validation point in the model is a pair consisting of a sequence neighborhood around a codon and the scaled count of footprints with their A site at that codon. This data set has the special property that the total number of possible data points is fixed by the size of the transcriptome. When we collect more footprints with a larger sequencing experiment, we do not increase the number of data points in our model, but rather the quality of our data points. This is because a large profiling experiment will result in a

larger number of raw footprint counts per codon, and a lower variance in scale count measurements. If we restrict ourselves to high coverage genes, we expect to train our model on more reliable data. Consequently, we chose the top 600 genes as ranked by average raw footprint counts per codon. We randomly sorted these genes into a validation set of 100 genes and a training and test set of 500 genes. We divided the training and test set into thirds, and performed three-fold cross validation to evaluate model training hyperparameters. We then evaluated and present model performance on the validation set genes.

All models were created with the Python packages Lasagne v. 0.2.dev and Theano v. 0.9.0[2, 33]. Models were trained with minibatch stochastic gradient descent (batch size 500) and nesterov momentum parameter 0.9.

Chapter 3

Results

3.1 Model dimensions

Our first model selection challenge was to determine an appropriate width and depth for the neural network. We started by training a series of models with between one and four hidden layers. Each of these layers ranged from ten to two hundred hidden units in width. We trained the models with minibatch gradient descent (batch size 500) and a squared error loss function, and selected a number of training epochs by overtraining and selecting the epoch with the lowest average squared error across each cross-validation series. Model performances are presented in tables 3.1-3, measured by mean squared error, and Pearson and Spearman correlations between true and predicted scaled counts on the validation set codons.

Model performance is generally decreasing with model depth and increasing with model width. Improvement at greater widths suggests that there is a large number of relevant features within the sequence neighborhood that contain information for predicting scaled footprint counts. Decrease in performance with increased depths suggests that the model is not benefiting from combining these features in increasingly complex ways. This could also suggest that deep models are overly complex for the data, particularly for the larger widths. We also observed that deeper models learn predictive features in the data with fewer iterations over training data. Our one layer models required an average of 95 training epochs, whereas the four layer models were trained in an average of 30 epochs. In contrast, models of equal depth required similar numbers of training epochs, irrespective of their width. We chose a model architecture with two hidden layers of 200 hidden units to proceed. This architecture was consistently the best performing across our correlation and error metrics.

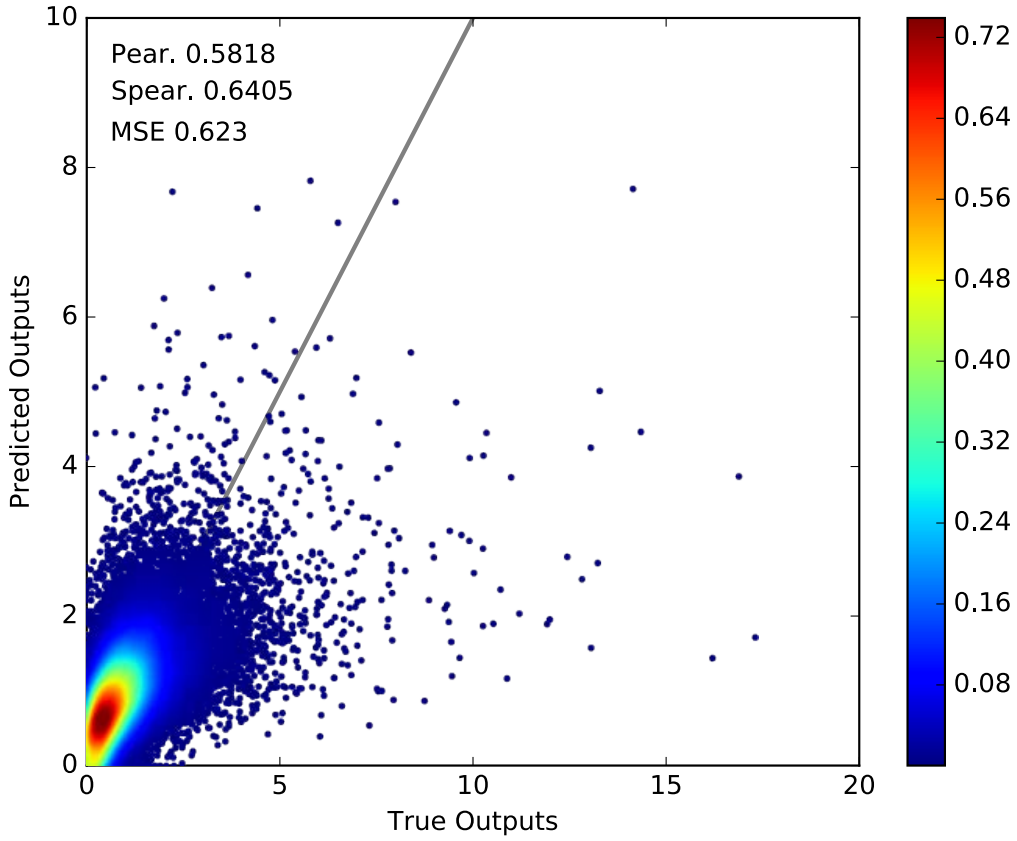

Figure 3.1: Validation set true vs. predicted scaled counts. After model selection, we chose a neural network model with 2 hidden layers, each of 200 units. Color bar indicates data density in arbitrary units. In gray, the identity line. Perfect predictions fall on the identity line. The data cloud lies along this line, although there is substantial variation.

Table 3.1: MSE by Model Dimensions

Depth	Width 10	Width 25	Width 50	Width 100	Width 200
1	0.6467	0.6416	0.6303	0.6292	0.6280
2	0.6643	0.6377	0.6347	0.6376	0.6230
3	0.6686	0.6559	0.6393	0.6404	0.6365
4	0.6728	0.6535	0.6571	0.6554	0.6379

Table 3.2: Pearson Correlation by Model Dimensions

Depth	Width 10	Width 25	Width 50	Width 100	Width 200
1	0.5546	0.5616	0.5734	0.5758	0.5769
2	0.5375	0.5627	0.5702	0.5730	0.5818
3	0.5351	0.5480	0.5645	0.5684	0.5688
4	0.5353	0.5474	0.5500	0.5544	0.5620

Table 3.3: Spearman Correlation by Model Dimensions

Depth	Width 10	Width 25	Width 50	Width 100	Width 200
1	0.6109	0.6196	0.6314	0.6311	0.6356
2	0.6013	0.6206	0.6275	0.6305	0.6405
3	0.6005	0.6165	0.6255	0.6313	0.6290
4	0.6027	0.6082	0.6129	0.6212	0.6229

3.2 Model Performance

We show the predictions of our chosen model in Figure 3.1. There is a fair amount of noise in predictions, but overall there is a clear trend between the true and predicted scaled counts. The data cloud lies along the identity line, which indicates perfect predictions. For this model, we observe a Pearson correlation of 0.58, and a Spearman correlation of 0.64. This performance is fairly high in the context of low frequency genomic count data. The correlations are comparable with our previous analysis of a neural network regression model on a per-gene basis, and substantially outperform both RUST and riboshape on this data.

We also observe that our model has difficulty fitting very large true scaled counts values. These points represent codons where we have sampled a large number of footprints relative to other codons in their gene. This indicates that these codons may be sites of very

slow translation. Translation stall sites are often of interest when studying regulation, and may reflect rare or idiosyncratic regulatory events that are not captured consistently in our input data[17]. These points would be better captured by ROSE, which is specifically designed to identify translation stall sites. This suggests a possibility of a two stage model, first classifying with ROSE, and then predicting scaled counts on non-stall sites with our model. It is also possible that high counts at these positions are a function of technical artifacts that are not captured in our input features. For example, our model does not explicitly account for the exponential amplification (copying) of footprint data during sequencing library preparation. It is possible that some of these high scaled count values are a function of uneven amplification, and could be corrected with a deduplication protocol

Plotting squared error as a function of the true scaled counts, we observe that the quality of our predictions is consistent for true scaled counts values between 0 and 2 (Figure 3.2). Over this set, which comprises 90% of the data, mean squared error is 0.2626, compared with an overall mean squared error of 0.6230. About 58% of our error is derived from codons with true scaled counts above 2, or codons that are more than twice as slow as average codons on their gene.

3.3 Poisson Error Model

We evaluated alternative error models to see if these could remediate some of the challenges that our model experienced in predicting scaled counts. In particular, we reasoned that a Poisson error model might be more suitable for our data. Our formulation of this system involves a small rate parameter at each codon that indicates the probability of generating a footprint from that codon, conditional upon generating a footprint within that codon's gene. The total counts generated at each footprint are low in most genes. This type of data is commonly modeled with a Poisson distribution. In contrast, the squared error loss function that we and others have used is implied by Gaussian errors. We extended lasagne to include a Poisson loss function, and trained a model dimension series under this loss criterion. Performance metrics on these models are listed in tables 3.4-6.

Overall we observe similar but slightly decreased performance when training our model series under a Poisson loss criterion. This was surprising to us, as this criterion is better suited to our data model. However, we did observe a decrease in positive outliers in our predictions, and also improved performance over the intermediate range of data points. Performance decreased in the domains of very high and low scaled counts, suggesting that these regions are less well modeled by a Poisson distribution.

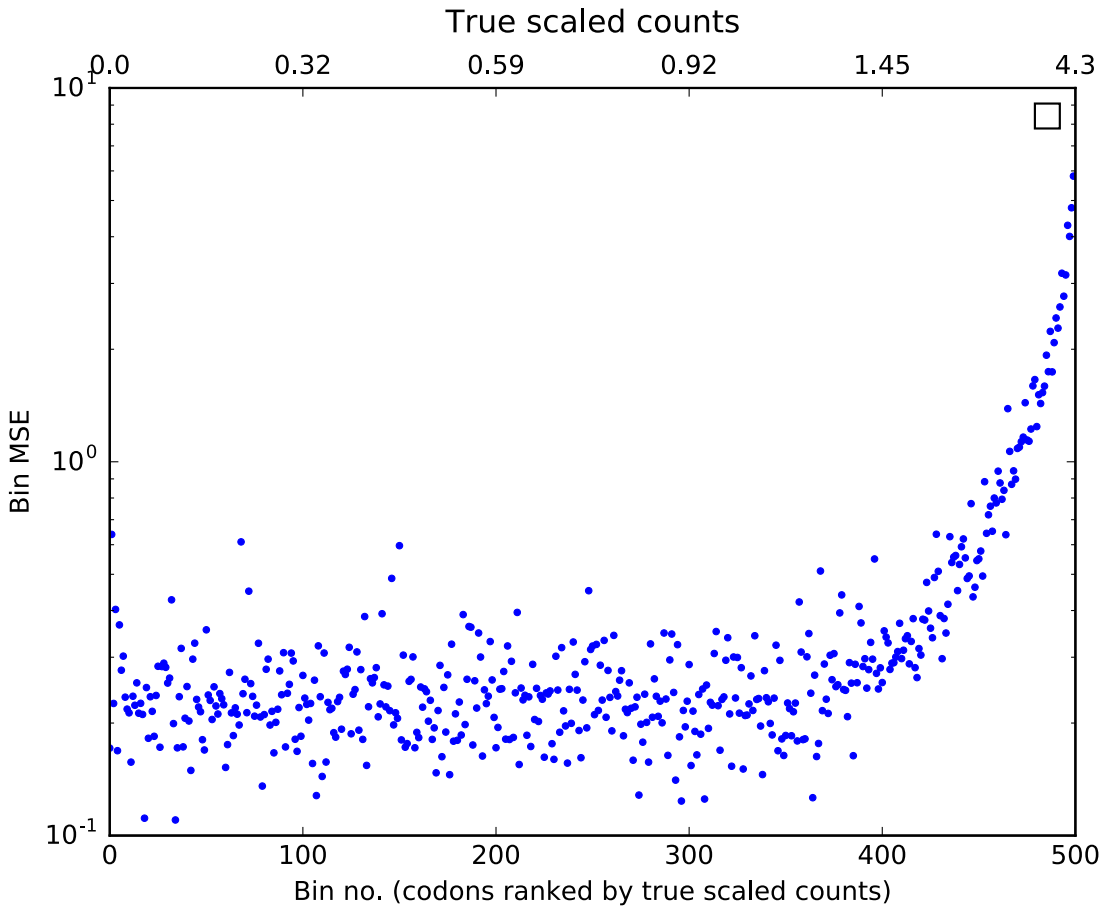

Figure 3.2: Mean squared error of codons binned by true scaled count values. Validation set codons are ranked by their true scaled counts, and split into 500 bins. A mean squared error is computed for each bin and displayed. Mean squared error is distributed similarly for codons with true scaled counts from 0-2, and steadily decreases past this domain.

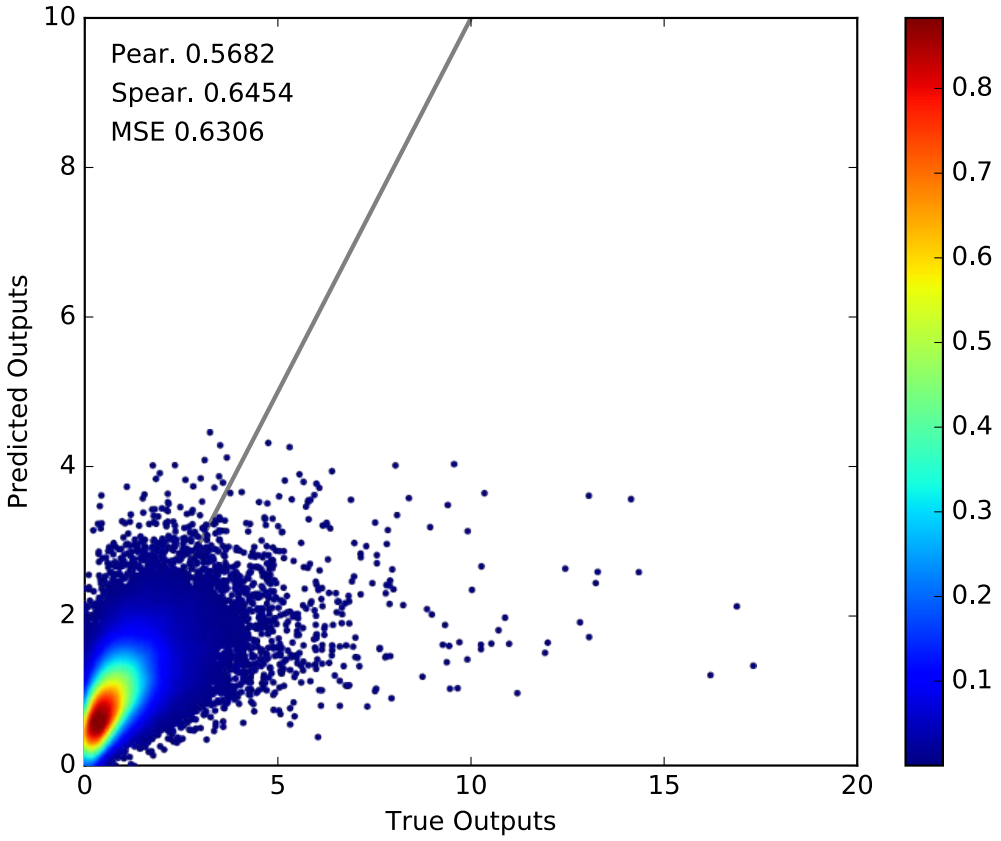

Figure 3.3: Validation set true vs. predicted scaled counts, Poisson loss criterion. After model selection, we chose a neural network model with 2 hidden layers, each of 200 units. Color bar indicates data density in arbitrary units. In gray, the identity line. Perfect predictions fall on the identity line. The data cloud lies along this line, although there is substantial variation.

Table 3.4: Poisson Loss MSE by Model Dimensions

Depth	Width 10	Width 25	Width 50	Width 100	Width 200
1	0.6809	0.6636	0.6625	0.6607	0.6563
2	0.6796	0.6666	0.6557	0.6593	0.6515
3	0.6670	0.6565	0.6561	0.6539	0.6642
4	0.6757	0.6631	0.6582	0.6514	0.6658

Table 3.5: Poisson Loss Pearson Correlation by Model Dimensions

Depth	Width 10	Width 25	Width 50	Width 100	Width 200
1	0.5184	0.5380	0.5376	0.5404	0.5448
2	0.5205	0.5348	0.5450	0.5443	0.5484
3	0.5300	0.5436	0.5443	0.5462	0.5391
4	0.5240	0.5370	0.5423	0.5520	0.5351

Table 3.6: Poisson Loss Spearman Correlation by Model Dimensions

Depth	Width 10	Width 25	Width 50	Width 100	Width 200
1	0.6002	0.6008	0.6212	0.6263	0.6279
2	0.5968	0.6062	0.6242	0.6245	0.6305
3	0.6069	0.6171	0.6192	0.6235	0.6180
4	0.5954	0.6105	0.6151	0.6245	0.6166

3.4 Testing for Overfitting

A general concern in developing neural network regression functions is that they are subject to overfitting. In our model dimensions series, we observed that the predictive performance of the models decreased at higher depths. This suggests that a deep model may be overfitting on the training data and failing to generalize to other genes. As a first test, we compared the validation and test errors across our model dimensions series (Table 3.7). The validation error was consistently higher than the training error, but there was no clear trend in the error over the range of model depths and widths. This suggested that wider and deeper models were not particularly overfitting the data.

As an additional test, we applied L2 regularization to our model weights. In the original model dimensions series, we chose a model of width 200. Reasoning that the widest models are also the most likely to overfit, we trained a series of models with width 200

Table 3.7: Increase in Validation Error over Test Error

Depth	Width 10	Width 25	Width 50	Width 100	Width 200
1	0.0295	0.0294	0.0309	0.0259	0.0288
2	0.0283	0.0290	0.0242	0.0325	0.0260
3	0.0241	0.0280	0.0324	0.0281	0.0342
4	0.0345	0.0292	0.0269	0.0262	0.0286

and depths of 1 to 4 hidden layers, with a series of L2 regularization parameters. Validation error increased after regularization in some cases, and decreased in others. The most consistent trend was that regularization improved model performance for the one layer models. After regularization, each of the one layer models performed better across all of our performance metrics, and the one layer model with a regularization parameter of 10^{-4} was the best performing overall.

Table 3.8: Change in Error after Regularization

Depth	$\lambda = 10^-7$	$\lambda = 10^-6$	$\lambda = 10^-5$	$\lambda = 10^-4$
1	-0.0025	-0.0050	-0.0074	-0.0091
2	0.0140	0.0048	0.0073	0.0073
3	0.0057	0.0032	-0.0073	-0.0073
4	0.0073	0.0259	0.0204	-0.0022

3.5 Coding Sequence Optimization

Having developed a model that predicts local translation rates, we applied this model to optimize coding sequences globally. There is considerable interest in designing coding sequences for optimal expression, both for biosynthetic applications and for understanding selective constraints on endogenous coding sequences[5, 10]. Most analyses have used fairly simple approaches that estimate the optimality of individual codons based on abundance in high expression genes, or tRNA copy number, and then test the substitution of a mix of preferred codons in a coding sequence [31]. We reasoned that our model could capture more information about the optimality of a whole sequence neighborhood, and could perform global optimization by determining the best set of overlapping sequence neighborhoods for fast or slow translation.

A key assumption of our optimization approach is that the defined sequence neighbor-

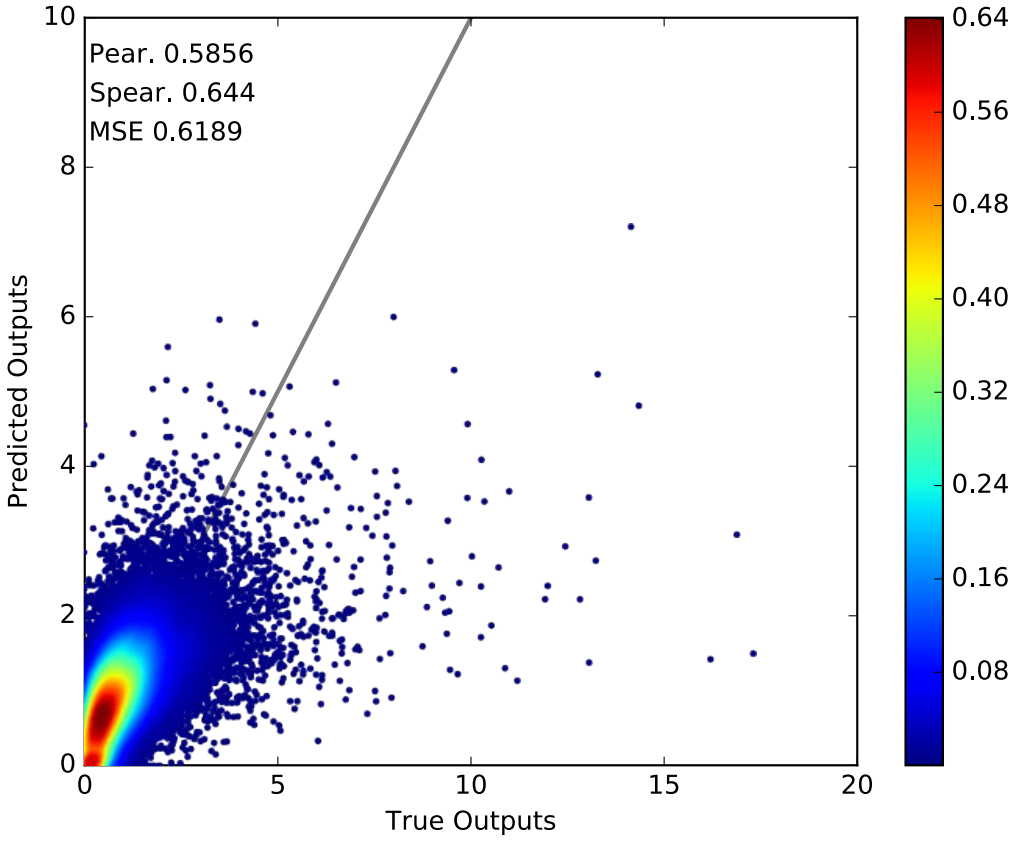

Figure 3.4: Validation set true vs. predicted scaled counts for model with L2 weight regularization. After testing a series of models with L2 regularization, we chose a neural network model with 1 hidden layer of 200 units, and a reguarization parameter of 10^{-4}. Color bar indicates data density in arbitrary units. In gray, the identity line. Perfect predictions fall on the identity line.

hood of a model is the relevant predictive region for local translation rates. We know that this assumption is not strictly true. For example, it has been shown that distant upstream sequence can affect translation elongation rates due to interactions between the growing protein and the ribosome's exit tunnel, and also due to stalls in protein folding events after the growing chain exists the ribosome[37, 9]. Nevertheless, this approach is likely to benefit from increased information in regions proximal to the A site, relative to existing approaches. The choice of sequence neighborhood is very important under this assumption. We should exclude sequence from the flanking ends of a ribosome footprint (codons -5, -4, +3, +4) when training our model, so that our model does not learn experimental artifacts due to ligation. In addition, we should apply a bias correction protocol to remove technical biases from our profiling data before model training. This will allow our model to learn biological relationships between sequence and translation rates rather than technical artifacts, and to improve the quality of sequence optimization.

The sequence neighborhood assumption allows us to develop a simple and efficient algorithm for coding sequence optimization. Formally our problem is to find the coding sequence for a given protein that will translate as fast or as slow as possible overall. We define the total translation time for a protein as the sum of the individual translation times at each codon, which is proportional to the scaled counts at each codon. Our goal is to optimize this quanitity over the search space of legal coding sequences for a protein. Each protein is a linear sequence of amino acids, typically on the order of hundreds of amino acids. We define the number of amino acids in a protein as length L. Each of these subunits can be encoded by between one and six codons. As a result, there is an exponential space of legal coding sequences for a given protein, on the order of 3^L. However, by assuming that the sequence neighborhood is the relevant context for predicting local translation rates, we reduce our problem to an (n-1)th order Markov chain, where n is the length of the sequence neighborhood in codons. Thus we can globally optimize the coding sequence a protein under our trained neural network model, using a simple dynamic programming algorithm specified below. This algorithm runs in $O(n^2L)$ time.

We tested this optimization strategy by designing a series of coding sequences for a yellow fluorescent protein, eCitrine. These coding sequences ranged from the fastest predicted coding sequence under a trained neural network model to the slowest predicted coding sequence, with a set of randomly generated intermediate values included to form a series. We cloned these sequences into yeast, and observed that the quantity of protein produced per unit of mRNA corresponded extremely well with our predicted total translation time per gene. This indicated that our neural network model successfully captured sequence features determining translation rates, and that optimization of these translation rates has a meaningful effect on the efficiency of protein production. See Tunney *et al.*, 2017 [35].

Translation rate optimization algorithm

L = length of coding sequence in codons
A = amino acid sequence of protein
$A[m:n]$ = slice of A from positions m to n, 1 indexed. Negative indices count from end.

i = index over A site codons in coding sequence
c_{rel}^{min} = min. index of a codon in the sequence neighborhood, relative to A site (e.g., -3)
c_{rel}^{max} = max. index of a codon in the sequence neighborhood, relative to A site (e.g., 2)

$\phi(a)$ = function that returns the set of synonymous codons for amino acid a
$\xi([a_1, a_2, \ldots, a_n])$ = $\phi(a_1) \times \phi(a_2) \times \cdots \times \phi(a_n)$

f = prediction function of the neural network model

Algorithm 1 Calculate fastest codon sequence under a predictive model†

for $i \in \{1 \dots L\}$ **do** ▷ Populate table with predicted translation rates
  c_i^{min} = max(1, i + c_{rel}^{min}) ▷ For each sliding sequence neighborhood
  c_i^{max} = min(i + c_{rel}^{max}, L)
  $Q_i = \xi(A[c_i^{min} : c_i^{max}])$ ▷ Get sequences encoding correct amino acids
  for $q \in Q_i$ **do** ▷ For each legal sequence in neighborhood
    $T_{i,q} \leftarrow$ f(q)†† ▷ Compute predicted translation rate under neural net
  end for
end for
for $i \in \{1 \dots L\}$ **do** ▷ For each sliding sequence neighborhood
  if $c_{min}^i == 1$ **then** ▷ If at beginning of protein sequence
    for $q \in Q_i$ **do**
      $P_{i,q} \leftarrow$ None ▷ Store null backtrack pointer
      $V_{i,q} \leftarrow T_{i,q}$ ▷ Store own predicted translation rate in DP table
    end for
  else if $c_{min}^i > 1$ **then** ▷ If not at beginning of protein sequence
    for $q \in Q_i$ **do** ▷ For each legal sequence in neighborhood
      $P_{i,q} \leftarrow \underset{p \in \phi(A[c_{min}^i - 1]) \times q[:-3]}{\text{argmin}}$ V[i-1](p) ▷ Find min. compatible sequence in
      $V_{i,q} \leftarrow V_{i-1,P_{i,q}} + T_{i,q}$ ▷ previous position
    end for ▷ Add own score to min previous, store in DP table
  end if
end for
$q_L = \underset{q \in Q_L}{\text{argmin}} V_{L,q}$ ▷ Find min. final score
i = L; $q_i = q_L$; cds = q_L
while $c_{min}^i > 1$ **do** ▷ Backtrack to recover optimal sequence
  i -= 1
  $q_i \leftarrow P_{i,q}$
  cds $\leftarrow q_i[:3]$ + cds
end while
return cds

† To calculate slowest sequence, change argmins to argmax

†† If the sequence neighborhood is truncated because it runs outside of the coding sequence, we input this part of the neighborhood to our model as all 0 values (i.e. no codons are encoded as 1)

Bibliography

[1] 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.

[2] E Battenberg et al. *Lasagne: First release*. Aug. 2015.

[3] Gloria A Brar and Jonathan S Weissman. "Ribosome profiling reveals the what, when, where and how of protein synthesis". en. In: *Nat. Rev. Mol. Cell Biol.* 16.11 (Nov. 2015), pp. 651–664.

[4] Patrick O Brown and David Botstein. "Exploring the new world of the genome with DNA microarrays". In: *Nature genetics* 21.1s (1999), p. 33.

[5] Nicola A Burgess-Brown et al. "Codon optimization can improve expression of human genes in Escherichia coli: A multi-gene study". In: *Protein expression and purification* 59.1 (2008), pp. 94–102.

[6] 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.

[7] J. Michael Cherry et al. "SGD: Saccharomyces Genome Database". In: *Nucleic Acids Research* 26.1 (1998), pp. 73–79. DOI: 10.1093/nar/26.1.73. eprint: /oup/backfile/content_public/journal/nar/26/1/10.1093_nar_26.1.73/2/26-1-73.pdf. URL: http://dx.doi.org/10.1093/nar/26.1.73.

[8] 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.

[9] 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.

[10] D Allan Drummond and Claus O Wilke. "Mistranslation-induced protein misfolding as a dominant constraint on coding-sequence evolution". In: *Cell* 134.2 (2008), pp. 341–352.

[11] Khanh Dao Duc and Yun S Song. "Identification and quantitative analysis of the major determinants of translation elongation rate variation". en. Mar. 2017.

[12] 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.

[13] Maxim V Gerashchenko and Vadim N Gladyshev. "Translation inhibitors cause abnormalities in ribosome profiling experiments". In: *Nucleic acids research* 42.17 (2014), e134–e134.

[14] 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.

[15] Daniel Horspool. *Central Dogma of Molecular Biochemistry with Enzymes*. 2008. URL: `https://commons.wikimedia.org/wiki/File:Central_Dogma_of_Molecular_Biochemistry_with_Enzymes.jpg`.

[16] 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.

[17] Nicholas T Ingolia. "Ribosome profiling: new views of translation, from single codons to genome scale". In: *Nature Reviews Genetics* 15.3 (2014), p. 205.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] Tzu-Yu Liu and Yun S Song. "Prediction of ribosome footprint profile shapes from transcript sequences". In: *Bioinformatics* 32.12 (2016), pp. i183–i191.

[24] Ronny Lorenz et al. "ViennaRNA Package 2.0". In: *Algorithms for Molecular Biology* 6.1 (2011), p. 26.

[25] Evan Z Macosko et al. "Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets". In: *Cell* 161.5 (2015), pp. 1202–1214.

[26] Ali Mortazavi et al. “Mapping and quantifying mammalian transcriptomes by RNA-Seq”. In: *Nature methods* 5.7 (2008), p. 621.

[27] 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.

[28] 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.

[29] 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.

[30] Premal Shah et al. “Rate-limiting steps in yeast protein translation”. en. In: *Cell* 153.7 (June 2013), pp. 1589–1601.

[31] 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.

[32] 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.

[33] The Theano Development Team et al. “Theano: A Python framework for fast computation of mathematical expressions”. In: (May 2016). eprint: 1605.02688.

[34] Cole Trapnell et al. “Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks”. In: *Nature protocols* 7.3 (2012), p. 562.

[35] Robert J Tunney et al. “Accurate design of translational output by a neural network model of ribosome distribution”. In: *bioRxiv* (2017), p. 201517.

[36] 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.

[37] Daniel N Wilson and Roland Beckmann. “The ribosomal tunnel as a functional environment for nascent polypeptide folding and translational stalling”. In: *Current opinion in structural biology* 21.2 (2011), pp. 274–282.

[38] Sai Zhang et al. *ROSE: a deep learning based framework for predicting ribosome stalling*. 2016.