Generative models for codon prediction and optimization

David K. Yang [*] [1] **Samuel L. Goldman** [*] [2] **Eli Weinstein** [3] **Debora Marks** [3] [4]

Abstract

Optimizing foreign DNA sequences for maximal protein production in a specified host organism is an important problem for synthetic biology and biomanufacturing. Experimental results have demonstrated that simply interchanging *codons*, triplets of three DNA bases, with synonymous alternatives can in fact amplify protein production several-fold while holding the produced protein constant. Previous methods for codon optimization are frequency based, which cannot consider factors such as RNA secondary structure that contribute to protein expression. Here, we apply a deep learning framework to model the distribution of codons in highly expressed bacterial and human transcripts. We show that our LSTM-Transducer model is able to predict the next codon of a genetic sequence with improved accuracy and lower perplexity on a held out set of transcripts, outperforming the previously state of the art frequency-based approach to modeling codon distribution.

1. Introduction

The ability to express recombinant proteins, proteins artificially designed or extracted from a different species, has been pivotal in all of biology. Expression of recombinant proteins in different host organisms has applications from basic research to industrial production of therapeutics (Xiao et al., 2014). Naturally, increasing the expression level of these recombinant proteins in a host organism such as *E.coli* is highly relevant as it can allow for more robust experimentation and more efficient biomanufacturing. Outside of manufacturing, codon optimization in humans will be important for the next wave of gene therapies (Mauro & Chappell, 2014).

Traditional biological approaches to increase the expression of recombinant proteins have included selecting strong promoters, enhanced cell proliferation, and codon optimization (Xiao et al., 2014; Sivashanmugam et al., 2009; Rosano & Ceccarelli, 2014; Zhou et al., 2016). In this work, we focus on codon optimization. Despite the importance of codon optimization, computational methods for this task remain naive and *ad hoc*, unable to capture deeper complexities of a given DNA sequence such as mRNA folding that have recently been shown to influence the optimal choice of codons (Kudla et al., 2009; Cambray et al., 2018; Goodman et al., 2013).

In this work, we propose combining an encoder network on the entire sequence of amino acids with a neural language model (NLM) to capture the native distribution of codon choice in highly expressed genes of a host organism. This yields higher accuracy and lower perplexity than the simpler, frequency-based approach.

2. Related Work

Related computational methods to perform codon optimization rely primarily upon the natural codon usage bias of the target organism. Methods like “Optimizer” finds the codon frequencies present in highly expressed genes to construct a “codon adaptation index” (CAI), which is then used to determine the codons (Puigbo et al., 2007a).

A recent platform, “Presyncodon” uses machine learning, specifically a local random forest classifier, to predict codons. They use their classifier to predict the middle amino acid in a sequence of up to 7 amino acids in *E. coli* (Tian et al., 2017). While the authors report a 97% accuracy in predicting the correct codon choice when trained with 64 available *E. coli* genomes on a held out genome, they do not exclude homologous genes from the same species in their test set nor use a non-local model. Still, these groups of frequency based methods are limited to consider only limited local context, motivating our use of N-gram baselines that consider only the N nearest amino acids to predict the codon of interest (Brown et al., 1992).

To our knowledge Fujimoto et al. present the only prior work that uses neural networks for the task of modeling natural codon distributions (Fujimoto et al., 2017). They

[*] Equal contribution [1] Harvard University [2] MIT Computational and Systems Biology [3] Harvard Medical School Department of Systems Biology [4] Broad Institute of Harvard and MIT. Correspondence to: David Yang <yangd@college.harvard.edu>, Samuel Goldman <samlg@mit.edu>.

frame codon optimization as a neural machine translation problem and broadly predict the codons related to all *E. coli* genomes. However, in the same vein of Presyncodon, they train their model on pooled sequences of 159 different *E. coli* strands before further fine-tuning on a specific *E. coli* genomes, which produces near 100% accuracy on the downstream task. We develop a closely related neural architecture that tightly couples amino acid encodings and prediction outputs, while also following a more principled test-train split.

A key use case for codon optimization is the expression of heterologous proteins that are very different from anything in the wild-type proteome, such as expressing human proteins in bacterial cells or vice versa (Gustafsson et al., 2004). We therefore need to ensure that our method can generalize well. Previous work has focused on intra-species generalization, using homologs of a given protein found in different strains of the same species to predict codon usage in a held-out strain. Since different strains can be nearly identical at the nucleotide level and previous work fails to quantify this similarity, their training and testing sets are potentially nearly identical. The question of how well neural codon optimization methods can generalize to proteins outside the wild-type proteome therefore remains open.

Additionally, a separate line of investigation has considered the task of optimizing codons in the presence of experimental data (Gonzalez et al., 2015; Tunney et al., 2018). Unlike these approaches, we consider the case in which only have access to a set or subset of native gene sequences.

3. Methods

3.1. Model

We view codon optimization through the lens of an neural language modeling problem. Let X be a sequence of N amino acids $x_1, \ldots, x_N$ defining a single gene, where $x_i \in \mathcal{X}, |\mathcal{X}| = 20$, denoting the 20 possible amino acids. We are interested in producing codons $Y = [y_1, \ldots, y_N]$ corresponding to each position, such that the sequence Y is functionally preserving for X. We say that $y_i \in \mathcal{Y}, |\mathcal{Y}| = 64$, denoting all possible triplet permutations of the DNA nucleotides, A, T, G, C. Since we are interested in modeling the codon usage patterns of highly expressed genes, we specify z as the protein expression level. Thus, we model $P(Y|X, z)$, and assume $z = z_{max}$ by training only on a subset of highly expressed genes.

In our neural models, we want to consider both information at the codon level and at the amino acid level. The intuition behind this is that amino acid x_j can inform the codon choice for y_i for $i \neq j$. By passing contextual information from across the amino acids $[x_1, ..., x_N]$, our models can incorporate long range dependencies.

To give our models these desired properties, we use a simplified architecture inspired by the transducer model (Graves, 2012). On top of a codon-level language model that models $p(y_i|y_1, y_2, \ldots, y_{i-1})$, the transducer adds an amino-acid level positional encoding. This encoding is used in combination with the codon-level language model to predict the distribution over the next codon in sequence, taking into account both previously generated codons and the future codons that must be produced. The amino-acid level encoder is trained jointly with the language model (Figure 1). More formally, if we refer to our encoder network over the amino acid sequence, X, as $Enc(X)$, our prediction network as $Pred$, and a simple linear transformation as g, then we get our model structure:

$$p(y_t|X, y_{1:t-1}) = \text{softmax}(g(Pred(y_{1:t-1}), Enc(X)_t))$$

3.2. Encoder Network

For the encoder, we define a Bidirectional LSTM that can model long-term dependencies, and tackle the limitations of n-gram models. Specifically, we consider a BiLSTM that explicitly takes into account information of local and distant amino acids. LSTM Networks are an especially promising model choice for biological sequence modeling, which may include components such as transcription factor binding sites that modulate gene expression and involve the interaction of DNA base-pairs several tens of codons away (Sønderby et al., 2015). We choose to use bidirectional models such that the encoding produced by this architecture can capture sequence level information at the whole sequence for a single encoding position $Enc([x_1, x_2, ..., x_N])_i$.

3.3. Prediction network

Next, we define a unidirectional, autoregressive LSTM over the produced codons that is able to produce hidden hidden representations:

$$\begin{aligned} Pred([y_0, y_1, .., y_{n-1}]) &= LSTM([y_0, y_1, .., y_{n-1}]) \\ &= [h_0, h_1, .., h_{n-1}] \end{aligned}$$

Each hidden state h_i of the prediction network is concatenated with the output of the encoder network, which is fed jointly as input to a final linear layer to predict the following codon.

3.4. Dataset

We retrieve the set of highly expressed genes of Escherichia coli strain K-12 substrain MG1655 and the set of all genes from NCBI (ref seq: NC_000913.3) (Puigbo et al., 2007b). We remove four genes from the set of highly expressed

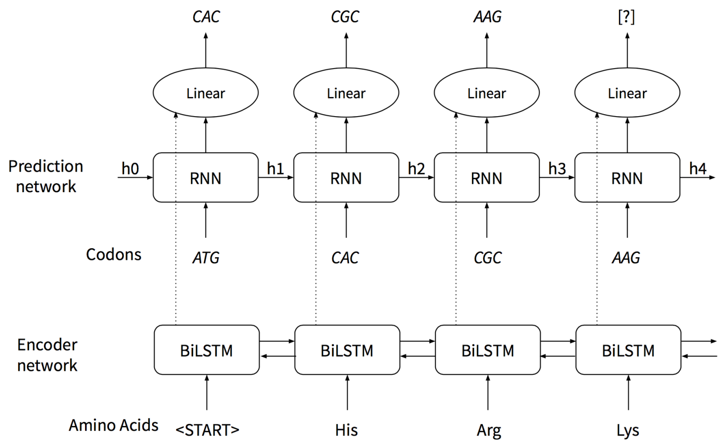

Figure 1. The codon-transducer architecture. An encoder network (BiLSTM) maps input amino acids $x_1, ...x_N$ to produce transcription vector $\mathbf{f}n$, which gets concatenated to outputs from a separate prediction network taking in previous outputs $y_{1:t-1}$ as input for language modeling.

genes not found in the current gene set, leaving a dataset of 249 highly expressed sequences with 85,899 codons (4,096 non highly expressed genes). We also retrieve a larger set of human housekeeping genes, which contains 3,769 sequences and 1,839,633 codons (Eisenberg & Levanon, 2013). Each codon was encoded in a one hot encoding of dimension 67 and the amino acids were encoded into a one hot encoding of dimension 23, with three extras ($\langle pad \rangle$, $\langle start \rangle$, and $\langle unk \rangle$) from the tokenizer. We note that we account for species-specific alternate start codons by switching the first amino acid to a $\langle start \rangle$ token. The high expression gene sequences were randomly separated into 80/20 splits and the human genes were filtered for highly similar sequences found through CD-HIT program (Li & Godzik, 2006).

3.5. Implementation Details

Because the set of highly expressed genes in *E. coli* is small, we first pretrain our *E. coli* models on the set of all non-high expression *E. coli* genes for up to 60 epochs. On the larger dataset of highly expressed human transcripts, we train for up to 60 epochs. Each model was trained with a batch size of 10. We also employ teacher-forcing for recurrent language models and codon-level masking for all models at test time because the true amino acid x_i is known at each step. Masking ensures the model is always constrained to make a choice among the corresponding codons for x_i. Loss was calculated by taking the cross entropy loss between the predicted codons and the target codons, and trained using the Adam Optimizer with default parameters.

n-grams Inspired by previous, frequency-based methods, for each n-gram model, we calculate the full distribution of the middle codon for every n gram set of amino acids. The 1-gram, "unigram" model is equivalent to choosing the most frequent codon for each amino acid in the training set. The $1, 3, 5$-gram denotes an ensemble method that weights the predictions of a unigram, trigram, and a 5-gram model for final output. The weights are identified through performing a grid search over possible weights. For this, the train set is further split into train and validation. We find $1, 3, 5$-gram weights of $[0.7, 0.3, 0]$ and $[0.4, 0.6, 0]$ perform best for *E. coli* and humans respectively.

LSTM Transducer We used a standard bidirectional LSTM architecture over the amino acid sequence as our encoder network. For both the *E. coli* and human gene models, this network has an hidden layer of size 100, 2 layers, and internal dropout of 0.1. We additionally test model performance using this encoder layer that does not depend on the set of previously outputted codons.

For the generative model over codon sequence, we use a one-directional LSTM. For the *E. coli* model, this network has an embedding dimension of size 50, hidden dimension of size 50, 1 layer, and internal dropout of 0.1. For the larger human model, we use an embedding dimension of size 50, 150 hidden dimension, 2 layers, and internal dropout of 0.1. The representation of the target amino acid and output of the generative model are concatenated. We apply dropout of 0.1 to this output, apply a linear layer, and take a softmax to generate outputs.

The models discussed in this paper were implemented using PyTorch. [1]

[1] https://github.com/samgoldman97/Codon-Optimization

	E.coli HEG				**Human HEG**			
Model	**Train Acc**	**Test Acc**	**Train Ppl**	**Test Ppl**	**Train Acc**	**Test Acc**	**Train Ppl**	**Test Ppl**
Unigram AA	0.6279	0.6246	2.7124	2.7490	0.4373	0.4313	3.6519	3.6655
Trigram AA	0.6910	0.6280	-	-	0.5060	0.4932	-	-
5-Gram AA	0.9829	0.3730	-	-	0.7951	0.3948	-	-
1,3,5-gram AA	0.6668	0.6380	2.5127	2.6786	0.5042	0.4920	3.5455	3.5718
BiLSTM Encoder Only	0.6930	0.6604	2.0004	2.1152	0.5623	0.5186	2.6189	2.7906
LSTM Transducer	0.7119	**0.6700**	1.9341	**2.0870**	0.5649	**0.5538**	2.6017	**2.6342**

Table 1. Evaluation metrics for our neural and n-gram models. AA denotes amino acid-level input. Perplexity is omitted for models that give zero probability weight to the correct codon in the test set, specifically trigram's and 5-gram's. We note that transducer architecture achieved lowest perplexity (PPL, the lower the better) and highest accuracy during testing. We include full model parameters and descriptions in Methods.

3.6. Evaluation Metrics

To evaluate performance, we compared both accuracy and perplexity (PPL) across trained models. Accuracy was calculated by generating a reverse translated DNA sequence given a protein, and examining how many of the codons were predicted correctly. The reduced perplexity with our transducer model indicates less confusion over the distribution of codons. Perplexity was computed by exponentiating average cross entropy loss for the neural models and exponentiating the average negative log likelihood for frequency baseline models. We find this to be important because in a biological context, we often desire to ancestrally sample several plausible sequences for experimentation from a distribution and experimentally test these. Finally, when evaluating accuracy and PPL on the test set, we use teacher forcing for the top level language model.

4. Results

We present our results with respect to accuracy and perplexity (PPL) in Table 1. We see that the deep BiLSTM and LSTM Transducer models both out perform the standard frequency based approaches in accuracy and perplexity on the test set, indicating that these models have learned more complex underlying features than frequency alone. Curiously, we see only marginal improvements in accuracy between the baseline and neural models, with larger boosts in perplexity. Because we evaluate our transducer model as a language model predicting next codon given the true previous codon at test time, the results are slightly biased toward this approach. Nonetheless, the BiLSTM encoder-only model demonstrates near equal performance and boosts in perplexity over the baseline methods with no knowledge of previous codons.

5. Discussion and Future Work

We introduce a generative modeling approach for codon prediction and optimization. We present an LSTM-transducer model which achieves better predictive accuracy and modeling performance on a left out test set of the *E.coli* and human sets of highly expressed genes. While we improve perplexity over these distributions substantially, improvements in accuracy are not drastic, leading us to be cautiously optimistic about the potential for neural models in this space and wonder if the selective pressure on natural codon choice is sufficient for this task.

In addition to applications for codon optimization, we also envision ways in which similar models over codon space can be used for horizontal gene transfer identification, similar to how the CAI has been used in the past (Lawrence & Ochman, 1997).

Dataset limitations A critical limitation of this work is that our training set is limited to the genes of the target organism. This problem is only exacerbated by our decision to target highly expressed subsets of genes that we believe may have a higher signal for modeling. This was particularly problematic for the small genome size of *E. coli*, which we supplemented with pretraining on the full gene set. In the future, this set of genes could be augmented with experimentally derived, non-endogenous sequences of high expression available from previous experimental projects, perhaps offering more signal (Goodman et al., 2013; Cambray et al., 2018). Modeling the untranslated region (UTR) in addition to the coding sequence could also prove important. Furthermore, experimental readouts invite additional work aimed at conditional generation models to produce codons with a desired expression.

As indicated by our results on a principled test-train split of the data, we find that the problem of modeling codon distributions remains an open question. Based on our findings, we are hopeful that the incorporation of different generative models and experimental datasets will prove useful in advancing this field.

6. Acknowledgments

We thank Alexander Rush for his input.

References

Brown, Peter F, Desouza, Peter V, Mercer, Robert L, Pietra, Vincent J Della, and Lai, Jenifer C. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479, 1992.

Cambray, Guillaume, Guimaraes, Joao C, and Arkin, Adam Paul. Evaluation of 244,000 synthetic sequences reveals design principles to optimize translation in escherichia coli. *Nature biotechnology*, 36(10):1005, 2018.

Eisenberg, Eli and Levanon, Erez Y. Human housekeeping genes, revisited. *TRENDS in Genetics*, 29(10):569–574, 2013.

Fujimoto, Masaki Stanley, Bodily, Paul M, Lyman, Cole A, Jacobsen, Andrew J, Snell, Quinn, and Clement, Mark J. Modeling global and local codon bias with deep language models. pp. 151–156, 2017.

Gonzalez, Javier, Longworth, Joseph, James, David C, and Lawrence, Neil D. Bayesian optimization for synthetic gene design. *arXiv preprint arXiv:1505.01627*, 2015.

Goodman, Daniel B, Church, George M, and Kosuri, Sriram. Causes and effects of n-terminal codon bias in bacterial genes. *Science*, 342(6157):475–479, 2013.

Graves, Alex. Sequence transduction with recurrent neural networks. *CoRR*, abs/1211.3711, 2012. URL http://arxiv.org/abs/1211.3711.

Gustafsson, Claes, Govindarajan, Sridhar, and Minshull, Jeremy. Codon bias and heterologous protein expression. *Trends in biotechnology*, 22(7):346–353, 2004.

Kudla, Grzegorz, Murray, Andrew W, Tollervey, David, and Plotkin, Joshua B. Coding-sequence determinants of gene expression in escherichia coli. *Science*, 324(5924): 255–258, 2009.

Lawrence, Jeffrey G and Ochman, Howard. Amelioration of bacterial genomes: rates of change and exchange. *Journal of molecular evolution*, 44(4):383–397, 1997.

Li, Weizhong and Godzik, Adam. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22(13):1658–1659, 2006.

Lorenz, Ronny, Bernhart, Stephan H, Zu Siederdissen, Christian Hoener, Tafer, Hakim, Flamm, Christoph, Stadler, Peter F, and Hofacker, Ivo L. Viennarna package 2.0. *Algorithms for Molecular Biology*, 6(1):26, 2011.

Mauro, Vincent P and Chappell, Stephen A. A critical analysis of codon optimization in human therapeutics. *Trends in molecular medicine*, 20(11):604–613, 2014.

Puigbo, Pere, Guzman, Eduard, Romeu, Antoni, and Garcia-Vallve, Santiago. Optimizer: a web server for optimizing the codon usage of dna sequences. *Nucleic acids research*, 35(suppl_2):W126–W131, 2007a.

Puigbo, Pere, Romeu, Antoni, and Garcia-Vallve, Santiago. Heg-db: a database of predicted highly expressed genes in prokaryotic complete genomes under translational selection. *Nucleic acids research*, 36(suppl_1): D524–D527, 2007b.

Rosano, Germán L and Ceccarelli, Eduardo A. Recombinant protein expression in escherichia coli: advances and challenges. *Frontiers in microbiology*, 5:172, 2014.

Sivashanmugam, Arun, Murray, Victoria, Cui, Chunxian, Zhang, Yonghong, Wang, Jianjun, and Li, Qianqian. Practical protocols for production of very high yields of recombinant proteins using escherichia coli. *Protein Science*, 18(5):936–948, 2009.

Sønderby, Søren Kaae, Sønderby, Casper Kaae, Nielsen, Henrik, and Winther, Ole. Convolutional lstm networks for subcellular localization of proteins. In *International Conference on Algorithms for Computational Biology*, pp. 68–80. Springer, 2015.

Tian, Jian, Yan, Yaru, Yue, Qingxia, Liu, Xiaoqing, Chu, Xiaoyu, Wu, Ningfeng, and Fan, Yunliu. Predicting synonymous codon usage and optimizing the heterologous gene for expression in e. coli. *Scientific reports*, 7(1): 9926, 2017.

Tunney, Robert, McGlincy, Nicholas J, Graham, Monica E, Naddaf, Nicki, Pachter, Lior, and Lareau, Liana F. Accurate design of translational output by a neural network model of ribosome distribution. *Nature structural & molecular biology*, 25(7):577, 2018.

Xiao, Su, Shiloach, Joseph, and Betenbaugh, Michael J. Engineering cells to improve protein expression. *Current opinion in structural biology*, 26:32–38, 2014.

Zhou, Zhipeng, Dang, Yunkun, Zhou, Mian, Li, Lin, Yu, Chien-hung, Fu, Jingjing, Chen, She, and Liu, Yi. Codon usage is an important determinant of gene expression levels largely through its effects on transcription. *Proceedings of the National Academy of Sciences*, 113(41): E6117–E6125, 2016.

7. Appendix

7.1. Impacts of pretraining

We test the impact of pretraining the *E. coli* model on the full set of non highly expressed genes by training a model exclusively on the small set of highly expressed genes. Curiously we see that these models still have comparable and favorable performance, indicating that pretraining offers only small boosts in performance (Table 2). Additionally, we tested the accuracy of the models on the highly expressed genes after pretraining on the non highly expressed genes, prior to fine-tuning to make sure there was no data leakage between train and test. The pre-fine tuning accuracy was lower than the fine-tuned values, 0.6001 accuracy (2.3469 ppl) for the only BiLSTM model and 0.6345 accuracy (2.221 ppl) for the LSTM Transducer model on the HEG training set.

Model	**Train Acc**	**Test Acc**	**Train Ppl**	**Test Ppl**
BiLSTM Encoder	0.6733	0.6513	2.0749	2.1468
LSTM-Transducer	0.6856	0.6553	2.0275	2.1331

Table 2. Modeling codon bias in E. coli without pertaining on non-highly expressed genes.

7.2. Secondary structure of predicted mRNA

Once we train our models, we generate full sequences of codon predictions over the test set with all models. Next, we predict the secondary structure, measured by the minimum free energy (MFE) through the ViennaRNA package (Lorenz et al., 2011). We calculate the MFE for the first 36 nucleotides of the sequences. This is motivated by existing experimental work that implicates the importance of the positions -4 to 37 of the sequence (Kudla et al., 2009). While further work is in order to verify that the model has learned anything about RNA structure or the actual biology of the free energy, we compare the MFE of the generated sequences and true sequences as a non-local metric to evaluate our sequence predictions.

We compute the Spearman rank correlation between the MFE of target sequences and MFE of model-generated sequences with the true sequences in the test set (Table 3). In addition, we report the codon level accuracy of these models on only the first 36 nucleotides (12 codons). Despite having comparable codon-level accuracy as the frequency models, we find that the MFE of sequences generated by our LSTM Transducer model has an increased Spearman rank correlation with the true MFEs than frequency based approaches. However, given the earlier discussion regarding the limitations of our model, this result is not suggestive of whether the model is learning long-term interactions and secondary structure of the RNA.

Species	**Model**	**Spearman's ρ**	**p-value**	**Acc $(+36)$**
E. coli	Unigram	0.432	0.0017	0.518
	1,3,5-gram	0.523	9.89e-05	**0.527**
	LSTM Transducer	**0.584**	**8.54e-6**	0.520
Human	Unigram	0.562	4.27e-64	0.469
	1,3,5-gram	0.584	4.43e-70	0.511
	LSTM Transducer	**0.639**	**1.11e-87**	**0.525**

Table 3. Evaluation of generated full length codon sequences. Rank correlation between the MFE values of target sequences and model predicted sequences in the test set. The accuracy is taken between the completely generated sequences (without teacher forcing) and the true sequence averaged over the first 36 nucleotides.