CodOpt: Enhancing Drug and Vaccine Development by Using Deep Learning and Natural-Language Processing to Optimize Recombinant Codon Sequences via a Host-Independent Data Pipeline

Columbia University Medical Center

Bhushan Mohanraj
The Lawrenceville School, Lawrenceville NJ

Cedars Sinai

With the mentorship of

Dr. Chao Lu, Assistant Professor in the Department of Genetics and Development
Columbia University Irving Medical Center

Dr. Ryan Urbanowicz, Assistant Professor in the Department of Computational Biomedicine
Cedars-Sinai Medical Center

Background: Recombinant Vaccines and Pharmaceuticals

Proteins are biomolecules that perform active functions in all living organisms. Every protein contains amino acids and is defined by the codons within a gene. Since there are twenty amino acids and sixty-four codons, some codons (synonymous codons) encode the same amino acid.

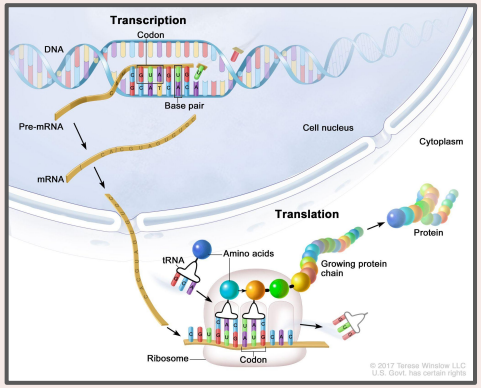
Recombinant DNA technology produces proteins, such as vaccines and pharmaceuticals, that are crucial to mitigating global health challenges. Examples include:

- COVID-19 vaccines
- Flu vaccines
- Recombinant granulocyte colony-stimulating factor for treating cancer
- Recombinant insulin for treating diabetes
- Recombinant human growth hormone for treating growth hormone deficiency

These recombinant proteins save millions of lives, especially during outbreaks such as the COVID-19 pandemic that require the rapid design of therapeutics to fight disease.

To design a recombinant gene for a vaccine or pharmaceutical, researchers must select codons to encode its amino acids. However, synonymous codons are not equivalent because they occur at different frequencies (codon bias). For example, the codons "CTG" and "CTA" both encode the amino acid leucine, but in *E. coli*, "CTG" occurs fifteen times more often than "CTA."

Research shows that choosing certain synonymous codons over others (codon optimization) can amplify protein expression hundreds of times. This enhancement can accelerate the rapid design and production of therapeutics during health emergencies.



The translation of the codons within a gene to the amino acids that form a protein. Image by Terese Winslow.

CCA(P) 0.83%	CGA(R) 0.37%	CAA(Q) 1.48%	CTA(L) 0.38%
CCG(P) 2.28%	CGG(R) 0.60%	CAG(Q) 2.94%	CTG(L) 5.22%
CCT(P) 0.72%	CGT(R) 2.03%	CAT(H) 1.27%	CTT(L) 1.13%
CCC(P) 0.57%	CGC(R) 2.12%	CAC(H) 0.93%	CTC(L) 1.08%
GCA(A) 2.04%	GGA(G) 0.86%	GAA(E) 3.91%	GTA(V) 1.08%
GCG(A) 3.27%	GGG(G) 1.15%	GAG(E) 1.81%	GTG(V) 2.62%
GCT(A) 1.53%	GGT(G) 2.43%	GAT(D) 3.24%	GTT(V) 1.82%
GCC(A) 2.57%	GGC(G) 2.87%	GAC(D) 1.92%	GTC(V) 1.53%
ACA(T) 0.78%	AGA(R) 0.25%	AAA(K) 3.35%	ATA(I) 0.50%
ACG(T) 1.49%	AGG(R) 0.16%	AAG(K) 1.08%	ATG(M) 2.74%
ACT(T) 0.90%	AGT(S) 0.93%	AAT(N) 1.87%	ATT(I) 2.97%
ACC(T) 2.33%	AGC(S) 1.62%	AAC(N) 2.16%	ATC(I) 2.43%
TCA(S) 0.77%	TGA(*) 0.13%	TAA(*) 0.21%	TTA(L) 1.38%
TCG(S) 0.90%	TGG(W) 1.52%	TAG(*) 0.03%	TTG(L) 1.31%
TCT(S) 0.86%	TGT(C) 0.52%	TAT(Y) 1.63%	TTT(F) 2.22%
TCC(S) 0.90%	TGC(C) 0.64%	TAC(Y) 1.20%	TTC(F) 1.60%

Codon bias within Escherichia coli. Image from CoGe Genomics.

Purpose and Hypothesis

Combating health emergencies such as COVID-19 requires the efficient production of high-efficacy vaccines and pharmaceuticals. Codon optimization is essential to ensuring the timely and cost-effective production of recombinant proteins.

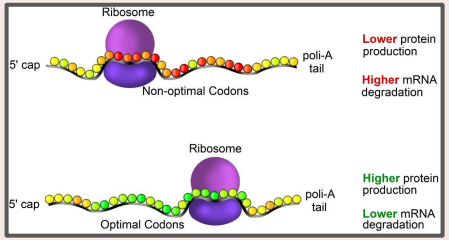
Common optimization techniques, used by tools such as GenSmart, replace all rare codons with frequent codons. However, this strategy ignores the significance of rare codons at certain locations and the evolutionary details that stabilize natural protein production.

- Many common optimization techniques entirely ignore rare codons. Although rare codons can slow translation, this strategy causes tRNA imbalance and removes rare codons that allow for protein folding.

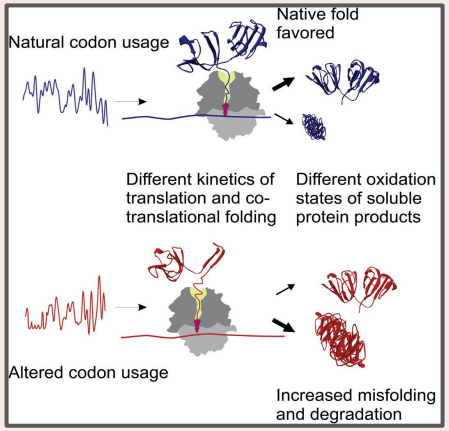
These consequences cause metabolic stress that hinders recombinant expression and protein misfolding that endangers patients.

- When a pharmaceutical is administered, misfolded proteins can trigger the production of anti-drug antibodies (ADAs) that hinder a patient's natural protein pool.
- Although recombinant proteins are intended to complement natural protein pools, misfolded proteins can harm patients instead of helping them.

Hypothesis: By learning to emulate high-expression genes from host organisms, neural networks can amplify protein expression while avoiding drawbacks such as metabolic stress and misfolding that natural genes avoid. A web application could provide global access to optimization tools based on deep learning, accelerating vaccine and pharmaceutical development and saving lives.



The benefits of codon optimization for recombinant protein production. Image from EurekAlert!.



The drawbacks of standard codon optimization techniques. Image from Buhr et al.

Methods: Genomic Data Pipeline

The genomic data pipeline was applied to three popular recombinant hosts: *Escherichia coli*, baker's yeast (*Saccharomyces cerevisiae*), and Chinese hamster ovary cells.

For each host, neural networks were trained to optimize codon sequences by learning to predict the codon sequences of natural, high-expression genes from the corresponding amino acids.

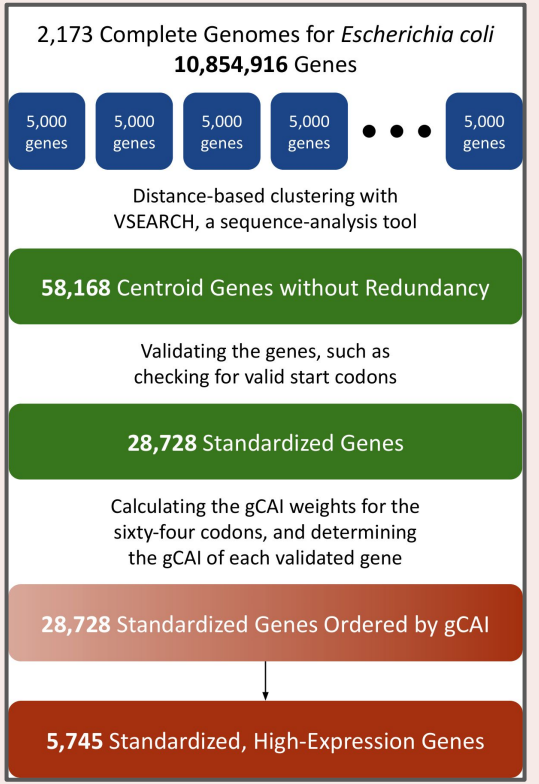
- Evolutionary pressure has tuned these genes to achieve both high expression *and* safe protein production without misfolding.
- By understanding abstract features within these genes, neural networks can optimize recombinant sequences to achieve the same efficiency and safety as natural protein production.

Up to ten million genes per host organism were downloaded from NCBI. VSEARCH was used to eliminate redundant genes within these large datasets.

- Sequences with pairwise similarities above 90% were clustered together.
- The centroid sequences, one from each cluster, were then stored in a FASTA file.

Sequences were ranked using the global Codon Adaptation Index (gCAI) algorithm for predicting expression. Genes were selected if their gCAI values were above the 80th percentile.

The data pipeline is host-independent, so researchers can easily train models for other host organisms.



The pipeline for distilling tens of millions of genes into a standardized and high-quality dataset for deep learning.

Methods: Model Architectures

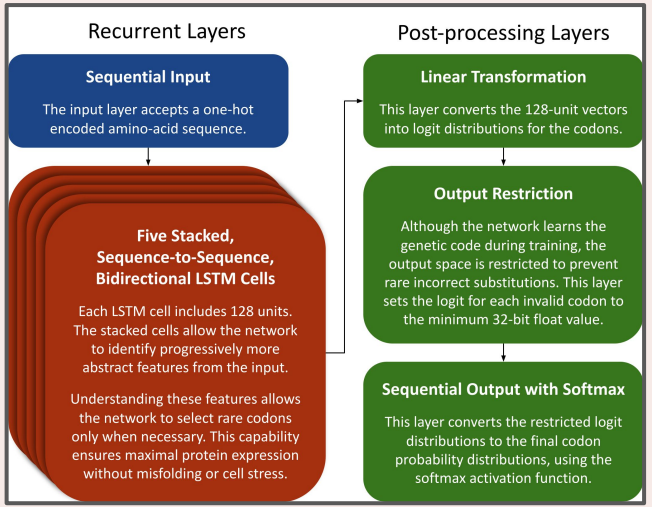
Neural network architectures are specific arrangements of the neurons and connections within a model.

Network Designs: Convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformers were built and compared to determine the best architecture.

- CNNs are used for computer vision, and their spatial invariance allows them to detect features or patterns at any location within an amino-acid sequence.
- RNNs (simple RNNs, GRUs, and LSTMs) are used for language processing and with their contextual understanding can select rare codons where necessary.
- Transformers with their attention mechanisms can identify which amino acids should affect a codon selection.

Architectures and Hyperparameters: Each neural network accepts one-hot encoded amino acids and returns probability distributions for the codon for each amino acid.

- A sequence-to-sequence CNN was built using convolutions with strides of one, with skip-layer connections between earlier convolutional layers and later ones.
- Multiple RNNs were built, including the gated recurrent unit (GRU) and long short-term memory (LSTM) variants of the traditional RNN. Unstacked and stacked models, with between 128 and 512 units per cell, were compared.
- A sequence-to-sequence transformer was built with four parallel attention heads followed by two convolutional layers.



The architecture of the stacked LSTM model.

Model	Hidden Output Sizes	Optimizer and LR	Encoding
CNN	16, 32, 64, 32, 16	Adam (10^{-5})	One-Hot
RNN	512	Adam (10^{-4})	One-Hot
Stacked RNN	256, 256, 256, 256, 256		One-Hot
GRU	512	Adam (10^{-4})	One-Hot
Stacked GRU	256, 256, 256, 256, 256		One-Hot
LSTM	256	Adam (10^{-4})	One-Hot
Stacked LSTM	128, 128, 128, 128, 128		One-Hot
Transformer	1024 (Four Attention Heads) 64 (Convolutional Layer)	Adam (10^{-5})	One-Hot

The hyperparameter configurations for the eight model architectures after training and tuning.

Methods: Model Training

The neural networks were trained to predict the natural codon sequences of the high-expression proteins for each host organism.

By learning to emulate natural high-expression genes, rather than relying on theoretical assumptions, the models can improve expression levels without ignoring the evolutionary details that stabilize natural protein production.

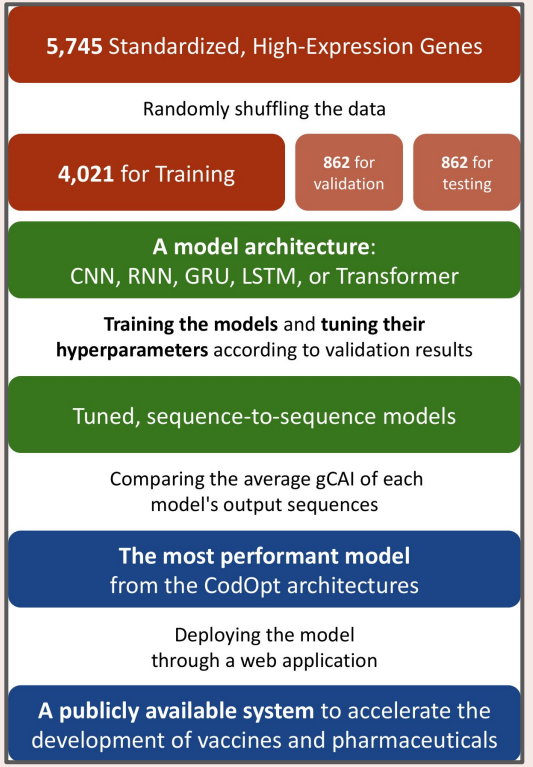
The networks were compared according to:

- Their predictions' categorical accuracy: how closely the predictions matched the natural genes that determine the inputted amino acids.
- Their predictions' average gCAI: the expected expression levels of the outputted genes.

The hyperparameters of the model architectures were carefully tuned to improve categorical accuracy according to validation performance. For each model architecture, these hyperparameters include:

- The network depth and layer sizes
- The encoding method (one-hot encoding or a linear embedding layer)

The training sequences had varying lengths, ranging from fifty amino acids to thousands, so the training batches contained one sequence each. The Adam optimizer was used with the categorical cross-entropy loss function and a learning rate between 0.0001 and 0.00001. The models were trained until the categorical accuracy increased negligibly between epochs.

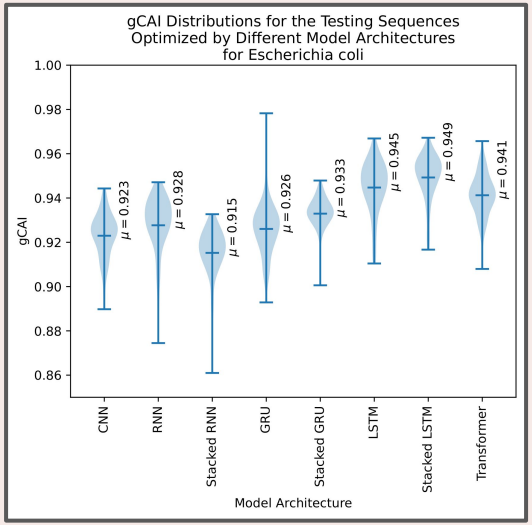


The pipeline for utilizing a dataset of highly expressed genes to train neural networks that generate optimized codon sequences.

Results: Model Performance and Statistical Analysis

Model	Global Codon Adaptation Index
CNN	0.923
RNN	0.928
Stacked RNN	0.915
GRU	0.926
Stacked GRU	0.933
LSTM	0.945
Stacked LSTM	**0.949**
Transformer	0.941

The average gCAI of the *Escherichia coli* sequences optimized by various models.



The global Codon Adaptation Index (gCAI) quantitatively predicts the protein expression of a genetic sequence. The gCAI algorithm uses the codon bias of a genome to recursively determine a gCAI weight ($w_{i,j}$) for each codon.

$$\mathrm{gCAI}(g) = \left(\prod_{k=1}^{L} \bar{w}_k\right)^{1/L}$$

$$\bar{w}_{i,j} = \frac{|S^i|}{|S|} \cdot \frac{x_{i,j}}{y_j}$$

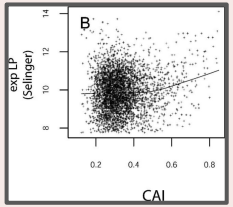
The average gCAI for the testing sequences increased significantly after optimization, demonstrating that the CodOpt models can substantially enhance protein expression.

Species	Original gCAI	Optimized gCAI
Escherichia coli	0.600	0.949
Baker's Yeast	0.723	0.970
CHO Cells	0.524	0.948

The average gCAI of the sequences for all three hosts optimized by the stacked LSTM.

Most Performant Architecture: The most performant model for *Escherichia coli* was the stacked LSTM, which achieved an average gCAI of 0.949 on testing data, an improvement of 58% over the original 0.600.

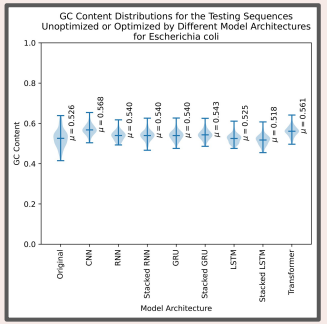
By a one-sided Wilcoxon signed-rank test, the gCAIs of the optimized sequences (μ = 0.949) were *significantly* greater than the gCAIs of the original testing sequences (μ = 0.600), with a p-value of 4.14×10^{-16}.



Expression levels versus CAI for *E. coli*. Image from dos Reis et al.

Additionally, the stacked LSTM achieved the highest average gCAI for baker's yeast and Chinese hamster ovary cells.

GC Content: GC content (the proportion of bases that are guanine or cytosine) relates to the stability of translation. Values below 30% or above 70% can cause secondary structure formation that inhibits translation. All the models produced sequences with GC contents between 30% and 70%.



Results: Evolutionary Phenomenon Analysis

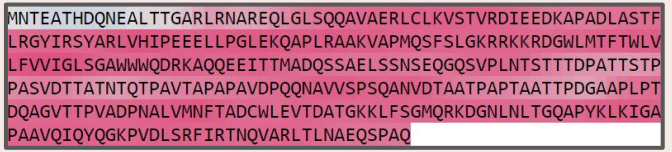
Feature analysis visualizes the patterns that neural networks learn during training, enabling researchers to understand a model's predictions. These visualizations can reveal new insights in developing domains such as codon optimization.

The most performant model, the stacked LSTM, contained five bidirectional recurrent cells with 128 units each. Each unit of each cell learned to identify specific sequence properties, such as evolutionary patterns that determine where rare codons are used.

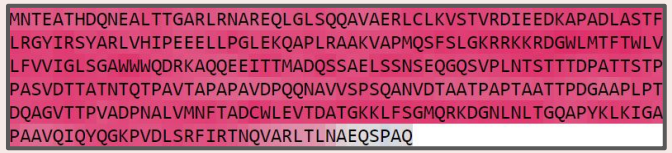
To understand the features learned by the stacked LSTM for *E. coli*, the output of each unit was captured as the model performed prediction on the testing dataset. Heatmaps of these output values were created and inspected for features learned by the model.

The five plots shown demonstrate that the network learned multiple, discernible evolutionary phenomena that affect rare-codon usage:

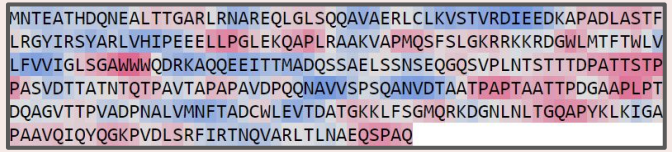
- Rare codons cluster at a sequence's start and end, ensuring efficient translation initiation and conclusion.
- The appearance of rare codons correlates with the Kyte-Doolittle hydropathy (a measure of hydrophobicity and hydrophilicity) of different protein regions. Rare codons cluster in hydrophobic areas where folding often must be slowed.
- Rare codons correspond with intrinsically disordered protein regions, ensuring the correct folding of such areas.
- In transmembrane proteins, rare codons cluster in positions 50 to 70 to allow for unhindered cotranslational insertion of the proteins.



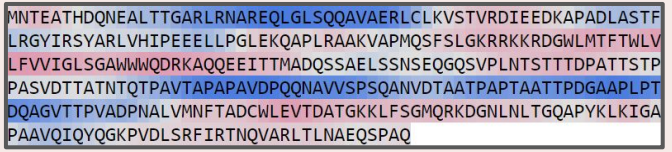
A unit identifying the amino acids toward the beginning of a sequence.



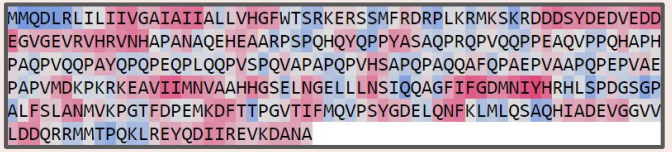
A unit identifying the amino acids toward the end of a sequence.



A unit whose value at each amino acid correlates with hydropathy.



A unit whose value at each amino acid correlates with protein disorder.



A unit activated at amino acids 50 to 70 (with some noise from connections with previous layers) for a transmembrane protein.

Results: Optimized Protein Expression in *E. coli*

To validate the CodOpt models experimentally, a recombinant protein was expressed in *Escherichia coli* cells, using both an original DNA sequence and a sequence optimized with CodOpt. The procedure for this lab trial was conducted by researchers at the Lu Lab in the Columbia University Irving Medical Center.

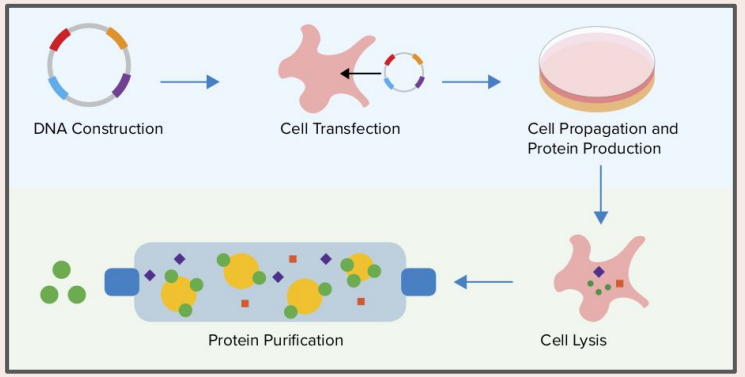
The protein expressed was pA-Tn5, a fusion protein containing both protein A and Tn5 transposase. pA-Tn5 is crucial to CUT&Tag, a low-cost method for studying protein–DNA interactions, which underlie many biological processes and diseases.

The original DNA sequence for pA-Tn5 was sourced from the widely used plasmid repository Addgene. The optimized sequence was created with the stacked LSTM, the most performant CodOpt model.

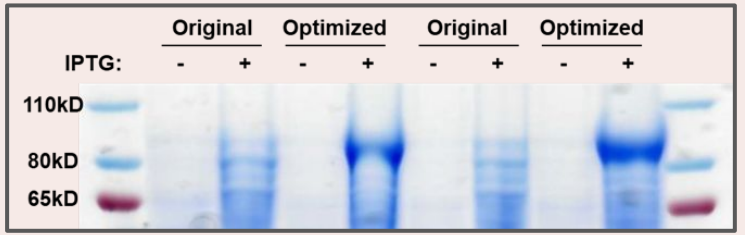
Both the original and optimized sequences were cloned into the same plasmid. The plasmids were introduced into separate *Escherichia coli* colonies, and recombinant protein expression was induced.

After expression and lysing, the pA-Tn5 was stained and isolated with gel electrophoresis. A fluorescent scanner was used to visualize protein expression.

The gene optimized by CodOpt resulted in *significantly greater expression* than the original gene. Therefore, codon optimization with CodOpt can greatly improve protein expression, accelerating the production of pA-Tn5 and other recombinant proteins such as vaccines and pharmaceuticals.



The procedure for recombinant protein production. Plasmids with a gene are introduced into host cells. After expression, the desired proteins are isolated. Image from Addgene.



The original and CodOpt-optimized plasmids were both expressed in *E. coli* colonies. As revealed by staining and gel electrophoresis, the gene optimized by CodOpt achieved significantly higher expression than the original gene.

CodOpt Web Application

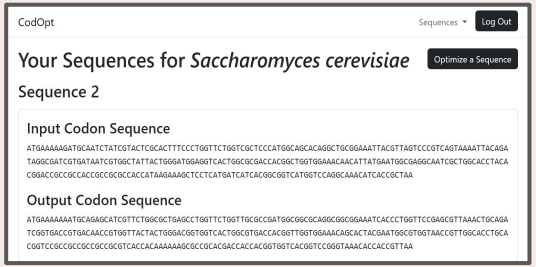
A web application was built to provide researchers around the globe with the functionality of the CodOpt models. Using the app, researchers developing vaccines and pharmaceuticals can accelerate their work, broadening the impact of these crucial treatments. Researchers can:

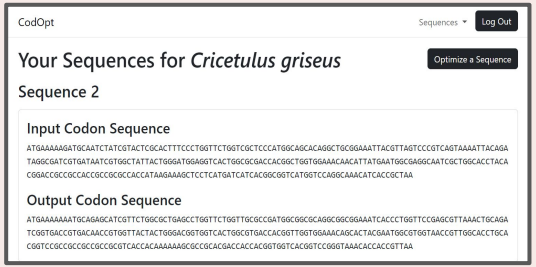
- Select the host species they are using for recombinant expression (*Escherichia coli*, *Saccharomyces cerevisiae*, or *Cricetulus griseus*)
- Input the recombinant DNA sequence for the vaccine, pharmaceutical, or other recombinant protein they intend to manufacture
- Generate a DNA sequence optimized by the stacked LSTM model for maximal expression within the selected host organism

For deployment, the stacked LSTM model for each species was saved in the Open Neural Network Exchange (ONNX) format, a cross-framework file format for saving neural networks.

- When a researcher accesses the application and submits an input codon sequence, the amino acids for the corresponding protein are determined using the genetic code.
- The saved model is loaded with the ONNX Runtime for Python and used to predict a codon probability distribution for each amino acid.
- The codons with the highest probabilities are returned as the output DNA sequence.

Some researchers or organizations may seek to integrate the models into their own pipelines. For this scenario, a JSON API was built, through which developers can request optimized sequences programmatically and forward the output to other applications or databases.





Conclusions and Future Work

Conclusions

The CodOpt networks were compared by their capability to enhance protein production. The stacked LSTM architecture achieved the highest gCAI after optimization.

Since the neural networks were trained to predict natural codon sequences, they successfully avoided the standard optimization technique of using only frequent codons, mitigating consequences such as metabolic stress and protein misfolding.

According to visual feature analyses, the stacked LSTM model learned several evolutionary features of amino acid sequences, including the effects of sequential location, hydropathy, and protein disorder on codon usage.

Applications

The CodOpt models can accelerate the development of safe vaccines and pharmaceuticals by addressing the drawbacks of current solutions for codon optimization. This enhancement could save millions of lives, especially during outbreaks that require the rapid design of therapeutics to fight disease.

Future Work

Feature Analysis: By visualizing the recurrent units of the stacked LSTM, five discernible features learned by the model were identified. Future research could investigate the other plots to better explain the model's decision-making and reveal unknown or ignored factors that affect codon optimization.

Transfer Learning: The models were trained for popular heterologous hosts with extensive, public sequencing data. However, less common hosts may require more data than available. Transfer learning, which tunes neural networks for new tasks, could solve this data disparity. Future research could apply transfer learning to build networks for less common hosts.

Optimizing Other Sequence Regions: The models were built to optimize the coding region of a gene. However, many auxiliary sequences, such as promoter sequences that initiate transcription, contribute to gene expression. Future research could apply deep learning to optimize these regions as well.

Experimental Trials: Currently, experimental evidence demonstrates that CodOpt-optimized sequences generate significantly more protein than unoptimized sequences. Further lab trials could establish this increase for a variety of proteins and compare CodOpt models to commercially available optimization techniques on metrics such as expression, protein folding, and protein solubility.

Bibliography

- Mohammed N. Baeshen, Ahmed M. Al-Hejin, Roop S. Bora, Mohamed M. M. Ahmed, Hassan A. I. Ramadan, Kulvinder S. Saini, Nabih A. Baeshen, and Elrashdy M. Redwan. Production of Biopharmaceuticals in E. coli: Current Scenario and Future Perspectives. Journal of Microbiology and Biotechnology, 25(7):953–962, 2015. Publisher: The Korean Society for Microbiology and Biotechnology.
- Suliman Khan, Muhammad Wajid Ullah, Rabeea Siddique, Ghulam Nabi, Sehrish Manan, Muhammad Yousaf, and Hongwei Hou. Role of Recombinant DNA Technology to Improve Life. International Journal of Genomics, 2016:2405954, 2016.
- Arnold L. Demain and Preeti Vaishnav. Production of recombinant proteins by microbes and higher organisms. Biotechnology Advances, 27(3):297–306, May 2009.
- Evelina Angov. Codon usage: Nature's roadmap to expression and folding of proteins. Biotechnology Journal, 6(6):650–659, 2011.
- P M Sharp and W H Li. The codon adaptation index – a measure of directional synonymous codon usage bias, and its potential applications. Nucleic Acids Research, 15(3):1281–1295, February 1987.
- Gila Lithwick and Hanah Margalit. Hierarchy of Sequence-Dependent Features Associated With Prokaryotic Translation. Genome Research, 13(12):2665–2673, December 2003.
- Vincent P. Mauro and Stephen A. Chappell. A critical analysis of codon optimization in human therapeutics. Trends in molecular medicine, 20(11):604–613, November 2014.
- Ming Gong, Feng Gong, and Charles Yanofsky. Overexpression of tnaC of Escherichia coli Inhibits Growth by Depleting tRNA2Pro Availability. Journal of Bacteriology, 188(5):1892–1898, March 2006.
- Vincent P. Mauro. Codon Optimization of Therapeutic Proteins: Suggested Criteria for Increased Efficacy and Safety. In Zuben E. Sauna and Chava Kimchi-Sarfaty, editors, Single Nucleotide Polymorphisms: Human Variation and a Coming Revolution in Biology and Medicine, pages 197–224. Springer International Publishing, Cham, 2022.
- Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. Nature, 521(7553):436–444, May 2015. Number: 7553 Publisher: Nature Publishing Group.

All images and graphics were created by Bhushan Mohanraj, unless otherwise specified with a citation.