Using Predictive Models to Engineer Biology: A Case Study in Codon Optimization

Alexey A. Gritsenko[1,2,3], Marcel J.T. Reinders[1,2,3], and Dick de Ridder[1,2,3]

[1] The Delft Bioinformatics Lab, Department of Intelligent Systems, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands
[2] Platform Green Synthetic Biology, P.O. Box 5057, 2600 GA Delft, The Netherlands
[3] Kluyver Centre for Genomics of Industrial Fermentation, P.O. Box 5057, 2600 GA Delft, The Netherlands

Abstract. Given recent advances in synthetic biology and DNA synthesis, there is an increasing need for carefully engineered biological parts (e.g. genes, promoter sequences or enzymes) and circuits. However, forward engineering approaches are thus far rarely used in biology due to lack of detailed knowledge of the biological mechanisms. We describe a framework that enables forward engineering in biology by constructing models predictive of properties of interest, then inverting and using these models to design biological parts.

We demonstrate the applicability of the proposed framework on the problem of codon optimization, concerned with optimizing gene coding sequences for efficient translation. Results suggest that our data-driven codon optimization (DECODON) method simultaneously considers the effects multiple translation mechanisms to produce optimal sequences, in contrast to existing codon optimization techniques.

Keywords: synthetic biology, codon optimization, support vector regression, genetic algorithms.

1 Introduction

In biotechnology, microorganisms such as yeast are genetically engineered for improved production of foods, beverages, fuels and pharmaceuticals. Recent advances in synthetic biology and dropping cost of DNA synthesis have led to a growing need for methods to engineer biological parts (promoter regions, gene *coding sequences* (CDSs) and even entire enzymes) with specific properties. Whereas in many engineering disciplines optimization techniques are routinely used to design such parts (e.g. aircraft wings [16]), in synthetic biology this is not yet the case. This stems from a lack of fundamental biological knowledge on the processes in which these parts are involved.

For some problems, this limitation can be overcome by constructing predictive models for properties of biological parts (e.g. promoter strength, mRNA translation rate or enzyme activity) and inverting the constructed models to design biological parts with desired properties. A successful use of such a "black-box"

A. Ngom et al. (Eds.): PRIB 2013, LNBI 7986, pp. 159–171, 2013.

modeling approach would enable forward engineering in areas of biology where detailed knowledge of the underlying processes is unavailable. We showcase the use of our proposed framework on the problem of codon optimization, in which a gene coding sequence is changed to obtain a desired translation rate of the mRNA into protein while keeping the amino acid sequence intact.

The degeneracy of the genetic code manifests itself in the differential use of synonymous codons in different organisms and different genes in the same organism. It has been long noticed that organisms preferentially use just one or two codons out of a family of codons translated into the same amino acid. This preference, termed *codon usage bias* (CUB), is more pronounced in highly expressed genes, which sometimes exclusively use only the preferred codons. For this reason it is believed that in unicellular organisms, such as baker's yeast *Saccharomyces cerevisiae* and the bacterium *Escherichia coli*, the codon bias of a gene is related to its translation rate [1]. Over the years numerous methods (called *indices*) summarizing the degree of CUB of a gene in a single number have been proposed and have been demonstrated to correlate with intracellular mRNA and protein levels [3].

These correlations have been used in a process called *codon optimization* to modify gene CDSs such that their translation rate is maximized, by introducing synonymous codon substitutions which increase one of the codon indices [9]. Codon optimization is routinely applied in biotechnology to overexpress genes for heterologous protein production and heterologous pathway expression [13]. However, CUB only partially explains the difference in translation rates among genes. Although the precise mechanisms influencing gene translation rates are not known, there is evidence suggesting that codon pair usage, tRNA recycling [2], mRNA secondary structure [19], adaptation to an organisms tRNA pool, mRNA untranslated regions (UTRs) and protein amino acid charge [19] may influence translation initiation and elongation rates. The relative influence of these factors on translation is not understood, making it difficult to combine them in a single codon optimization strategy. To our knowledge only Maertens et al. [15] have successfully combined multiple codon optimization objectives, by equally weighting them.

We present DECODON (data-driven codon optimization), an approach to codon optimization that combines multiple optimization objectives in a data-driven way by constructing a regression model. We use *Support Vector Regression* (SVR) [7] to predict *ribosome density*, a measure related to translation rate, based on coding sequence features of *S.cerevisiae* genes. We then invert this predictor by using it inside a genetic algorithm to optimize gene CDSs for desired ribosome density.

2 Materials and Methods

2.1 Dataset

To our knowledge no datasets with direct measurements of translation rates are available. However, Ingolia et al. [11] performed genome-scale measurements of

average *ribosome density*, defined as the number sequencing reads originating from parts of mRNA molecules covered by ribosomes in all mRNA copies of a particular gene, divided by the length of the gene transcript. Ribosome density is indicative of translation rate, as genes with higher densities are expected to produce more protein per copy of mRNA.

The number of gene mRNA copies per cell depends on its transcription rate and the stability of its mRNA. Although the relationship is poorly understood, the latter may be influenced by the secondary structure of the mRNA, which can differ between synonymous (i.e. encoding the same peptide) versions of a gene. In order to take the potential influence of coding sequence on the transcript levels into account, we propose to directly (i.e. without normalizing by the mRNA *read density*) use ribosome density as a measure of gene translation rate.

Yeast gene CDSs were obtained from the Saccharomyces Genome Database and the matching 5′- and 3′-UTR sequences were obtained from Nagalakshmi et al. [17] and Yassour et al. [21] (preference given to the former in cases when the two studies were not in agreement). The resulting dataset contains 5,048 yeast genes, each associated with coding and UTR sequences and a measured ribosome density.

2.2 Sequence Features

In order to construct a predictor of ribosome density from gene sequences a number of candidate sequence-based features identified from the literature have been computed for each gene in the dataset. These features were then used in a multivariate regression training step. Selected candidate features (Table 1) include a subset of existing codon bias indices (13 features); protein indices and protein properties (12 features); and nucleotide, codon and amino acid composition features (122 features). Prior to training, features as well as the ribosome density to be predicted were standardized to zero mean and unit variance.

2.3 Regression Model Training

ϵ-SVR [4] has been chosen as a regression method as it supports nonlinear regression through the use of kernels, allowing for complex models, and because efficient training algorithms are available. SVR relies on the choice of several parameters, including the cost parameter C, the error in sensitivity ϵ, the regression kernel and its parameters. Often, due to the lack of a theoretical framework for choosing these parameters, a grid search approach is used to find a combination of parameters that minimizes the regression error. This training procedure, if performed inside cross-validation (CV), becomes computationally very expensive.

As a performance measure we calculate the coefficient of determination R^2. Normally this measure approaches 1 with increasing model complexity regardless of its validity and is therefore not suitable for assessing quality of complex (nonlinear, many features) models. However, if the coefficient of determination is computed using CV (denoted R^2_{CV}), it becomes a measure of the amount of variance in *unseen* data explained by the model. Similar to the coefficient of

Table 1. Sequence-based features used as initial input for regression model training. CF and SF respectively stand for the number of candidate features in the feature group and the number of features selected for the final ribosome density predictor. Description of codon indices can be found in Cannarozzi and Schneider [3].

Name	Description	SF	CF
CAI	*Codon Adaptation Index* measures the extent to which a gene is composed of codons from the highly expressed genes.	0	1
tAI	*tRNA Adaptation Index* measures the extent to which a gene consists of codons recognized by abundant tRNAs. It is computed for the full CDS and its first 14, 17 and 19 codons (tAI, tAI_{14}, tAI_{17} and tAI_{19} respectively) [19].	3	4
N_c	*Effective number of codons* estimates the number of uniformly used codons that would produce the CUB observed in a gene.	0	1
D_{ncu}	*Distance to native codon usage* [18] measures the difference between codon usage of a gene and the overall codon usage of the organism.	1	1
E_w	*Weighted sum of relative entropy* measures the degree of deviation from equal usage of synonymous codons using the Shannon entropy.	1	1
CPB	*Codon Pair Bias* score [5] is computed as the sum of log-ratios of observed and expected codon pair counts.	0	1
TPI_2	*tRNA Pairing Index* measures the extent of potential tRNA re-use during gene translation.	1	1
F_{op}	For computing the *Frequency of optimal codons*, optimal codons were chosen as corresponding to the most abundant tRNA species.	1	1
RCBS	*Relative codon usage bias* measures codon usage difference of a gene with respect to the its nucleotide composition.	0	1
P_1	Mean number of non-specific tRNA interactions per elongation cycle.	1	1
prot	Protein hydrophobicity, aromaticity, aliphatic and instability indices.	3	4
Q_{port}	Protein net charge, isoelectric point and weight.	3	3
Q_{side}	Mean amino acid side chain charge computed for the full protein and its first 4, 11, 15 and 40 amino acids [19].	0	5
len	Lengths of the CDS, the 5′- and the 3′-UTR regions.	3	3
nuc	Nucleotide and dinucleotide frequencies of the CDS regions.	7	20
GC_{15}	GC-content computed for the first 15 codons of the CDS	1	1
RSCU	*Relative Synonymous Codon Usage* is computed for each codon (except ATG) as the ratio between the observed number of its occurrences and the mean number of occurrences for codons encoding the same amino acid.	41	63
codon^2	tAI and CAI weights of the second codon in the CDS (denoted tAI^2 and CAI^2).	2	2
amino	Amino acid frequencies.	6	21
ΔG	Gibson free energy for mRNA secondary structures predicted by the Vienna RNA package [10]. It is computed for the 5′-/3′-UTR sequences; and the first 17, 34, and 53 codons of the CDS [19] with ($\Delta G_{\text{5′-UTR,CDS}_{17}}$, $\Delta G_{\text{5′-UTR,CDS}_{34}}$ and $\Delta G_{\text{5′-UTR,CDS}_{53}}$) and without ($\Delta G_{\text{CDS}_{17}}$, $\Delta G_{\text{CDS}_{34}}$ and $\Delta G_{\text{CDS}_{53}}$) 5′-UTR sequence	4	12

determination computed without CV, the cross-validation R^2_{CV} approaches 1 as *generalization* becomes better, but can be negative if the trained model explains less variance in unseen data than a constant model. We believe that R^2_{CV} is a suitable measure for assessing quality of nonlinear models and use it to optimize and assess performance of our regression models.

Parameter Preselection: To keep the amount of computation tractable, we first *screened* the parameter space by training predictors with different parameter settings and assessing their coefficient of determination computed by 10-fold CV ($R^2_{10\mathrm{CV}}$) on the complete dataset (Figure 1, block A). Screening results (data not shown) indicated that the performance of RBF and polynomial kernels on the considered dataset is comparable, which led us to consider only polynomial kernels $K(u, v) = (\gamma \cdot \langle u, v \rangle + 1)^d$ with degrees $d = 2, 3, 4$ for the actual parameter selection stage. Based on the screening $R^2_{10\mathrm{CV}}$ results, ranges for parameters C, γ and ϵ were set to $\{1\} \cup \{0.001 \cdot 3^i\}$ for $i = 0, \ldots, 6$.

Parameter Estimation: The preselected parameter ranges were used to estimate optimal SVR parameter settings (Figure 1, block B) in a grid search procedure. For each combination of parameters an SVR is trained and its $R^2_{4\mathrm{CV}}$ is computed to select a *single* combination of SVR parameter settings with the

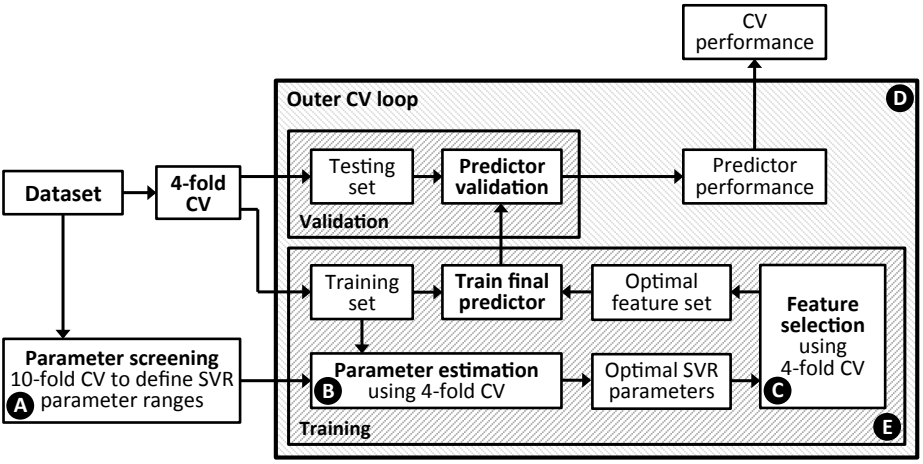


Fig. 1. Predictor training and evaluation scheme (adapted from [20]). The full dataset is used to preselect SVR parameter ranges (block A) and evaluate the training protocol using CV (block D). Predictor training consists of parameter estimation (block B) used to find an optimal set of SVR parameters, for which feature selection is performed (block C). The optimal parameters and the selected features are used to train the final predictor which is evaluated on the testing set of the CV loop. The same training procedure (block E) is used to train the final predictors used for sequence optimization on the *complete* dataset.

best performance. This combination is then used in the subsequent feature selection step.

Feature Selection: Feature selection was used to eliminate features that do not contribute to the model's generalization capability. This also allowed for selecting a concise set of features which can be interpreted biologically. While generally yielding good results, wrapper approaches to feature selection are computationally very demanding. To lower the computational load, backward feature elimination [12] was performed only on the SVR parameter settings obtained as discussed above (Figure 1, block C). At every step of the feature elimination procedure, given n features, we computed $R^2_{4\mathrm{CV}}$ for n predictors trained on subsets of $n-1$ features (i.e. obtained by removing one of the features). A subset with the highest R^2_{CV} was then selected for the next step of the feature elimination procedure. After the procedure was complete, the number of features (and the corresponding subset) with the best performance was chosen. If multiple subsets gave optimal performance, the smallest one was selected. The selected features were used to train the final predictor on the available data (Figure 1, block E).

Training Strategy Evaluation: In order to obtain an unbiased estimate of the predictor performance we used a second 4-fold CV loop (Figure 1, block D) around the described parameter estimation and feature selection strategies. The $R^2_{4\mathrm{CV}}$ values computed in the outer CV loop are reported in Section 3 as estimates of predictor generalization.

2.4 Sequence Optimization

In order for the constructed predictor $y = f(x)$ to be useful for sequence optimization, it first needs to be "inverted" such that it can be used to find sequences x that have the desired ribosome density $\check{y}$. Constructing the inverse function $x = f^{-1}(y)$ for SVR is impossible. Moreover, solving this function for a given $\check{y}$ would yield multiple nonsynonymous sequences x, thereby presenting an additional problem of selecting the suitable sequences from a large pool of solutions. Instead we implicitly invert the predictor by searching through the space of sequences x_i synonymous to the original sequence x to find $\check{x}$ such that its predicted ribosome density $f(\check{x})$ is close to the desired $\check{y}$.

Genetic Algorithm: The space of all nucleotide sequences synonymous to a given sequence x grows exponentially with the length of the sequence. Typically, it is too large to evaluate all possible x_i and requires an efficient search strategy to find (an approximation of) $\check{x}$ in a timely manner. *Genetic Algorithms* (GAs), specifically tailored for large discrete optimization problems, use computational equivalents of genetic crossover, mutation and selection concepts from biological systems to evolve a pool of potential solutions to a given optimization problem. The problem of finding an $\check{x}$ whose predicted ribosome density $f(\check{x})$ is as close

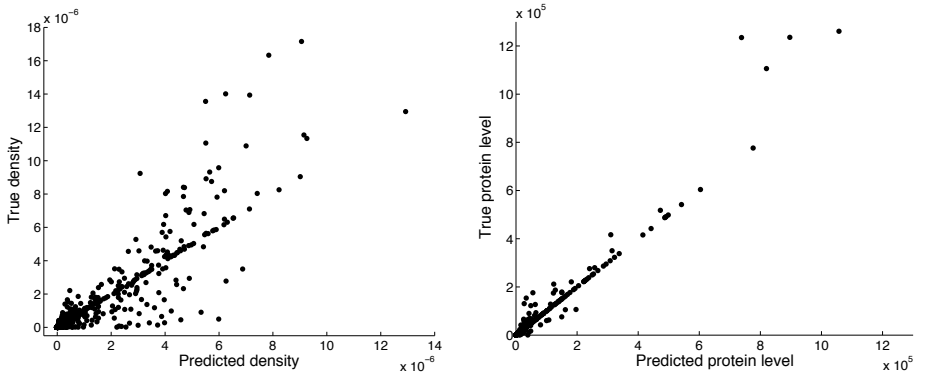
as possible to a desired level $\check{y}$ can be cast into an optimization problem and tackled using GAs if $g(x) = |f(x) - \check{y}|$ is used as an objective to be minimized. In practical applications, optimized gene sequences are synthesized and cloned into living cells in the wet lab. It is then required that the sequences do not contain certain motifs, such as restriction sites of enzymes used in cloning. This presents an optimization constraint that has to be taken care of by the GA. Treating this constraint as an additional objective of minimizing the number of undesired motifs present in the sequence allows to refrain from banning parts of the search space at the cost of casting the problem of finding $\check{x}$ into a multi-objective discrete optimization problem with two objectives. If it exists, the solution to the original problem will then be among the non-dominated solutions (i.e. solutions that cannot be improved in both objectives simultaneously) of the multi-objective optimization problem.

NSGA-II [6], a multi-objective GA, was chosen to solve the optimization problem as previously it has been successfully applied to DNA sequence optimization. It was implemented using multi-point crossover with a rate of 0.9; a mutation operator synonymously changing every sequence codon with probability $\frac{1}{n}$, where n is the number of degenerate codons in the sequence; and a binary tournament selection operator. For the genes optimized in this paper, the number of crossover points was set to 100.

3 Results

3.1 Regression Model

The cross-validation loop used to evaluate the regressor training strategy described in Section 2.3 gave an $R^2_{4\mathrm{CV}} = 0.66 \pm 0.03$, suggesting that the proposed



(a) Ribosome density predictor. (b) Protein level predictor.

Fig. 2. Predicted vs. true (a) ribosome density and (b) protein level plotted for *S.cerevisiae* genes.

strategy produces regressors that generalize well on unseen data. This strategy was employed to train the *final* ribosome density predictor (shown in Figure 2(a)) for use in codon optimization on the complete dataset.

Selected Features: The final predictor contained 78 features (Table 1, Figure 3), including codon indices, protein features, sequence composition and mRNA structure features selected to best explain the data. While black-box predictors are generally hard to interpret in biological terms, the fact that a certain feature was selected in the final predictor suggests that the mechanism it describes could indeed be used by the translation machinery. In this way, selection of the tRNA Pairing Index (TPI_2) suggests presence in yeast of a tRNA recycling mechanism, in which outgoing tRNA molecules stay bound to the ribosome to be recharged and reused in the course of translation [3]. Selection of the CAI^2 and tAI^2 features, describing respectively the extent to which the second codon of a gene is used in highly expressed genes of *S.cerevisiae* and its adaptation to the organisms tRNA pool, suggests that choice of the second codon influences ribosome density. Fredrick and Ibba [8] observe that the second codon is usually a highly frequently used codon that is translated more quickly, and speculate that this mechanism may be required for efficient recycling of the initiator tRNA.

Similarly, the selected tAI_{17}, tAI_{19}, and the $\Delta G_{5'\text{-UTR,CDS}_{17}}$, $\Delta G_{5'\text{-UTR,CDS}_{53}}$ and $\Delta G_{\mathrm{CDS}_{53}}$ features suggest that the mechanism of slowly translated "ramp" in the beginning of the CDS [19] influences gene translation rate. It is believed that the role of this "ramp" is to generate space between translating ribosomes and thereby prevent ribosome collision [8, 19]. The same mRNA structure features

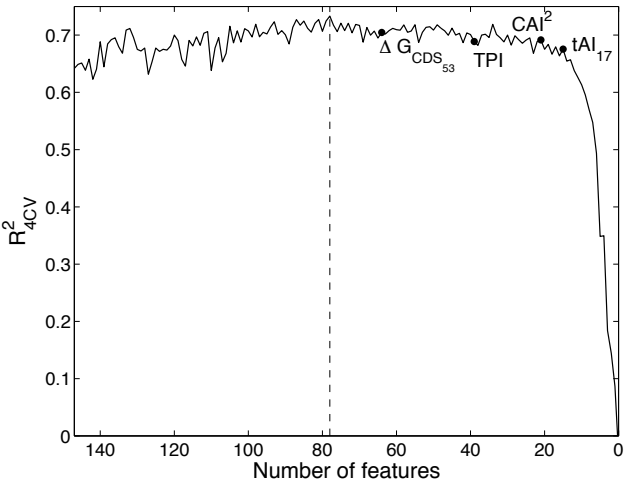


Fig. 3. Cross-validated $R^2_{4\mathrm{CV}}$ for the backward feature elimination procedure during final predictor training. Features eliminated at a particular step are marked with black circles. The maximum $R^2_{4\mathrm{CV}}$ is achieved at 78 features (see Table 1).

also describe the accessibility of the 5′-UTR for translation initiation by the ribosome machinery, suggesting it as another *S.cerevisiae* mechanism influencing gene translation.

3.2 Codon Optimization

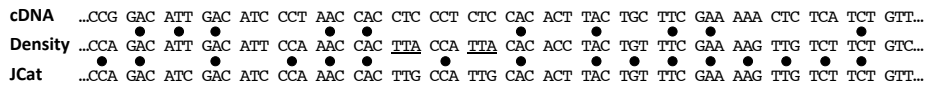
The final ribosome density predictor (Section 3.1) was used to optimize sequences of the genes 4CL (*4-coumaric acid-CoA ligase*, 562 codons) and PAL1 (*phenylalanine ammonia lyase*, 726 codons) involved in flavonoid biosynthesis [13]. The genes' cDNA, obtained from the plant *Arabidopsis thaliana*, was optimized using the described GA for *maximum* ribosome density. Based on preliminary experiments, optimization was performed for 200 generations with a population size equal to the gene length in codons. An initial population was generated by backtranslating genes from their amino acid sequences by choosing codons with probabilities proportional to their CAI weights. The 5′- and 3′-UTR sequences were set based on the respective sequences of the GPD promoter and CYC1 terminator sequences used in the pAG416GPD yeast expression vector. The *SpeI* and *XhoI* restriction site sequences used for cutting the expression vector were treated as undesired motifs.

Table 2 shows that the predicted ribosome density of the optimized sequences is significantly higher than that of the plant cDNA. As a sanity check, we compared sequences optimized using our method DECODON to sequences optimized by JCat [9], a well-known codon optimization tool that optimizes sequences for high CAI. The constructed predictor also predicts a significant increase in ribosome density for the JCat-optimized sequences (Table 2), showing that the trained predictor agrees with the currently used codon optimization methods. Note that the predicted ribosome density for the DECODON-optimized sequences is nearly two-fold higher than that of the JCat-optimized sequences.

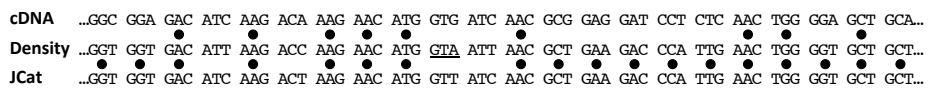
Sequence Analysis: Compared to the cDNA sequences, the DECODON- and JCat-optimized versions have roughly the same number of codon substitutions. To highlight the specific differences between the sequences, we compared them

Table 2. Sequence optimization results for the 4CL and PAL1 genes. Predicted ribosome densities are shown for the plant cDNA, sequences codon-optimized using JCat [9] and sequences optimized using DECODON. The number of different codons and the fold increase in the predicted density are computed relative to the cDNA sequences.

	4CL			PAL1		
Type	Different codons	Predicted density	Fold inc.	Different codons	Predicted density	Fold inc.
cDNA	N/A	0.0000000090	1	N/A	0.0000000524	1
JCat	338 (60.14%)	0.0000101491	1128	414 (57.02%)	0.0000079718	152
DECODON	361 (64.23%)	0.0000201560	2240	444 (61.16%)	0.0000172657	329



(a) 4CL gene, codons 40 to 61.



(b) PAL1 gene, codons 20 to 41.

Fig. 4. Comparison of part of the codon-optimized sequences (JCat and ribosome density optimized using DECODON). Matching codons are marked with black circles. Underscored codons are not explained by the "one amino acid - one codon" rule.

to each other. It can be seen from Figure 4 that codon usage in the DECODON sequences is more similar to that of the JCat-optimized genes than to that of the original sequences.

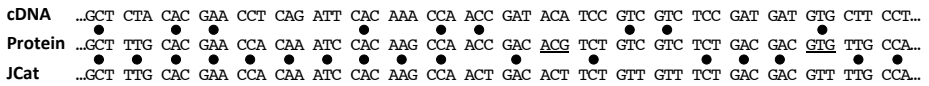
When optimized for maximum ribosome density, codon usage of the optimized sequences follows the "one amino acid - one codon" rule meaning that for each amino acid only a single (preferred) codon is used to encode it. The preferred codons in the genes optimized by DECODON mostly correspond to the codons with high CAI weights (the JCat- and density-optimized 4CL and PAL1 genes differ only in 126 and 150 codons respectively) with a few notable exceptions: (a) ACC is preferred for the amino acid threonine; (b) GTC is preferred for valine; (c) TGC is preferred for cysteine; and (d) ATT is preferred for isoleucine.

The preference rules account for all but a few codon differences (underscored in Figure 4) between the optimized sequences. These substitutions, when introduced in the sequences optimized using the "one amino acid - one codon rule", influence codon indices and mRNA features ($\Delta G_{\mathrm{CDS}_{53}}$ and $\Delta G_{5'\text{-UTR},\mathrm{CDS}_{53}}$), according to which the mRNA secondary structures at the 5′-UTR become less stable. This further suggests that the constructed predictor takes into account multiple translation mechanisms, even when used to optimize genes for maximum ribosome density.

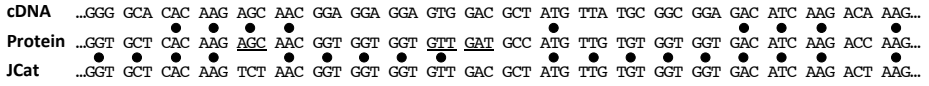
3.3 Applicability to Other Datasets

To demonstrate the applicability of the framework proposed in this paper to different datasets, we used it to optimize codon use based on the predicted absolute protein level measurements of 756 proteins [14]. All the training steps (parameter preselection, training strategy evaluation and final predictor training) were repeated, yielding an cross-validation $R^2_{4\mathrm{CV}} = 0.65 \pm 0.09$ and a final predictor with 138 features (Figure 2(b)). This large number of features, explained by the relatively high variance in the $R^2_{4\mathrm{CV}}$ used for feature selection due to the limited size of the dataset, hampers further biological interpretation.

The 4CL and PAL1 gene sequences optimized for maximum protein levels using the constructed predictor show a "one amino acid - one codon"' rule



(a) 4CL gene, codons 5 to 26.



(b) PAL1 gene, codons 5 to 26.

Fig. 5. Comparison of part of the codon-optimized sequences (JCat and absolute protein levels optimized using DECODON)

behavior similar to the density-optimized genes with several differences: (a) TGT is preferred for cysteine (as in JCat); (b) ATC is preferred for isoleucine (as in JCat); and (c) GCT and GCC are preferred for alanine. Similarly, these rules explain all but a few codon substitutions near to the $5'$ end of the CDS (Figure 5). The codon usage similarities between the protein- and density-optimized gene sequences show that the proposed framework can be applied to various types of biological data to enable forward engineering approaches. However, wet-lab experiments are required in order to determine which of the constructed predictors is better suited for codon optimization.

4 Discussion

We have described a generic framework for forward engineering of biological systems and demonstrated its use by optimizing genes for maximum ribosome density and maximum protein levels using predictors constructed from the corresponding yeast datasets. The general agreement between the optimized gene sequences obtained by us and gene sequences optimized using an existing codon optimization method suggests that the proposed approach can be successfully utilized for forward engineering of biological parts, whereas the differences between the sequences suggest that our codon optimization method DECODON simultaneously considers the effects of multiple translation mechanisms to produce optimal sequences. Time complexity of DECODON is much higher than that of JCat, however, it is negligible compared to the time involved in ordering and experimenting with the synthesized DNA.

Features selected for the final ribosome density predictor and the exceptions to the "one amino acid - one codon" rule in the optimized sequences show that data-driven models can combine multiple features describing (competing) biological mechanisms in a way that best explains the available data. While the effect of combining multiple mechanisms in a single predictor is hard to observe in sequences optimized for maximum ribosome density (or protein level), we believe that it would be more pronounced in sequences optimized for intermediate ribosome density, in which no one single mechanism would have a dominating influence.

Using black-box models for combining multiple (potential) mechanisms in a single predictor is particularly useful in areas where precise workings of a system are not known, but hypotheses on its important aspects can be generated and described by features. Note that a danger associated with the interpretation of the results is that the constructed model will select features that correlate with the property it is trained to predict, rather than the features describing the actual underlying mechanisms. For example, Qian et al. [18] suggest that strong CUB in highly expressed genes is not related to translation rate of those genes, but is rather a consequence of random mutations and the evolutionary pressure to keep codon usage and tRNA availability of an organism balanced. Nevertheless our models exhibit the "one amino acid - one codon" behavior when genes are optimized for maximum density/protein levels. It is, therefore, crucial to validate predictive models by testing their predictions in the wet-lab prior to their application.

For the constructed predictors (especially in the case of the protein level predictor) we observed that a single codon substitution often leads to changes in many features. These changes are often difficult to interpret and to link to the effect a substitution has on the prediction. Nevertheless, we believe that by trading interpretability for general applicability, our framework will enable forward engineering of various parts essential for synthetic biology such as promoters, coding sequences and UTRs.

References

[1] Angov, E.: Codon usage: Nature's roadmap to expression and folding of proteins. Biotechnology Journal 6(6), 650–659 (2011)
[2] Cannarozzi, G., Schraudolph, N.N., Faty, M., von Rohr, P., Friberg, M.T., Roth, A.C., Gonnet, P., Gonnet, G., Barral, Y.: A role for codon order in translation dynamics. Cell 141, 355–367 (2010)
[3] Cannarozzi, G.M., Schneider, A.: Codon evolution: mechanisms and models. OUP Oxford (2012)
[4] Chang, C.-C., Lin, C.-J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 1–27 (2011)
[5] Coleman, J.R., Papamichail, D., Skiena, S., Futcher, B., Wimmer, E., Mueller, S.: Virus attenuation by genome-scale changes in codon pair bias. Science 320(5884), 1784–1787 (2008)
[6] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
[7] Drucker, H., Burges, C.J.C., Kaufman, L., Smola, A., Vapnik, V.: Support vector regression machines. In: Advances in Neural Information Processing Systems, pp. 155–161 (1997)
[8] Fredrick, K., Ibba, M.: How the sequence of a gene can tune its translation. Cell 141(2), 227–229 (2010)
[9] Grote, A., Hiller, K., Scheer, M., Münch, R., Nörtemann, B., Hempel, D.C., Jahn, D.: JCat: a novel tool to adapt codon usage of a target gene to its potential expression host. Nucleic Acids Research 33(suppl. 2), 526–531 (2005)

[10] Hofacker, I.L., Fontana, W., Stadler, P.F., Bonhoeffer, L.S., Tacker, M., Schuster, P.: Fast folding and comparison of rna secondary structures. Monatshefte für Chemie/Chemical Monthly 125(2), 167–188 (1994)
[11] Ingolia, N.T., Ghaemmaghami, S.A., Newman, J.R.S., Weissman, J.S.: Genome-wide analysis in vivo of translation with nucleotide resolution using ribosome profiling. Science 324(5924), 218–223 (2009)
[12] Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence 97(1), 273–324 (1997)
[13] Koopman, F., Beekwilder, J., Crimi, B., van Houwelingen, A., Hall, R.D., Bosch, D., van Maris, A.J.A., Pronk, J.T., Daran, J.-M.: De novo production of the flavonoid naringenin in engineered *Saccharomyces cerevisiae*. Microbial Cell Factories 11(1), 155 (2012)
[14] Lu, P., Vogel, C., Wang, R., Yao, X., Marcotte, E.M.: Absolute protein expression profiling estimates the relative contributions of transcriptional and translational regulation. Nature Biotechnology 25(1), 117–124 (2006)
[15] Maertens, B., Spriestersbach, A., von Groll, U., Roth, U., Kubicek, J., Gerrits, M., Graf, M., Liss, M., Daubert, D., Wagner, R., et al.: Gene optimization mechanisms: A multi-gene study reveals a high success rate of full-length human proteins expressed in *Escherichia coli*. Protein Science 19(7), 1312–1326 (2010)
[16] Mohammadi, B., Pironneau, O.: Shape optimization in fluid mechanics. Annu. Rev. Fluid Mech. 36, 255–279 (2004)
[17] Nagalakshmi, U., Wang, Z., Waern, K., Shou, C., Raha, D., Gerstein, M., Snyder, M.: The transcriptional landscape of the yeast genome defined by rna sequencing. Science 320(5881), 1344–1349 (2008)
[18] Qian, W., Yang, J.R., Pearson, N.M., Maclean, C., Zhang, J.: Balanced codon usage optimizes eukaryotic translational efficiency. PLoS Genetics, 8(3), e1002603 (2012)
[19] Tuller, T., Veksler-Lublinsky, I., Gazit, N., Kupiec, M., Ruppin, E., Ziv-Ukelson, M.: Composite effects of gene determinants on the translation speed and density of ribosomes. Genome Biology 12(11), R110 (2011)
[20] Wessels, L.F.A., Reinders, M.J.T., Hart, A.A.M., Veenman, C.J., Dai, H., He, Y.D., Van't Veer, L.J.: A protocol for building and evaluating predictors of disease state based on microarray data. Bioinformatics 21(19), 3755–3762 (2005)
[21] Yassour, M., Kaplan, T., Fraser, H.B., Levin, J.Z., Pfiffner, J., Adiconis, X., Schroth, G., Luo, S., Khrebtukova, I., Gnirke, A., Nusbaum, C., Thompson, D.-A., Friedman, N., Regev, A.: *Ab initio* construction of a eukaryotic transcriptome by massively parallel mRNA sequencing. Proceedings of the National Academy of Sciences 106(9), 3264–3269 (2009)