Molecular Cell, Volume *65*

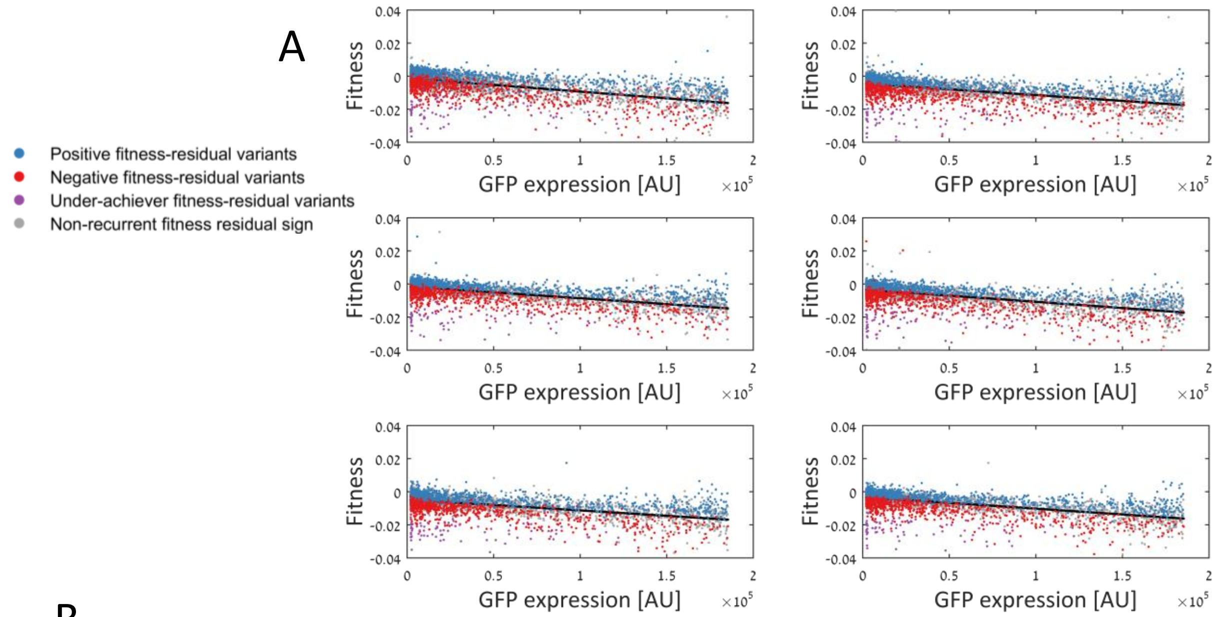
Supplemental Information

Gene Architectures that Minimize Cost of Gene Expression

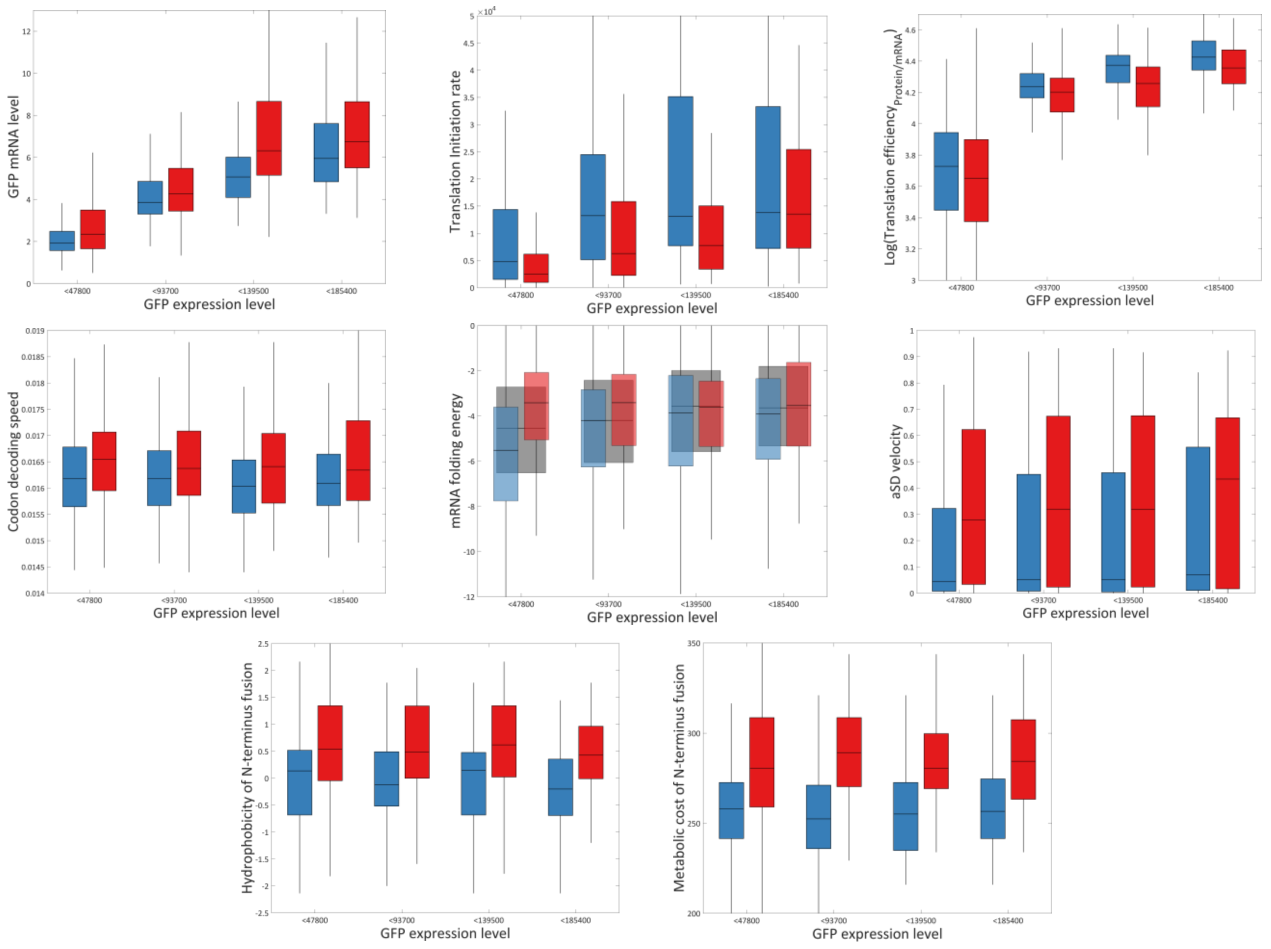
Idan Frumkin, Dvir Schirman, Aviv Rotman, Fangfei Li, Liron Zahavi, Ernest Mordret, Omer Asraf, Song Wu, Sasha F. Levy, and Yitzhak Pilpel

Supplementary Figure 1

A



B

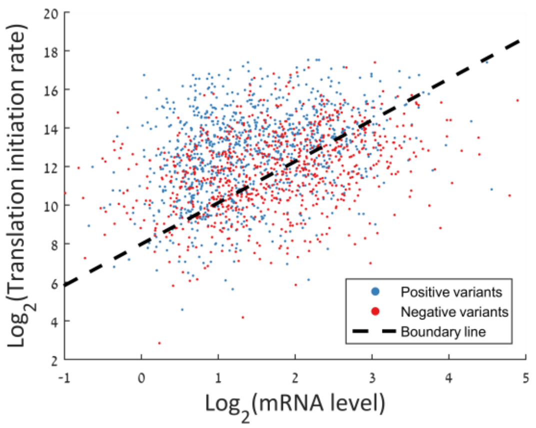


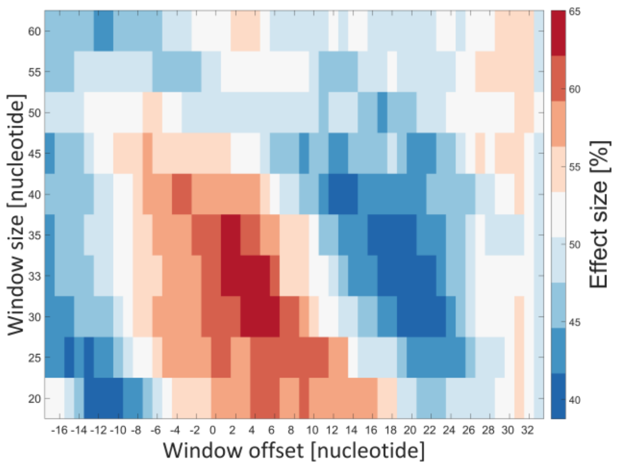
Supplementary Figure 1, related to Figures 1-4

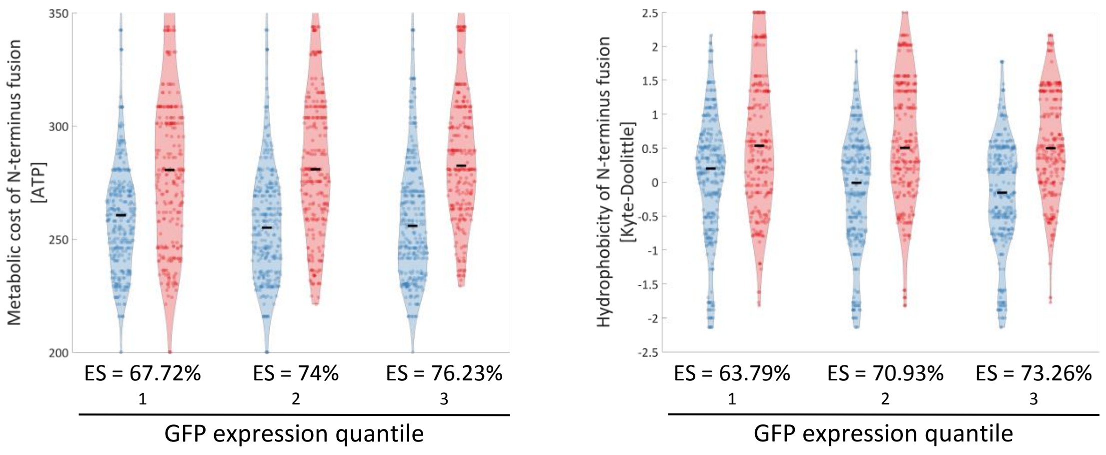
A| Correlation between fitness and GFP level for each FitSeq repeat. The correlation between fitness and GFP expression level is presented for each independent competition of the synthetic library. The Pearson's r for each repeat is: -0.76, -0.76, -0.79, -0.78, -0.74 and -0.77, respectively. All p-Values are lower than 10^{-200}.

B| Contribution of each feature to fitness residual in bins of GFP expression level. Library variants were binned according to GFP protein expression level and further split between positive and negative fitness residual variants. Positive (blue) and negative (red) variants from each GFP expression bin were then compared according to each of the eight features that affect fitness residual. Gray boxplots represent entire library variants.

Supplementary Figure 2

A


B


C


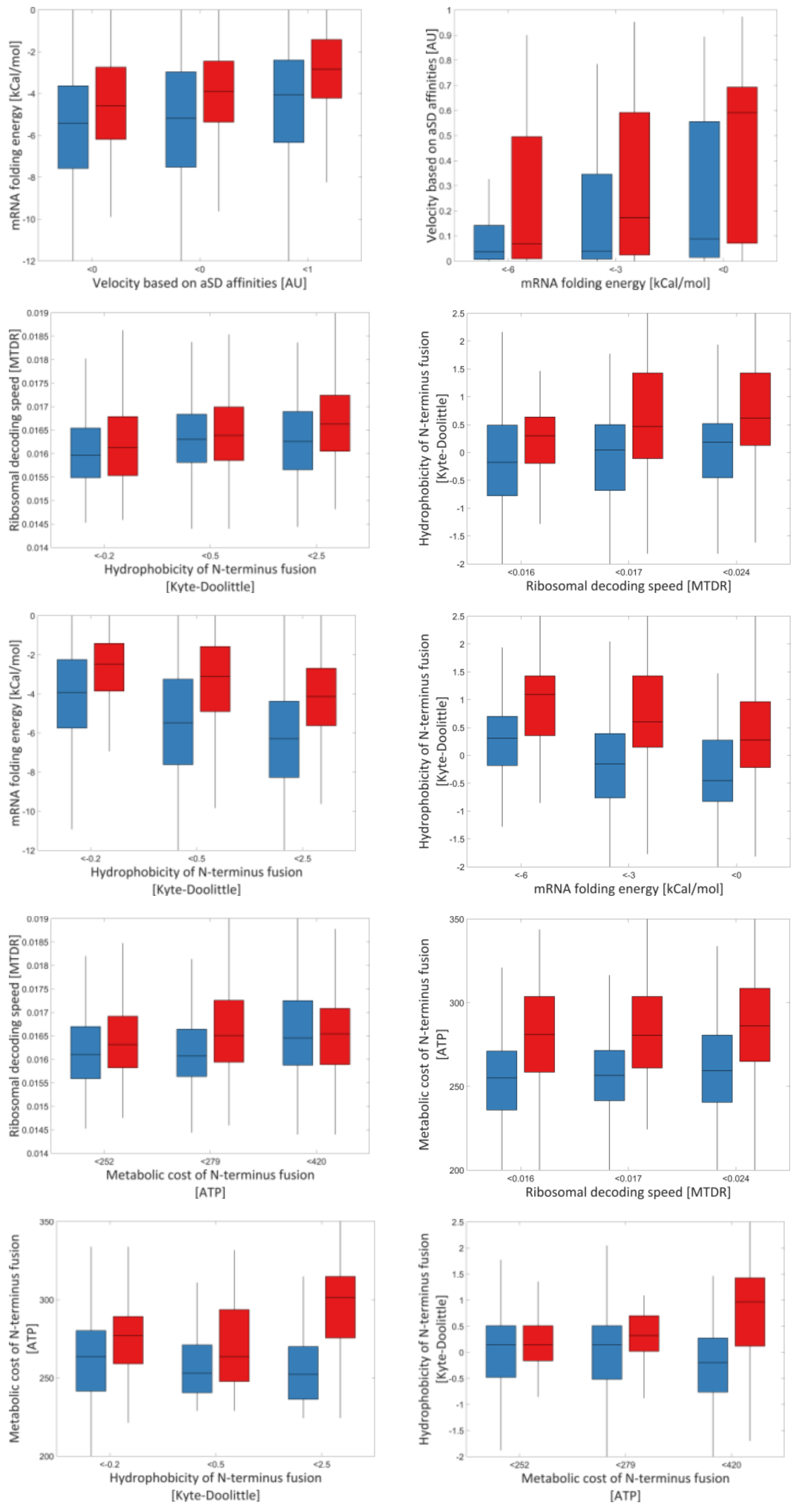
Supplementary Figure 2, related to Figure 2+3+4

A| Positive fitness residual variants have lower GFP mRNA levels but higher initiation rates compared to negative variants. Positive (blues dots) and negative (red dots) variants are drawn according to their mRNA level of the GFP gene (X-axis) and their translation initiation rate (Y-axis). The dashed line represents the optimal linear boundary line between the positive and negative variants, as was computed by training an SVM classifier with a linear kernel function on all the variants, using the two axes as features.

B| Color map of effect size for mRNA folding energy comparison between positive and negative fitness residual variants. X-axis is the window starting position and Y-axis is the window size. The largest effect size of the difference in folding energy between positive and negative variants is observed when the window is positioned exactly at the variable region, just after the AUG codon.

C| The higher the GFP expression, the more beneficial it is to utilize cheap or hydrophilic amino acids. **Left** Positive and negative variants were split into three equally-populated quantiles according to GFP expression levels. Then, the effect size for hydrophobicity between positive and negative variants was calculated for all quantiles. The difference in effect sizes between 1st and 2nd and between 1st and 3rd quantiles was calculated and found to be significant compared to random split of the variants (p-Values=0.018 and 0.0022, respectively). “ES” denotes Effect Size. **Right** Same as left, only for amino acid synthesis cost (p-Values=0.0088 and 0.0008).

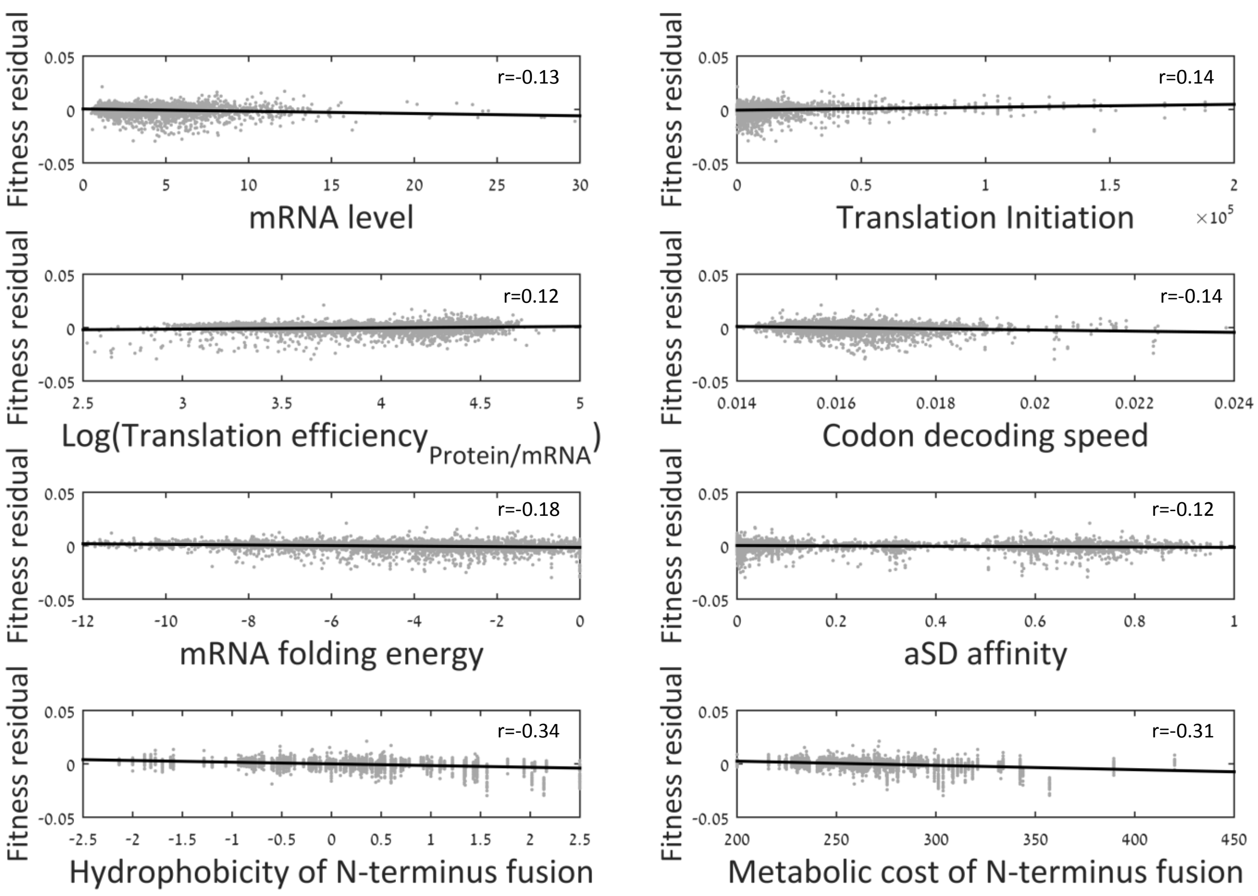
Supplementary Figure 3



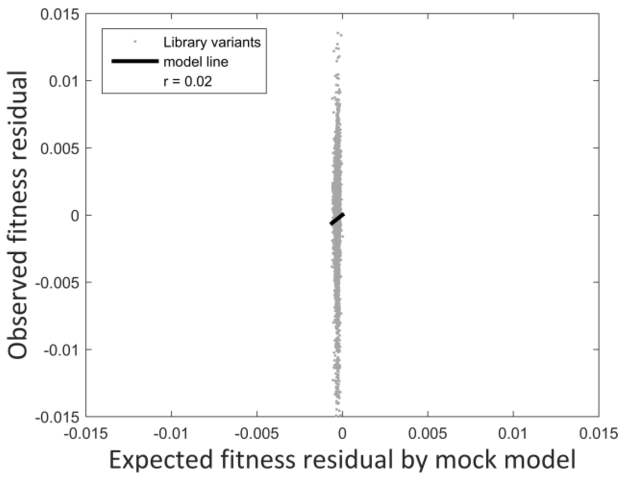
Supplementary Figure 3, related to Figure 5 – Controlling the correlation between each feature
To test the independent contribution of each feature to fitness residual, we checked the Pearson correlation between each pair of features (Figure 5). For features with r>0.1, we binned library variants according to the first feature, and then compared in each bin the difference between positive and negative fitness residual variants in the second feature. We then reciprocally binned the data according to the second feature and compared the first feature. List of feature pairs for which this analysis was performed: mRNA folding energy with aSD velocity, ribosomal decoding speed with hydrophobicity, mRNA folding energy with hydrophobicity, ribosomal decoding speed with cost of N-terminus fusion and hydrophobicity with cost of N-terminus fusion. For example, binned positive and negative fitness residual variants according to anti-Shine Dalgarno affinities still demonstrate mRNA folding energy difference in all bins (Wilcoxon rank-sum p-Value=$2.8 \cdot 10^{-5}$, $2 \cdot 10^{-7}$ and $2.5 \cdot 10^{-10}$). The reverse (bin according to folding energy and significant difference in aSD affinities) is also observed (Wilcoxon rank-sum p-Value=$8.9 \cdot 10^{-3}$, $4.5 \cdot 10^{-6}$ and $3 \cdot 10^{-11}$).

Supplementary Figure 4

A



B



Supplementary Figure 4, related to Figure 7

A| Individual sequence features demonstrate weak correlations with fitness residual

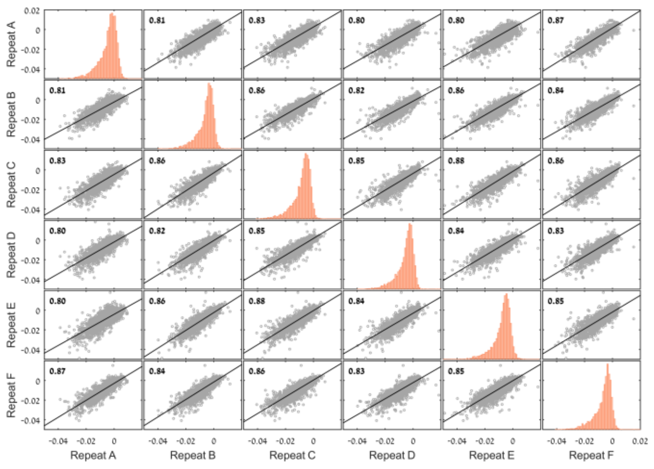
Correlation of each feature, which was shown to differentiate between positive and negative variants, with fitness residual. Pearson's r was calculated for each correlation.

B| Mock model and cross validation for linear regression model

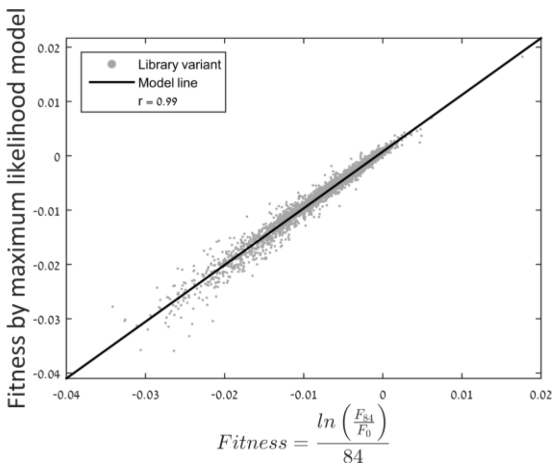
Mock regression model that was trained on randomly scrambled library data fails to accurately predict fitness residual.

Supplementary Figure 5

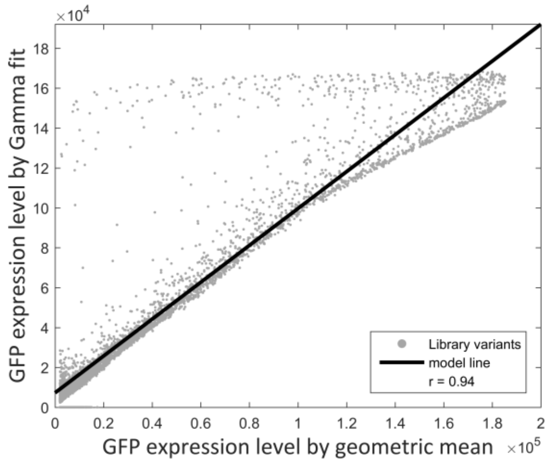
A



B



C



Supplementary Figure 5, related to Methods

A+B| comparison of fitness evaluation

A| Comparison of fitness estimation of all library variants between the different repeats of the experiment.

B| The fitness estimations for all library variants with the two methods that were used in this study are highly correlated (see Methods, Pearson's r=0.99, p-Value<10^{-200}).

C| Two methods of evaluation GFP expression level from FACS data

In *Goodman et al.*, cells were sorted into 12 expression bins using FACS, and in each bin the relative frequency of each variant was measured using deep-sequencing. The estimated expression level of each variant was then calculated by computing the weighted geometric mean of the bins' median expression level, using the relative frequency of each variant at each bin as the bin's weight. In order to validate these data, we estimated the GFP expression level from the raw data by fitting gamma distribution parameters to the histogram of each variant's frequencies in all bins (see Methods). These two estimation methods are highly correlated (r=0.94, p-Value<10^{-200}), yet ~600 variants showed high expression levels according to the gamma fit method, while coming from the entire range of expression level using the geometric mean method. We thus excluded these variants from our analyses since their GFP measurement is not accurate.

Table S1, related to Figure 1 – Raw sequencing data.

Table S2, related to Figure 1 – Fitness per library variant.

Table S3, related to Figures 2-4 – Fitness residual and feature values per library variant.

Table S4, related to Figure 7 – Predicted fitness residual and feature values per *E. coli* or *B. subtilis* gene.

Supplementary Materials and Methods File

Library architecture

The synthetic library was provided to us by Goodman *et al.* (Goodman et al., 2013) and is fully described there. In short, each variant in the library harbors a unique 5' gene architecture that is composed of a promoter, a Ribosome Binding Sites (RBS) and an N'-terminus amino acid fusion of 11 amino acids followed by a sfGFP gene. The library as a whole includes: two promoters with either high or low transcription rate. Three synthetic RBSs with strong, medium, or low translation initiation rates, as well as 137 different genomic RBSs that were defined as the 20bps upstream to the ORF of 137 *E. coli* genes. And finally, 137 coding sequences (CDS) consisting of the first 11 amino acids from the same genes. Each CDS appears in the library in 13 different nucleotide sequences representing alternative synonymous forms. All combinations amounted in 14,234 distinct library variants.

Competition Assay

Competition experiment was carried out by serial dilution. The library was grown on 1.2ml of LB + 50μg/ml kanamycin at 30°C, the exact same conditions as was used in Goodman *et al.* to measure GFP expression level. We grew six parallel, independent lineages and each was diluted daily by a factor of 1:120 into fresh media (resulting in ~6.9 generations per dilution). This procedure was repeated for 12 days and samples were taken from each lineage every four days (~27 generations), mixed with glycerol and kept at -80°C.

Library preparation and sequencing

Plasmids from time zero (library "ancestor") and all other samples were purified with a QIAgene mini-prep kit and used as templates for PCR to amplify specifically the variable region of all variants in the population. To minimize PCR and sampling biases, we used a large amount of template, ~500ng of DNA, and a relatively short PCR of 26 rounds. The forward primer (sequence: CAGCTCTTCGCCTTTACGCATATG) was paired with 5 different reverse primers that are one bp shifted from each other to insure that library complexity was high enough for Illumina sequencing:

R1: GACAATGAAAAGCTTAGTCATGGCG ; R2: ACAATGAAAAGCTTAGTCATGGCG

R3: CAATGAAAAGCTTAGTCATGGCG ; R4: AATGAAAAGCTTAGTCATGGCG

R5: ATGAAAAGCTTAGTCATGGCG

PCR products were then run on BluePipin to capture the correct amplicon size of ~140 bps and remove any un-specific amplicons. Then, DNA buffer was exchanged using Agencourt AmPure SPRI bead cleanup protocol. Hiseq library was prepared next using the sequencing library module from *Blecher-Gonen. et al.* 2013 (Blecher-Gonen et al., 2013). In short, blunt ends were repaired, Adenine bases were added to the 3' end of the fragments, barcode adapters containing a T overhang were ligated, and finally the adapted fragments were amplified. The process was repeated for each sample with a different Illumina DNA barcode for multiplexing, and then all samples were pooled in equal amounts and sequenced. We performed a 125bp paired end high output run on HiSeq 2500 PE Cluster Kit v4. Base calling is performed by RTA v. 1.18.64, and de-multiplexing is carried out with Casava v. 1.8.2, outputting results in FASTQ format.

Data processing

De-multiplexed data was received in the form of FASTQ files split into samples. First, SeqPrep (https://github.com/jstjohn/SeqPrep) was used to merge paired reads into a single contig, to increase sequence fidelity over regions of dual coverage. The size of each contig was then compared to the theoretical combined length of the forward primer, the reverse primer and the variable region of the variants. Next, the forward and reverse primers were found on each contig (allowing for 2 mismatches) and trimmed out. This step was performed for both the forward and reverse complement sequences of the contig, to account for non-directional ligation of the adaptors during library preparation. Then, the reverse primer was searched at the last 5 nucleotides of the contig to account for different primer lengths. After primers were trimmed, the contig was tested again for its length to ensure no indels had occurred. Contigs were then compared sequentially to the entire library, comparing the sequence of each contig to the sequence of each variant. Any contig without a matching variant within two mismatches or less was discarded. Contigs with more than a single matching variant with the same reliability were also discarded due to ambiguity. Each contig that passed these filters was counted in key-value data structure, storing all variants in the library and their frequency in each sample. These data were then used for all downstream analyses. See raw data in Table S1, related to Figure 1.

Fitness estimation

Fitness effect is derived from the following equation:

$$f(t) = f(anc) \cdot (1+s)^t \approx f(anc) \cdot e^{st}$$

Where f is the variant frequency, t is the generation number and s is the fitness effect.

To extract fitness effect, we took two independent approaches. First, we took the logarithm of the ratio between the frequency of a variant at a certain time point and its frequency at time zero. We then divided this value by the number of generations. This calculation was performed both for generation ~84 and generation ~56. See fitness per variant in Table S2, related to Figure 1.

Second, we derived fitness by employing a Maximum-Likelihood (ML) algorithm on all frequency measurements along the competition experiment per variant. A key challenge to accurately estimating each variant's fitness over many generations is that the mean fitness of the population (against which each variant competes) changes (improves) over time. This is caused because more fit variants expand in the population at the expense of other (less fit) strains. To overcome this challenge, we use a Poisson likelihood maximization strategy (see full description below). Briefly, we make a first fitness estimate of each variant using a simple log-linear regression over the first three time points. Based on these estimations, the initial relative frequencies of each variant, and a noise model that accounts for experimental errors (Levy et al., 2015), we estimate the expected trajectory of each variant and compare this to the measured trajectory. We next make small changes to our fitness estimates, repeat this comparison, accept updated fitness estimates if they better fit the data (higher likelihood), and perform this procedure iteratively until fitness estimates are stable (maximized likelihood).

These two fitness-estimation methods were highly correlated (Supplementary Figure 5A+B, r=0.99, p-Value<10^{-200}) and resulted with the same conclusions throughout our analyses.

GFP expression level estimation

GFP expression levels were taken from *Goodman et al.* (Goodman et al., 2013) data, in which it was calculated using the method described in *Kosuri et al.* (Kosuri et al., 2013). In short, cells were sorted into 12 expression bins using FACS, and in each bin the relative frequency of each variant was measured using deep-sequencing. The estimated expression level of each variant was then calculated by computing the weighted geometric mean of the bins' median expression level, using the relative frequency of each variant as the bin's weight.

In order to validate this data, we estimated the GFP expression level from the raw data in *Goodman et al.* by fitting gamma distribution parameters (suggested before as a model to capture noise, or spread of expression values of a gene within an isogenic population (Friedman et al., 2006)) to the histogram of each variant's frequencies in all bins. This gamma distribution follows this equation: $P(x) = \frac{x^{a-1}e^{-\frac{x}{b}}}{b^{a}\Gamma(a)}$ where Γ denotes the gamma function.

These two estimation methods are highly correlated (Supplementary Figure 5C, r=0.94, p-Value<10^{-200}). However, we noticed that ~600 variants showed high expression levels according to the gamma fit method, while coming from the entire range of expression level using the geometric mean method. When closely examining these cases, we noticed that the source for the disagreement between the two methods is that these variants were observed only in two bins, with one of them being the highest bin, and the other not being the second highest. Therefore, we decided that the expression estimation for these variants is unreliable and excluded them from our analyses.

Calculation of fitness residuals and classifying variants according to their positive or negative fitness residual sign

We defined "fitness residual" of a variant as the difference between the observed fitness by FitSeq and the fitness predicted by a linear model given the variant's GFP expression level. To calculate fitness residual, we performed the following steps:

First, we filtered out variants that demonstrate a lower GFP level than 2^{11}[AU], since below this threshold the GFP measurement method is not sensitive to accurately measure GFP. We also excluded variants with a GFP level above $2^{17.5}$[AU], as above this threshold the measurement method saturates. Notably, only variants with the "high promoter" were included in the analysis, since almost all "low promoter" variants did not pass the protein level filter. This decision was essential as the few "low promoter" variants that did pass this threshold show biased values of sequence features, such as a very low GC-content, which could mask real signals.

Next, we fitted a linear regression model between fitness and GFP expression levels for each of the six independent FitSeq repeats separately at each of the last two time points (generations ~56 and ~84). Then, variants for which fitness residual was in the top or bottom 5% were excluded and a new regression line was fitted in order to reach a better fit. These outliers were excluded only for the sake of fitting a regression line and were still included in our downstream analyses. Then, a fitness residual score was calculated for each variant at each repeat of the experiment and on each of the two time points.

We then split the variants into two groups: positive or negative fitness residual variants. To account for random processes (experimental errors and drift), "positive" or "negative" class was assigned for a given variant only if it showed a positive/negative fitness residual sign in at least 5 lineages in both time points. The set of all the above filters resulted in 975 variants in the positive variant group and 815 in the negative variant group.

Since we noticed that some of the negative variants have extreme negative fitness residual values, we further classified them as "underachievers". Underachiever variants were defined as variants that repeatedly showed fitness residual values in the bottom 5% of the entire library.

Similar to the positive/negative classification, a variant is assigned as “underachiever” only if it is found in the bottom 5% in at least 5 out of the 6 linages in both time points, which resulted in 80 variants.

Parameter comparison between two fitness residual groups

A one-sided Wilcoxon rank-sum test was used to compare the distributions of different sequence parameters between the positive and negative fitness residual groups. We also tested the effect size of each parameter using the “Probability of superiority” method (Ruscio, 2008) that calculates the probability to randomly choose a member from group A with a higher value than a random member from group B.

To compare between effect sizes according to GFP expression levels, we split the positive and negative variant groups into three quantiles according to GFP expression levels. Then, the effect size for hydrophobicity or amino acid synthesis cost between positive and negative variants were calculated for each quantile. We then performed an empirical p-Value estimation by randomly choosing three data sets with the same number of variants, and computed the effect size at each set. This sampling was performed 10^4 times, and p-Value was estimated by counting the number of times the difference in effect sizes between the first and second sets and between the first and third sets were lower in the real data than the difference in effect sizes of the random groups.

Calculating translation initiation rate per variant

We estimated the translation initiation rate of each variant with the “RBS calculator” (Espah Borujeni et al., 2014; Salis et al., 2009), which simulates initiation rates given a UTR and a coding sequence. This calculation is achieved by using a bio-mechanic model combining the affinity to the anti-Shine Dalgarno sequence of the ribosome, mRNA secondary structure of the UTR and coding sequence, and steric interference of the ribosome and the mRNA.

Mean of the Typical Decoding Rates (MTDR) estimation

To evaluate codon-decoding times by the ribosome we used a published index of Mean of the Typical Decoding Rates (MTDR) values (Dana and Tuller, 2014), which were derived from ribosome profiling data (Li et al., 2012). MTDR values for each of the 61 sense codons are driven from measured ribosome density, when the ribosome A site is on a codon, averaged over all the appearances of the codon within mRNAs. This measurement estimates the translation speed of each codon, and it correlates significantly (r=0.46 for *E. coli)* with tRNA availability. The final score given to each variant was the harmonic mean of its MTDR values of the first 11 amino acids.

Folding energy estimation of mRNA secondary structure

We calculated folding energy of mRNA secondary structure for each variant by using the ViennaRNA package algorithm (Lorenz et al., 2011). Each sequence was computed by a sliding window, whose starting position ranged from position -18 to position 32. The calculation was repeated with different window sizes (20-60bps). All calculations were done assuming a temperature of 30^{o}C.

Model for estimating translation velocity based on anti-Shine Dalgarno affinity

The Shine-Dalgarno affinity was calculated identically to Li *et al.* (Akashi and Gojobori, 2002). In short, for each position we calculated the affinity of 8-11bps upstream of that position (the distance between the ribosome A site and the aSD site) to the anti-Shine Dalgarno motif. The free energy of interaction between the aSD motif and the mRNA sequence (ΔG) was calculated for all possible 10mer sequences for that position using the RNA annealing function from the ViennaRNA package algorithm (Lorenz et al., 2011), and the highest affinity (lowest energy) score was used. We calculated the affinity for all positions for which the annealing with the aSD motif resides in the 11-amino acid fusion (positions 19-33) and then transformed all affinities of a given variable sequence to estimated ribosomal velocity as follows.

We converted the ΔG estimates into the equilibrium constant of the interaction, K by:

(i) $$K = e^{-\frac{\Delta G}{RT}}$$

Where ΔG denotes the SD affinity, R denotes the gas constant and T denotes the temperature.

This equilibrium constant, at the nth codon along a sequence, is defined in turn, given the association reaction rate (k_f) which represents the association to the current site (n), and a dissociation reaction (k_b) that represents the dissociation to the current site as:

(ii) $$K = \frac{k_{f_n}}{k_{b_n}}$$

The elongation velocity (v) as the ribosome moves from current site n to the n+1 site is given by the harmonic mean of the dissociation reaction of site n and the association reaction of site $n+1$:

(iii) $$\frac{1}{v_{n\to n+1}} = \frac{1}{k_{b_n}} + \frac{1}{k_{f_{n+1}}}$$

(iv) $$v_{n\to n+1} = \frac{k_{b_n} k_{f_{n+1}}}{k_{b_n} + k_{f_{n+1}}}$$

We further assume that the association reaction rate is not dependent on the sequence, therefore for every n, $k_{f_n} = k_f$. Introducing equations (i)-(ii) to the equation (iv), results in a term for the ribosomal velocity at a specific position by the anti-Shine Dalgarno affinity:

(v) $$v_{n\to n+1} = \frac{k_f \cdot k_f K^{-1}}{k_f(1+K^{-1})} = k_f \frac{e^{\frac{\Delta G}{RT}}}{1+e^{\frac{\Delta G}{RT}}}$$

To calculate the average ribosomal velocity across the entire N-terminus fusion sequence of each library variant, we calculated the harmonic mean of the velocity values for all positions. The analysis was performed also at a codon resolution, taking into account only positions of the sequence that are the first nucleotide of codons, which yielded similar results to the nucleotide-based analysis.

Amino acid property estimation of N-terminus fusion amino acids

Hydrophobicity of each 11-amino acid N-terminus peptide was calculated according to its score on the Kyte-Doolittle scale (Kyte and Doolittle, 1982). Amino acid cost was derived from Akashi and Gojobori (Akashi and Gojobori, 2002) in the form of the amount of energy consumed for its production in high energy ATP or GTP bonds. Cost was either evaluated per amino acid or summed for the whole peptide.

Supply of amino acids were derived from *Bennet et al.* (Bennett et al., 2009), which measured cellular concentrations of amino acid in exponnentially grown *E. coli*. Notably, two amino acids are missing from this table (Gly & Cys), and two amino acids are indistinguishable (Lys & Ile). Therefore, those 4 amino acids were excluded from the this analysis. The demand per amino acid was calculated by multiplying the frequency of each amino acid in each *E. coli* gene by the median ribosome profiling score of the gene (Li et al., 2012). The sum of all genes was defined as the total amino acid demand.

Amino acid enrichment in positive and negative variant groups

To calculate the frequency of the various amino acids in the collective proteome in either the positive or the negative fitness residual group, we counted the occurrences of each amino acid in each variant. We then summed this number for each amino acid across all variants in each group and divided the sum by the number of variants in each group multiplied by 11. To quantify the relationship between amino acid enrichment and energetic-cost or availability we calculated the frequency ratio of each amino acid by dividing the amino acid frequency of the positive fitness residual group by the frequency of the negative group. We then calculated the Pearson correlation between the log2 amino acid enrichment ratio and the amino acid energetic-cost or their availability.

Comparing fitness residual among variants with the same N-terminus fusion by Δfitness-residual

We defined Δfitness-residual as the difference between the fitness residual of a given variant with the average fitness residual of the variant group with the same N-terminus amino acid fusion. Therefore, Δfitness-residual measures the expression cost of a variant normalized to its GFP expression level and its N-terminus amino acid sequence. We then spilt each variant group of the same N-terminus fusion to above-average and below-average variants and calculated for each sub group the mean value for six features (RNA levels, translation initiation rates, translation efficiency, codon decoding speed (MTDR), mRNA secondary structure, and anti-Shine Dalgarno affinity). For each feature, the mean value of the below-average (x-axis) and above-average (y-axis) Δfitness-residual groups were depicted as a scatter plot, in which each point represents a different N-terminus fusion. Then, the deviance of all dots from the identity (X=Y) line was calculated and tested for significance with a one-tailed Student's t-test. To compute an effect size for this enrichment, we used Cohen's d: $d = \frac{\bar{x}_{high} - \bar{x}_{low}}{S}$ where $\bar{x}_{high\backslash low}$ represents the mean of the feature in the above- or below-average group, and S represents the standard deviation of the feature in the entire set of library variants that was used in this study.

A multiple linear regression model to predict fitness residual

We performed a multiple linear regression using all eight features as independent variables (RNA levels, translation initiation rate, translation efficiency (GFP protein/mRNA), mRNA secondary structure, codon-decoding speed, aSD affinities, amino acid metabolic cost and hydrophobicity) and the mean fitness residual across six repeats of FitSeq as the dependent variable. The regression yielded a coefficient for each feature, which were all used in order to predict fitness residual of a given variant.

As a negative control for this model we randomly shuffled each of the features in the library, trained a mock model on this shuffled library, and computed the Pearson correlation coefficient between the observed fitness residual and the expected fitness residual according to the mock model. In order to compute a p-Value we repeated this process 10^5 times, and counted the number of times the correlation coefficient from the mock model was higher than the correlation coefficient from the real model.

To predict fitness residual of natural *E. coli* and *B. subtilis* genes, a second regression model was performed, in which we excluded translation efficiency (due to lack of data for the entire ~4000 *E. coli* genes) and hydrophobicity (due to the fact that hydrophobic motifs in membrane proteins are functional, hence including this feature might lead to wrong estimation of membrane proteins). We also used Lasso regularization and feature selection method (Tibshirani, 1996) with Matlab's "lassoglm" function from the "Statistics and Machine Learning" toolbox to avoid overfitting of the model. The λ value was chosen as the value for which the deviance was one standard deviation higher than the minimum deviance achieved in a 1000-fold cross validation. Out of the six features used for this model, none were excluded by Lasso method. This model performed well in predicting fitness residual of the library variants and a cross validation test resulted in correlation of r=0.3 (p-Value=10^{-10}).

This model was then used to predict fitness residual scores for natural *E. coli* (strain MG1655) and *B. subtilis* (strain 168) genes. For each gene of these species, we computed a score for each feature of the model. We used RNA levels for *E. coli* from a previous RNA-seq experiment in which cells ware grown in LB and were harvested at the logarithmic growth phase. We used published RNA data for *B. subtilis* (Cohen et al., 2016). Translation initiation rates was computed with the same initiation rate model as was used for the library variants (Espah Borujeni et al., 2014; Salis et al., 2009). mRNA secondary structure, codon-decoding speed and aSD affinities were calculated as explained for the library variants. MTDR values for both species were taken from published data (Dana and Tuller, 2014). Amino acid metabolic cost was calculated as the mean value for the entire protein, and for both species the same cost was assigned for each amino acid (Akashi and Gojobori, 2002). Protein expression levels for both species were taken from the integrated datasets in Pax-Db (Wang et al., 2012). See Table S4, related to Figure 7 for feature and fitness residual scores for genes of these two organisms.

As a negative control for the prediction of fitness residual for natural *E. coli* genes, we generated a mock model by randomly shuffling each of the features in the library, training a linear regression model on this shuffled library and using it to predict fitness residual for all *E.*

coli genes. We then compared the standard deviation of the fitness residual predictions by the real model to the one of the mock model. This analysis was repeated 10^5 times to compute a p-Value for the chance of the real model to show a higher standard deviation than the mock model.

Cross validation sets

Cross validation tests of the regression model were performed by randomly choosing training and test sets, in proportions of 70% and 30% of the entire library variants, respectively. In order to account for the fact that some of the information lays in the amino acid sequence, the training/test sets were also separated by the N-terminus amino acid peptide sequences with 41 peptide sequences (~30%) chosen as test set, and the rest as training sets. 10-fold cross validation was performed by randomly generating ten different pairs of training and test sets. The results are based on the average across these 10 repeats.

RNA fitness residual calculation

To evaluate mRNA fitness residuals we repeated the same calculation as described for fitness residual only with the mRNA levels instead of protein levels placed on the x-axis.

Fitness residual and feature values per variant

See Table S3, related to Figures 2-4.

Bibliography

Akashi, H., Gojobori, T., 2002. Metabolic efficiency and amino acid composition in the proteomes of Escherichia coli and Bacillus subtilis. Proc. Natl. Acad. Sci. U. S. A. 99, 3695–3700. doi:10.1073/pnas.062526999

Bennett, B.D., Kimball, E.H., Gao, M., Osterhout, R., Van Dien, S.J., Rabinowitz, J.D., 2009. Absolute metabolite concentrations and implied enzyme active site occupancy in Escherichia coli. Nat. Chem. Biol. 5, 593–9. doi:10.1038/nchembio.186

Blecher-Gonen, R., Barnett-Itzhaki, Z., Jaitin, D., Amann-Zalcenstein, D., Lara-Astiaso, D., Amit, I., 2013. High-throughput chromatin immunoprecipitation for genome-wide mapping of in vivo protein-DNA interactions and epigenomic states. Nat. Protoc. 8, 539–54. doi:10.1038/nprot.2013.023

Cohen, O., Doron, S., Wurtzel, O., Dar, D., Edelheit, S., Karunker, I., Mick, E., Sorek, R., 2016. Comparative transcriptomics across the prokaryotic tree of life. Nucleic Acids Res. gkw394. doi:10.1093/nar/gkw394

Dana, A., Tuller, T., 2014. The effect of tRNA levels on decoding times of mRNA codons. Nucleic Acids Res. 1–11. doi:10.1093/nar/gku646

Espah Borujeni, A., Channarasappa, A.S., Salis, H.M., 2014. Translation rate is controlled by coupled trade-offs between site accessibility, selective RNA unfolding and sliding at upstream standby sites. Nucleic Acids Res. 42, 2646–2659. doi:10.1093/nar/gkt1139

Friedman, N., Cai, L., Xie, X.S., 2006. Linking stochastic dynamics to population distribution: an analytical framework of gene expression. Phys. Rev. Lett. 97, 168302. doi:10.1103/PhysRevLett.97.168302

Goodman, D.B., Church, G.M., Kosuri, S., 2013. Causes and effects of N-terminal codon bias in bacterial genes. Science 342, 475–9. doi:10.1126/science.1241934

Kosuri, S., Goodman, D.B., Cambray, G., Mutalik, V.K., Gao, Y., Arkin, A.P., Endy, D., Church, G.M., 2013. Composability of regulatory sequences controlling transcription and translation in Escherichia coli. Proc. Natl. Acad. Sci. U. S. A. 110, 14024–9. doi:10.1073/pnas.1301301110

Kyte, J., Doolittle, R.F., 1982. A simple method for displaying the hydropathic character of a protein. J. Mol. Biol. 157, 105–32.

Levy, S.F., Blundell, J.R., Venkataram, S., Petrov, D. a., Fisher, D.S., Sherlock, G., 2015. Quantitative evolutionary dynamics using high-resolution lineage tracking. Nature advance on. doi:10.1038/nature14279

Li, G.-W., Oh, E., Weissman, J.S., 2012. The anti-Shine-Dalgarno sequence drives translational pausing and codon choice in bacteria. Nature 484, 538–41. doi:10.1038/nature10965

Lorenz, R., Bernhart, S.H., Höner Zu Siederdissen, C., Tafer, H., Flamm, C., Stadler, P.F., Hofacker, I.L., 2011. ViennaRNA Package 2.0. Algorithms Mol. Biol. 6, 26. doi:10.1186/1748-7188-6-26

Ruscio, J., 2008. A probability-based measure of effect size: robustness to base rates and other factors. Psychol. Methods 13, 19–30. doi:10.1037/1082-989X.13.1.19

Salis, H.M., Mirsky, E. a, Voigt, C. a, 2009. Automated design of synthetic ribosome binding sites to control protein expression. Nat. Biotechnol. 27, 946–950. doi:10.1038/nbt.1568

Tibshirani, R., 1996. Regression Shrinkage and Selection via the Lasso. J. R. Stat. Soc. 58, 267–

288.

Wang, M., Weiss, M., Simonovic, M., Haertinger, G., Schrimpf, S.P., Hengartner, M.O., von Mering, C., 2012. PaxDb, a database of protein abundance averages across all three domains of life. Mol. Cell. Proteomics 11, 492–500. doi:10.1074/mcp.O111.014704

Simulation for measurement errors in GFP expression level

To calculate fitness residual, we correlated fitness to expression level in order to learn the expected fitness of each library variant. Variants were classified as positive/negative fitness residual variants, if their fitness value was repeatedly above/below the expected line, respectively. This approach allowed us to factor out the effect of GFP expression level on fitness and elucidate mechanisms that reduce cost at a given expression level. Yet, a potential bias in our method could arise because of experimental errors in the GFP measurement. Indeed, *Goodman et al*.(Goodman et al., 2013) and *Kosuri et al*.(Kosuri et al., 2013) report an estimated coefficient of variation $\left(\frac{\sigma}{\mu}\right)$ of 0.22 for the GFP level in each variant. We set to assess the potential effect of such measurement error on fitness residual estimation.

We simulated our experimental design with a range of measurement errors for GFP expression level (see below a description of the simulation). This simulation allowed us to evaluate how many variants would be wrongly classified with either positive or negative fitness residual simply due to error in GFP expression levels measurements.

Our results show that for all simulated error levels, even those that far exceed the actual reported error level, we observed much less positive and negative variants in the simulation compared with the actual group size of the positive and negative variants in the FitSeq experiment (Simulation Figure 1A). This result means that our classification of variants into positive and negative fitness residual groups could not be due to error in measurements.

Additionally, our simulation predicts that GFP measurement error alone would result in a negative variant group that is larger than the positive variant group and only upon introduction of fitness residual signal to the simulation, more positive than negative variants were observed (Simulation Figure 1B). Reassuringly, our data resulted in a greater number of positive variants compared to negative, suggesting that library variants show a real phenomenon of fitness residual that minimize cost of gene expression.

Next, we turned to test whether the features that we discovered to differ between positive and negative variants, and thus affect fitness residual, could be observed due to measurement errors (at the reported error level of *Goodman et al.*). For all features, except GFP mRNA levels (p-Value=0.8), we observed that the effect size that separates between the positive and negative variant groups is much higher than would appear simply because of experimental errors (Simulation Figure 1C). P-Values for initiation rate$<10^{-4}$, translation efficiency$_{\text{protein}\backslash\text{mRNA}}<10^{-4}$, codon decoding speed$=3.2\text{x}10^{-3}$, mRNA folding$=2\text{x}10^{-4}$, aSD velocity$<10^{-4}$, amino acid metabolic cost$<10^{-4}$ and hydrophobicity$<10^{-4}$. These results mean that the molecular mechanisms we revealed in this work are not observed due to an experimental error, but rather reflect a genuine biological phenomenon relating to expression cost.

Regarding mRNA levels, we present three arguments for its relevance to fitness residual: First, translation efficiency, defined as GFP protein/mRNA, at the variant level was still observed as significant in this analysis, a result which strengthens our claim that producing more proteins per mRNA reduces cost of gene expression. Second, we calculated "RNA fitness residual" based on fitness and two independent GFP mRNA measurements (Simulation Figure 1D and see methods) and observed that positive variants demonstrated lower mRNA levels compared to negative (Effect size=55.13%, rank-sum p-Value$=7.2\text{x}10^{-5}$), suggesting that higher mRNA levels are costly and reduce fitness residual. Third, we observed 80 variants with consistent extremely low fitness residual ("underachievers", see main text). While these very low fitness residual variants cannot be explained by measurement error (they do not appear in the simulation), they also demonstrate even higher mRNA levels than the negative variant group. All of these points suggest that mRNA level, as the other eight parameters, indeed reduce expression cost and increase fitness.

Detailed description of the simulation

For each single run of the simulation, 12 independent repeats are performed that simulate the 12 sampling points we used to classify each library variant with either positive or negative fitness residual. The simulation steps are as follows:

1. GFP expression level is randomly assigned for 4115 simulated variants from a log uniform distribution of GFP levels, which is similar to the distribution of the synthetic library we used in this study.
2. Fitness score is assigned to each simulated variant, according to the observed correlation between GFP expression level and fitness in our experiment.

(i) $\widehat{f_i} = a\widehat{GFP_i} + b$

$\widehat{f_i}$ – fitness predicted from GFP level according to linear model

$\widehat{GFP_i}$ – GFP expression levels drawn from the experimental distribution of GFP levels

a,b – confidents of the linear model as extracted from the measured data.

3. For each simulated variant, 12 independent measurement errors are added to the assigned fitness, in order to simulate the 12 repeats (6 from each time point) that were used for classifying the library variants. This fitness measurement error is drawn from a normal distribution with a mean of zero and a standard deviation of 0.03, N(0, 0.03). This SD was used as it is the mean SD we observed for the library variants based on the independent repeats of our experiment.

(ii) $f_i = \widehat{f_i} + N(0{,}0.03) = a\widehat{GFP_i} + b + N(0{,}0.03)$

f_i - simulated fitness

4. A measurement error is added to the GFP level of each simulated variant by drawing a measurement error from a normal distribution. Since the absolute size of the measurement error is dependent on expression level (higher expressions mean larger errors) the simulated measurement error is chosen from a normal distribution with a mean of zero and an SD that is the multiplication of the simulated GFP level by the noise factor, N_x, which is a parameter of the simulation.

(iii) $GFP_i = \widehat{GFP_i} + N\left(0, N_x \cdot \widehat{GFP_i}\right)$

GFP_i – simulated GFP levels

N_x – GFP noise factor

5. All simulated variants are then classified with positive or negative variants with the same approach as described in the methods section and the size of each group is counted.
6. The Pearson correlation between simulated GFP levels and simulated fitness scores is also calculated and recorded.

   The above steps describe a single run of the simulation. We performed 10^4 runs to calculate p-Value to the group size we observed in our study.

   We then turned to simulate the effect size of each feature we observed to affect fitness residual:

   For each simulated variant (as described in steps 1-6) we assigned a random feature score (taken from the actual values in the library) based on its randomly assigned GFP expression level. We performed this step while maintaining the correlation between expression level and the specific feature. For example, in the synthetic library we used in this study, mRNA folding energy is correlated with GFP expression with r=0.15. We maintained this correlation in the random assignment of folding energies to the simulated variants.

   In order to produce a correlation r between a given feature and the simulated GFP levels, we used the following method:

   a. Using Matlab's Statistics and Machine Learning toolbox copularnd() function we generated two random vectors with a normal distribution and 4115 samples each: $\{U_1, U_2\}$ with a correlation r between them.
   b. Each of the vectors was sorted and a vector of the indices of U mapping to the sorted vector was returned, such that: $U_i(I_i) = S_i$, where S_i is the sorted vector, and I_i are the indexes of U_i ordered according to their rank.

c. The feature vector and the GFP vector were also sorted, we mark their sorted version as $S_{feature}$,S_{GFP} respectively.

d. The correlated vectors of the feature and GFP vectors $X_{feature}$,X_{GFP} were created by sampling the sorted vectors, using the indexes mapping to the sorted correlated normal vectors.

$$X_{feature} = S_{feature}(I_1)$$
$$X_{GFP} = S_{GFP}(I_2)$$

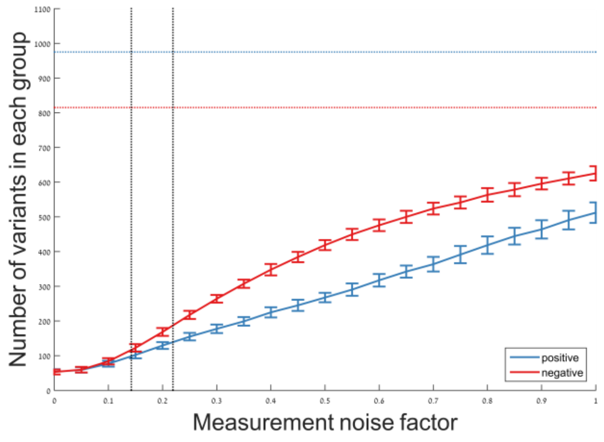
After creating a pair of vectors representing the feature values and simulated GFP values which have the same correlation as the measured feature have with the GFP expression levels, we repeated steps 2-5 in order to extract fitness residual values and positive/negative classification for the new permuted GFP vector.

We repeated the above process for each of the eight features.

7. Then for each feature, we calculated the effect size between the positive and negative fitness residual variants in the simulation. Since there are only experimental errors and no real signal in the simulation, this measured effect size is the threshold for our experimental design. To calculate p-Value to the effect size that we observed in the experiment data, we performed 10^4 runs of the simulation and counted the number of times the effect size was higher than the one observed in the experiment.

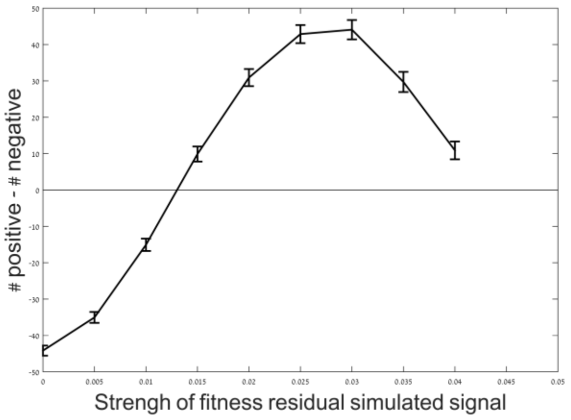
Simulation Figure 1

A



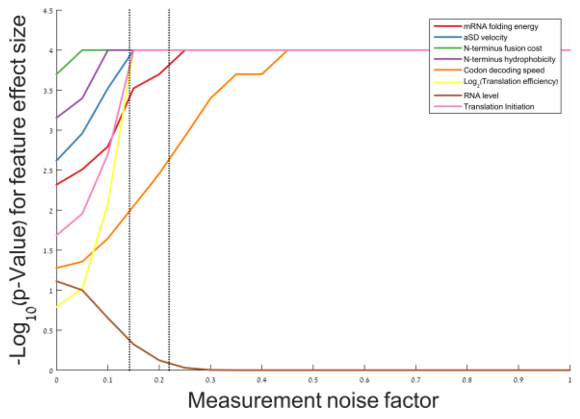
Blue horizontal line - number of positive variants in the library.
Red horizontal line - number of negative variants in the library.
Left vertical line - noise factor corresponding to measured r between fitness and expression level in our data.
Right vertical line – noise factor as estimated from *Kosuri et al*.

B

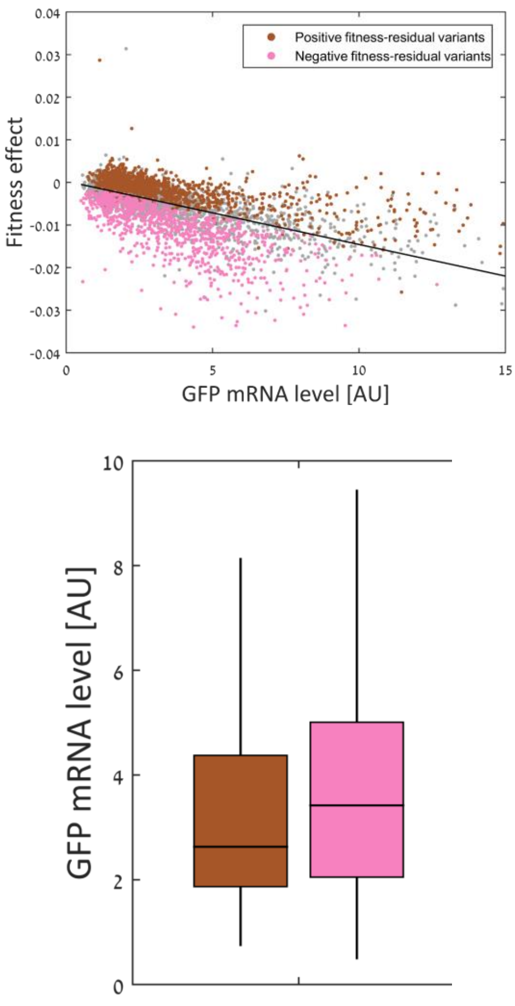


Noise factor = 0.22 (as measured by from *Kosuri et al.*)

C



D



Poisson likelihood maximization strategy to estimate fitness

Here, each genetic design is called a lineage and the total number of lineages, l, is 14234. Let $n_i(t)$ be the cell number of lineage i at the t-th generation, and $\overline{x}(t)$ be the mean fitness of the population at the t-th generation. In a limited-resource environment, the growth of lineage i follows,

$$n_i(t) = n_i(0) \cdot e^{\int_0^t (x_i - \overline{x}(\tau))d\tau},$$

where $\overline{x}(t) = \frac{\sum_{i=0}^{l} n_i(t) \cdot x_i}{\sum_{i=0}^{l} n_i(t)}$.

Therefore, the growth of lineage i can be rewritten as,

$$n_i(t + \Delta t) = n_i(0) \cdot e^{\left(\int_0^t (x_i - \overline{x}(\tau))d\tau + \int_t^{t+\Delta t} (x_i - \overline{x}(\tau))d\tau\right)} \approx n_i(t) \cdot e^{\Delta t \cdot (x_i - \overline{x}(t))}.$$

We estimate the fitness of all lineages using the following method:

1. Let x_i be fitness of lineage i. Let $r_i(t)$ and $f_i(t)$ be the experimental read number and read frequency of lineage i at the t-th generation, respectively, i.e., $f_i(t) = \frac{r_i(t)}{\sum_{i=0}^{l} r_i(t)}$. Let Δt be the number of generations passed between two bottlenecks. Here, $\Delta t \approx 28$. Make an initial guess of x_i by linear regression of $\ln f_i(0), \ln f_i(\Delta t)$, and $\ln f_i(2\Delta t)$. Note that we only do the linear regression for the lineage with an initial experimental read number larger than 10, i.e., $r_i(0) > 10$. Lineages with lower read numbers are too noisy to get an accurate estimation.

2. Let $\hat{n}_i(t)$ be the estimated cell number of lineage i at the t-th generation. Assume that $\hat{n}_i(0) = \frac{r_i(0) \cdot N}{\sum_{i=0}^{l} r_i(0)}$, where N is the total cell number after bottleneck. Here, $N = 9.37 \times 10^7$. Calculate $\hat{n}_i(t)$ and $\overline{x}(t)$ for the first 4 sequencing time points using,

$$\begin{cases} \overline{x}(k\Delta t) = \frac{\sum_{i=0}^{l} \hat{n}_i(k\Delta t) \cdot x_i}{\sum_{i=0}^{l} \hat{n}_i(k\Delta t)}, \\ \hat{n}_i(k\Delta t + \Delta t) = \hat{n}_i(k\Delta t) \cdot e^{\Delta t \cdot (x_i - \overline{x}(k\Delta t))}, \end{cases} \qquad k = 0,1,2,3.$$

3. Let $\hat{r}_i(t)$ be the estimation of $r_i(t)$. Thus,

$$\hat{r}_i(k\Delta t) = \frac{\hat{n}_i(k\Delta t) \cdot \sum_{i=0}^{l} r_i(k\Delta t)}{\sum_{i=0}^{l} \hat{n}_i(k\Delta t)}, \qquad k = 0,1,2,3.$$

4. Define the Poisson likelihood function as,

$$F(x_1, x_2, \cdots, x_l) = \sum_{i=1}^{l} \sum_{k=0}^{3} \ln\left(\frac{\hat{r}_i(k\Delta t)^{r_i(k\Delta t)} \cdot e^{-\hat{r}_i(k\Delta t)}}{r_i(k\Delta t)!}\right).$$

   Here, $\frac{\hat{r}_i(t)^{r_i(t)} \cdot e^{-\hat{r}_i(t)}}{r_i(t)!}$ gives the probability of observing the experimental read number $r_i(t)$ given the estimated read number $\hat{r}_i(t)$.

5. Obtain the optimal fitness for all lineages by maximizing $F(x_1, x_2, \cdots, x_l)$. We use the *fminunc* function in the MATLAB Optimization Toolbox to do the optimization. The function *fminunc* uses Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm with cubic line search to solve unconstrained nonlinear optimization problems. BFGS algorithm is an iterative method, which seeks a stationary point (with the derivative or gradient of the function being zero) of a function as Newton's method. The BFGS algorithm is a fast-converging algorithm and we find that all replicates nearly converge by 21 iterations (Maximum Likelihood Figure 1A).

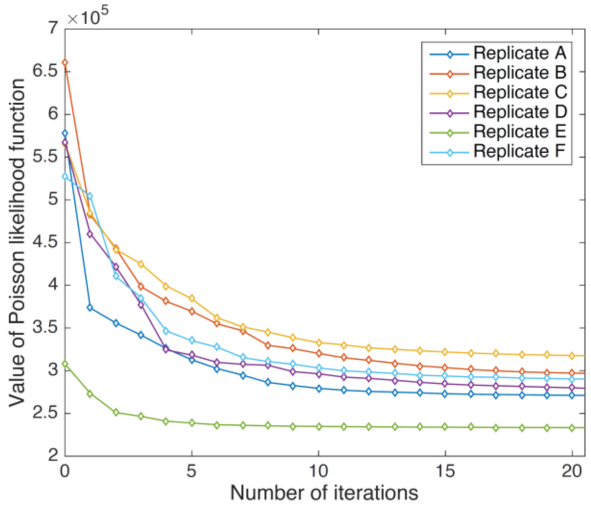
This Poisson likelihood optimization method provides a more accurate estimation of fitness compared with log-linear regression method. We define the relative error of the estimation of read number of lineage i as $\frac{1}{4} \cdot \sum_{k=0}^{3} \frac{2|\hat{r}_i(k\Delta t) - r_i(k\Delta t)|}{\hat{r}_i(k\Delta t) + r_i(k\Delta t)}$ and then calculate the relative error between the measured lineage trajectories and the lineage trajectories that would be expected based on our fitness estimates. We find that the fitnesses estimated using Poisson likelihood optimization have lower errors than the log-linear regression method (Maximum Likelihood Figure 1B).

To test the consistency of the estimated fitnesses, we compared the Poisson likelihood optimization fitness estimates between all replicates. We find that replicates generally correlate well (Pearson's correlation > 0.76 for all replicates), with higher fitness designs having higher correlations between replicates. Additionally, we find that variance between fitness estimates across the six replicates is generally low (Maximum Likelihood Figure 1C).

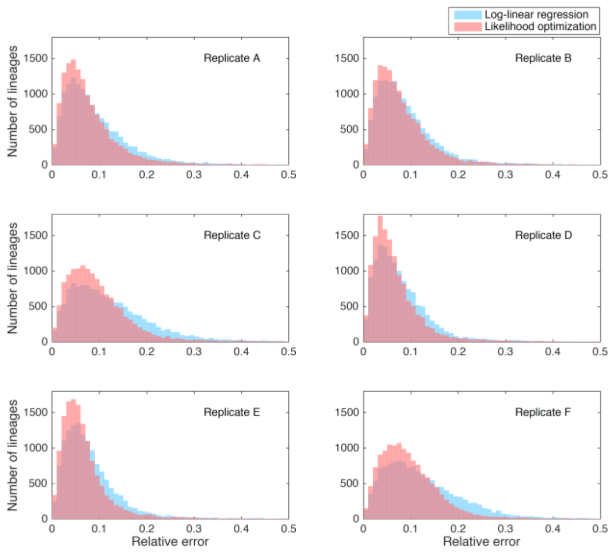
To validate that the method worked as expected, we ran a simulation of the evolutionary process that would be expected in replicates E and F, given the fitness of each cell in lineage i has the same fitness x_i and that x_i is the fitness estimated from real data using the Poisson likelihood optimization method. We assumed that there are no mutations and that the offspring per generation of an individual follows Poisson distribution with mean $1 + x_i$. Let $n_i^{simu}(t)$ be the cell number of lineage i at the t-th generation in the simulation. In the simulation, the cell number of lineage i at the beginning is set as $n_i^{simu}(0) = \frac{r_i(0) \cdot N}{\sum_{i=0}^{l} r_i(0)}$, where $r_i(0)$ is the initial read number of lineage i. The evolution is simulated for 84 generations. To simulate the sequencing process where the data is sequenced every 28 generations, we let $r_i^{simu}(t)$ be the read number of lineage i at the t-th generation, assuming that $r_i^{simu}(t)$ follows the Poisson distribution with mean $\frac{n_i^{simu}(t) \cdot r_i(t)}{\sum_{i=0}^{l} n_i^{simu}(t)}$, where $r_i(t)$ is the initial read number of lineage i. We then used the read number data obtained from these simulations to estimate the fitness for each lineage using the Poisson likelihood optimization method. We find that our fitness estimates correlate extremely well with the true (assigned) fitness of each lineage in both simulations (Maximum Likelihood Figure 1D).

Maximum Likelihood Figure 1

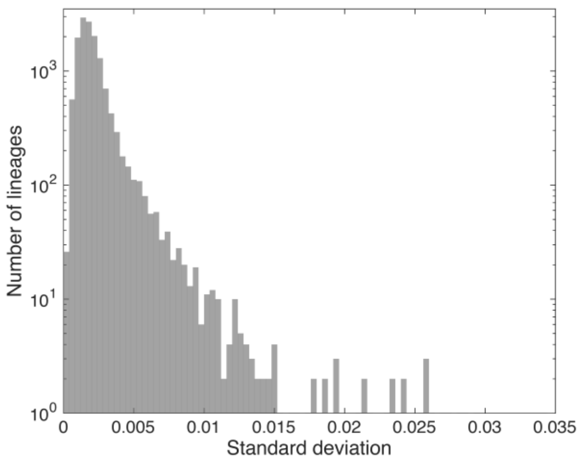
A



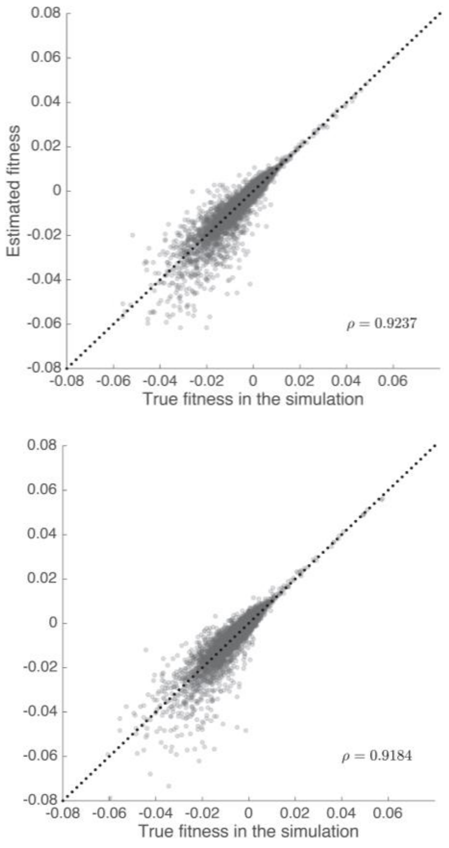
B



C



D



Maximum Likelihood Figure 1

A| Convergence of Poisson likelihood function. A plot of the value of the Poisson likelihood function at each iteration for all six growth replicates.

B| Distribution of relative error. The distribution of the relative error for each replicate pooled growth using Poisson likelihood optimization method (pink) and the log-linear regression method (blue).

C| Histogram of standard deviations of estimated fitness across six replicates using the Poisson likelihood optimization method.

D| Poisson likelihood optimization method performance on simulated data. Scatter plots of the true and estimated fitnesses for evolutionary simulations with starting parameters that (up) match replicate E, or (down) match replicate F. ρ is the Pearson correlation coefficient.