Supplementary Information

Supplementary Information Guide

Supplementary Notes

Gigantic Parallel Reporter Assay (GPRA) experimental details

Expression measurements were performed as described in (de Boer *et al.*, 2020) (**Supplementary Fig. 1**). Briefly, a library of ~200,000,000 random 80 bp promoters was cloned in front of a YFP reporter construct within the -160:-80 region of a synthetic promoter scaffold. The promoter scaffold used throughput this study included a distal poly-T tract (5 or more Ts), and a proximal poly-A tract (5 or more As) surrounding the random 80 mers; these features are common in yeast promoters. Furthermore, the scaffold sequences were designed to exclude strong binding sites for TFs. The dual reporter plasmid used is available from AddGene (AddGene:127546) and was derived from the plasmid used by (Sharon *et al.*, 2012). This plasmid contains *URA3*, which we use as a selectable marker, a constitutive RFP (with which to control for extrinsic noise), and the YFP under variable control. Random 80 mers (and designed 80 mer libraries) were cloned into an XhoI site using Gibson assembly. The resulting libraries were transformed into *S. cerevisiae* strains lacking *URA3* using the lithium acetate method (De Boer, 2017), selecting on SD-Ura media, and ensuring that at least 100,000,000 transformants were achieved for the random high-complexity libraries and >100x coverage for designed libraries. Because this is a low copy number CEN plasmid that is segregated like a chromosome during cell division, if a yeast cell is transformed with two different promoters, subsequent cell divisions will ensure with a very high probability that the two plasmids end up in different descendant cells. For random libraries, the strain Y8205 was used, but later experiments including the designed libraries were performed in S288C::*ura3*, which is less auxotrophic. Accordingly, all cases except that in random test dataset (complex media), the models were trained on sequences assayed in one strain of yeast and tested on sequences assayed in another, likely leading to underestimation of the model's performance due to *bona fide* differences between the strains.

Yeast were grown continuously in SD-Ura over the course of two days, and kept in log phase for ~10 generations to allow for reporters to reach equilibrium prior to sorting, diluting the media by 1:4 three times during this period as necessary to keep cells in log phase (OD below 0.8). All cultures were grown in a shaker incubator, at 30°C and approximately 250 RPM. Yeast were harvested by centrifugation, washed once in ice-cold PBS, resuspended in ice-cold PBS, and kept on ice prior to and during sorting. Sorting was performed with a Moflo Astrios (Beckman Coulter) sorting in three sets of 6 bins (all equal width and adjacent) each over the course of ~8 hours, dividing the time equally for the three sets. Cells were sorted by the log ratio of RFP to YFP signal (using mCherry and GFP absorption/emission), which controls for extrinsic sources of variation that affect both reporters (*e.g.*, cell size, plasmid copy number). Once sorted, cells were kept on ice. Sorted samples were centrifuged to pellet sorted cells, the PBS/sheath fluid aspirated, leaving ~0.5 mL remaining, then the cells resuspended in 1 mL SD-Ura, transferred to a 50mL conical tube containing 9mL media, and the sorting tube washed once with SD-Ura, and transferred to the same conical tube. This produced 18 50 mL tubes each containing ~10 mL of SD-Ura and sorted yeast cells; one per sorting bin. These were allowed to grow for 2-3 days, until all samples reached saturation. Plasmids were isolated using Qiagen spin miniprep kits, as adapted for yeast according to the manufacturer's website (https://www.qiagen.com/ca/resources/resourcedetail?id=5b59b6b3-f11d-4215-b3f7-995a95875fc0&lang=en). Nextera adaptors and multiplexing indices were added by PCR, indexed samples were mixed in proportion to the number of cells sorted per bin, and the resulting libraries sequenced paired-end, 76 bp each, using an Illumina Nextseq 500 and 150 cycle kits so that complete coverage of the promoter could be achieved, including overlap in the center.

The sorting bins differed slightly each time FACS was performed for a promoter library. This resulted from the inability of the cell sorter (MoFlo Astrios) to accurately preserve bin configurations on different days and between calibrations. Consequently, the 18 bins were re-assigned for each experiment, and/or laser intensities were adjusted, such that the distribution of RFP:YFP ratios were correctly positioned within the bins. Sorting bins were defined to be uniform in width (expression range) and included the vast majority (>98%) of the distribution and the entirety of the high end of expression, but leaving out the bottom tail of expression. The bottom end of expression tended to be dominated by noise and outliers with abnormally low YFP:RFP ratios. However, the sensitivity at the low end of expression increased in our experiments over time, such that model predictions (in particular for the complex media model) were squished at the low end (e.g. **Extended Data Fig. 1, 3** lower left corners).

The paired reads representing both sides of the promoter sequence were aligned using the overlapping sequence in the middle, constrained to have 40 (+/-15) bp of overlap, and discarding any reads that failed to align well within these constraints. This was not required for the designed libraries. Promoters were aligned to themselves using Bowtie2 (Langmead *et al.*, 2009) to identify clusters of related sequences, merging these clusters and taking the sequence with the most reads as the "true" promoter sequence for each cluster. The designed library reads were aligned to the promoter sequences we ordered using Bowtie2, and only perfect matches were considered in further analysis. Mean expression level for each promoter (as in the processed files) was taken as the average of the bins, weighted by the number of times the promoter was observed in each bin. For the designed libraries (that included all the high-quality test data experiments), we calculated expression for all promoters for which any reads were seen, but used only those for which we saw at least 100 reads for the analyses described to reduce the amount of measurement error present in the data. For high-complexity random libraries, all promoters were used.

Biochemical models

The biochemical models were created and used as described previously (de Boer *et al.*, 2020). Code is available on GitHub (https://github.com/de-Boer-Lab/CRM2.0). Briefly, the models are trained using the "makeThermodynamicEnhancosomeModel.py" program within the https://github.com/de-Boer-Lab/CRM2.0/blob/master/usefulScripts/makeProgressiveBiochemicalModels.bat script (using the "110 -eb" parameters which describe the sequence length (110) and the expected binding TF model (-eb)). Training happens in 5 stages, with each subsequent stage restoring the parameters learned in the previous step before continuing training, and optimizing the noted new parameters as well as all others that were previously learned: (**1**) potentiation and activity parameters are learned, after having initialized the motifs to known motifs for each TF, and TF concentrations initialized to the min K_d possible with each motif (corresponding to 50% occupancy of a perfect binding site); (**2**) concentration parameters are optimized; (**3**) motif models are optimized; (**4**) TF binding/activity limits are introduced and optimized; and (**5**) position-specific activities are introduced and optimized.

Each training round is performed with a full epoch of the training data (5 epochs total). Inference is performed using the "predictThermodynamicEnhancosomeModel.py" program. In order to get regulatory strength for a TF, the "-dotf" parameter was used, and inference run again. This parameter sets the concentration parameter of the indicated TF to 0, and then predicts expression. An example for how to use these parameters and programs to calculate regulatory complexity is included here: https://github.com/de-Boer-Lab/CRM2.0/tree/master/usefulScripts. For all analyses, the biochemical models using position-specific activities were used with the exception of the biochemical model-derived ECC, where the non-positional model was used, because the position-specific activity parameters we had previously found are partly dependent on the surrounding sequence context (de Boer *et al.*, 2020). The decrease in % error (100% x (1-r^2)), that is, the fraction of variance unexplained relative to the biochemical model is around ~45% ((0.96^2 - 0.926^2) / (1- 0.926^2) (positional biochemical model) for the Native test data. The biochemical models were used in sections where we required model interpretability (**Fig. 2d**), but the deep learning models were used elsewhere, since the biochemical models are slower than the deep learning models to run inference on and have lower predictive performance on the test data.

ECC calculation details and considerations

The ECC depends on both simulated and natural variation in promoter sequences. The natural variation in promoters is not independently sampled, since promoters from closely related strains often have identical sequences. Consequently, even when there are 1,011 orthologous promoters for each gene in the 1,011 whole yeast genomes dataset, there will typically be many fewer unique promoter sequences. Meanwhile, each sequence in the simulated variation is sampled independently (to increase robustness of the estimation of the null expectation), so, here, there are often 1,011 unique promoter sequences. The simulated variation was generated by placing random mutations within the gene's promoter consensus (the most abundant base at each position in the orthologous set), while preserving the Hamming distance distribution observed in the natural sequences (**Methods**). Despite these sets each having the same Hamming distance distribution relative to the consensus, the standard deviation (SD) calculated from N independently sampled sequences (as in the simulation) is biased towards being greater than that for N dependently sampled sequences (as in evolution), resulting in the raw ECC values being biased in favor of "conservation" as a result of a statistical bias rather than due to selection.

To demonstrate that this bias is not evolutionary in nature we calculated a "mock" ECC where both the numerator and denominator represent simulated variation. In the mock ECC, the sequences in the numerator are sampled independently and match the Hamming distance distribution of the natural variation (as in the standard ECC), but the denominator (normally the natural variation) is sampled in a way that matches *both* the Hamming distance distribution *and* the number of unique sequences at each Hamming distance (relative to the natural variation). Despite both sets of sequences being randomly sampled and having matched Hamming distance distributions, the mock ECC is slightly positively biased (**Supplementary Fig. 2a**), highlighting the need for a correction factor. Consequently, we used the median of these mock ECCs $\left(\log_2(\frac{\sigma_{c_i}}{\sigma_{c'_i}})\right)$ as the correction factor.

While it is theoretically better to have gene-specific correction factors, *these* are much more computationally intensive to calculate and provide little benefit in practice. To generate gene-specific correction factors, we need to make many instances of simulated variation for each gene, estimate the gene-specific bias, and use it to correct the observed ECC. Doing this with 1,111 simulations for each gene showed that there was little difference in the resulting ECC values, compared to a global correction factor (**Supplementary Fig. 2b**). Given the computational intensity of this approach (which, after optimization, still takes several days to run) and low practical utility, we favored the approach with a global correction factor. We do provide the gene-specific corrected ECCs in **Supplementary Table 1**.

The substitution rate in the genome is not uniform, but we use a uniform substitution rate when calculating the ECC. To test for the impact of this choice, we re-calculated the ECC using the substitution rates observed in the 1,011 yeast genomes promoters and found that the ECCs were largely concordant (**Supplementary Fig. 2c**). Since the mutations we observe in promoters are themselves biased (having survived selection), both approaches yield similar ECC values, and it is much easier to use a uniform base substitution rate, we use the uniform substitution rate ECC throughout the study.

Finally, we note that our approach for computing the ECC assumes that the relative effects of mutations within a sequence are similar regardless of the surrounding sequence context.

Comparison of ECC to RNA-seq expression

We examined the robustness of our finding that ECC distributions differ significantly between genes with conserved and divergent expression (by RNA-seq) to the threshold we chose to define expression conservation. To this end, we performed the Wilcoxon rank sum test analysis across a range of thresholds for each dataset. Both the *Saccharomyces* and Ascomycota results were significant ($P<0.05$) at all thresholds, and much more significant ($p<10^{-5}$) at a threshold of 10% and above (**Supplementary Fig. 3a-c**).

For mammals, we used the threshold of 25% applied in the original publication (Chen *et al.*, 2019). In addition, we performed the Wilcoxon rank sum test analysis across a range of thresholds and found that the results were similarly significant for the full range of thresholds bar one (5%, the lowest threshold; **Supplementary Fig. 3d**). The null hypothesis could not be rejected at the 5% threshold, given the smaller number of yeast gene one-to-one orthologs in mammals in both the expression conservation classes.

In principle, the ECC can be calculated across orthologous regulatory sequences from many different species (as opposed to individuals within a species, as we did here), but we advise caution if doing so. The ECC assumes that the function relating sequence to gene expression is the same across the orthologous sequences being compared. Since regulatory sequences evolve much faster than the regulators themselves(Weirauch and Hughes, 2010), this assumption is likely a reasonable approximation within a species, but as evolutionary distances increase, regulators will diverge, gradually eroding this assumption. An alternative is to use gene orthology to infer the extent of expression conservation in one species using ECCs calculated in another species (**Extended Data Fig. 4b**). However, such relations would extend only to well-mapped orthologs.

Benchmarking of sequence-to-expression models

We examined different neural network architectures for their ability to predict expression when trained on our data. We compared our transformer model to three model architectures from the literature: DeepAtt (Li *et al.*, 2020), DeepSEA (Zhou and Troyanskaya, 2015), and DanQ (Quang and Xie, 2016). (We focus here on comparison to the transformer model, as the convolutional model was not used for some of the compared tasks, such as calculation of the ECC. However, equivalent comparisons can be made with the convolutional using the code shared) Although these models differ from our own and from each other, we adopted each of the model architectures for our application to the best of our ability using the source code (https://github.com/jiawei6636/Bioinfor-DeepATT) from each original publication (the adopted model architecture implementation can be found on our GitHub repo at: https://github.com/1edv/evolution/tree/master/manuscript_code/model/benchmarking_models) for the purpose of this benchmarking analysis. The precise details of the benchmarking architectures can be found in the code, and are described below. Note, that the input and output layers (which are the same for each model) are omitted from the lists below.

1) DeepATT :
   - Convolution (filters=256, kernel_size=30)
   - MaxPool (pool_size = 3, strides = 3)
   - Dropout (0.2 probability)
   - BiDirectional LSTM (16 units)
   - MultiHeadAttention
   - Dropout (0.2 probability)
   - Dense (16 units)
   - Dense (16 units)

2) DeepSEA :
   - Convolution (filters=320, kernel_size=8)
   - MaxPool (pool_size = 3, strides = 3)
   - Dropout (0.2 probability)
   - Convolution (filters=480, kernel_size=8)
   - MaxPool (pool_size = 3, strides = 3)
   - Dropout (0.5 probability)
   - Dense (64 units)
   - Dense (64 units)

3) DanQ :
   - Convolution (filters=320, kernel_size=26)
   - MaxPool (pool_size = 3, strides = 3)
   - Dropout (0.2 probability)
   - BiDirectional LSTM (320 units)
   - Dropout (0.5 probability)
   - Dense (64 units)
   - Dense (64 units)

Next, we trained each of these adapted models using the same training data (in complex media) as the original convolutional and transformer model, and tested each of the model's predictive power on a set of high-quality native DNA sequences measured in our system. We found that our transformer model outperformed the other three architectures on these data (**Supplementary Fig. 4a**), as expected given that these other approaches were designed for other purposes.

We also used each of these models to calculate the ECC, finding that the resulting ECC values are highly correlated to the ECCs predicted by the transformer model (**Supplementary Fig. 4b-d**). This shows that our framework leads to equivalent biological conclusions when used with model architectures that have overall comparable predictive performance.

To rule out the possibility that the transformer model's increased performance results from learning of technical biases, we compared the transformer model's ECC to an ECC calculated using the interpretable biochemical model (de Boer *et al.*, 2020), also trained using GPRA data, which, with a single convolutional layer and many fewer parameters, is presumably less able to capture technical biases. Here too, we found that the ECCs are highly similar between the two models (**Supplementary Fig. 5g**). Finally, we found that the ECC values computed using the transformer model are better at predicting expression conservation as measured by RNA-seq across the range of possible thresholds considered (**Supplementary Fig. 5h**).

Ablation analysis of the sequence-to-expression transformer model

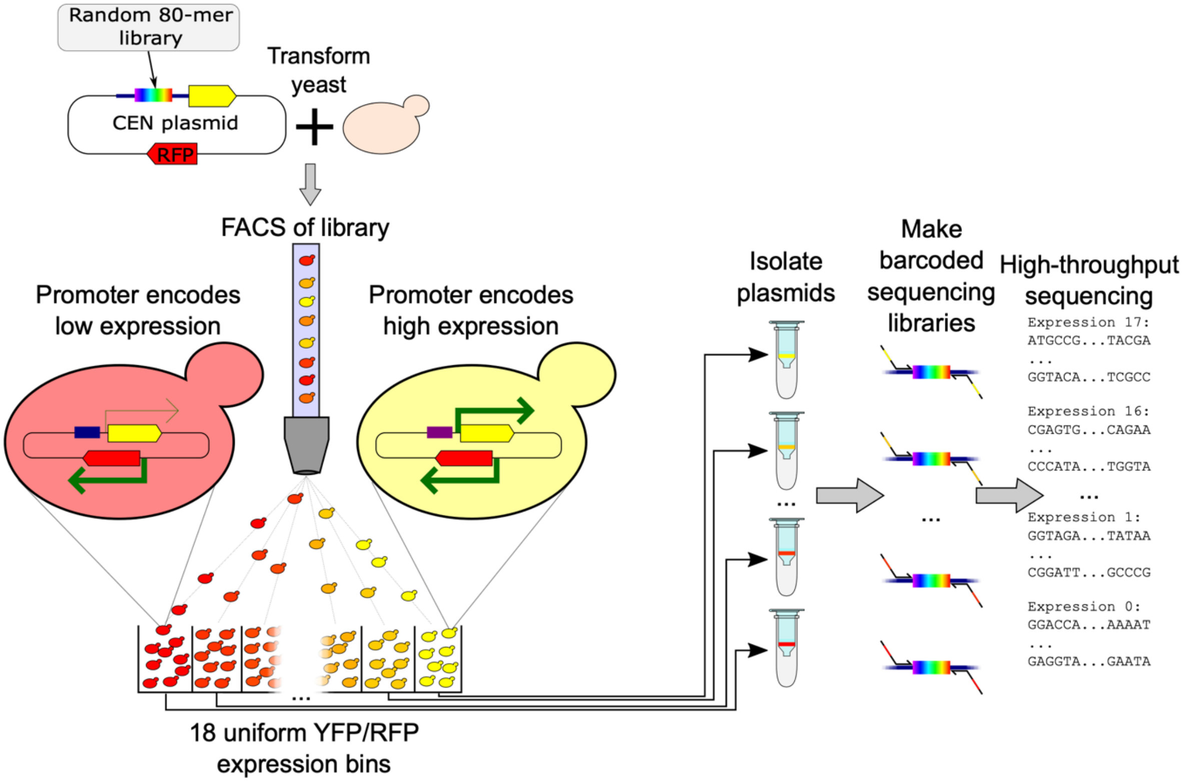
The transformer model was motivated by several intuitions aimed to help it leverage known aspects of *cis*-regulation(Weirauch *et al.*, 2013; Brodsky *et al.*, 2020), but which may or may not be explicitly captured. The first convolutional block with three layers, was motivated by the idea to identify sites that are important for computing the expression target, and could be analogous to a TF scanning the length of the sequence for binding sites. The first layer was aimed towards an abstract representation of first order TF-sequence interactions by operating with convolutional kernels on the sequence in the forward and reverse strands separately to generate strand-specific features (each individual kernel in the first layer can be thought of as possibly learning the motif of one TF, or a combined representation of the motifs)(Alipanahi *et al.*, 2015; Zhou and Troyanskaya, 2015; Shrikumar, Greenside and Kundaje, 2017; Quang and Xie, 2019) and we designed the width of the first convolutional layer (30 bp) to be sufficient to capture the largest TF motifs known in yeast(de Boer and Hughes, 2012); the second was aimed towards capturing interactions between strands, by using a 2D convolution (implemented using the *tf.keras.layers.Conv2D* layer, and convolving along the sequence dimension) on the combined features from the individual strands; and the third layer was aimed towards capturing higher order interactions, such as TF-TF cooperativity. We zero-pad the convolution blocks to allow the convolutional filters to detect motif instances near the edges of the input sequence. The second block was motivated by an analogy to combining the biochemical activities of multiple bound TFs and accounting for their positional activities. Its transformer-encoder with a multi-head self-attention module(Vaswani *et al.*, 2017) could capture relations between features extracted by the convolutional block at different positions in the sequence, by attending to them simultaneously using a scaled dot product attention function. This could be analogous to the model learning ‘where to look’ within the sequence. Then, a bidirectional Long Short-Term Memory (LSTM) layer in this block was motivated by the idea of capturing long range interactions between the sequence regions. Finally, a multi-layer perceptron block was motivated by the idea of capturing cellular operations that occur after TFs are recruited to the promoter sequence, by pooling all the features extracted from the sequence through the previous layers and learning a scaling function that transforms these abstract feature representations of biomolecular interactions into an expression estimate. While these were our motivations in architecting the model, because our focus was predictive ability and not interpretability of regulatory mechanisms, we do not know if the model in fact captured these relations in this way.

In order to determine whether any of the transformer model’s layers were superfluous, we conducted an ablation study. For each ablation experiment, we initialized a new model from scratch after removing the ablated layer individually from the original transformer model architecture, while retaining every other component of the original transformer model. Then, we trained this new model using the same training data (in complex media) as the original transformer model, and tested the resulting models on the high-quality random DNA test data. We found that each layer has non-trivial individual contributions to our predictions, with the full model performing better than any of the ablated models (**Supplementary Fig. 6**).

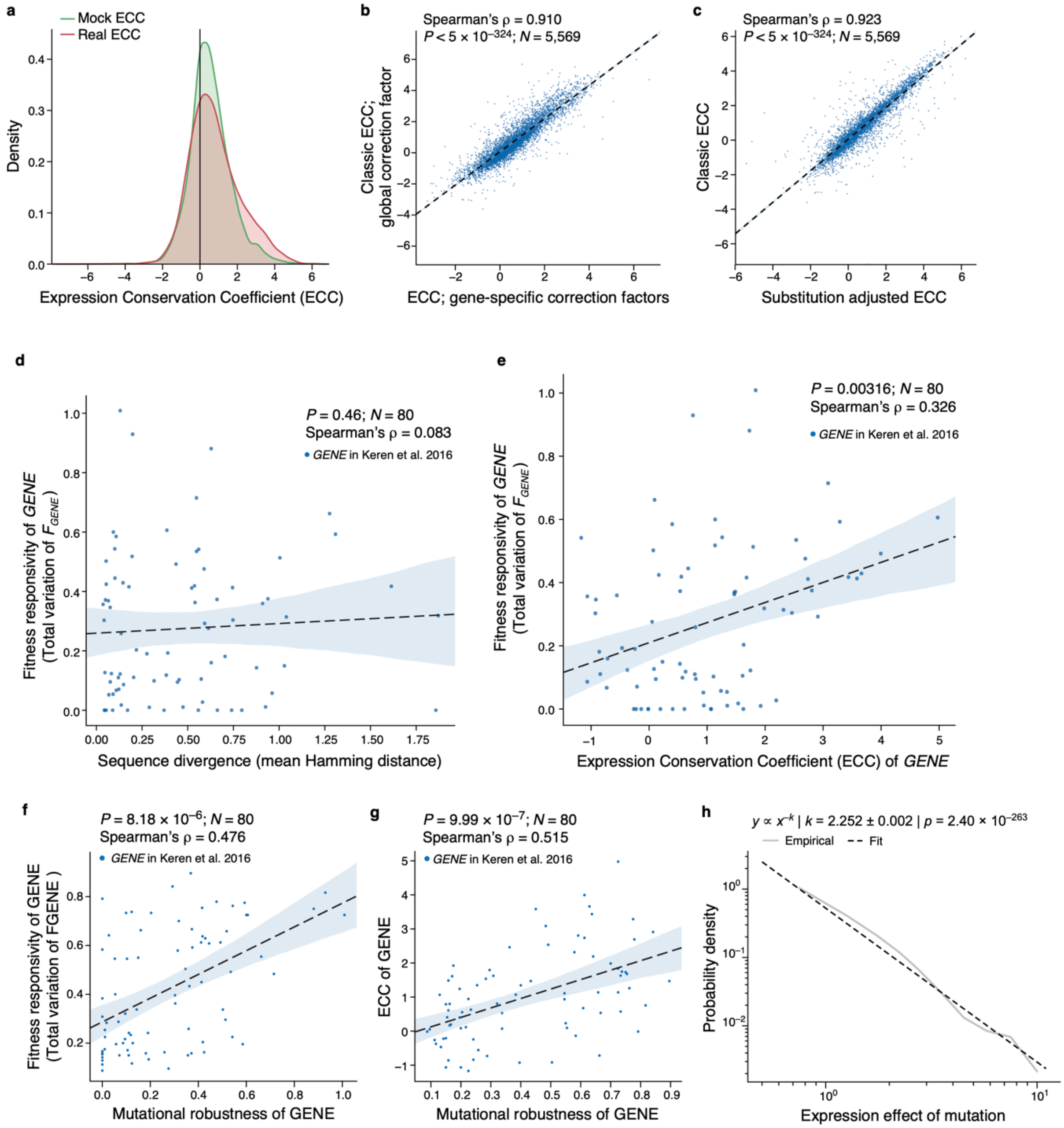
Expression distribution at the robustness cleft and the malleable archetype

While our observation that sequences with intermediate expression levels are move likely to be near the malleable archetype ($A_{\text{malleable}}$) and depleted near the robustness cleft (**Fig. 4d**), could in theory result from a saturation artifact of our reporter construct, our ratiometric sorting strategy allowed us to detect saturation and none was observed. Instead, the robustness cleft could reflect sequences at the stable extremes of one or more activation steps of gene expression (e.g. near 100% or 0% nucleosome occupied), while the malleable archetype could reflect instability around the inflection points.

Supplementary Figures

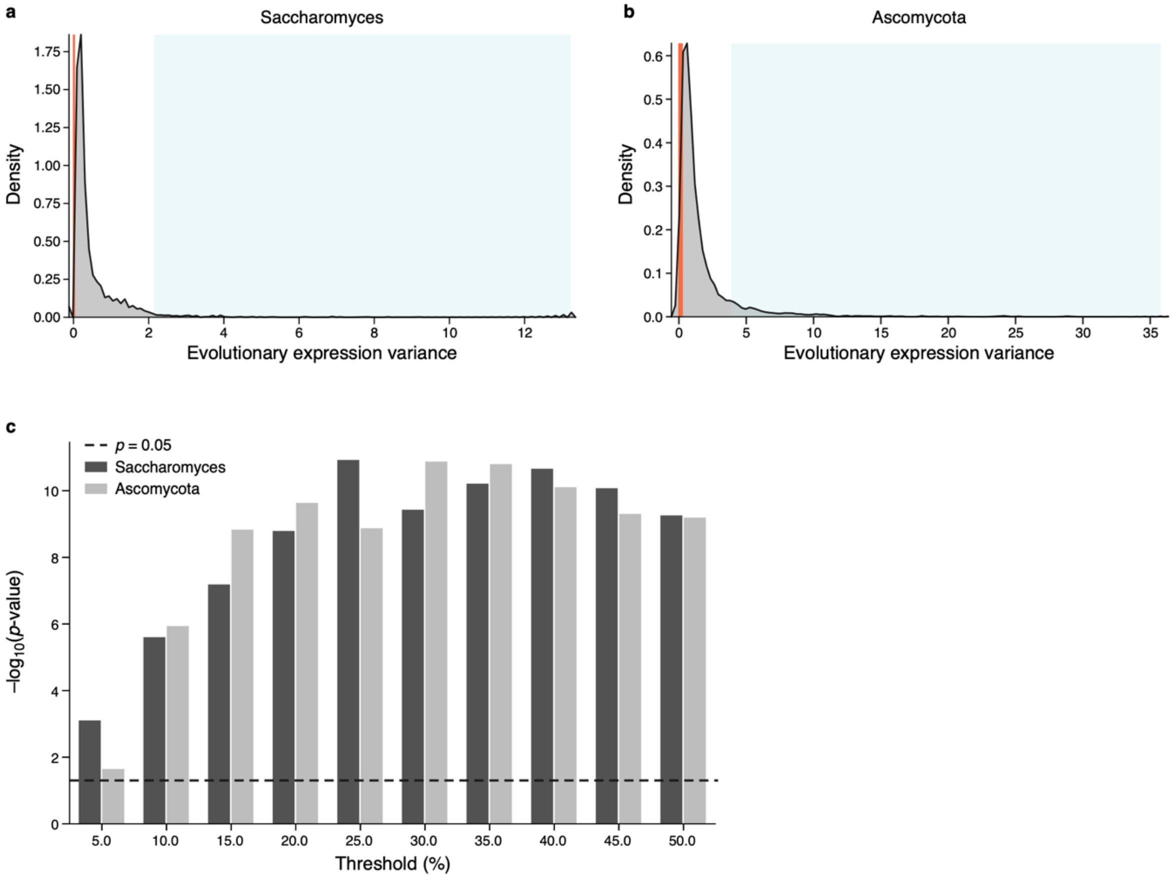


Supplementary Fig. 1 | GPRA experiment overview. Yeast are transformed with a library of random 80 bp sequences driving YFP expression, the cells recovered and selected for successful transformants, and grown in the target media in log phase. Yeast are then sorted by the ratio of YFP to RFP into 18 different uniform expression bins. Yeast are then recovered in selection media (SD-Ura), plasmids isolated, sequencing libraries created, and the promoters in each expression bin sequenced with high-throughput sequencing.

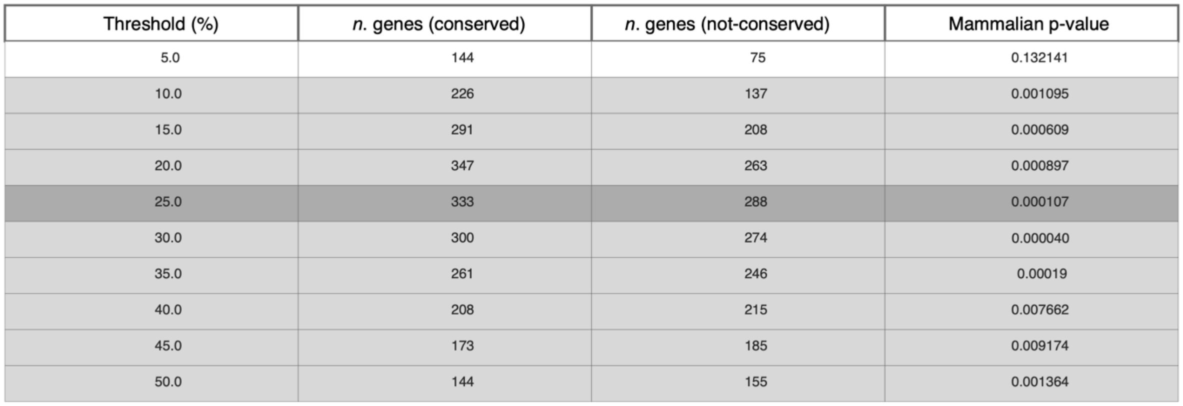


Supplementary Fig. 2 | a, Comparison of raw ECC distributions for natural variation (red) and matched simulated variation (green, "mock ECC"). Both are biased towards having an ECC above 0. **b,** Comparison of ECCs with global correction (x axis) and gene-specific correction factors (y axis). **c,** ECC with uniform substitutions (*y* axis) is highly correlated to the ECC computed using the observed substitution rate (*x* axis). **d, e,** Fitness responsivity is not associated with simple sequence diversity, but is associated with ECC. Fitness responsivity (*y* axes) and mean Hamming distance (**d,** *x* axis) or ECC (**e,** *x* axis) for each of 80 genes (points). **f, g,** Genes whose expression changes have stronger effects on organismal fitness have mutationally robust regulatory sequences. Mutational robustness (*x* axes) and fitness responsivity (**f**, *y* axis) or ECC (**g**; *y* axis) for each of 80 genes (points) for which the expression-to-fitness curves were quantified (Keren *et al.*, 2016). (**b-g**) Spearman's ρ and associated two-tailed p-values are shown. The light blue error bands represent the respective 95% confidence intervals. **h,** Mutational effects follow a power law distribution.

Probability density (*y* axis) and expression effect of mutation (magnitude) (*x* axis) plotted on log-log axes (solid line) alongside the goodness of fit (dash line) of the power law distribution.

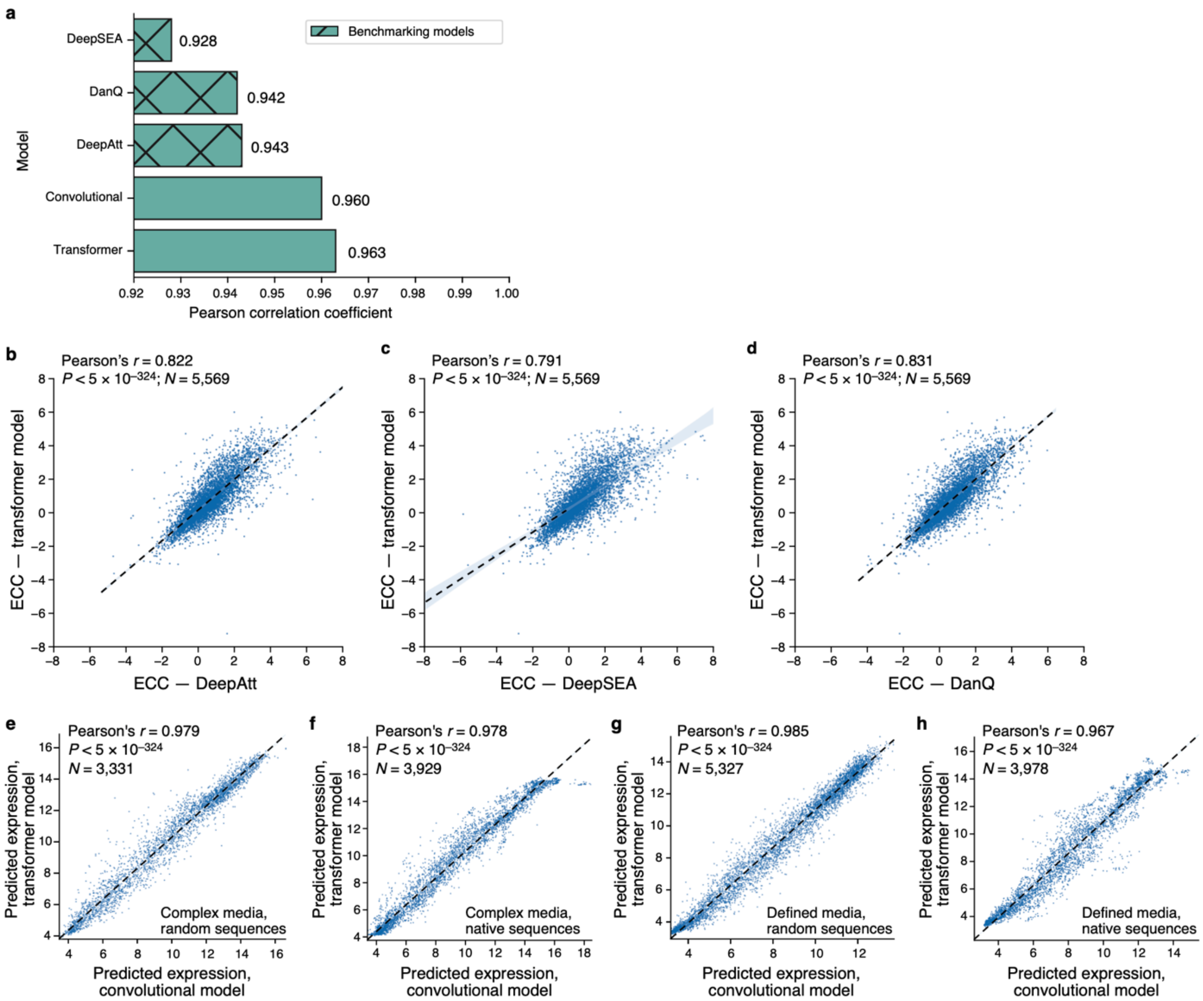


d

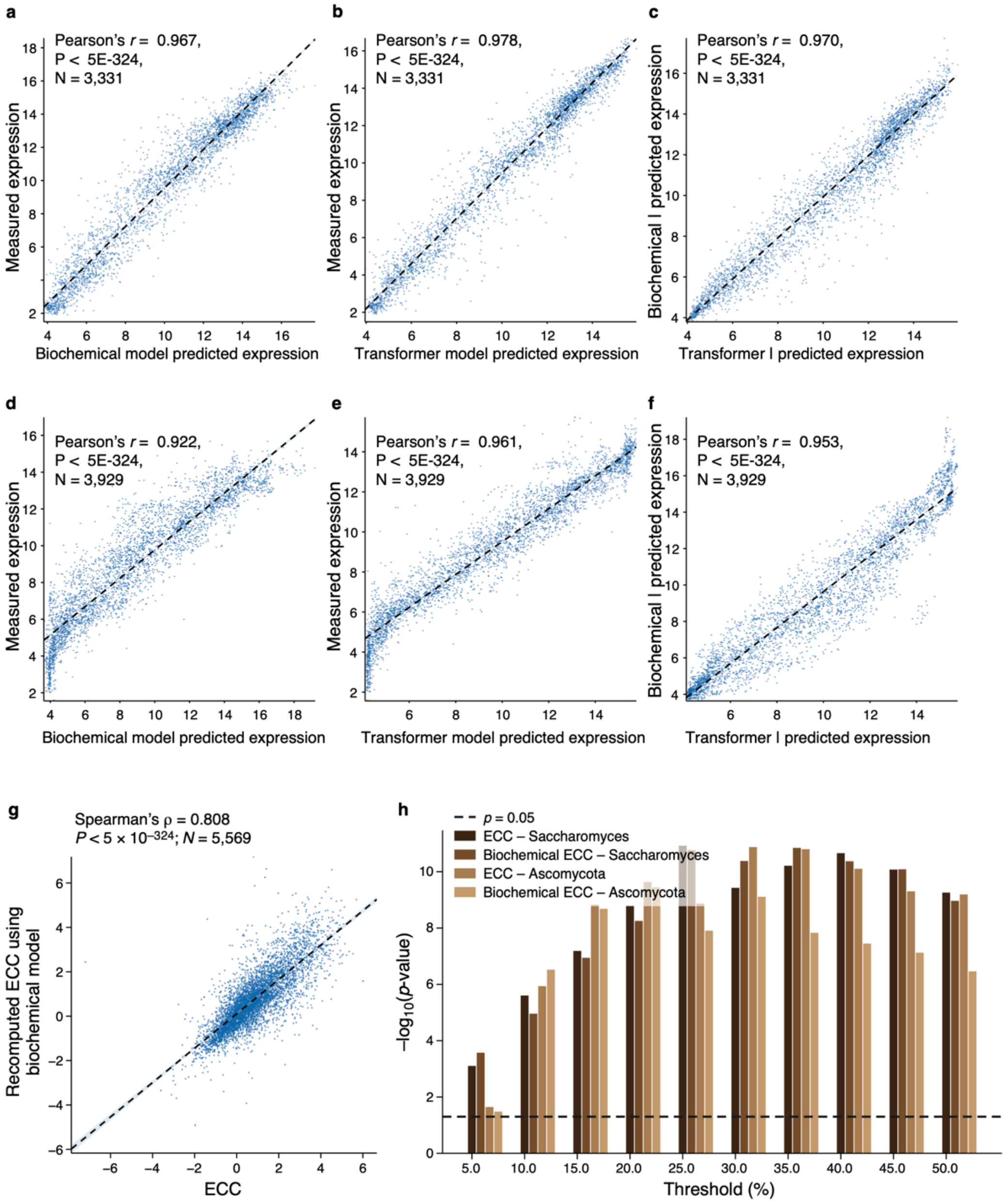


Threshold (%)	*n.* genes (conserved)	*n.* genes (not-conserved)	Mammalian p-value
5.0	144	75	0.132141
10.0	226	137	0.001095
15.0	291	208	0.000609
20.0	347	263	0.000897
25.0	333	288	0.000107
30.0	300	274	0.000040
35.0	261	246	0.00019
40.0	208	215	0.007662
45.0	173	185	0.009174
50.0	144	155	0.001364

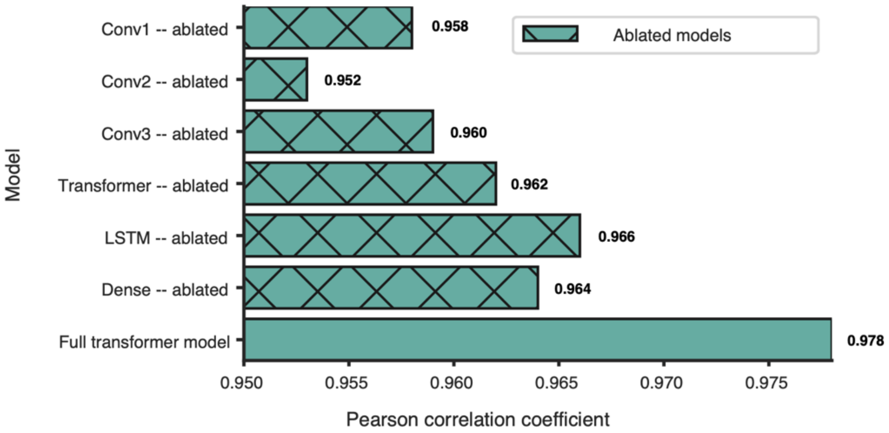
Supplementary Fig. 3 | a,b, Expression variance (by RNA-seq) for *Saccharomyces* (**a**) and Ascomycota (**b**). Green boxes: genes called as divergent; orange: genes called as conserved by the thresholds in this study (as in **Extended Data Fig. 4b**). **c,** Sensitivity of ECC enrichment significance (Wilcoxon rank sum test -$\log_{10}$(P-values); *y* axis) to "conserved" *vs*. "divergent" thresholds (*x* axis) for Ascomycota (light gray) and *Saccharomyces* (dark gray). P=0.05: dashed line. **d,** Sensitivity of ECC enrichment significance (Wilcoxon rank sum test, “Mammalian p-value”) to “conserved” *vs*. “divergent” thresholds (“Threshold (%)”) in mammals. The columns display the number of genes determined to be in each class at each threshold.



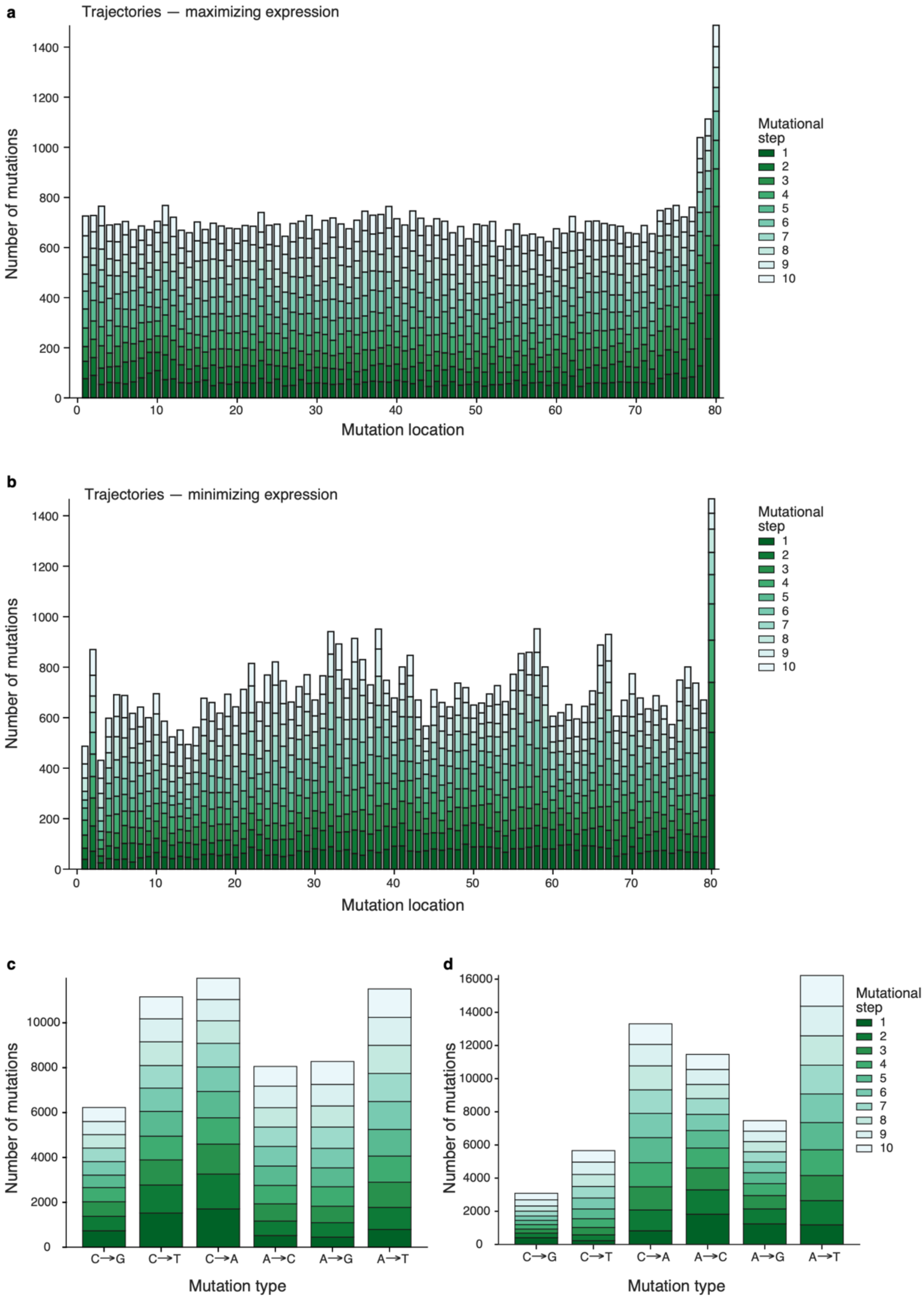
Supplementary Fig. 4 | a, Benchmarking of performance against existing neural network architectures. Pearson correlation coefficient between model predictions and test data (*x* axis) for four model (*y* axis). All models were trained on the same training dataset, and tested on the same set of native promoter test sequences in complex media. While all approaches performed reasonably well, the transformer model architecture used in this paper out-performed the others on the native test sequence dataset. **b-d,** Comparison of ECC calculated with our model (*y* axis) and with (**b**) DeepAtt, (**c**) DeepSEA and (**d**) DanQ (*x* axis). In each case, the ECC predictions are highly correlated between each approach and our model. (Outliers not shown for the panel (**c**) to maintain scaling and visibility; Pearson's r was computed using all of the data including outliers.). **e-h,** The convolutional and transformer models have highly correlated predictions. Predicted expression from the convolutional (*x* axis) and transformer (*y* axis) models in complex (**e-f**) and defined (**g-h**) media for random (**e-g**) and native (**f-h**) test datasets. (**b-h**) Pearson's r and associated two-tailed p-values are shown.



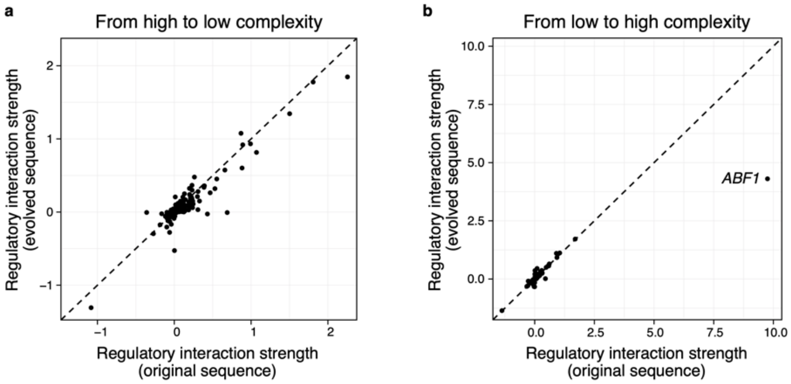
Supplementary Fig. 5 | Comparison of the biochemical and transformer models. Measured and predicted expression in complex media for **(a-c)** random test data as, and **(d-f)** native test data. (**a,b,d,e**) Measured (*y*-axes) and predicted (*x*-axes) expression, for (**a,d**) biochemical and (**b,e**) transformer models. (**c,f**) transformer (*x*-axes) and biochemical (*y*-axes) model predictions. (**a-f**) Pearson's r and associated two-tailed p-values are shown. **g,h,** The transformer model outperforms the biochemical model in differentiating expression conservation status. **g,** Comparison of ECCs calculated for each gene (points) for the transformer model (x axis) versus the biochemical model (y axis). Spearman's ρ and associated two-tailed p-values are shown. **h,** Significance (y axis) of rank sum statistics for how well ECCs calculated with each method separates conserved versus not conserved genes across *Saccharomyces* (dark brown) and Ascomycota (light brown).



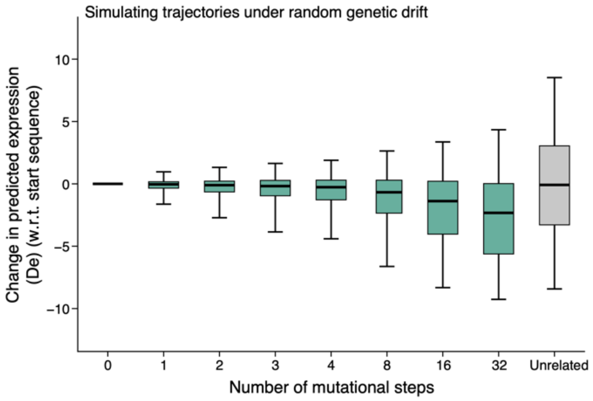
Supplementary Fig. 6 | Each layer individually contributes to model performance. Performance (*x* axis, Pearson's *r* between the model predictions and random test data) of the transformer model variants (*y* axis) with each layer individually ablated, and the full transformer model (bottom). The full transformer model outperforms all other versions with any model component ablated. The two-tailed p-value corresponding to each performance metric shown is $< 5*10^{-234}$.



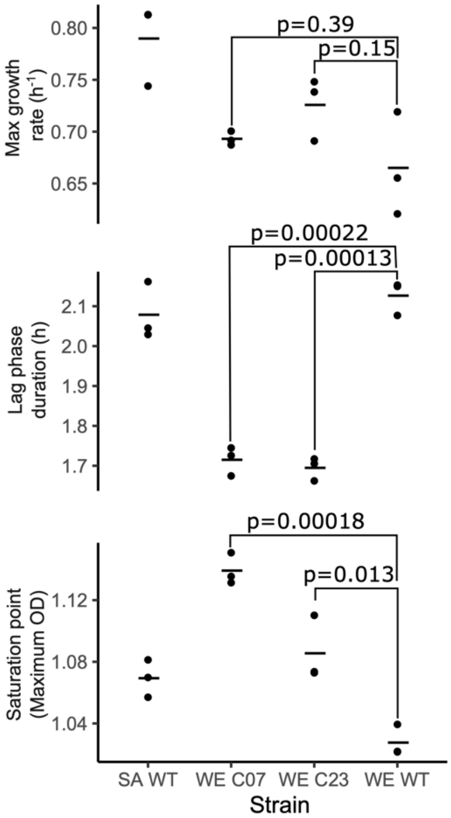
Supplementary Fig. 7 | Sequences took diverse paths to evolve extreme expression. a-b, The number (*y* axis) of mutations across each promoter position (*x* axis; -160 to -80 region) for trajectories under the SSWM regime starting with native promoter sequences when (**a**) maximizing or (**b**) minimizing expression in defined media using the convolutional model. Some of the observed bias to TSS-proximal mutations may be related to prior observations of proximal repressor activity bias(de Boer *et al.*, 2020). **c,d,** The number (*y* axis) of mutations of each type (*x* axis) for trajectories under the SSWM regime starting with native promoter sequences when (**c**) maximizing or (**d**) minimizing expression in defined media. Colors represent the mutational step (1-10).



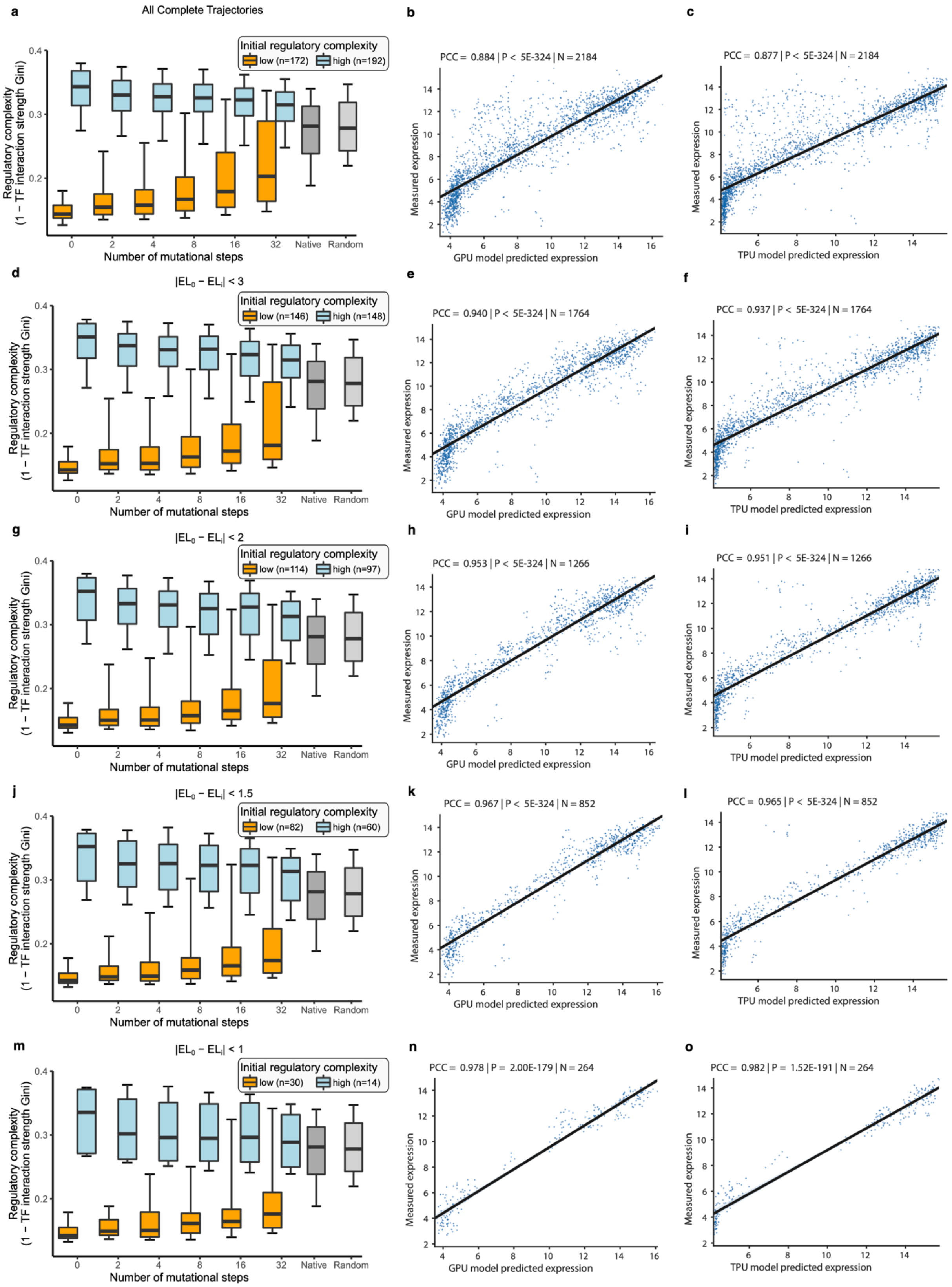
Supplementary Fig. 8 | a,b, Examples of regulatory complexity changes under stabilizing selection. TF regulatory interaction strengths for original (*x* axes) and evolved (*y* axes) sequences after 32 neutral (expression maintaining) mutations for each TF (points) for -160:-80 promoter regions for (**a**) *YDR476C*, whose regulatory complexity was high and decreased (from 0.3 to 0.25), and (**b**) *AIF1*, whose complexity was low (dominated by the TF Abf1p) and increased (from 0.14 to 0.21). Both have approximately the same predicted expression levels (13.7 and 14.3 respectively).



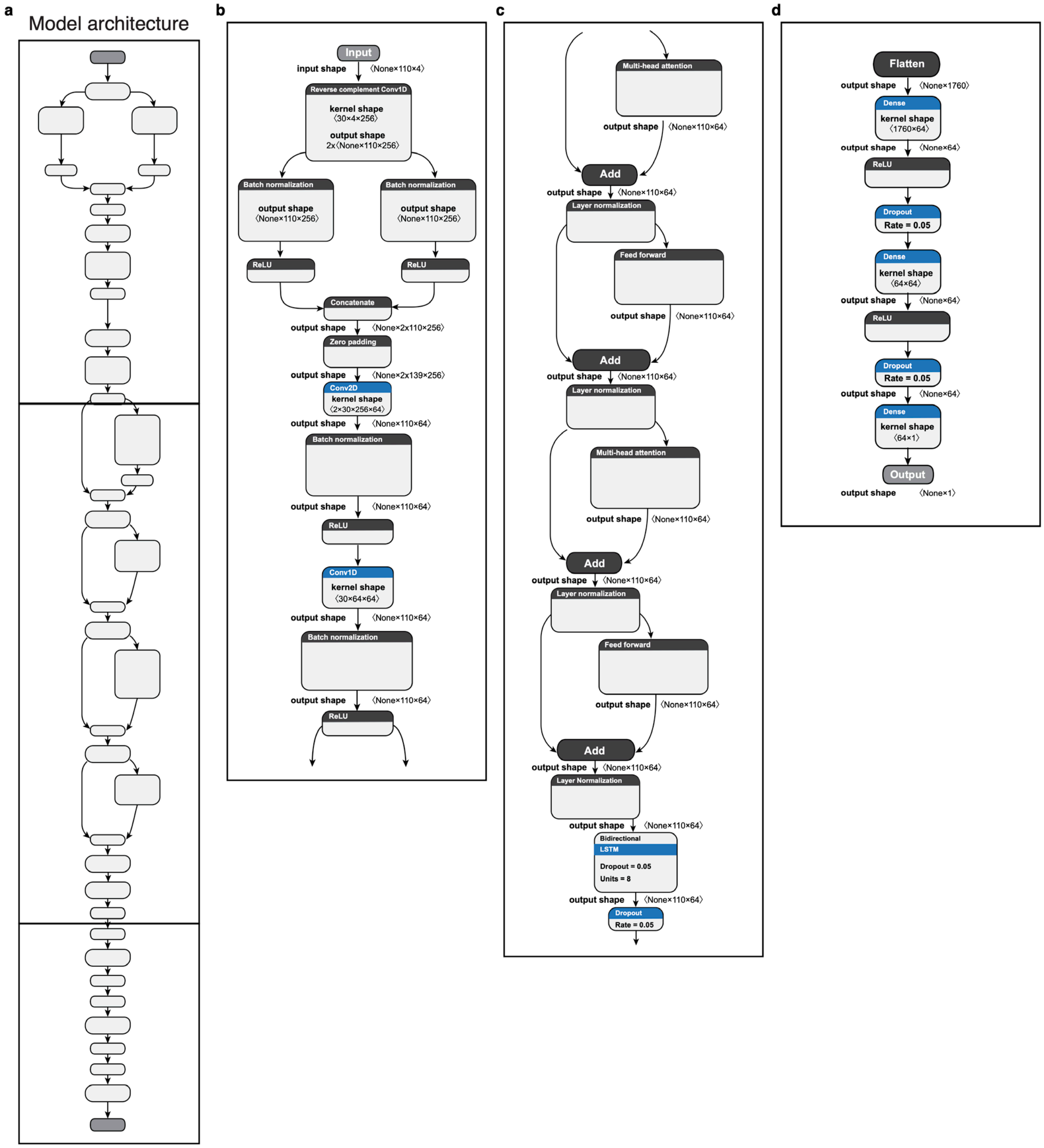
Supplementary Fig. 9 | Predicted expression divergence under random genetic drift. Distribution of the change in predicted expression (*y* axis) for native yeast promoter sequences (n=5,720) at each mutational step (*x* axis) for trajectories simulated under random mutational drift using the transformer model. Silver bar: differences in expression between unrelated sequences. Expression decreases with increasing mutation number because the average expression of the starting set of native sequences is greater than for random DNA, and so including random mutations are more likely to decrease expression than increase it. Midline: median; boxes: interquartile range; whiskers: 5th and 95th percentiles.



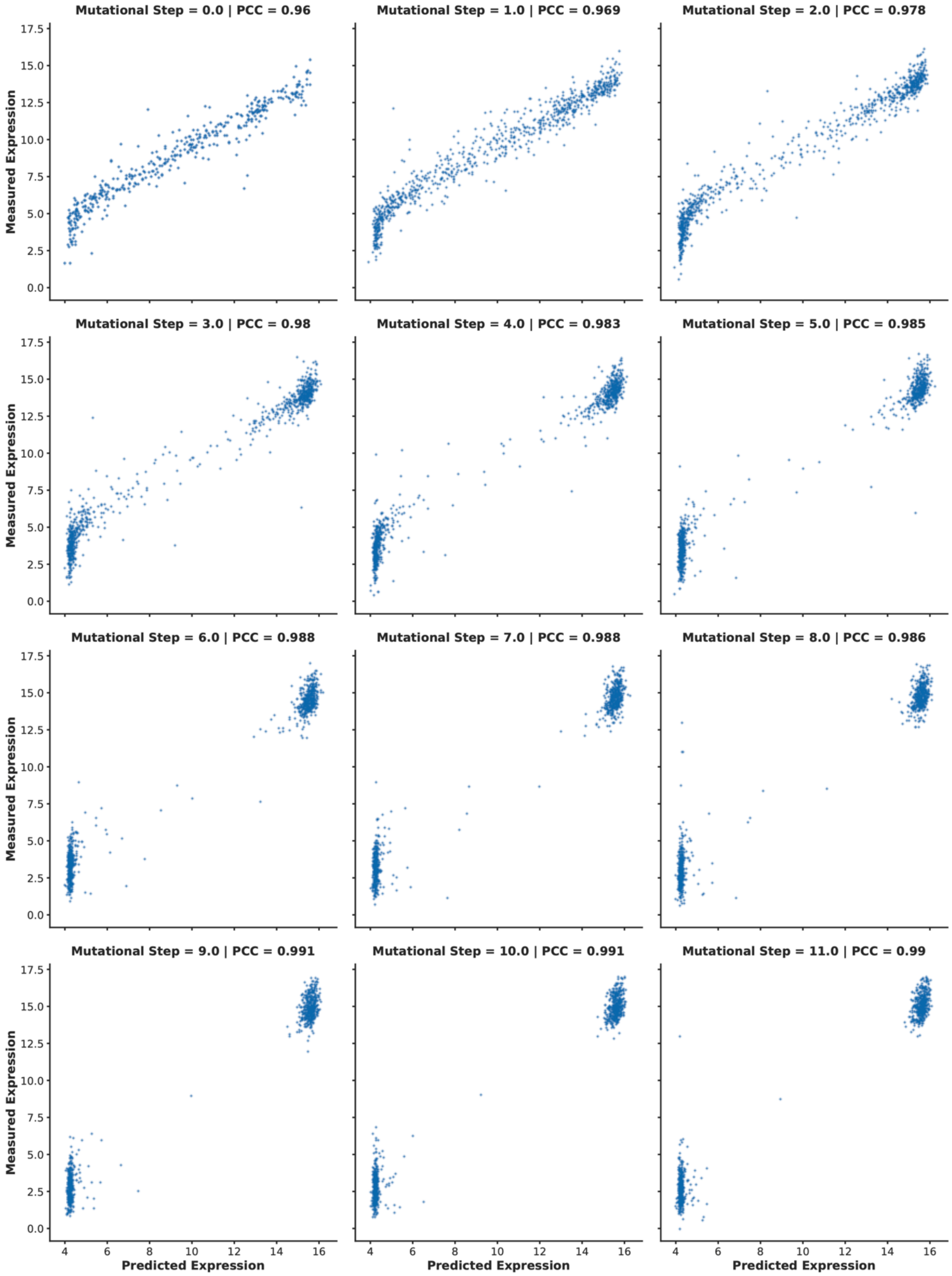
Supplementary Fig. 10 | Growth phenotypes of *CDC36* promoter mutant strains. Maximum growth (*y* axis, top), duration of lag phase (*y* axis, middle) and saturation of growth (*y* axis, bottom) for two WT strains and two engineered strains (*x* axis). Bars: means, dots: replicate measurements. P-values: Student's t-test; two-sided, unpaired, equal variance. *n*=3 replicates/strain.



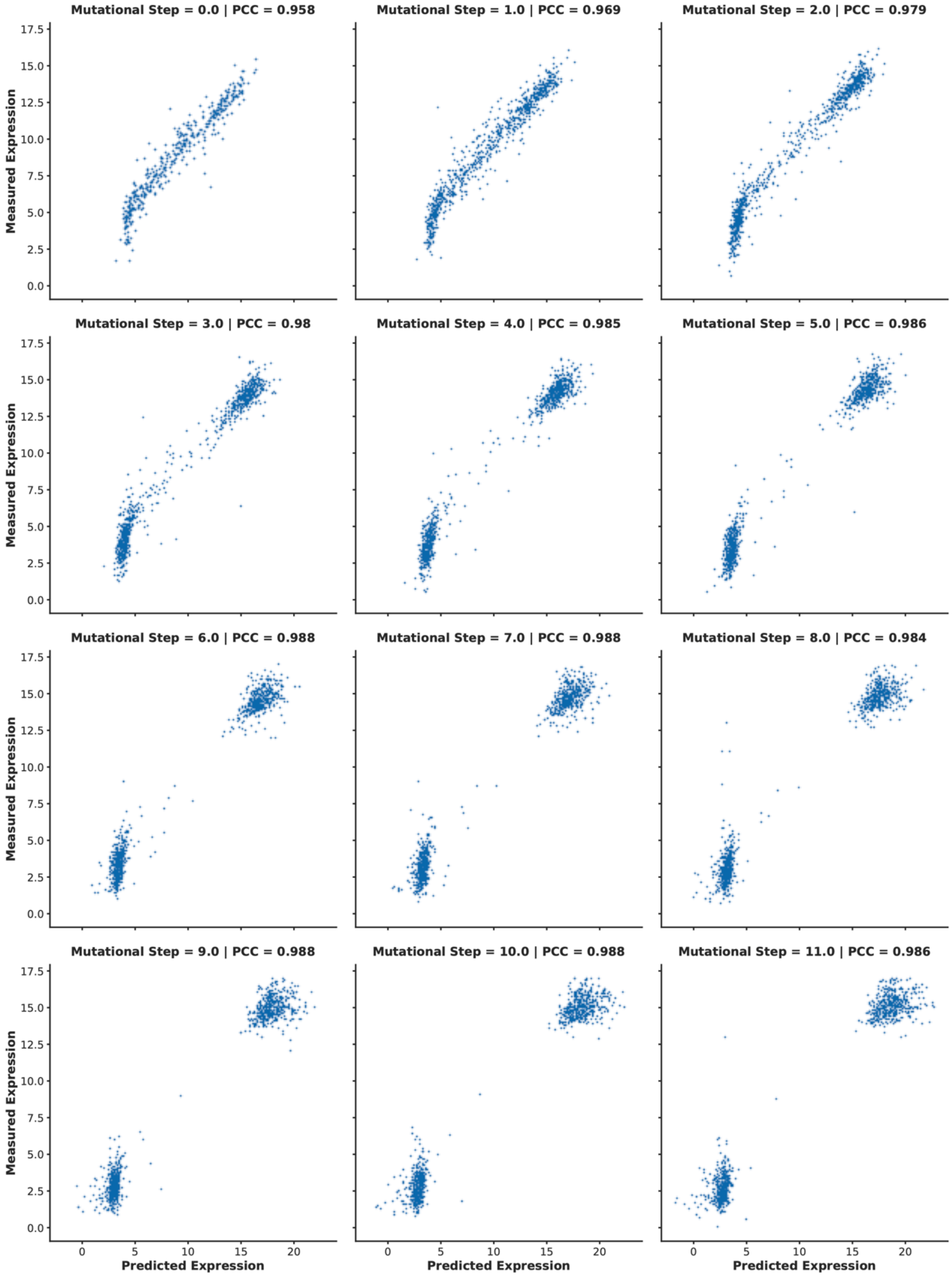
Supplementary Figure 11: Robustness of moderation of regulatory complexity to the degree of stabilizing selection. (**a,d,g,j,m**) Distributions of regulatory complexity (*y*-axes) for sets of sequences with initial high (light blue) and low (orange) regulatory complexity, and evolved sequences at different mutation steps, with native and random sequences shown for reference (dark and light gray respectively). Here, *n* is the number of trajectories included. All evolved sequences were designed to mimic stabilizing selection by requiring that expression changes by no more than 0.5 expression units relative to the original using the GPU model. Also shown are the measured (*y*-axes) and model predicted (*x*-axes) expression levels for the convolutional (**b,e,h,k,n**) and transformer (**c,f,i,l,o**) models. Results are shown for all complete experimental trajectories (**a-c**), or when including only trajectories where no evolved sequences had measured or transformer model-predicted expression that differed from the measured expression of the original sequence by more than 3 (**d-f**), 2 (**g-i**), 1.5 (**j-l**) or 1 (**m-o**) expression units. All data are for complex media (YPD). (**a,d,g,j,m**) Midline: median; boxes: interquartile range; whiskers: 5th and 95th percentile range. (**b,c,e,f,h,i,k,l,n,o**) Pearson's r and associated two-tailed p-values are shown.



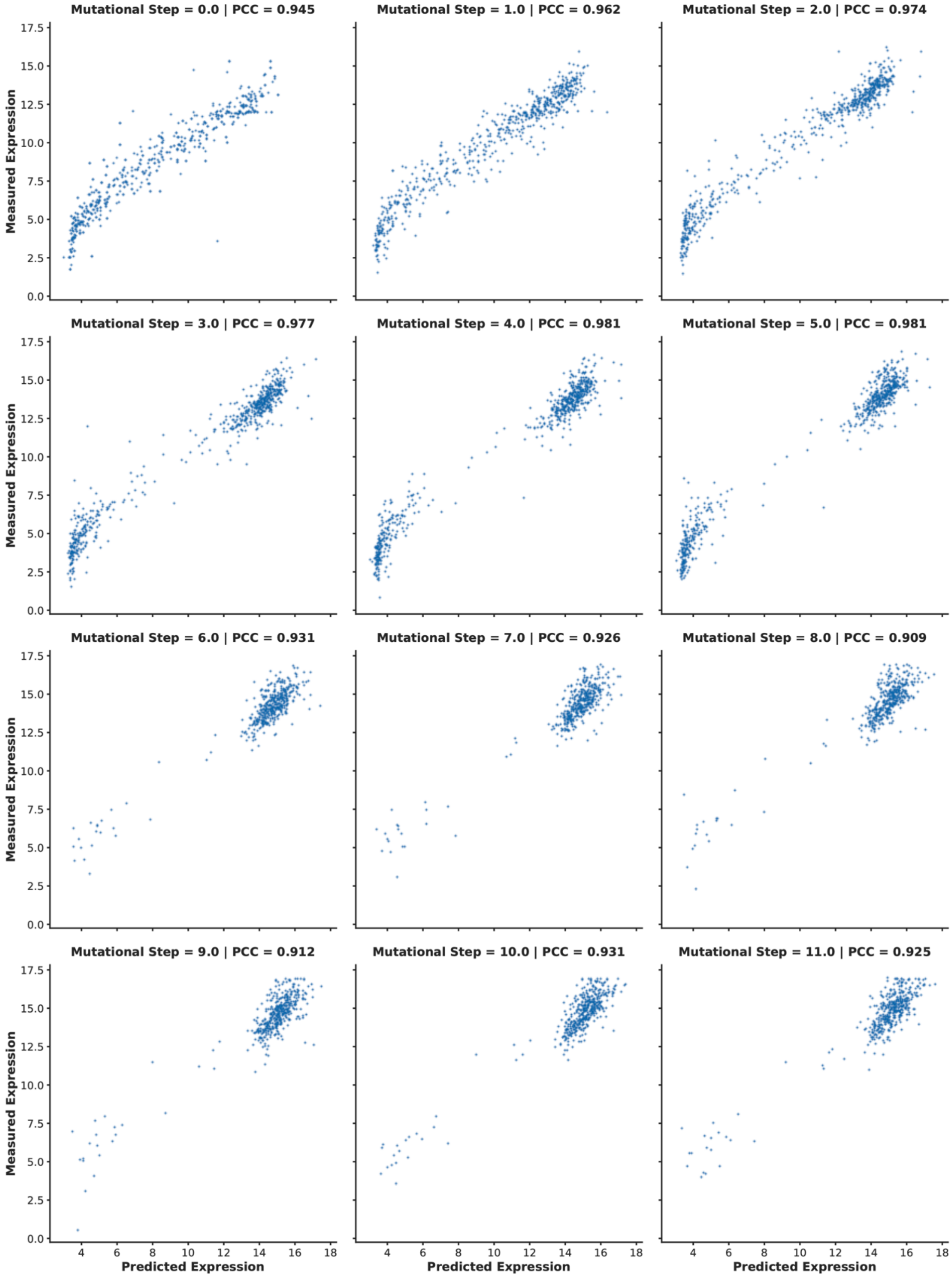
Supplementary Fig. 12 | The deep transformer neural network architecture for the sequence-to-expression model. a, Model architecture with three blocks (horizontal lines) and multiple layers (boxes). b-d. Expanded architecture (**Methods**) for the convolutional (**b**), transformer encoder (**c**) and multi-layer perceptron (**d**) blocks in our transformer model.



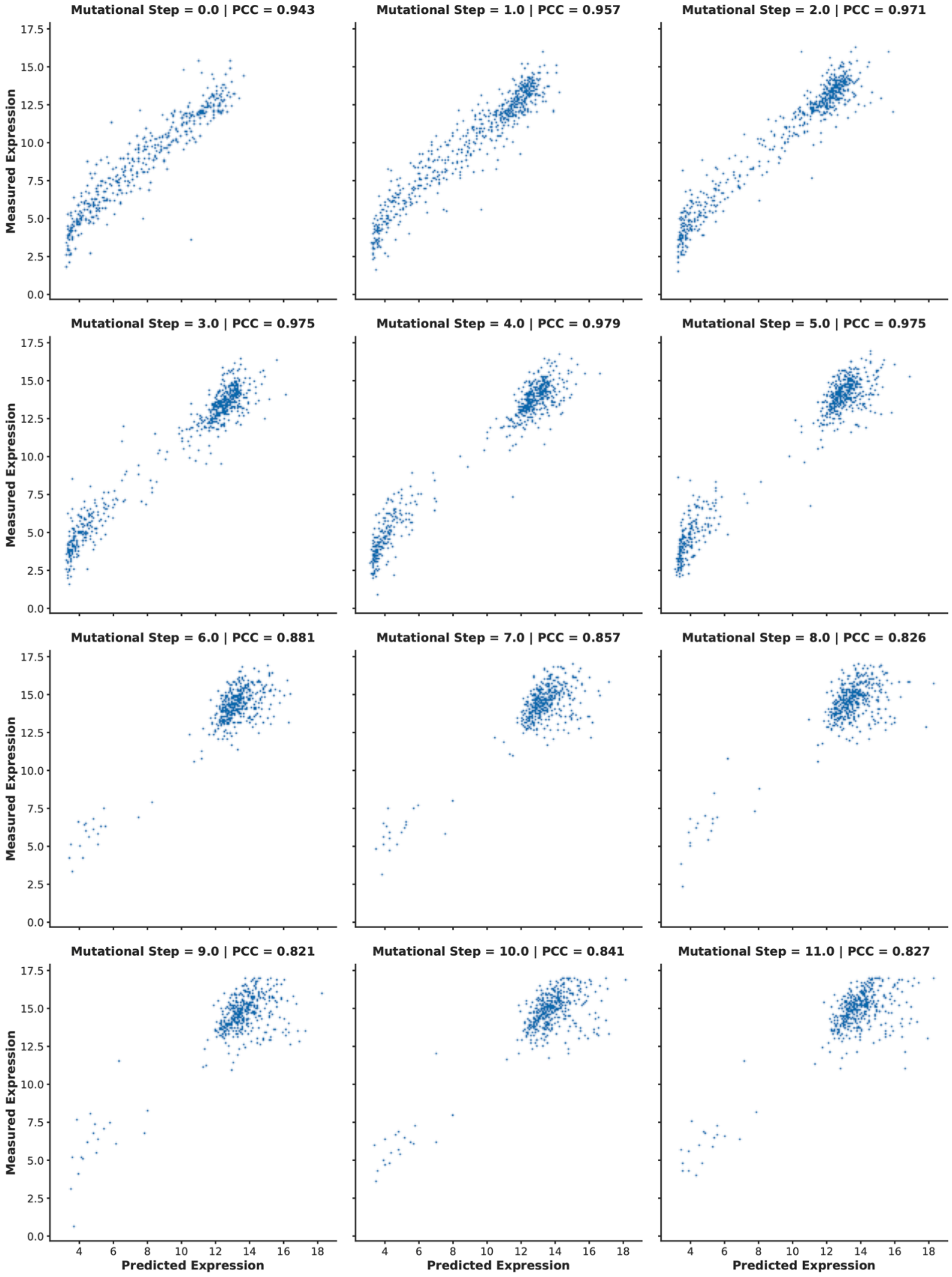
Supplementary Fig. 13 | Comparison between predictions and measurements of expression at each mutational step in complex media. Transformer model predicted (*x*-axes) and measured (*y*-axes) expression for each mutational step (plots) for trajectories in **Fig. 2g** (n=10,322 sequences, divided amongst plots). The Pearson's correlation coefficient (PCC) associated with each panel is also shown. The two-tailed p-value corresponding to the performance metric shown in each panel is < $5*10^{-234}$.



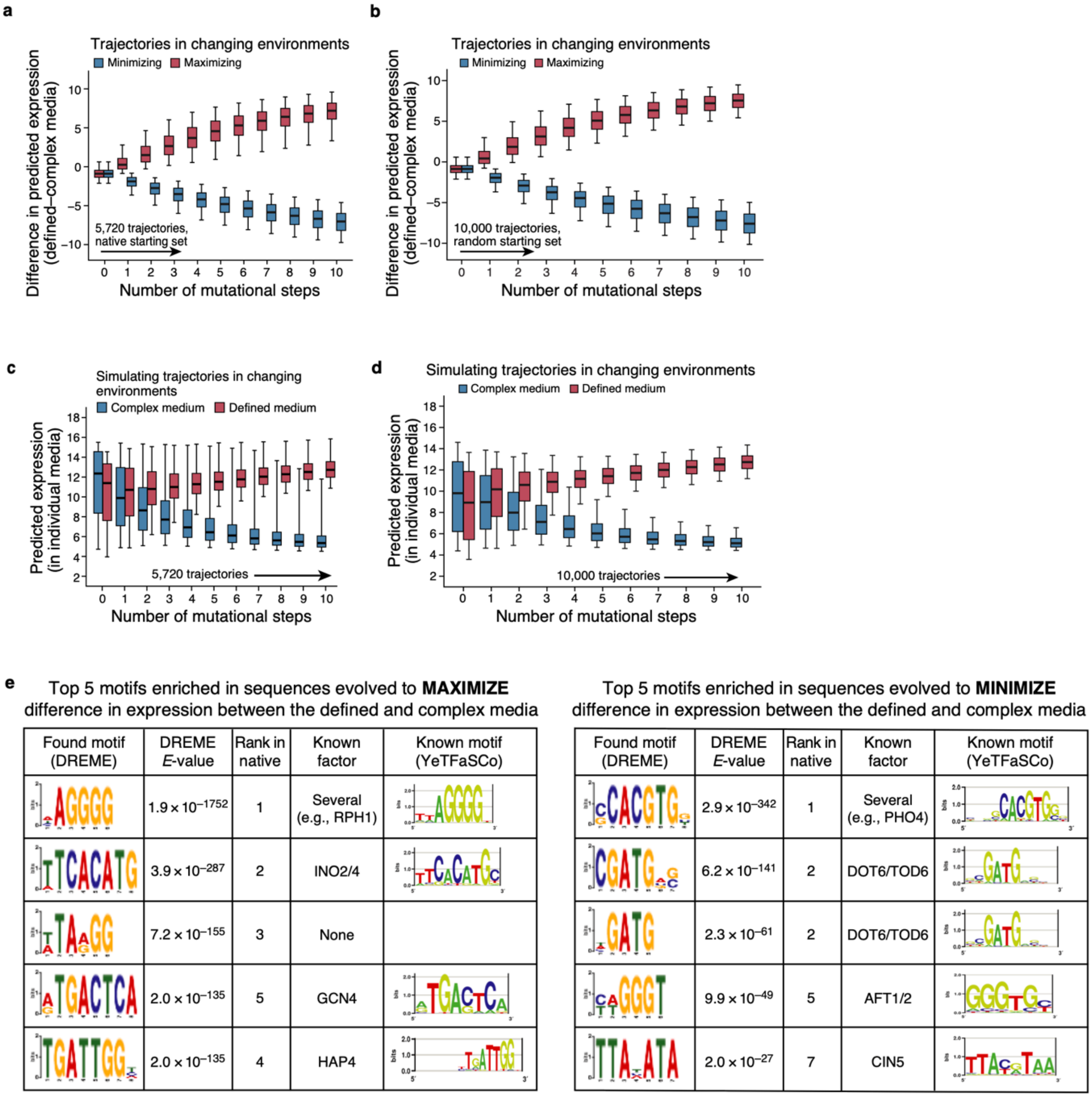
Supplementary Fig. 14 | Comparison between predictions and measurements of expression at each mutational step in complex media. Convolutional model predicted (*x*-axes) and measured (*y*-axes) expression for each mutational step (plots) for trajectories in **Fig. 2g** (n=10,322 sequences, divided amongst plots). The Pearson's correlation coefficient (PCC) associated with each panel is also shown. The two-tailed p-value corresponding to the performance metric shown in each panel is $< 5*10^{-234}$.



Supplementary Fig. 15 | Comparison between predictions and measurements of expression at each mutational step in defined media. Transformer model predicted (*x*-axes) and measured (*y*-axes) expression for each mutational step (plots) for trajectories in **Extended Data Fig. 1g** (n=6,304 sequences, divided amongst plots). The Pearson's correlation coefficient (PCC) associated with each panel is also shown. Due to limitations in the number of sequences we could test per experiment, we only tested the decreasing expression defined media trajectory to 5 mutations. The two-tailed p-value corresponding to the performance metric shown in each panel is < $5*10^{-234}$.



Supplementary Fig. 16 | Comparison between predictions and measurements of expression at each mutational step in defined media. Convolutional model predicted (*x*-axes) and measured (*y*-axes) expression for each mutational step (plots) for trajectories in **Extended Data Fig. 1g** (n=6,304 sequences, divided amongst plots). The Pearson's correlation coefficient (PCC) associated with each panel is also shown. Due to limitations in the number of sequences we could test per experiment, we only tested the decreasing expression defined media trajectory to 5 mutations. The two-tailed p-value corresponding to the performance metric shown in each panel is < $5*10^{-234}$.



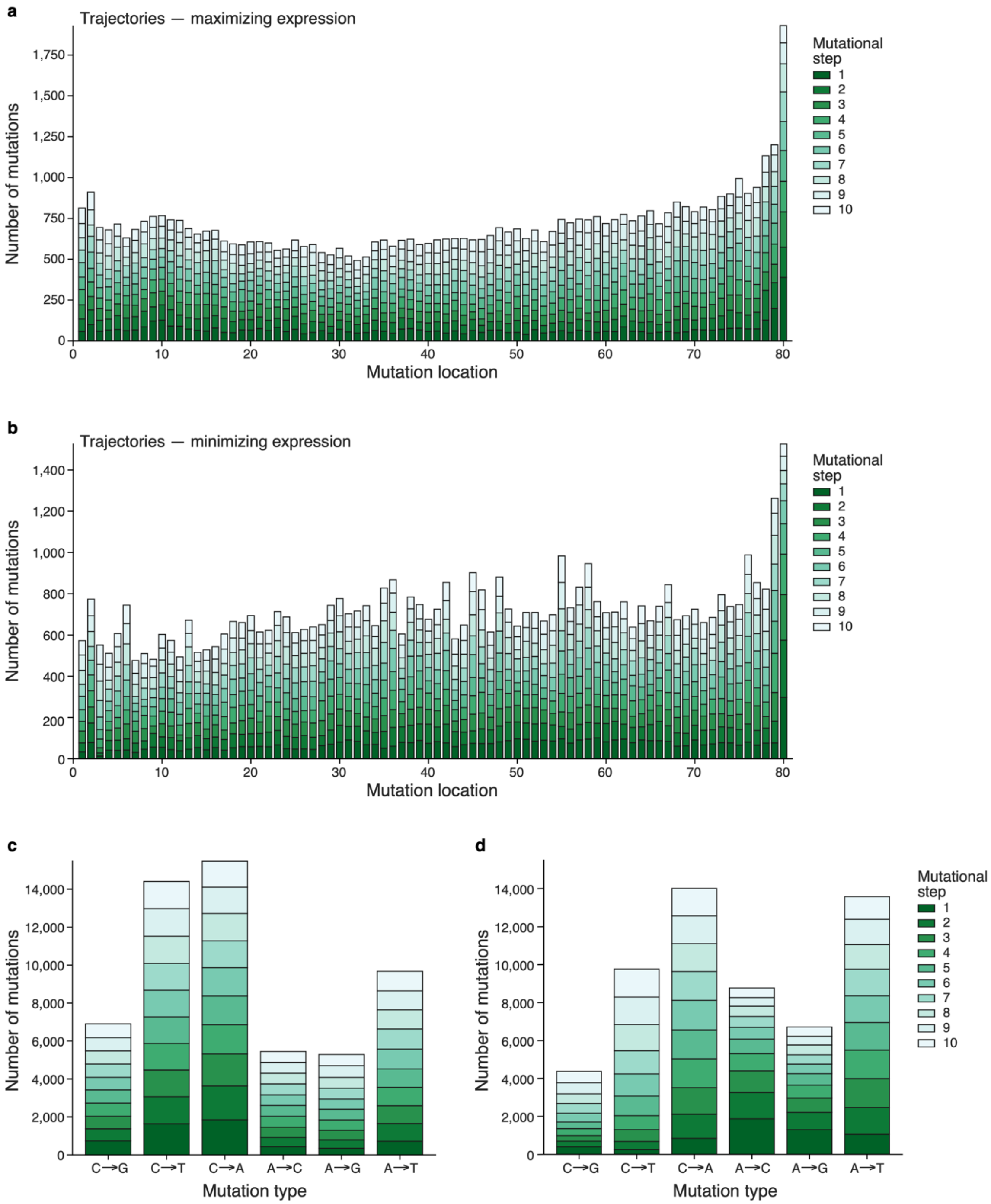
Top 5 motifs enriched in sequences evolved to **MAXIMIZE** difference in expression between the defined and complex media

Found motif (DREME)	DREME *E*-value	Rank in native	Known factor	Known motif (YeTFaSCo)
AGGGG	1.9×10^{-1752}	1	Several (e.g., RPH1)	AGGGG
TTCACATG	3.9×10^{-287}	2	INO2/4	TTCACATGC
TAAGG	7.2×10^{-155}	3	None	
ATGACTCA	2.0×10^{-135}	5	GCN4	ATGACTCA
TGATTGG	2.0×10^{-135}	4	HAP4	TGATTGG

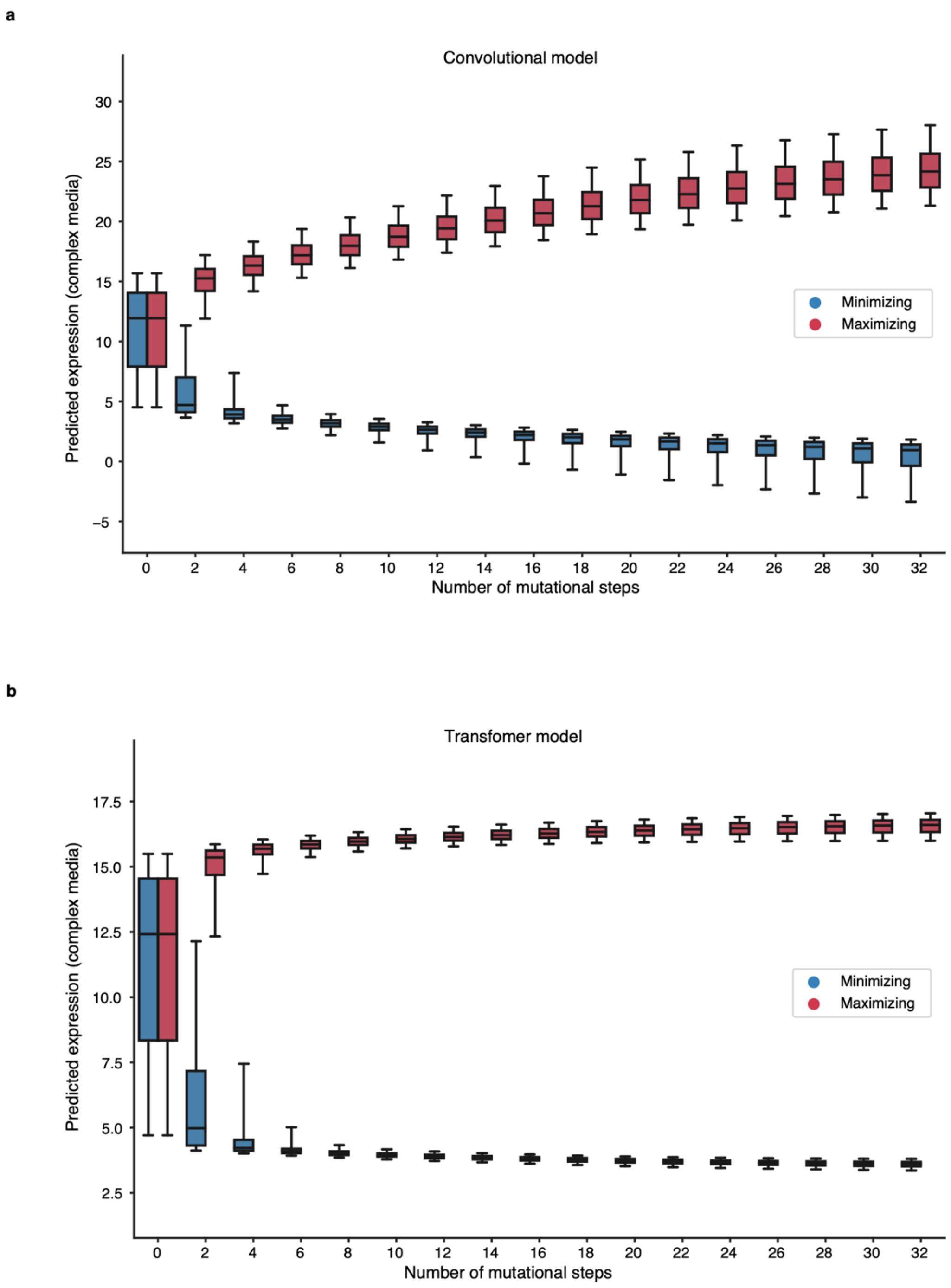
Top 5 motifs enriched in sequences evolved to **MINIMIZE** difference in expression between the defined and complex media

Found motif (DREME)	DREME *E*-value	Rank in native	Known factor	Known motif (YeTFaSCo)
CCACGTG	2.9×10^{-342}	1	Several (e.g., PHO4)	CACGTG
CGATG	6.2×10^{-141}	2	DOT6/TOD6	GATG
GATG	2.3×10^{-61}	2	DOT6/TOD6	GATG
CAGGGT	9.9×10^{-49}	5	AFT1/2	GGGTGC
TTA ATA	2.0×10^{-27}	7	CIN5	TTACGTAA

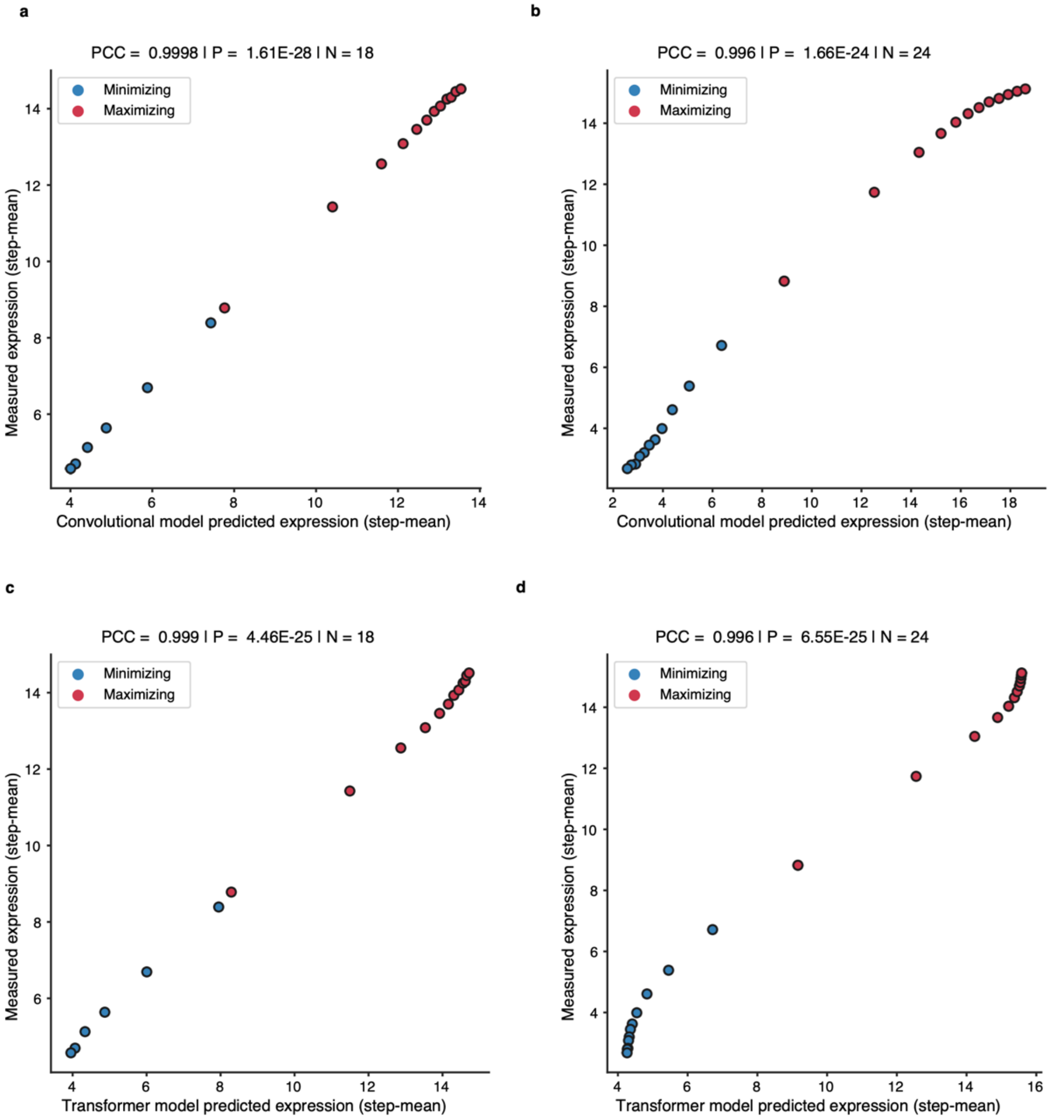
Supplementary Data Fig. 17 | Characterization of sequence trajectories under strong competing selection pressures using the transformer model. a-d, Competing expression objectives are slow to reach saturation. **a,b,** Difference in predicted expression (*y* axis) at each evolutionary time step (*x* axis) under selection to maximize (red) or minimize (blue) the difference between expression in defined and complex media, starting with either native sequences (**a**, n=5,720 trajectories) or random sequences (**b**, n=10,000 trajectories). **c-d,** Distribution of predicted expression (*y* axis) in complex (blue) and defined (red) media at each evolutionary time step (*x* axis) for a starting set of native sequences (**c**, n=5,720 trajectories) and random sequences (**d**, n=10,000 trajectories). Midline: median; boxes: interquartile range; whiskers: 5th and 95th percentile range. **e** Motifs enriched within sequences evolved for competing objectives in different environments. Top five most enriched motifs, found using DREME(Bailey, 2011) (**Methods**) within sequences computationally evolved from a starting set of random sequences to either maximize (left) or minimize (right) the difference in expression between defined and complex media, along with DREME E-values, the corresponding rank of the same motif when using native sequences as a starting point, the likely cognate TF and that TF's known motif.



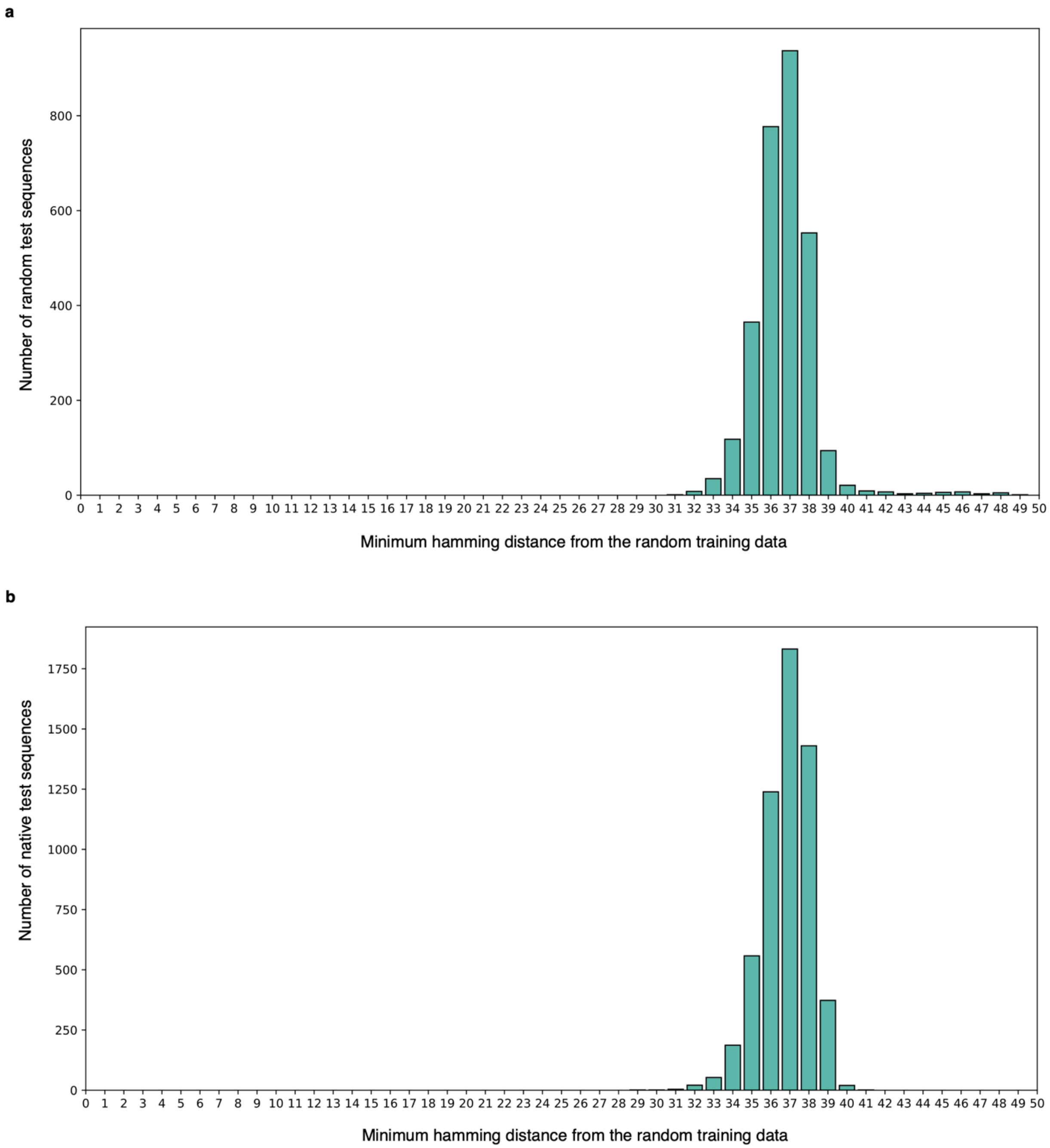
Supplementary Fig. 18 | Sequences took diverse paths to evolve extreme expression in simulations with the transformer model. a-b, The number (*y* axis) of mutations across each promoter position (*x* axis; -160 to -80 region) for trajectories under the SSWM regime starting with native promoter sequences when (**a**) maximizing or (**b**) minimizing expression in defined media using the transformer model. **c,d,** The number (*y* axis) of mutations of each type (*x* axis) for trajectories under the SSWM regime starting with native promoter sequences when (**c**) maximizing or (**d**) minimizing expression in defined media. Colors represent the mutational step (1-10).



Supplementary Fig. 19 | The transformer model captures expression plateau better than the convolutional model when simulating trajectories under SSWM for 32 mutational steps. Distribution of predicted expression levels (*y* axis) in complex media at each mutational step (*x* axis) for sequence trajectories under SSWM favoring high (red) or low (blue) expression, starting with native promoter sequences using the convolutional (**a**, n=5,720 trajectories) or transformer (**b**, n=5,720 trajectories) models. The transformer model predicts an expression level plateau (like the measured expression in **Fig. 2g**), while the convolutional model predictions do not plateau at higher mutational distances. Midline: median; boxes: interquartile range; whiskers: 5th and 95th percentile range.



Supplementary Fig. 20 | Summary statistic scatterplot for trajectories under SSWM. Mean measured expression (*y* axis) and mean predicted expression (*x* axis) at each step in the mutational trajectories for native sequences under SSWM for the convolutional (**a,b**) and transformer (**c,d**) model in the complex (**b,d**) and defined (**a,c**) media, as in Supplementary Fig. 19. The Pearson's correlation coefficient (PCC) and the corresponding two-tailed p-value are shown.



Supplementary Fig. 21 | Sequence differences between training and test data. The distribution of the Hamming distance between each sequence in the (**a**) random or (**b**) native test sets and the closest sequence in in the random training set.

Supplementary Tables

Supplementary Tables

Supplementary Table 1 | The Expression Conservation Coefficient (ECC), mutational robustness, evolvability vector archetypal coordinates, predicted expression, ECC using gene-specific correction factors, and ECC non-neutrality p-values corresponding to all native promoter sequences.

Supplementary Table 2 | The GO terms enriched by the ECC ranking. One-sided p-values were computed using minimum hypergeometric statistics, taking into account multiple testing as previously described(Eden *et al.*, 2009).

Supplementary Tables 1 and 2 are provided as an Excel file.

Supplementary Table 3 | Primers used in this study. The list of single stranded oligonucleotides used. This table can be found in the Supplementary Information document.

Name	Sequence (5'-3')	Orientation	Description	Reference
pCDC36_DBVPG6765_WT_fw	ATCCATACACAAGACTCATAGAA	Fw	WE gRNA	This study
pCDC36_DBVPG6765_WT_rv	AACTTCTATGAGTCTTGTGTATG	Rv	WE gRNA	This study
D6765_to_Y12_ssODN	TTCCATCTCTATATAACAAAGTATTTCTTTATTTTCTAATAGTTCCTTTCTACGAGTCTTGTGTATGTTTATAAAGAGTGAGCTCTTTTGTTATGAAGT	Duplex	ssODN SA allele	This study
pCDC36_seq_F	TCACACGTAGACGACTTGCCA	Fw	Sequencing	This study
pCDC36_seq_R2	CCTTGTAGTTTTTGCATATCTAGT	Rv	Sequencing	This study
Seq_3_Fw	ACTTGCCACATCCTGGTGTT	Fw	Sequencing	This study
Seq_3_Rv	ATGTTTCTGCCCACGGTGAT	Rv	Sequencing	This study
CDC36_Fw	CATGACCTTAGGAGCGGACT	Fw	qPCR	This study
CDC36_Rv	TCCACTTCGCTTCTGGATGT	Rv	qPCR	This study
ACT1_Fw	TTGGCCGGTAGAGATTTGAC	Fw	qPCR	Teste et al.
ACT1_Rv	CCCAAAACAGAAGGATGGAA	Rv	qPCR	Teste et al.

RPN2_Fw	GCGGATACAGGCACATTGGATACC	Fw	qPCR	Teste et al.
RPN2_Rv	TGTTGCTACCTTCTCTACCTCCTTACC	Rv	qPCR	Teste et al.
pT-pA_GibsRI	GAACTGCATTTTTTTCACATCNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNGGTTACGGCTGTTTCTTAA	Fw	Random promoter oligo for use in pTpA promoter context	de Boer et al
R-pT_GibsDS	TTAAGAAACAGCCGTAACC	Rv	For double-stranding pT-pA_GibsRI	de Boer et al
Nextera_i5LN5_GpT	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGNNNNNTGCATTTTTTTCACATC	Fw	Nextera adaptor addition, with 5 random bases to help clustering	de Boer et al
Nextera_i7R_GpA	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGAACAGCCGTAACC	Rv	Nextera adaptor addition	de Boer et al

Supplementary Table 4 | Strains. The list of yeast strains used.

Strain	Genotype	Reference
Y8205	*MATalpha, can1delta ::STE2pr-Sp_his5 lyp1delta ::STE3pr-LEU2 his3delta1 leu2delta0 ura3delta0*	Charles Boone Lab – strain verified by auxotrophy
S288C::*ura3*	*MATα SUC2 gal2 mal2 mel flo1 flo8-1 hap1 ho bio1 bio6 ura3delta0*	de Boer et al. 2020 – strain verified by PCR of URA3
DBVPG6765 (WE)	*MATalpha*, ho::NatMX, ura3::KanMX	Cubillos, Louis & Liti (DOI: 10.1111/j.1567-1364.2009.00583.x)
Y12 (SA)	*MATalpha*, ho::NatMX, ura3::KanMX	Cubillos, Louis & Liti
WE C7	DBVPG6765 derivate with SA Upc2 binding site	This study – pCDC36 genotype verified by Sanger sequencing
WE C23	DBVPG6765 derivate with SA Upc2 binding site	This study – pCDC36 genotype verified by Sanger sequencing

References

Alipanahi, B. *et al.* (2015) "Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning," *Nature Biotechnology*, 33(8), pp. 831–838.

Bailey, T. L. (2011) "DREME: motif discovery in transcription factor ChIP-seq data," *Bioinformatics (Oxford, England)*, 27(12), pp. 1653–1659.

de Boer, C. G. *et al.* (2020) "Deciphering eukaryotic gene-regulatory logic with 100 million random promoters," *Nature biotechnology*, 38(1), pp. 56–65.

de Boer, C. G. and Hughes, T. R. (2012) "YeTFaSCo: a database of evaluated yeast transcription factor sequence specificities," *Nucleic Acids Res*, 40(Database issue), pp. D169-79.

Brodsky, S. *et al.* (2020) "Intrinsically Disordered Regions Direct Transcription Factor In Vivo Binding Specificity," *Molecular Cell*, 79(3), pp. 459-471.e4.

Chen, J. *et al.* (2019) "A quantitative framework for characterizing the evolutionary history of mammalian gene expression," *Genome research*, 29(1), pp. 53–63.

De Boer, C. (2017) "High-efficiency S. cerevisiae lithium acetate transformation v1 (protocols.io.j4tcqwn)," *protocols.io*. ZappyLab, Inc. doi: 10.17504/protocols.io.j4tcqwn.

Eden, E. *et al.* (2009) "GOrilla: a tool for discovery and visualization of enriched GO terms in ranked gene lists," *BMC bioinformatics*, 10, p. 48.

Keren, L. *et al.* (2016) "Massively Parallel Interrogation of the Effects of Gene Expression Levels on Fitness," *Cell*, 166(5), pp. 1282-1294.e18.

Langmead, B. *et al.* (2009) "Ultrafast and memory-efficient alignment of short DNA sequences to the human genome," *Genome biology*, 10(3), p. R25.

Li, J. *et al.* (2020) "DeepATT: a hybrid category attention neural network for identifying functional effects of DNA sequences," *Briefings in bioinformatics*. doi: 10.1093/bib/bbaa159.

Quang, D. and Xie, X. (2016) "DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences," *Nucleic acids research*, 44(11), p. e107.

Quang, D. and Xie, X. (2019) "FactorNet: A deep learning framework for predicting cell type specific transcription factor binding from nucleotide-resolution sequential data," *Methods (San Diego, Calif.)*, 166, pp. 40–47.

Sharon, E. *et al.* (2012) "Inferring gene regulatory logic from high-throughput measurements of thousands of systematically designed promoters," *Nature biotechnology*, 30(6), pp. 521–530.

Shrikumar, A., Greenside, P. and Kundaje, A. (2017) "Reverse-complement parameter sharing improves deep learning models for genomics," *bioRxiv*, p. 103663.

Vaswani, A. *et al.* (2017) "Attention is All you Need," in Guyon, I. et al. (eds.) *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., pp. 5998–6008.

Weirauch, M. T. *et al.* (2013) “Evaluation of methods for modeling transcription factor sequence specificity,” *Nature Biotechnology*, 31(2), pp. 126–134.

Weirauch, M. T. and Hughes, T. R. (2010) “Conserved expression without conserved regulatory sequence: the more things change, the more they stay the same,” *Trends in genetics: TIG*, 26(2), pp. 66–74.

Zhou, J. and Troyanskaya, O. G. (2015) “Predicting effects of noncoding variants with deep learning-based sequence model,” *Nature methods*, 12(10), pp. 931–934.