Supplementary Materials:

Materials and Methods

Figures S1-S11

Tables S1-S2

Materials and Methods:

U3 mutant library construction (saturation mutagenesis)

The library of U3 mutants was constructed in a two-step PCR approach that included assembly and amplification PCR reactions. To generate the "Big" library we performed assembly PCR with 6 overlapping, “doped” oligonucleotides (IDT, all sequences in Table S2). In the doped oligonucleotides, each position contained the wild-type nucleotide at 97% frequency and a 1%:1%:1% mix of the three other nucleotide types. This protocol resulted in a 3% mutation rate per position. To generate the "Small" library we performed 6 independent assembly reactions, each using 1 doped and 5 non-doped oligonucleotides, and we mixed the assembly PCR products in equimolar ratios. The Small library had an approximately 1% mutation rate per position. The assembly PCR products were used as a template for amplification PCR, to add 20-nucleotide random barcodes and restriction sites to each variant (primers: U3_start_SalI and U3_end_bar_EcoR, in Table S2). Nucleotides 1-6 of the U3 gene represent a SalI restriction site, and we did not introduce any mutations in this fragment. The product was cloned using SalI and EcoRI sites into the pU3-empty vector, under the control of native U3 promoter. Plasmid pU3-empty was constructed by replacing wild-type U3 coding sequence and 70 downstream nucleotides with 40 nt double stranded oligonucleotide Sal_oligo_Eco (Table S2) in the ARS-CEN vector pU3-wt, carrying an ADE2 marker (*25*), created on the backbone of pASZ11 (*26*).

Ligation products were transformed into DH5a/TOP10 competent cells and plated on 450 plates (~400 colonies per plate), for a total of 180,000 colonies. Colonies were pooled in PBS and used directly for MIDIprep (18 columns) (Qiagen, UK) to generate the U3 mutant plasmid library. MiSeq 300-nt paired-end sequencing of the entire insert (U3 mutated sequence, non-mutated linker and barcode) was performed to check the complexity of library and to associate random barcode sequences with U3 sequences.

Yeast strains

The D343 yeast strain (leu2, ura3, ade2, can1, his1, his3, trp1, U3aΔ, UASGAL:U3A::URA3, U3B::LEU2), where the wild type genomic version of U3 is expressed under the control of galactose promoter, was transformed with 80 μg of U3 mutant plasmid library using LiAc/SS-DNA/PEG High Efficiency Transformation Protocol (*27*). The whole transformation mix was transferred to 1l of liquid synthetic medium without adenine, supplemented with 2% galactose. To estimate the efficiency of transformation, 100ul of this liquid culture was plated just after inoculation and incubated for 2 days in 30C (based on this, the efficiency of the main transformation was usually 6250 colonies / 1 μg DNA). The liquid culture was grown in 30C until it reached OD600=2.0, diluted to OD600=0.2 (at least $5x10^8$ cells were transferred) and grown again to OD=1.0. This stage we called population G0 and performed HiSeq Illumina sequencing of barcodes (Edinburgh Genomics, UK) to check the complexity of the library.

Assigning U3 sequences to random barcodes

To identify the sequences of U3 variants associated with random barcodes, we used the plasmid library to amplify a ~400nt fragment containing the U3 mutated sequence, non-mutated linker

and random 20-nt barcode (primers: IndexX_PCR_U3_seq and R_PCR_U3bar_seq) and used this as a template for 300-nt paired-end MiSeq Illumina sequencing (Edinburgh Genomics, UK), using the Custom_Read1_seq_primer_U3 and Illumina Read2_sequencing_primer (Table S2). Candidate barcode sequences were extracted from between flanking sequences using blastall and bedtools programs. A large proportion of these candidate barcodes represented Illumina sequencing errors and were only present at very low frequency in the sequencing. To identify bona-fide barcodes associated with the U3 variants, we filtered the list of candidate barcodes according to their frequency in several sequencing runs (the original U3 mutant plasmid library and the G0 yeast population). To avoid errors in barcode assignment, we also filtered barcodes by similarity to other barcode sequences. We then used bowtie2 (*28*) and samtools (*29*) to map all reads to wild type U3 and to identify the consensus U3 sequence corresponding to each random barcode. We called a mutation if the same change at a given position appeared in at least 80% of reads with a given barcode. To avoid ambiguous assignments of barcodes to the U3 sequence, we discarded all barcodes for which we found any mutation in U3 present in between 20-79% of reads. We also discarded barcodes for which the linker sequence between U3 and the barcode was mutated, and barcodes for which the U3 sequence was not fully covered by reads.

Competition experiments

The competition experiments were performed in synthetic medium without adenine (Formedium, UK), supplemented with 2% glucose (Sigma-Aldrich, UK) (or 2% galactose (Sigma-Aldrich, UK) for control conditions) inoculated with ~$5x10^8$ cells from population G0. Cells were grown in 500 ml of liquid medium at 30C (or 37C) and 230 RPM for 4 days. To keep the culture between OD600=0.1 and OD600=1.0, ~$5x10^8$ cells were transferred into fresh medium every 12

hours. Five competition experiments were performed, and samples were collected as shown in Table S1. Throughout the manuscript, unless otherwise noted, we show the results of experiment "Small_1_30C_Glu", but the results and conclusions were reproducible between experiments.

During competitive growth, selection could in principle increase the copy number of U3-containing plasmids. This would lead to unreliable fitness estimates, and to overestimation of fitness, particularly for variants whose fitness can be compensated by increased plasmid copy numbers. We estimated the magnitude of these effects by examining variation in fitness estimates among replicate measurements of the same U3 variant, both within and between experiments. Fitness measurements were reproducible, even among low-fitness variants (Figs 1C,D; S3C; S6), suggesting that random effects, such as changes in plasmid copy number, do not play a major role. In addition, copy number changes would lead to overestimation of low fitness effects, and in consequence, a positive bias in epistasis estimates. Thus, selection for copy number cannot explain the observed overall enrichment of negative interactions, nor can it explain any of the other major conclusions of the study.

Sequencing sample preparation

Yeast cells collected at each time point were treated with zymolyase in 3 ml of 20mM KPi pH=7.4 for 1 hour in 37C to remove the cell wall, and plasmid DNA was isolated from remaining spheroplasts using Qiagen MAXIprep, omitting the steps prior to addition of P2 buffer. We amplified 16 ng of template per sample, corresponding to ~ 1.5 x 10^9 plasmid molecules. Illumina (or Ion Torrent) adapters were added in a 25-cycles PCR reaction, performed with Phusion® High-Fidelity PCR Master Mix with HF Buffer (Fisher Scientific, UK) in 50 µl in each of 8 reaction tubes (HiSeq primers: IndexX_PCR_bar_seq with

R_PCR_U3bar_seq; Ion Torrent Proton primers: Proton_trP1_PCR_bar_seq_F with Proton_A_PCR_bar_seq_R and Proton_A_PCR_bar_seq_F with Proton_trP1_PCR_bar_seq_R Table S2). PCR products from all reaction tubes were pooled into 50 µl using a PCR Cleanup column (Qiagen). The appropriate PCR band was isolated by 2% E-Gel SizeSelect agarose gel electrophoresis (Life Technologies, UK) and quantitated by Bioanalyzer (Agilent, IGMM technical support/Edinburgh Clinical Research Facility, UK). For Illumina high-throughput sequencing, we usually pooled 6 samples (for HiSeq) or 2 samples (for MiSeq) mixed in equimolar ratios. In the case of Ion Torrent Proton (IGMM technical support/Edinburgh Clinical Research Facility, UK) we sequenced 1 sample per chip**.**

Counting of barcodes

FASTQ files from 50 bp Illumina HiSeq (or Ion Torrent) sequencing were demultiplexed. The template for deep sequencing of barcodes contained 25 nt fragments flanking the 20 nt barcode sequence of interest. The HiSeq sequencing was performed so that the first position in the read corresponds to the first nucleotide of the barcode and the 3' flanking sequence was found at the 3' end of reads (in the case of Ion Torrent sequencing both flanks were present in the read). We used blast (*30*) and bedtools (*31*) to locate and remove the 3' flanking sequence (or both flanking sequences) from reads and we counted unique barcodes. We recovered barcode sequences that could be uniquely matched to exactly one of the barcode sequences from the filtered list (see "Associating U3 sequences with random barcodes" above), allowing at most two sequencing errors per barcode.

Fitness estimation

In laboratory experiments with continuous growth and overlapping generations, the population size can be represented by the formula $N_t=N_0 \exp(m\ t)$, where N_t is the number of individuals at time t, N_0 is the initial number of individuals, and m is the exponential growth rate, also known as "Malthusian parameter" (*32, 33*). When two or more populations compete with each other, the Malthusian parameter of each population is equivalent to the logarithm of fitness (log fitness), and the difference in Malthusian parameters is equivalent to the difference of log fitness. Throughout this manuscript, we define the relative "log fitness" of a genotype as the Malthusian parameter of that genotype minus the median Malthusian parameter of the wild-type genotypes in the same experiment.

To obtain log fitness estimates from data, we used a Poisson regression approach (*34*) with exponentially decaying mean. Count numbers of barcode *l* at time *t* were modeled as Poisson random variables

$$n_l(t) \sim Po(m_l^t)$$

$$m_l^t = \frac{\exp(\lambda_l t + b_{l0})}{b_t}$$

Here λ_l represents the unknown (un-normalized) log fitness of barcode *l*, and b_t and b_{l0} are normalization factors accounting for different library sizes and different initial counts. We placed Gaussian priors over the λ_l, b_t and b_{l0} variables and obtained Bayesian posterior estimates using the Expectation-Propagation approximation (*35*). Log fitness estimates were then adjusted by subtracting the median log fitness estimate of barcodes corresponding to wild type U3 to produce normalized log fitness estimates.

Reproducibility of fitness measurements for over 2,000 wild-type variants of U3 with different barcodes was used to set the minimal number of reads at G0 which assure reliable results, and we filtered the rest of the barcodes accordingly.

Because mutations known to completely inactivate U3 (in C' and D boxes) all had average log fitness estimates between -2 and -2.5, we reasoned that barcode log fitness estimates below -3 were unreliable. Such estimates showed the largest variation between replicate experiments and were typically based on low numbers of reads. Overall, about 700/22,000 variants in each of the Small library datasets, and 5,000/37,000 variants in the Big library dataset, have log fitness estimates below -3. We therefore replaced these values by -3. Omitting this step increased the noise associated with the data, but did not change any of the conclusions. Replacing this step with a smooth tanh transformation of log fitness estimates, $\lambda'=3*\tanh(\lambda/3)$, with parameters chosen so that the transformation is approximately linear in the [-2,0] range and maps values below -3 to values close to -3, led to the same conclusions.

Positional fitness effects

f_i, the average log fitness effect of mutations in position i in an otherwise wild-type background, was defined as the mean log fitness of variants that had a single substitution or deletion at position *i*, and no other mutations elsewhere.

p_i, the aggregate log fitness effect of position *i* across all genetic backgrounds, was defined as the mean log fitness of variants that had a substitution or deletion at position *i* (and possibly other mutations elsewhere) minus the mean log fitness of variants that had no mutation at position *i*.

Explaining fitness from mutation patterns

In order to attribute the fitness changes to the underlying mutation pattern, we used a regression-based approach. We associate each mutant i with a feature vector z_i, a binary vector whose individual entries correspond to all positions and pairs of positions in the U3 gene. Thus z_i is a vector with 333+(333x332)/2= 55,611 dimensions. z_i will have 1 at entry j<334 if and only if the corresponding U3 variant is mutated in that position. Notice that this is a redundant representation: the binary code in the first 333 entries of z_i uniquely determines the remaining 55,278 entries, nevertheless this redundancy is necessary to disentangle the effects of single mutations from epistatic effects of pairs of mutations.

We then modeled the response variable (log fitness) as a linear function of the corresponding feature vector:

$$\lambda_i = w \cdot z_i + \varepsilon \qquad (1)$$

where w is a weight vector to be learnt from the data and ε is an error term. The weight vector has the same dimensions as the feature vectors z. The weight vector encodes the effect on fitness of the mutations: the first 333 entries code for the effect of single point mutations, while the remaining 55,278 capture the epistatic effects of having a mutation in two different locations.

One can highlight the different type of terms (single point and epistatic terms) by reformulating the response equation (1) in terms of features z_l and z_{lm}, and corresponding weights w_l and w_{lm}. A barcode j would have a non-zero feature l if it was mutated in position l and a nonzero feature lm if it was mutated in both positions l and m. Consequently, the coefficients w_l and w_{lm} would represent the single point and epistatic effects learned by the model. To extract the single point

and epistatic effect terms from the measured fitness values, we solve the following regularized least squares problem

$$w = \arg\min\left(\sum_j \left\|\lambda_j - wz_j\right\|^2 - L(w)\right)$$

where *L(w)* is a regularization term which is needed since the number of parameters which must be estimated exceeds or is comparable to the number of barcodes in each library. We consider two types of regularizers:

$$L(w) = a\sum_j w_j^2$$

so called Tikhonov regularization or ridge regression (RR), avoids overfitting by penalizing large coefficients;

$$L(w) = a\sum_j \left|w_j\right|$$

L1 penalty giving rise to the Lasso regression (*36*), a popular choice (*16*) as it returns sparse estimates where irrelevant coefficients are set to zero. In both cases, the regularization coefficient was chosen by five-fold cross-validation on a grid of values. Out of sample predictions at the optimal regularization value (a=5) indicated a good predictive power, with on average 55% of total test variance explained by the model across the cross-validation runs. These values indicate that the model achieves a good fit without overfitting to the training data. To quantitatively measure the performance of the trained model, we measured the explained variance, i.e. the difference between the initial sample variance and the sum of the squared model residuals, relative to the initial variance. As the distributions of fitnesses and residuals are strongly non-Gaussian, we removed the top and bottom 5% of data to eliminate outliers which

may dominate the estimated variances. Using this procedure, the optimal epistatic model explained 86% of the initial variance, while a nonepistatic model, which used only log fitness effects associated with single mutants, explained 49% of the variance. It should be remarked that using the measured effects of single mutations is not necessarily optimal from the point of view of explained variance. To ensure a fair comparison, we also repeated the fitting procedure using solely features associated with single mutations. This optimal nonepistatic model explained approximately 55% of variance, in line with previous reports (*16*) and considerably below the variance explained by an epistatic model.

We found in our experiments that both Lasso and RR were able to compute reproducible estimates of single-site coefficients, which were in good agreement with the experimentally measured values (Pearson $R=0.87$, $p<0.05$). Estimation of pairwise effects with Lasso was however less successful, as Lasso consistently retained almost exclusively the large epistatic effects associated with pairs of positions with large single-site effects. By contrast, RR captured both positive and negative epistatic effects. Comparisons with directly measured pairwise effects on a subset of mutants with only two mutations reveals that the empirical distribution of pairwise effects appears to encompass both large positive epistatic effects, and smaller (but still important) negative epistatic effects. It therefore appears that Lasso, in the limited data regime we operate, is not well suited to estimate relevant coefficients of highly heterogeneous size, and effectively eliminates most of the negative epistasis by shrinking such coefficients to zero.

Further analysis of the RR results revealed a bias in the estimation of single-site effects (see Fig. S8), which is more pronounced for the Big library (where few single-site mutants were present). This is possibly due to the redundant encoding of mutants, which creates correlated features where explaining away is possible. We therefore further modified the RR model by directly

inputting the w_j coefficients as measured on single-site mutants, and estimated the remaining entries of the w_j vector from the data. The results shown in the manuscript come from this modified model.

The code used in our numerical experiments is available on github: https://github.com/terembura/EpistaticInteractionsYeastU3.

Resolution of fitness estimates

The standard deviation of log fitness estimates of wild-type variants ranges from 0.11 to 0.18 in the small library experiments in glucose. As a result, very small effects, both negative and positive, could not be detected experimentally even if they did play some role in evolution.

There were no single mutants with a fitness estimate greater than 2 standard deviations above wild-type in any experiment. Although 17 single mutants had fitness estimates greater than 1 standard deviation above wild-type, none of these mutants were reproducibly fitter than wild-type in all three replicate experiments. Thus, no mutations increased fitness to a degree detectable with our technology.

Visualization of fitness effects and epistatic interactions

To visualize the fitness effects of individual mutations and their epistatic interactions, we used JavaTreeView (*37*) to generate 2D heatmap plots, VARNA (*38*) to display the effects of mutations along the secondary structure, and Circos (*39*) to generate circular interaction plots. To generate the heatmap shown in Fig. 1D, the fitness effect of each mutation was calculated as the median fitness of single mutant variants that contained the focal mutation, and of the subset of

double mutants that contained the focal mutation plus a low-effect second mutation (|log fitness effect| of the second mutation <0.01). Inclusion of double mutants significantly improved the coverage and reduced noise, without altering the conclusions.

Prediction of U3 secondary structure using epistatic coefficients

To obtain folding constraints for secondary structure prediction, we averaged the epistatic coefficients w_{ij} from four glucose competition experiments, and we filtered these coefficients in two ways. First, we identified all pairs of residues that could potentially interact within uninterrupted stems of 3 or more nucleotides (allowing Watson-Crick and G-U interactions). This retained 7,637 of the 55,278 epistatic interactions. Second, we removed interactions involving at least one large-effect site (defined as regions 80-87 (Box C'), 252-257 (Box C) and 325-329 (Box D). This further reduced the dataset to 6,639 interactions between potentially basepaired sites. We then computed epistatic support scores for each 3-nt stem by averaging the relevant w_{ij} values, as in (*40*). In the resulting set, known interactions had significantly larger support scores than noninteracting pairs (Wilcoxon test, $p<2*10^{-16}$), and in particular, 5 out of 6 top scores corresponded to known basepairing interactions. These 5 scores were used as constraints in RNA structure prediction by the Vienna program (*41*).