

Supplementary Information

Contents

1	Design of the renewable barcoding system	3
1.1	Strains	5
1.2	Construction of barcoding plasmids	5
1.3	Barcoding procedure	7
1.4	Evolution experiment	8
1.5	Barcode sequencing	10
2	Read parsing, error correction, UMI removal, assembly, and chimera removal	12
3	Fitness measurements using fluorescent labels	13
4	Construction of a model for barcode count trajectories	13
4.1	Branching process model for the lineage read count trajectory	14
4.2	Modeling drift and selection during the barcoding procedure	16
4.3	The relative importance of drift and selection	17
4.4	Inference of the parameters of the branching process model	18
5	Inference of lineage fitness and the arising times of new mutations from barcode count trajectories	20
5.1	Identification of selected lineages	22
5.2	Joint estimation of fitnesses of selected lineages	23
5.3	Identification of barcodes that carry new mutations	26
6	Analysis of clone fitness, fate, and overall population dynamics	28
6.1	Joint inference of clone fitnesses	28
6.2	Impact of the barcoding procedure on the evolutionary dynamics	29
6.3	Replication of dynamics without barcoding	35
6.4	Genetic and fitness diversity	36
6.5	Probability of acquiring further established beneficial mutations	38
6.6	Comparison with fitness measurements using competitive fitness assays	41
7	Metagenomic sequencing and sequence data analysis	42
7.1	Consistency between metagenomic and barcode sequencing data	43
	Supplementary Data Tables	49
	References	50

List of Supplementary Figures

1	Landing pad locus.	6
2	Barcoding plasmids.	7
3	Detailed schematic of the barcoding procedure.	9
4	Statistics of barcoded lineages immediately after barcoding.	10
5	Comparison of observed and best-fit theoretical distributions of read counts.	19
6	Inferred parameters of the branching process.	20
7	Number of epochs needed to detect a positively selected lineage.	28
8	Comparison of the effect sizes of individual mutations on fitness in the evolutionary and barcoding conditions	35
9	Changes in the log-frequencies of each called clone over the course of (top) a single barcoding interval or (bottom) a single evolution epoch.	36
10	Average fitness compared to the evolution environment fitness for each called clone.	37
11	Traveling wave dynamics in the evolution environment.	38
12	Traveling wave dynamics in the barcoding environment.	39
13	Replication of evolutionary dynamics in the YPA population.	44
14	Variance in fitness through time	45
15	Probabilities of accumulating an equal or larger number of mutations in each epoch based on a clone's frequency.	46
16	Probabilities of accumulating an equal or larger number of mutations in each epoch based on a clone's frequency and fitness.	46
17	Probabilities of accumulating an equal or larger number of mutations over the course of the experiment.	47
18	Simulated metagenomic sequencing datasets.	48

List of Supplementary Tables

1	Earlier gene stacking methods.	4
2	Inferred parameters of the branching process.	21
3	Clone fitness estimates in the YPD population.	31
4	Clone fitness estimates in the YPA population.	34
5	List of all barcoded lineages and their frequencies in each sequencing time-point.	49
6	List of total lineage read counts in each sequencing time-point.	49
7	List of all mutations called from metagenomic sequence data.	49

1 Design of the renewable barcoding system

Several requirements must be met when designing a renewable barcoding system for tracking microbial lineages. In this section, we outline considerations which were important in motivating our design choices for this study. Briefly, they include an appropriate choice of organism/strain, use of high efficiency genetic modification systems, and minimal perturbation during barcoding.

One important requirement in any renewable barcoding system is to maintain the barcode locus at a single copy per cell, which requires a stable ploidy. Although in principle it would be possible to track cells with a varying number of barcode loci, this would not allow faithful lineage tracking with re-barcoding. This requirement makes it challenging to work in any bacterial model system. Instead, we chose to work in the budding yeast *Saccharomyces cerevisiae*, a fast-growing model microbe that has a more stable ploidy. However, because diploids are more fit than haploids in many common laboratory environments, experimental evolution in haploid *S. cerevisiae* often results in autodiploidization or other types of aneuploidies [1]. For this reason, we chose to work in diploid lines. Despite studies showing that tetraploids can evolve faster than diploids [2], preliminary experiments indicated that these aneuploids do not provide a strong fitness advantage over diploids, and do not routinely invade diploid populations. Despite this preliminary work, we did find that over the course of long-term evolution, tetraploids do indeed appear in some populations, and these must be discarded from our lineage tracking analysis.

Working in diploid lines limits the technological choices for barcode introduction: high-efficiency CRISPR or endonuclease-based approaches [3] are no longer possible due to the high frequency of homology-mediated repair that occurs through homologous chromosomes. We instead take advantage of recombinase-based technologies to selectively introduce barcodes into the desired locus. Our approach is inspired by earlier gene stacking methodologies, which typically use at least two enzymes [3–8]. One enzyme is unidirectional and introduces a plasmid carrying the desired fragment into a target genomic site. The other enzyme removes the extraneous plasmid sequences and recycles the resistance marker. The enzyme that is unidirectional typically requires two orthogonal sites (i.e. sites that do not react with each other). The general schemes are similar to Figure 1b, with particulars differing for the enzymes of choice (see Supplementary Table 1 for a summary).

These earlier gene stacking methods use sequential enzyme activation (they first integrate the plasmid, and then excise and recycle extraneous plasmid sequence). This reduces their efficiency to the product of transformation and integration efficiency, which is typically $\sim 10^{-5}$ or less. For our purposes, this would lead to large population size bottlenecks which would interfere with experimental evolution. To avoid this problem, our approach decouples transformation and integration. We first transform (with efficiency $\sim 10^{-3}$) and then allow the transformed population to grow, after which we perform a single integration and recycling step (also with efficiency $\sim 10^{-3}$). Because of the intermediate growth step, the population is never bottlenecked by a factor larger than $\sim 10^{-3}$.

To perform our single integration and recycling step, we must rely on a single enzyme. We developed a method to do so using the Cre recombinase and three orthogonal Lox sites. Several orthogonal Lox sites have previously been identified [9] and the molecular basis for orthogonality is well understood: each Lox site contains a spacer sequence that determines the specificity of Cre-mediated recombination, and two Lox sites that differ by both a mutation in position 2-5 and a mutation in position 6-7 will be orthogonal. In addition to this simple rule, several other orthogonal spacer sequences that differ in other ways have been described [10]. We tested the efficiency of Cre-mediated recombination in yeast at several of these sites, and found that LoxP (ATGTATGC) had the highest efficiency, several had $\sim 50\%$ lower efficiency (Lox5171 (ATGTgTaC), Lox2272 (AaGTATcC), Lox4361 (ATGgAcGC), and Lox5261 (ATGTtCGC)), and six others very low efficiency ($< 25\%$ compared

Reference	Integration Enzyme	Recycling Enzyme	Notes
Nandy et. al. (2015) [4]	Cre (Transient Expression)	I-SceI, ZFN	Transient Cre expression integrates the plasmid. Double-strand break inducing enzymes promote excision of the plasmid via NHEJ or homologous recombination.
Lin et. al. (2013) [5]	Cre (Transient Expression)	I-SceI, PI-SceI	<i>Idem.</i>
Lee et. al. (2014) [6]	Cre (Transient Expression)	PiggyBac	PiggyBac is a transposase, which can mediate unidirectional excision of the integrated plasmid.
Hou et. al. (2014) [7]	Bxb1	Cre	Bxb1 is an integrase, which mediates unidirectional integration of the plasmid.
Chen and Ow (2016) [8]	phiC31	Cre	phiC31 is an integrase, which mediates unidirectional integration of the plasmid.
Wingler and Cornish (2011) [3]	I-SceI, HO	—	Yeast homologous recombination is exploited in this method to mediate unidirectional integration and recycling via alternating double-strand break inducing enzymes.

Supplementary Table 1: **Earlier gene stacking methods.** Brief summary of enzymes used in previous gene stacking methods.

to LoxP). Based on these results, we chose to work with LoxP, Lox5171, and Lox 2272, since these were the most well studied of the high-efficiency sites. However, we note that Lox5171 and LoxP have been shown to react at low frequency; this can lead to the excision of integrated fragments if the Lox sites are oriented in the same sense direction and are far enough apart. To mitigate this, our system ensures that the Lox5171 and LoxP sites are always anti-sense and close to each other.

Using this Cre-Lox system for gene stacking required us to develop a method to make the Lox sites unidirectional. This is critical for renewable barcoding approaches: if it is not correctly performed, Lox sites will react in subsequent barcoding reactions and the desired constructs will not be obtained. Unfortunately, Cre-Lox recombination is reversible. To make unidirectional Lox sites, we must shift the equilibrium of this reaction to the desired product. To do so, we take advantage of the fact that Cre binding to its Lox site is cooperative. The Lox sites contain two palindromic “arms” (flanking the internal spacer) that bind to Cre. Earlier work has argued that the presence of certain mutations in only one of the two arms only slightly diminishes Cre binding, while mutations in both arms strongly reduce it [11]. The most well-known arm mutants with this property are the Lox66 left arm mutation (`taccgTTCGTATA-spacer-TATACGAAGTTAT`) and Lox71 right arm mutation (`ATAACTTCGTATA-spacer-TATACGAAcggta`). Recombination between two of these sites leads to Lox72 with mutations in both arms (`taccgTTCGTATA-spacer-TATACGAAcggta`). In principle, this could be used to shift the equilibrium: Lox66 and Lox71 sites recombine to produce Lox72 sites in a unidirectional fashion.

However, despite the assumptions of some earlier work [12], we found that the doubly-mutated

Lox72 has only slightly lower reactivity than LoxP (this is consistent with the original description of the arm mutants, which showed only a 7-fold shift in reaction equilibrium when the mutations are in cis versus in trans [11]). Thus these Lox66 and Lox71 mutations do not result in strongly unidirectional sites. We therefore sought to identify arm mutations that would maintain similar Cre efficiency when the mutated arms are in trans, while completely abolishing its activity when in cis. To do so, we systematically screened a small set of potential arm mutants using an in vitro LacZ excision assay. These mutations were either described in the literature to be more resistant to recombination than Lox72 when in cis (LoxKR3 [13], Lox66/71 [11], LoxJT5 [14], LoxJT21 [14], LoxJT510 [14], and LoxJTZ17 [14]), or had been shown to retain comparable Cre activity when in trans (Lox2 [15]). We screened these seven arm mutations and all of their pairwise combinations, finding a single arm mutant (Lox2 [15]) that reduced Cre activity by at least six orders of magnitude when combined with itself in cis, while having no detectable reduction in efficiency compared to wild-type (LoxP) when in trans. We confirmed these results in vivo in budding yeast using a Ura3 excision system.

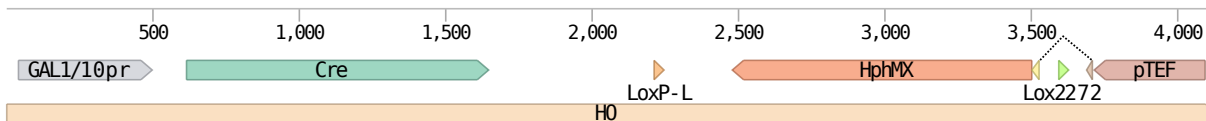
Motivated by these results, we used Lox2 arm mutants as the basis for our unidirectional Lox sites. We chose to make LoxP and Lox5171 the unidirectional sites, which flank the barcodes, because this allows us to keep these sites close together and hence further mitigate cross-reactivity. We use Lox2272 as the recycling site downstream of the resistance marker. We note that the Lox2 arm mutants represent a significant improvement to the Cre site-specific recombinase technology which may be useful for other applications using Cre (e.g. Brainbow [16]). However, some Cre reactions that work on wild-type LoxP sites do not work on Lox2 arm mutants even in trans; these are not relevant for our applications here but may be relevant in other studies.

1.1 Strains

Strains in this study are derived from the BY4742 strain background [17] (S288C: *MAT α* , *his3 Δ 1*, *ura3 Δ 0*, *leu2 Δ 0*, *lys2 Δ 0*) with several modifications. We chose to work in the S288C background due to its high transformation efficiency. The *RME1pr::ins-308A* mutation identified in a previous study to increase sporulation efficiency [18] was introduced by Delitto Perfetto [19], yielding YAN404. A *MAT α* version of this strain was obtained using HO endonuclease mediated mating type switching, yielding YAN407. This *MAT α* strain was then marked at HO with the NAT marker [20] (resistance to nourseothricin) driven by the TEF promoter from *Lachancea waltii*, and terminated with tSynth7 [21] (*HO::LwpTEF-NatMX-tSynth7*), creating strain YAN499. The *MAT α* version of the strain was marked at *HO* using the construct *HO::Gal1/10pr-Cre-tCyc1::LoxP2L::HygIN::Lox2272::AgpTEF*, creating strain YAN504 (note we use R and L in Lox site notations to indicate the location of the mutated palindromic arms with respect to the middle spacer). This construct forms the initial “landing pad” into which barcodes will be inserted (Supplementary Figure 1); it contains the Cre recombinase under the control of the galactose promoter and the hygromycin resistance marker driven by the TEF promoter from *Ashbya gossypii*, with an artificial intron containing the Lox2272 site [9] following the initial methionine (described in more detail in the next section). The *CAN1* locus was then replaced with mating type reporters [22] (*can1::Ste2pr-SpHIS5::Ste3pr-LEU2*), forming strain YAN508. These final *MAT α* and *MAT α* strains YAN499 and YAN508 were then mated to form the strain YAN517, which we used as the ancestor in the evolution experiment.

1.2 Construction of barcoding plasmids

Our barcoding system makes use of two alternating landing pads, and therefore two types of barcoding plasmids were constructed (one for each landing pad). We used a centromeric plasmid



Supplementary Figure 1: **Landing pad locus.**

(pAN316a) containing the URA3 gene (with BsaI restriction endonuclease site removed) as the backbone of these barcoding plasmids. The first type of barcoding plasmid had the configuration pAN316a-LoxPR-KanPX-ccdB-Lox5171L-KanIN-Lox2272, while the second type had the configuration pAN316a-Lox5171R-ccdB-HygPX-LoxPL-HygIN-Lox2272 (Supplementary Figure 2). Here KanIN and HygIN are geneticin and hygromycin B resistance markers respectively, with modifications. First, the initial methionine was replaced with an artificial intron containing the Lox2272 recombination site. Second, the methionine at position 17 in the geneticin resistance marker was mutated to a leucine. In this way, these resistance markers are non-functional on the plasmid, but become functional when recombined into the landing pad. This strategy is similar to the promoter trap often used in mammalian cell line genetic engineering, with the added stringency from the necessity of correct splicing. We constructed six different versions of each of these two types of barcoding plasmids, each containing a different priming site (KanPX or HygPX, where X ranges from 1 to 6).

We next introduced a diverse barcode library into each version of each type of plasmid using a Golden Gate reaction [23], exploiting the fact that the ccdB gene (with BsaI site removed) is flanked by two BsaI restriction endonuclease sites (see Supplementary Fig. 3a for a schematic illustration). Barcodes were ordered as single-stranded oligos from IDT (Integrated DNA Technologies) with degenerate bases selected as “hand-mixed”. The following oligo was used: P_BC=CTAGTTATTGCTCAGCGGAGGTCTCAtactNNN TANNNNATNNNNNTANNNcgctAGAGACCGT CATAGCTGTTTCCTG. This contains 16 degenerate bases, separated by bases that prevent BsaI restriction enzyme sites from being randomly formed, and is flanked by two BsaI sites that leave TACT and CGCT as overhangs. Another oligo was purchased corresponding to the M13 reverse (-27), which is complementary to the 3' end of the barcoding oligo. To convert the single-stranded oligo into double-stranded DNA, the two oligos were mixed at a 1:2 molar ratio (two times more of the M13 reverse (-27) primer), using about 1µg of the barcoding oligo, 5µL of NEBuffer 2.1 (New England Biolabs), topped to 50µL in a thermocycler using the following cycling conditions: 95°C for 2 minutes, with a reduction of 1°C every 30 seconds until room temperature. To the annealed oligos, dNTPs were added to 0.5mM. The PCR tube was cooled down to 0°C, and 1.5U of T4 DNA polymerase (New England Biolabs) was added, quickly mixed and cycled: 5 minutes at 0°C, 5 minutes at 22°C, 30 minutes at 37°C, and then held at 0°C. As soon as the reaction terminated, EDTA was added to 10mM to stop the reaction, and purified with PCR cleanup kits.

Following conversion to dsDNA, the plasmids were barcoded using Golden Gate reaction (New England Biolabs) using about 200ng of entry vector and 1µL of the purified dsDNA barcoding oligo, purified and electroporated into *Escherichia coli* DH10b cells (Invitrogen). To maintain library diversity, for each library the cells were recovered in 800mL of molten LB (0.5% yeast extract, 1% Tryptone) containing 100µg/mL carbenicillin (GoldBio) containing 0.3% SeaPrep agarose (Lonza) spread into a thin layer (about 1 cm). After mixing to homogenize, the cells were left in an ice bath for an hour to gel the agarose and then moved to a 37°C incubator for one day. This procedure allowed dispersed growth of colonies in 3D, and we routinely obtained over a million transformants. After clear colony growth, the soft-agar gel was shaken in baffled flasks to homogenize, and the

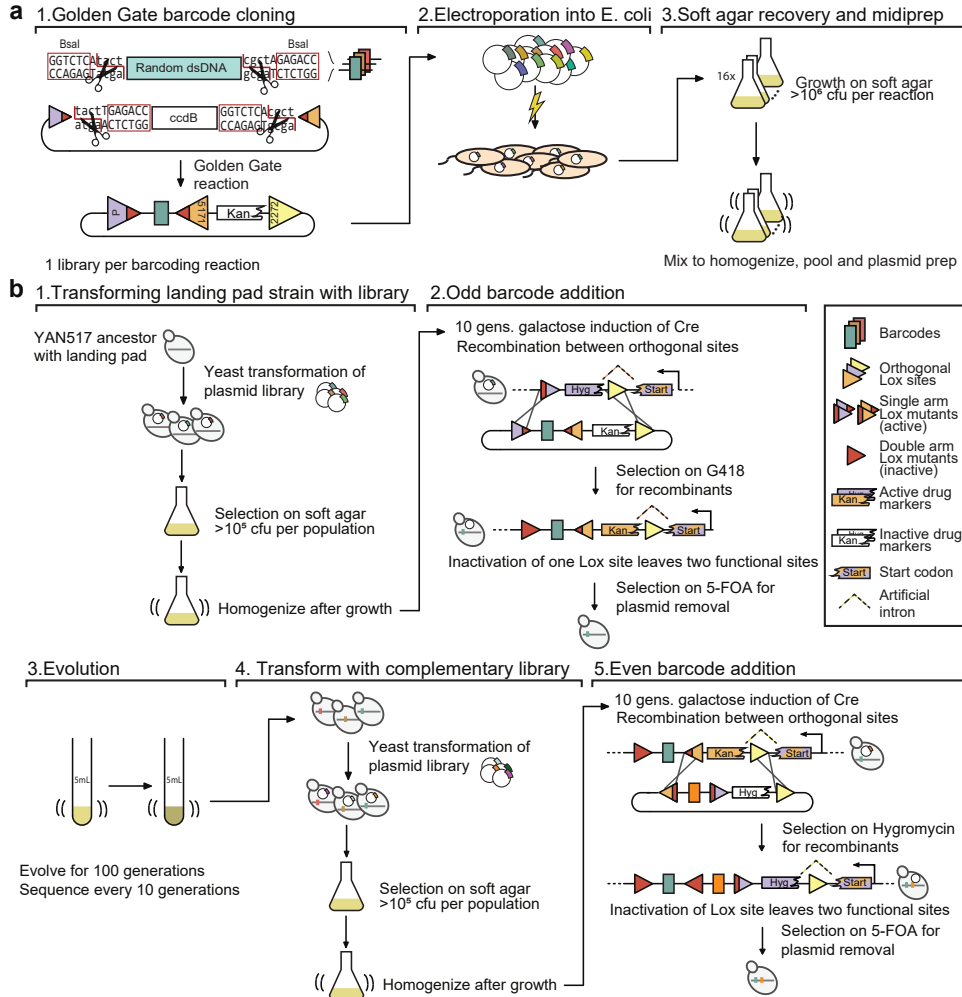
nants, we frequently observe that about 50 000 new barcodes are introduced into the population at the end of the process using this procedure (Supplementary Figure 4). This loss in diversity is presumably due to drift or selection during the barcoding process; we quantify these effects in detail in later sections below, and show that they lead to a similar scale of genetic drift as in the rest of our experiment and hence do not significantly affect the long-term effective population size.

We note that our barcoding process attempts to minimize potential mutagenic effects of yeast transformation: our strategy does not involve any host protein, such that DNA damage repair pathways that are typically used during homologous recombination transformation techniques are not exploited. While we cannot exclude the possibility that some mutagenic effects remain, because barcoding happens at frequent intervals these should simply lead to an overall elevation of the mutation rate in our experiment, which would not affect our analysis. Given the total number of mutations we observe, which are broadly consistent with earlier evolution experiments that did not involve barcoding, any such effects are likely to be small.

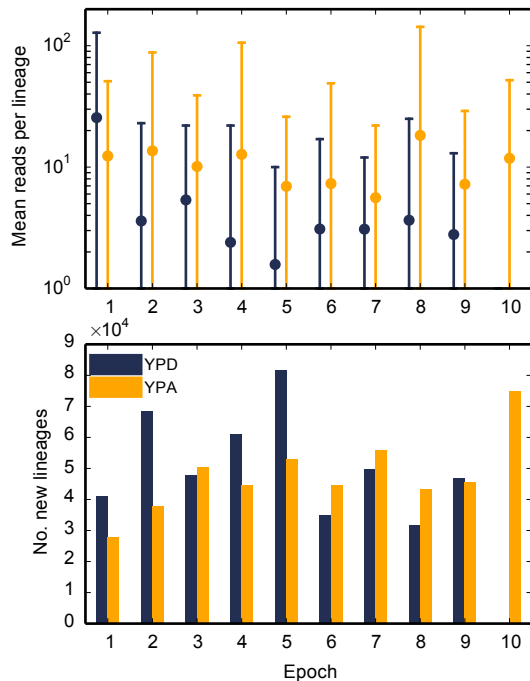
1.4 Evolution experiment

To allow detection of cross-contamination during the experiment, we first barcoded the ancestor as described above and founded two populations from single colonies carrying unique barcodes. These founding populations were then re-barcoded to begin the evolution with high barcode diversity. We propagated these lines at 30°C in 15mL glass culture tubes on a roller drum in 5mL of media (either YPD or YPD+50mM Acetic Acid (referred to as YPA), containing 10µg/mL nourseothricin (GoldBio) and either 300µg/mL hygromycin B or 200µg/mL G418 (GoldBio)). We diluted these populations by 1:2¹⁰ daily by passaging 4.88 µL into fresh media. Whole population pellets were stored daily at -20°C for later sequencing. At 30 generation intervals, we mixed an aliquot with 5% glycerol and froze at -80°C for long-term storage. As previously described [25], this protocol results in about ten generations per day, with a daily bottleneck size of about 10⁶ in YPD and about 5 × 10⁵ in YPA, corresponding to an effective population size of 10⁷ in YPD and 5 × 10⁶ in YPA. After each 100 generations of growth, we re-barcoded the population using the previously described procedure. We refer to each 100-generation interval as an “epoch”.

We evolved both YPD and YPA populations for a total of 10 epochs. After each barcoding procedure, we verified the integrity of the barcode locus by whole-population PCR and Sanger sequencing. This initial verification provides a gross assessment of barcode diversity and can diagnose general failures of the barcoding process. In addition, we isolated 32 clones from each population 90 generations after barcoding, conducted PCR of the barcode locus in each clone, and used gel electrophoresis to confirm that each clone contains a barcode locus of the correct configuration. These checks confirmed correct barcoding in all cases (including correct locus length in at least 27 of 32 clones), with one exception: after the tenth and final barcoding of the YPD population, we found that 23 of 32 clones lacked any detectable PCR product corresponding to the final barcode addition. This apparent failure could reflect leakage of unbarcoded cells through the barcoding process due to tetraploidization, inappropriate retention of the barcoding plasmid from the previous barcoding step, or some other mechanism. For this epoch, barcoding failure is also reflected in FACS-based fitness assays showing a dramatic decline in fitness of the YPD population at this timepoint. We therefore excluded this final epoch of the YPD population from all further analysis, leaving us with a total of 900 generations of evolution (9 epochs) in the YPD population and 1000 generations of evolution (10 epochs) in the YPA population.



Supplementary Figure 3: **Detailed schematic of the barcoding procedure.** **a.** Generating the barcode plasmid library. 1) Oligonucleotides containing random nucleotides are flanked by BsaI restriction endonuclease sites, converted to dsDNA and cloned into a recipient plasmid using a Golden Gate reaction. The recipient plasmid contains a *ccdB* gene, which is toxic to sensitive *E. coli* strains, to eliminate unbarcoded plasmids from the library. 2) After purification, the plasmid is electroporated into *E. coli*, and 3) the cells are recovered in a thin layer of media containing soft agar. After growth, the soft agar is shaken to homogenize the culture and the plasmid is extracted and purified. **b.** Barcoding the landing pad strain. 1) Yeast cells containing a landing pad are grown and transformed with the plasmid library to initialize the barcoding process. Cells are recovered on selective media containing soft agar. After growth, the culture is homogenized, and 2) passaged to galactose-containing media to induce Cre. Barcoding takes place by double recombination between orthogonal Lox sites, which inserts barcodes into the genome and swaps selectable drug markers. The plasmid, no longer containing a barcode, is removed by 5-FOA selection, and 3) cells are transferred into the evolution environment. After 100 generations in the evolution environment, this process of: 4) transformation, and 5) Cre/lox recombination, is repeated to introduce additional barcodes. After each subsequent evolution interval, the barcoding procedure is repeated (alternating between the even and odd protocols).



Supplementary Figure 4: **Statistics of barcoded lineages immediately after barcoding.** In the top panel, dots denote mean numbers of reads per barcoded lineage immediately after barcoding, and error bars denote 95% confidence intervals (n = number of unique new lineages, which is shown in the bottom panel using a separate bar for each barcoding procedure and population). All confidence intervals include 0 reads (i.e. lineages not seen in the first time-point, but seen in a subsequent timepoint), which cannot be shown on the logarithmic axis. In the bottom panel, bars represent the total number of new barcoded lineages seen in the epoch.

1.5 Barcode sequencing

Genomic DNA from cell pellets was extracted using zymolyase mediated cell lysis (5mg/mL Zymolyase 20T (Nacalai Tesque), 1M Sorbitol, 100mM Sodium Phosphate pH 7.4, 10mM EDTA, 0.5% 3-(N,N-Dimethylmyristylammonio)propanesulfonate (Sigma, T7763), 200µg/mL RNase A, and 20mM DTT), and binding on silica mini-preparative columns (IBI scientific, IB47207) with guanidine thiocyanate (4 volume of 100mM MES pH 5, 4.125M Guanidine Thiocyanate, 25% isopropanol, and 10mM EDTA). 3mL of cells eluted into 100µL of elution buffer (10mM Tris pH 8.5) routinely provided about 3µg of total DNA.

PCR of the barcodes was performed using a two-stage procedure previously described to attach unique-molecular identifiers to PCR fragments [26]. To avoid any loss of resolution from this process, we aimed to “tag” at least $\sim 10^5$ genomes with this PCR, corresponding to our daily dilution bottleneck size. We found that modestly sized primers are routinely carried over in PCR clean-up kits, which would prevent the first-round primers from acting as unique molecular identifiers. To avoid this problem, we used AMPure beads (Beckman Coulter) in place of silica-based cleanup kits; we found the former to be multiple orders of magnitude more efficient at removing primers. To obtain the necessary molecular counts, we found that Q5 polymerase (New England Biolabs) is the most sensitive polymerase (when compared to Phusion (Thermo Scientific), OneTaq (New England Biolabs), Phire (Thermo Scientific), Kapa HiFi (KAPA Biosystems) and SuperFi (Thermo Scien-

tific)) and that a carrier is necessary to achieve maximum performance with the AMPure beads. We used pure *E. coli* ribosomal RNA for this purpose at a final concentration of about 50ng/ μ L of RNA in the AMPure bead cleanup reaction, which significantly increases the yield of the desired fragment.

Primers used in the first stage PCR contained a priming sequence, a multiplexing index, 8 random nucleotides as unique molecular identifiers (UMIs), and an overhang that matched the Tn5 transposomes used in the standard Nextera (Illumina) sequencing kits. Following recommendations when sequencing amplicons [27], we used multiplexing indexes of different length (7 to 12bp), which generates base diversity for next-generation sequencing. The two primers had the configurations: P1=TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGNNNNNNNNXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX and P2=GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGNNNNNNNNXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX. Here N corresponds to degenerate bases used as unique molecular identifiers, X corresponds to multiplexing indexes, and Y to the annealing sequence.

The barcode locus increases in size with each barcode addition, quickly outgrowing the typical length of reads in next-generation sequencing. However, we take advantage of the fact that it is not necessary to sequence the complete locus because barcodes corresponding to previous epochs are uniquely associated with their descendant barcodes. Since we can sequence the previous epochs directly, the genealogy can be recursively reconstructed. Because barcodes are inserted adjacent to each other (separated by Lox sites), we can sequence up to four barcodes in each paired-end sequencing read. Using the custom primers introduced in the barcoding plasmids (KanP1 to KanP6, HygP1 to HygP6), the barcode of the current epoch is sequenced along with two or three of its ancestral barcodes (except for the first epoch, where only one ancestral barcode, corresponding to the population specific barcode, is sequenced in addition to the epoch barcode). Due to the high barcode diversity in the experiment, these ancestral barcodes allow unambiguous assignment of the current barcodes to the complete genealogy of the lineage (see below for details).

We describe here the final optimized procedure. Briefly, about 300ng of genomic DNA (about 10^7 genomes) was used in 20 μ L PCR reactions using Q5 polymerase. Only two cycles were performed to decrease any potential PCR bias during subsequent counting procedures. In all cases, we use an annealing temperature of 50°C (about 5°C lower than the suggested Tm) to ensure that all priming sites are saturated. The products of this first-stage PCR were then supplemented with 2 μ g of purified ribosomal RNA from *E. coli* as precipitation carrier (20 μ L of 100ng/ μ L was added directly to the first reaction product), purified using AMPure beads at 1.25x, and eluted into 33 μ L of water. The elution was used directly as template for the second stage PCR using KAPA HiFi polymerase with 35 amplification cycles and primers that contained multiplexing indexes and adapters that anneal to the Illumina flowcells. The two primers here had configurations: P5=AATGATACGGCGACCACCGAGATCTACACXXXXXXXXTCGTCGGCAGCGTC and P7=CAAGCAGAAGACGGCATACGAGATXXXXXXXXGTCTCGTGGGCTCGG. These final PCR products were then purified using AMPure beads at 0.85x. We found that this procedure amplifies at least 10^6 unique templates.

After quantification, PCR products were pooled to equimolar concentration and bead purified once more using AMPure beads at 0.85x to remove all traces of primer dimers. The PCR products were then sequenced using a NextSeq 500 high-output v.2 (Illumina) without further modifications. We sequenced to an average depth of approximately 5×10^5 reads per timepoint.

2 Read parsing, error correction, UMI removal, assembly, and chimera removal

We first processed our raw sequencing reads to identify and extract the barcode sequences, discarding Lox sites and other extraneous sequence. To do so, we developed custom Python scripts using the approximate regular expression library `regex`, which allowed us to handle complications in identifying the barcodes that arise from the irregular lengths of the indices and from sequencing errors. We used the following mismatch tolerances: 2 mismatches in the multiplexing index, 4 mismatches in the priming site, 1 mismatch in the barcode overhangs, 1 mismatch in the barcode spacers and 4 mismatches in the Lox sites.

This initial processing results in a set of putative barcodes. However, these putative barcodes do not all correspond to true barcodes within the evolving population. Instead, each true barcode will result in many exact reads but also many artifactual barcodes that are close to the true barcode but contain some sequencing errors. Ideally, we wish to “merge” barcode sequences that represent sequencing errors of a single true barcode, while at the same time avoid erroneously merging two true barcodes. To achieve this, we can exploit the diversity of the barcode library and the fact that we expect sequencing errors to be rare. By taking advantage of the $\sim 10^7$ sequencing reads we obtained that correspond to the known population-specific barcodes, we found that about 0.1% contain a single mismatch while 0.01% contained two mismatches. Hence we expect that allowing two mismatches should capture $\sim 99.9\%$ of reads corresponding to a true barcode while removing the vast majority of artifactual barcodes. Based on the diversity of our barcode library, the chance that two true barcodes randomly picked from our library differ by two or fewer mismatches is 0.2%. This represents an upper bound on the erroneous merging introduced at this threshold distance; in practice we can reduce this by exploiting frequency information and requiring that putative artifactual barcodes are rare compared to the true ones. In addition, because our identification of lineages based on genealogical assembly using ancestral overlapping barcodes uses the information of at least two previous epochs (see below), erroneous merging of true barcodes will not affect any further analysis unless these barcodes also share these ancestral barcodes, which is extremely unlikely.

Motivated by this, we developed an algorithm that deterministically builds consensus sequences using read counts; we refer to this as “error correction”. Candidate barcode sequences are sorted and processed by descending read counts. Barcode sequences within two Levenshtein distance units from a previously seen barcode are merged with that barcode if they were found at much lower frequency (here set to $1/32$). Finally, barcodes that could not be error corrected but that were present in fewer than 10 reads in the whole epoch were discarded. This error correction procedure was performed independently for each epoch. We note that this method is often conservative in merging barcodes; we often found barcodes at similar frequencies that differ by a single mismatch, where one is clearly derived from a sequencing error due to the presence of the same parent barcodes. We merge these cases in a later step using population genetics inference procedures (see below).

We next calculated the frequency of each error-corrected true barcode. To avoid double counting reads, we took advantage of the unique molecular identifiers, and removed any duplicate reads. Overall, our duplicate read rate was low (about 10%), indicating high coverage of the population. We then removed signatures of chimeric PCR, by requiring lineages to have a read in at least two time-points, or to have at least ten reads in the epoch. Although this does not remove all chimeras, any remaining chimeras are likely to be present at very low frequencies and would be considered as effectively neutral in our population genetics analyses. These modest thresholds for removing chimeras can in principle also remove real lineages, and therefore may affect calculated frequencies.

However, we found that although removing chimeras removes about 90% of all lineages, these lineages collectively only account for a small fraction of all reads (about 10%) and are therefore unlikely to affect the overall inference of the evolutionary dynamics.

As explained previously, we can only sequence four barcodes in a single paired-end Illumina read. However, larger barcode loci can be assembled from the shorter reads by taking advantage of the high barcode diversity. In practice, we only sequence the four most recently added barcodes and reconstruct the complete lineage history by “aligning” the older two barcodes to previously identified lineages. For example, given a pair of reads with barcodes A-B in the first read and C-D in the second read (the paired-end read is thus A-B-C-D), we assemble the paired-end read using a perfect A-B match to the previously assembled barcode genotypes (ending in A-B). Sometimes the B barcode does not appear in the previously assembled lineages, presumably due to being at low frequency at that time. In this case, only the A barcode is used. Any read that does not have a perfect match to A is discarded regardless of whether we can identify a matching B barcode; these reads are very rare and typically correspond to cross-indexing from other populations. Additionally, in rare cases, two different previously identified lineages will end with A-B, presumably by obtaining barcodes A and B in parallel. In these cases, the parent of A-B-C-D is ambiguous and we assign it to the A-B lineage with the highest frequency.

Following this procedure, we obtained on average 3.6×10^5 barcoded lineage counts per time-point. This effective depth of sequencing was chosen to be roughly equivalent to the size of the population at each transfer bottleneck, which we estimate to be about 10^6 individuals, and is sufficient to resolve any lineage large enough that its dynamics are potentially affected by natural selection (as opposed to being dominated by genetic drift; note that this assumes that selection pressures are typically of order a few percent or less, as we observe, see Section 4.3).

3 Fitness measurements using fluorescent labels

The fitness of populations during the evolution was measured using competitive fitness assays relative to reference strain YAN518, a fluorescent unbarcoded ancestor (marked with *can1::RPL39pr_ym GFP::Ste2pr_SpHIS5::Ste3pr_LEU2* instead), using the protocol described in previous work [28]. To confirm that the addition of barcodes does not affect fitness, we barcoded this ancestor 11 times, bottlenecking to three clones at each barcoding step. We found no detectable fitness difference between the ancestor and the final three clones that have 11 barcodes each.

4 Construction of a model for barcode count trajectories

The obtained trajectories of read counts of all barcoded lineages reflect the effects of selection on the lineages, as well as both the noise that is inherent in the growth and transfer cycle, and the noise arising from sample preparation and sequencing. In this section, we introduce a minimal model of this process, which we use to distinguish these stochastic effects from the effects of selection, and to infer the fitnesses and times of mutational events from barcode sequence data.

Suppose that immediately following a transfer after generation t , the population consists of N_b individuals, which carry m unique barcode arrays, and that $n_{i,t}$ individuals carry barcode array i , which we will refer to as lineage i . Over the course of the following growth cycle, the population will grow in size as the individuals in the population reproduce, reaching a final size of about $2^{\Delta t} N_b = 2^{10} \cdot N_b$, when a subset of N_b cells will be randomly sampled and transferred into fresh media. Thus, the average individual transferred at the beginning of the day will have left behind 2^{10} offspring by the end of the day. However, mutations arising over the course of evolution will

lead to selective differences that result in some individuals contributing more offspring than others. We denote the average absolute competitive fitness of lineage i by x_i , defined so that immediately before the next transfer the expected number of individuals in the lineage $n'_{i,t+\Delta t}$ is equal to

$$\langle n'_{i,t+\Delta t} \rangle = e^{x_i \Delta t} \cdot 2^{\Delta t} n_{i,t}. \quad (1)$$

Of course, if lineage i is not genetically homogeneous, x_i will change over time, but for small enough lineages and on short enough timescales, it can be considered a convenient effective parameter. In addition, noise in the growth cycle (e.g. in the duration of lag times at the beginning of the transfer) may lead to variance in $n'_{i,t+\Delta t}$ (see Ref. [12]), which we can also model explicitly. During the transfer, the probability that an individual from lineage i is sampled is proportional to the frequency of lineage i , and so the distribution of the sizes of the lineages immediately after the transfer, $n_{i,t+\Delta t}$, follows a multinomial distribution

$$p(\{n_{i,t+\Delta t}\}) = \frac{N_b!}{\prod_i n_{i,t+\Delta t}!} \prod_i \left(\frac{n'_{i,t+\Delta t}}{\sum_j n'_{j,t+\Delta t}} \right)^{n_{i,t+\Delta t}}. \quad (2)$$

The frequencies of all of the lineages are resampled multiple times as the DNA is extracted, PCR amplified and sequenced. In principle, we can model each of these steps explicitly as a multinomial sampling step analogous to the one in Equation 2, with N_b replaced with an appropriate bottleneck size for the sampling step. By integrating over all of the hidden variables (the numbers of individuals or reads at each sampling step), it would in principle be possible to calculate the likelihood of all of the observed read trajectories conditioned on the x_i and the bottleneck sizes, and therefore to infer the fitnesses of all of the lineages. However, because this likelihood does not have an analytic form, this is a computationally demanding problem even in the simplest case of a single biallelic marker (i.e. two lineages) and a single source of binomial sampling noise. This has inspired a large number of authors to look for computationally efficient approaches to inferring selection coefficients and population sizes that often rely on the use of dynamic programming algorithms and application-dependent approximations [29–40].

In our case, computational efficiency is essential since we need to jointly infer the fitnesses of hundreds of thousands of barcoded lineages. However, our task is greatly simplified by the fact that the majority of our barcoded lineages are present in the population at very small frequencies, which makes it possible to treat the fluctuations in the sizes of different lineages introduced in the multinomial sampling steps as independent. Thus, we can treat the lineage trajectories as independent branching processes that interact only via the mean fitness, $\bar{x}(t)$. In the next section, we describe how the noise in the growth/transfer and sequencing steps combines to determine the parameters of this branching process. This derivation closely resembles the approach of Ref. [12]. We restate the assumptions and necessary intuition briefly for completeness. We then proceed to describe how the parameters of the branching process can be inferred from the data itself. Finally, we describe our approach to jointly inferring the fitnesses of all the barcoded lineages, and therefore also the mean fitness of the population over time.

4.1 Branching process model for the lineage read count trajectory

In the limit that the frequency of a lineage is small, each of these sampling steps can be modeled by a branching process. Under relatively mild assumptions, this branching process is well-characterized by the mean and variance in the numbers of offspring per individual, independent of the actual distribution of offspring per individual. These assumptions essentially require that fitness differences

in the population are smaller than $\mathcal{O}(100\%)$, and that the distribution of offspring per individual has a tail with a finite second moment, which are satisfied by our experimental system.

Suppose for concreteness that each individual in a lineage i of $n_{i,t}$ individuals is expected to yield on average a_i descendants at the beginning of the next transfer cycle, with a variance of $2a_i b$ ($\approx 2b$) individuals. The parameter b therefore characterizes the overall noise in the transfer cycle. If this noise is dominated by the stochasticity of resampling, we expect that $b = 0.5$, corresponding to a Poisson process, but any noise occurring during the lag, growth or saturation phases will tend to increase b and lead to over-dispersion in the allele frequency trajectory. Given these assumptions, standard methods can be used to derive the probability distribution of the number of individuals in the lineage after the transfer, $n_{i,t+\Delta t}$ (see, e.g., Supplementary Information of Ref. [12] for a context-specific review). As long as $n_{i,t+\Delta t}$ is not too small, its probability density is well approximated by

$$p[n_{i,t+\Delta t}|n_{i,t}] \approx \sqrt{\frac{(a_i n_{i,t})^{1/2}}{4\pi b n_{i,t+\Delta t}^{3/2}}} \exp\left[-\frac{(\sqrt{n_{i,t+\Delta t}} - \sqrt{a_i n_{i,t}})^2}{b}\right]. \quad (3)$$

Suppose that the population from generation $t + \Delta t$ is then sequenced, so that an overall number of reads $R_{t+\Delta t}$ are recovered, and that the DNA extraction, PCR and sequencing process adds an additional variance of 2β for every sequenced read in the lineage. We can model this measurement step using an analogous branching process. The expected number of reads mapping to this lineage $r_{i,t+\Delta t}$ is simply

$$\langle r_{i,t+\Delta t}|n_{i,t+\Delta t} \rangle = \frac{n_{i,t+\Delta t}}{N_b} R_{t+\Delta t}, \quad (4)$$

and the variance in this number is

$$\text{var}(r_{i,t+\Delta t}|n_{i,t+\Delta t}) = 2\beta \langle r_{i,t+\Delta t}|n_{i,t+\Delta t} \rangle, \quad (5)$$

with the overall distribution given by

$$p[r_{i,t+\Delta t}|n_{i,t+\Delta t}] \approx \sqrt{\frac{\langle r_{i,t+\Delta t}|n_{i,t+\Delta t} \rangle^{1/2}}{4\pi\beta r_{i,t+\Delta t}^{3/2}}} \exp\left[-\frac{(\sqrt{r_{i,t+\Delta t}} - \sqrt{\langle r_{i,t+\Delta t}|n_{i,t+\Delta t} \rangle})^2}{\beta}\right], \quad (6)$$

as long as $r_{i,t+\Delta t}$ is not too small. Expressions analogous to Equation 4, Equation 5 and Equation 6 characterize the distribution of reads at the previous time-step, $p[r_t|n_{i,t}]$. By combining these expressions, we can calculate the probability of observing $r_{i,t+\Delta t}$ reads at generation $t + \Delta t$, conditioned on having observed $r_{i,t}$ reads in the previous sequencing time-point:

$$p[r_{i,t+\Delta t}|r_{i,t}; R_{i,t+\Delta t}, R_t] = \sqrt{\frac{\langle r_{i,t+\Delta t}|r_{i,t} \rangle^{1/2}}{4\pi\kappa_t r_{i,t+\Delta t}^{3/2}}} \exp\left[-\frac{(\sqrt{r_{i,t+\Delta t}} - \sqrt{\langle r_{i,t+\Delta t}|r_{i,t} \rangle})^2}{\kappa_t}\right], \quad (7)$$

where we have defined

$$\kappa_t = \beta + \frac{R_{t+\Delta t}}{N_b} b + \frac{R_{t+\Delta t}}{R_t} \beta \quad (8)$$

and

$$\langle r_{i,t+\Delta t}|r_{i,t} \rangle = a_i r_{i,t} \frac{R_{t+\Delta t}}{R_t}. \quad (9)$$

Thus, we have found that the distribution of read counts of a lineage in consecutive timepoints is characterized by two parameters: a single parameter, κ_t , that captures the total noise in the

process, and a parameter, a_i , that quantifies the effect of selection. We will refer to this noise as “drift”, as is customary in the population genetics literature. Note that even if the parameters that characterize the amount of noise present in the growth and transfer cycle (b) and sample preparation (β) are constant over the course of experiment, $\mathcal{O}(1)$ variation in read depth from time-point to time-point will lead to variation in κ_t .

As we explain in the previous section, a_i is related to the fitnesses of all of the lineages x_j according to

$$a_i = \frac{e^{x_i \Delta t}}{\sum_j f_{j,t} e^{x_j \Delta t}}, \quad (10)$$

where $f_{j,t} = \frac{n_{j,t}}{N_b}$ denotes the frequency of lineage j at time t , which we estimate as $f_{j,t} \approx \hat{f}_{j,t} = \frac{r_{j,t}}{R_t}$. Note that when the fitness differences between lineages lead to small per-interval increments ($(x_i - x_j)\Delta t \ll 1$), Equation 10 can be rearranged to yield

$$a_i \approx \frac{e^{x_j \Delta t}}{\sum_j f_{j,t} (1 + x_j \Delta t)} \approx \frac{e^{x_j \Delta t}}{1 + \bar{x}(t) \Delta t} \approx e^{(x_j - \bar{x}(t)) \Delta t}, \quad (11)$$

where we have used $\bar{x}(t)$ to denote the mean fitness of the population at time t . We will see that this is a good approximation for the majority of intervals during the experiment, and we will explain its use below in some steps in which we aim only to obtain rough estimates of the fitnesses of the lineages relative to the mean, $x_j - \bar{x}(t)$, though we will use the unapproximated form in Equation 10 when obtaining our final estimates of the fitnesses of all of the lineages.

4.2 Modeling drift and selection during the barcoding procedure

Thus far we have described our model with explicit reference to the experimental evolution protocol. However, as described in Section 1.3 above, the barcoding procedure involves a number of intermediate growth and dilution steps during which the population is resampled in a number of different environments. This means that each barcoding also involves effects of natural selection and genetic drift. Thus the populations experience what is, strictly speaking, an environment that fluctuates between an “evolution” condition and a “barcoding” condition. It is not clear *a priori* what effect these fluctuations will have on the overall evolutionary dynamics. To assess this, we need to quantify in detail the effects of drift and selection during each barcoding; we will later use these results in Section 6.2 to show that the effect of barcoding is relatively modest and we can consider the evolution as occurring in an “average” environment.

To quantify the effects of barcoding, we use an analogous model to that described above. Just like growth/transfer and sequencing in the evolution environment, each intermediate step in the barcoding procedure can be modeled as a branching process in which each lineage has a fitness which quantifies its tendency to increase or decrease in size compared to other lineages in the population, and a parameter that quantifies the variance in the distribution of offspring that lineages leave at the end of that step. Finally, for small lineages and as long as the effect of selection during barcoding is not too strong, the net effect of selection and drift during this procedure can also be quantified using an effective branching process.

Though it is not possible to sequence the population during these intermediate steps, we can quantify the cumulative effect of drift and selection during the barcoding procedure. In analogy to the quantities κ_t and x_i defined for the evolution period above, we denote the parameter characterizing the overall amount of drift during barcoding interval following timepoint t as $\kappa_t^{(\text{bc})}$, and the overall per-cycle “barcoding fitness” of lineage i as $x_i^{(\text{bc})}$. Thus $x_i^{(\text{bc})}$ is analogous to the product of the “evolution” fitness and the number of generations that elapse between two sequencing

timepoints, $x_i \Delta t$. Note that this product, rather than the per-generation fitness, is also the only observed quantity in the evolution condition. However, because the number of doublings per individual is precisely known during the evolution condition, we divide the observed per-cycle fitness with this number of doublings to report a per-generation fitness, as is conventional. Given the complexity of the growth during the barcoding procedure, we do not attempt to quantify the number of generations that elapse during barcoding.

Before we explain our approach to inferring the parameters of the branching process, κ_t and $\kappa_t^{(\text{bc})}$, and the fitnesses of the barcoded lineages, $\{x_i\}$ and $\{x_i^{(\text{bc})}\}$, we begin in the next section by outlining some key intuition about the relative importance of drift and selection on the read trajectory of a lineage. We will return to this intuition repeatedly in later sections when describing our inference procedure.

4.3 The relative importance of drift and selection

An important property of branching processes of the form we are considering here is that at sufficiently low counts, trajectories are almost entirely dominated by drift. The minimal size that a lineage needs to reach to be impacted by the effect of selection can be calculated by considering the timescales on which drift and selection can substantially impact the allele frequency trajectory. This can be done formally by analyzing how the distribution of lineage counts changes over multiple sampling steps starting from Equation 3. However, since this is a standard result [41, 42], we do not reproduce the proof here, but only quote the result that the effects of drift are expected to dominate over the effects of selection in lineage i on timescales of $\frac{1}{x_i - \bar{x}}$ generations (or $\frac{1}{(x_i^{(\text{bc})} - \bar{x}^{(\text{bc})})}$ barcoding procedures) whenever the lineage consists of fewer than $\frac{2b}{(x_i - \bar{x})\Delta t}$ individuals at the bottleneck. As in the section above, b parametrizes the amount of noise in the growth and transfer cycle and excludes the sequencing noise. Note that even when the lineage exceeds a size of $\frac{2b}{(x_i - \bar{x})\Delta t}$ individuals, the effects of drift may dominate on timescales shorter than $\frac{1}{x_i - \bar{x}}$ generations.

The threshold of $\frac{2b}{(x_i - \bar{x})\Delta t}$ individuals at the bottleneck corresponds to a threshold number of reads mapping to a lineage equal to $\frac{2bR_t}{N_b} \cdot \frac{1}{(x_i - \bar{x})\Delta t}$ (or $\frac{2b^{(\text{bc})}R_t}{N_b^{(\text{bc})}} \cdot \frac{1}{(x_i^{(\text{bc})} - \bar{x}^{(\text{bc})})}$ reads in the barcoding condition). Thus, lineages not exceeding $\frac{2bR_t}{N_b} \cdot \frac{1}{(x_i - \bar{x})\Delta t}$ counts have trajectories that will be dominated by drift on timescales of $\frac{1}{x_i - \bar{x}}$ generations (or $\frac{1}{x_i^{(\text{bc})} - \bar{x}^{(\text{bc})}}$ barcoding procedures). On shorter timescales, this threshold is conservative, both because drift can have a stronger effect than selection on shorter timescales, and because sequencing contributes an additional source of noise.

We will see in the following section that in both populations we typically have $\frac{2bR_t}{N_b} \gtrsim 1$ and $\frac{2b^{(\text{bc})}R_t}{N_b^{(\text{bc})}} \gtrsim 10$. Because the variance in fitness in our populations is on the order of $\sigma^2 = 10^{-4}$ or less in both the YPD and the YPA evolution condition and $(\sigma^{\text{bc}})^2 \lesssim 0.5$ (see Supplementary Fig. 14), this means that the majority of lineages in our experiments that are present at less than a few multiples of $\frac{2bR_t}{N_b} \cdot \frac{1}{\sigma \cdot \Delta t} = 10$ counts will have trajectories dominated by drift. Note that because new barcodes appear at average read depths of about 10 counts per lineage (Supplementary Fig. 4), this has important consequences. First, it means that it is in principle impossible to infer the fitnesses of very low-frequency lineages before they reach higher frequencies. However, it also makes it possible to use these low-frequency barcodes to infer the parameters of the branching process κ_t and $\kappa_t^{(\text{bc})}$ at frequencies where drift is the dominant force. We also note that this result implies that given the strength of selection observed in our experiments, our barcoding sequencing depth of on average 3.6×10^5 per timepoint is adequate to resolve any lineage whose dynamics are dominated by selection rather than drift.

4.4 Inference of the parameters of the branching process model

To infer the parameter κ_t for each pair of consecutive time-points in both the evolution and barcoding condition, we compile pairs of counts $(r_{i,t}, r_{i,t+\Delta t})$ of all lineages i for which $r_{i,t}$ is in the range $(r_{\min}, r_{\max}) = (40 - 2 \cdot \sqrt{40}, 40 + 2 \cdot \sqrt{40})$. Thus, we simultaneously exclude very low counts for which the distribution in Equation 7 is inaccurate, and also counts over ~ 50 , because this ensures that the overwhelming majority of included data points are not significantly impacted by selection. We fit κ_t to this observed distribution of pairs of read counts, by finding the κ_t that maximizes the likelihood of the pairs of counts under the model given by Equation 7, with $a = 1$, and estimate the error in the estimate of κ_t from the curvature of the second derivative of the likelihood. This likelihood can be analytically maximized, yielding:

$$\kappa_t = 2 \frac{\sum_i (\sqrt{r_{i,t+\Delta t}} - \sqrt{r_{i,t}})^2 \theta(r_{i,t} - r_{\min}) \theta(r_{\max} - r_{i,t})}{\sum_i \theta(r_{i,t} - r_{\min}) \theta(r_{\max} - r_{i,t})}, \quad (12)$$

$$\Delta\kappa_t = \kappa_t \cdot \sqrt{\frac{2}{\sum_i \theta(r_{i,t} - r_{\min}) \theta(r_{\max} - r_{i,t})}},$$

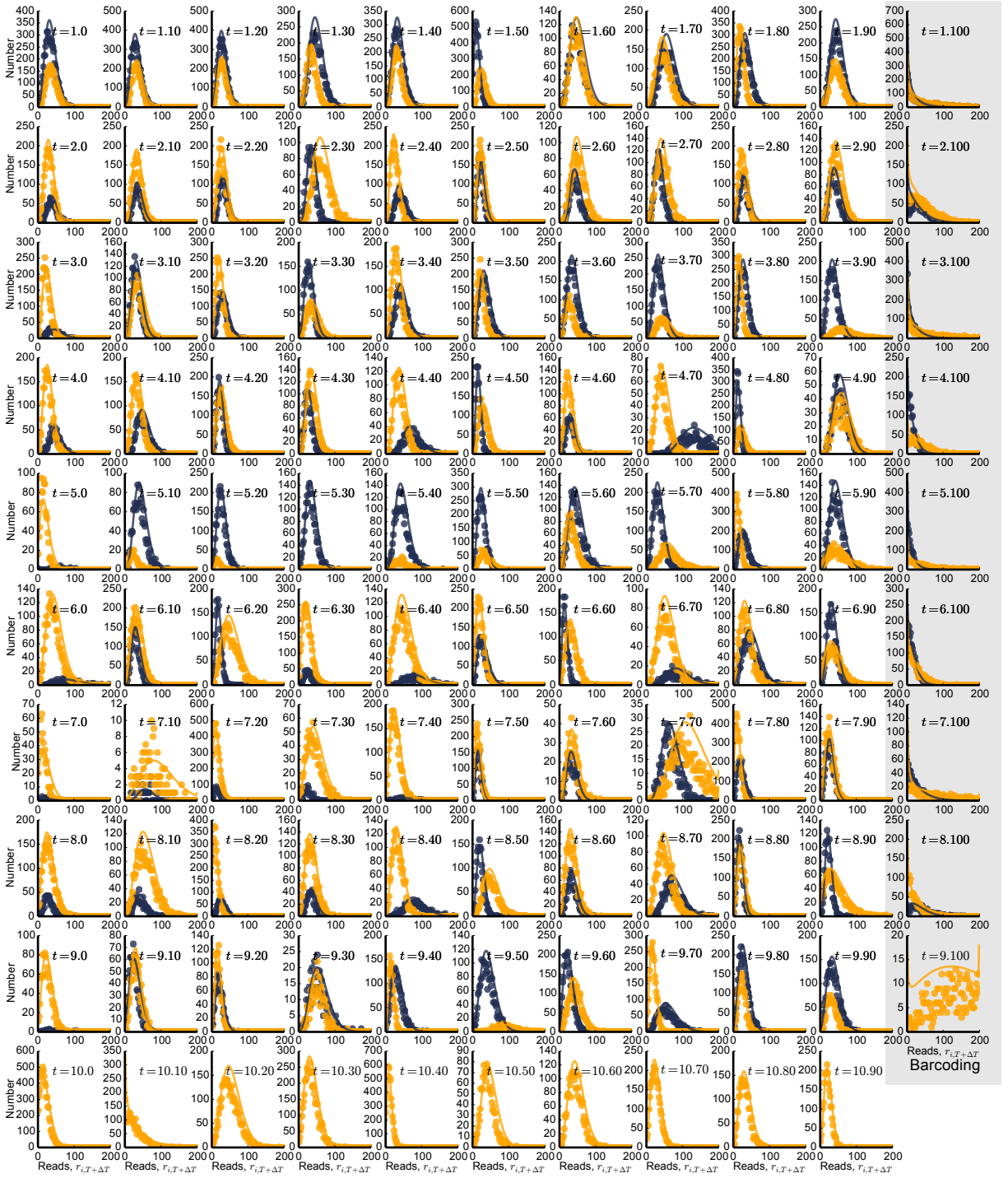
where $\theta(x)$ denotes the Helmholtz theta function and the summation runs over all lineages. We take 95% confidence intervals for the κ_t to be $(\kappa_t - 2\Delta\kappa_t, \kappa_t + 2\Delta\kappa_t)$.

Supplementary Figure 5 shows a comparison between the observed distribution and the distribution predicted using our maximum likelihood κ_t . As can be seen in the left panel of Supplementary Fig. 6, though κ_t varies over the course of the experiment, there are no strong systematic trends. Using these estimates of the κ_t , we can fit the model in Equation 8 and obtain estimates for the variance per individual that are introduced during the growth and transfer cycle ($2b$) and during DNA extraction and sequencing (2β), assuming that these are constant throughout the experiment. We fit these parameters separately for each population and condition (evolution and barcoding), and summarize the best-fit parameters and confidence intervals Supplementary Table 2.

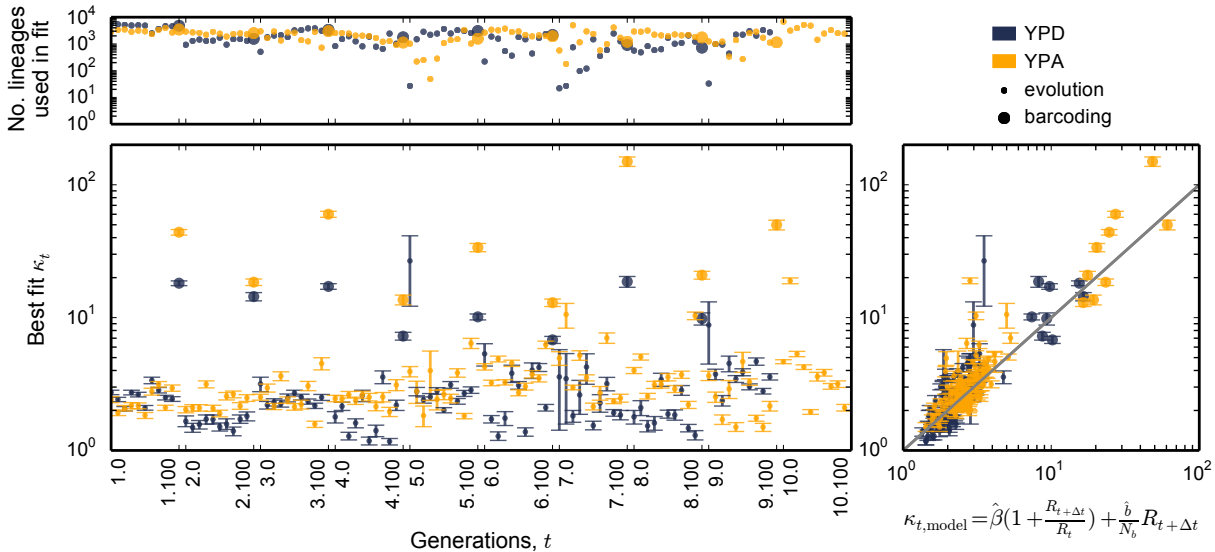
The model in Equation 8 provides a reasonable fit (see right panel of Supplementary Fig. 6). The estimates of $\frac{2b}{N_b}$ in the evolution conditions are consistent with our intuition that the variance per individual introduced during the growth-transfer cycle, $2b$, is of order 1 and that the size of the population at the bottleneck is roughly 10^6 in the YPD population and slightly smaller in the YPA population.

In the barcoding environment, the estimates of $\frac{2\hat{b}}{N_b}$ are consistent with an effective bottleneck that is 50-fold lower than the daily transfer bottleneck in the YPD evolution condition, and slightly less than 10-fold lower than the daily transfer bottleneck in the YPA condition. Note that these bottlenecks are broadly consistent with what we expect them to be from the barcoding protocol; they lead to an overall scale of genetic drift which is comparable to the combined effects of the ten daily transfers during each evolution epoch.

Finally, the best-fit parameters are consistent with the noise added during DNA extraction and sequencing having slightly larger than Poisson variance, and are in rough agreement with one another given the precision of the estimates, with the exception of the YPA barcoding condition, which appears slightly larger. There are multiple possible reasons for this discrepancy. For instance, the model for κ_t in Equation 8 is clearly simplistic in assuming that all sequencing runs are equivalent. However, DNA extraction and sequencing are a composite procedure, consisting of multiple steps including a PCR reaction during which a variable number of molecules may be recovered for different timepoints, introducing a variable PCR bottleneck. This would lead to additional variability in the parameter β that is not captured by our model. However, these discrepancies are not relevant for our purposes here since we are not interested in characterizing the details of the origins



Supplementary Figure 5: **Comparison of observed and best-fit theoretical distributions of read counts.** Each panel shows the distribution of read counts $R_{t+\Delta t}$ of lineages that were observed to have between $40 - 2\sqrt{40}$ and $40 + 2\sqrt{40}$ reads at timepoint t . Lines represent the model prediction based on Equation 7 with the maximum likelihood κ_t . Blue points and lines correspond to the YPD population, and orange corresponds to YPA, consistent with Figure 4 of the Main Text.



Supplementary Figure 6: **Inferred parameters of the branching process.** (Left) Best estimates of κ_t over the course of the experiment. Error bars show 95% confidence intervals (calculated as described in Equation 12 and the text immediately below). The top left panel shows the total sample size used to estimate κ_t in each population and timepoint, which is equal to the number of lineages of the appropriate size. (Right) Variation in these estimates of κ_t is partly explained by the variation in read depth over the course of the experiment. Estimates of $\hat{\beta}$ and $\frac{\hat{b}}{N_b}$ were obtained by weighted-least squares regression. The weights for data point were estimated from the uncertainty in the best fit κ_t , $\Delta\kappa_t$.

of the drift and sequencing noise during the experiment. Instead, we are primarily interested in quantifying its net effect using a single parameter, κ_t , which we can use to distinguish its effects from the effect of selection.

5 Inference of lineage fitness and the arising times of new mutations from barcode count trajectories

Having established a model for the trajectory of a lineage, we are now in a position to infer the fitnesses of all the barcoded lineages in a population in the evolution and in the barcoding condition. Note that this is still a challenging problem because at any point during the experiment, a majority of the recently barcoded lineages are segregating at such low frequencies that their trajectories are primarily impacted by drift. Thus, if we were to subdivide the population at the finest possible level (using all the available barcode information), we would not be able to infer the fitnesses of the majority of lineages in the population. This would lead to an underestimation in the fitness variance and, consequently, the mean fitness of the population $\bar{x}(t)$. Since the expected trajectory of a lineage only depends on its fitness relative to the mean, this would also lead to the underestimation of the fitnesses of the lineages that *are* large enough to feel the effect of selection.

Fortunately, because lineages retain all barcodes that they receive over the course of evolution, we have abundant genealogical information that we can use to group sublineages according to their earlier barcodes. Because barcodes are inserted into the population at a rate that is far higher than

condition	$\frac{2\hat{b}}{N_b}$	$2\hat{\beta}$
YPD evolution	2.3×10^{-6} (2.16, 2.43) $\times 10^{-6}$	1.69 (1.66, 1.72)
YPD barcoding	1.16×10^{-4} (1.06, 1.27) $\times 10^{-4}$	1.79 (0.61, 2.97)
YPA evolution	6.37×10^{-6} (6.05, 6.69) $\times 10^{-6}$	1.48 (1.41, 1.54)
YPA barcoding	4.48×10^{-5} (3.32, 5.64) $\times 10^{-5}$	12.48 (9.58, 14.3)

Supplementary Table 2: **Inferred parameters of the branching process.** The best-fit parameters of the model in Equation 8, as obtained by weighted least squares regression. Confidence intervals for each of the parameters are provided in parentheses.

we expect beneficial mutations to establish (10^5 barcodes every 100 generations), the majority of barcoded lineages will not obtain a new beneficial mutation between consecutive barcoding “epochs”. This means that by grouping sublineages according to earlier (“parental”) barcodes, we will largely recover sub-populations that carry the same beneficial mutations, and are therefore homogeneous in fitness. This allows us to average over the drift impacting the trajectories of individual sublineages, and obtain more accurate estimates of fitness.

However, a small fraction of lineages will receive new beneficial mutations between barcoding periods. Once established, these beneficial mutations will cause the “child” sublineage in which they established to have a different fitness than the parental lineage. Thus differences in the fitnesses of parental and child lineages are indicative of new beneficial mutations. Thus by looking for lineages that have fitness larger than their parents, but whose relative benefits compared to the parents are shared by all their children, we can narrow the establishment time of a mutation to within a barcoding epoch (100 generations). This will leave us with a set of all of the beneficial mutations that established in the population.

Note that by using the phylogenetic information encoded in the barcode locus to narrow the establishment time of a mutation, we deviate from previous methods that inferred the establishment time by extrapolating backwards the long-term trajectory to find the effective time when the lineage size exceeds roughly $\frac{1}{(x-\bar{x})}$ individuals [12]. We make this choice for a number of reasons. First, given that typical differences between the fitnesses of lineages in our experiment are of order $x - \bar{x} \approx 1\%$, and that the establishment of a lineage is not a discrete event, but rather an extended period of time of duration of order $\frac{1}{x-\bar{x}} \approx 100$ generations during which the impact of genetic drift becomes less important than the impact of selection [12, 43], the accuracy of single 100-generation-long barcoding epoch is sufficient for describing this process. Second, given that our population is subjected to a sequence of two environments (the evolution and the barcoding environment), it is not obvious what principled approach could be taken to extrapolate backwards the long-term trajectory, and it is also not clear how meaningful any finer resolution of the establishment time that could be achieved in this way would be.

To accomplish these goals of inferring the beneficial mutations and their approximate establishment times, we must first obtain estimates of the fitnesses of all the lineages in the population. We start by identifying all lineages whose trajectories are not primarily impacted by drift at least in some portion of the experiment in either the evolution or the barcoding condition (i.e. those for which there is evidence for selection). We then jointly infer the relative fitnesses of these lineages, and compile a list of those that are *positively* selected in at least some portion of the experiment in at least one of the conditions (evolution or barcoding). These lineages are all likely to contain some beneficial mutations, but multiple of them will reflect the effects of the same co-inherited mutation. We then proceed to identify which of these lineages appear to have acquired a new beneficial mutation that was not present in the parental background. Finally, with this information

we can split the population into “clonal” lineages that do not differ by any beneficial mutations that we can detect. We then jointly infer the fitnesses of these clones with respect to the ancestor independently for the evolution and for the barcoding condition, and therefore the effects of all of the detected mutations in both of the conditions. We describe these steps in detail in the following sections.

5.1 Identification of selected lineages

We begin by first identifying a set of lineages for which we can reliably estimate fitness. To do this, we first split all trajectories into segments (“epochs”) corresponding to the 100-generation evolution periods (between barcoding intervals) and the barcoding intervals (during which we can measure changes in the sizes of the already existing barcoded lineages). We will refer to the evolution period during which each individual carries a total of E barcodes as “epoch E ”, and the barcoding interval during which the $(1 + E)^{\text{th}}$ barcode is added as “barcoding interval E ”. As in the Main Text, we denote generation x of epoch E with “ $E.x$ ”, so that 1.90 represents generation 90 of the first epoch.

Because the changes in the frequency of a lineage between consecutive measurements are independent (conditioned on the fitnesses of all of the lineages in the population), the likelihood of a segment of the trajectory of lineage i in epoch E under the hypothesis that its fitness is equal to $x_i(t)$, conditioned on the fitnesses of all of the other lineages being equal to $x_j(t)$ is

$$\mathcal{L}_{i,E}(\{r_{i,t}\}|\kappa_t, a_i(t)) = \prod_{t=E.0}^{E.90} p(r_{t+\Delta t}|r_t, \kappa_t, R_{t+\Delta t}, R_t; a_i(t)), \quad (13)$$

where $p(r_{t+\Delta t}|r_t, \kappa_t, R_{t+\Delta t}, R_t; a_i(t))$ is given by Equation 7 with $a_i(t)$ given by Equation 10. The likelihood of a segment of a trajectory during barcoding interval E has an equivalent form to Equation 7, with the only difference being that it consists of a single interval, as opposed to the 10 intervals observed during the evolution epoch. In both cases, the likelihood of the segment of the trajectory under the null hypothesis that it is primarily affected by drift can be obtained by setting $a = 1$ in Equation 13. Because Equation 7 is valid only when both $\{r_t\}$ and all hidden variables are not too small, we exclude from further analysis lineage segments that fall below 20 counts in any of the timepoints. Note that this does not exclude the entire lineage, but only those segments that contain extremely low counts.

As mentioned in our discussion above, both the absolute fitness of lineage i , $x_i(t)$, and the absolute mean fitness of the population, $\bar{x}(t)$, will change over time, as new beneficial mutations establish in the lineage and population, respectively, and increase in frequency. Thus, $a_i(t)$ will also change over time. We are interested in modeling these changes explicitly, but we must first identify the subset of these lineages for which fitness (and, therefore, changes in the fitness) can be meaningfully inferred.

To do this, we construct the log-likelihood ratio test statistic, T , for the segment of lineage i in epoch E , defined by

$$T = 2 \log \left[\frac{\mathcal{L}_{i,E}(\{r_{i,t}\}|\kappa_t, a = \hat{a})}{\mathcal{L}_{i,E}(\{r_{i,t}\}|\kappa_t, a = 1)} \right], \quad (14)$$

where \hat{a} represents the (constant) parameter a for which the likelihood of the segment of the trajectory in epoch E (or barcoding interval E) is maximized. We obtain \hat{a} by evaluating the likelihood on a grid of 2001 points between e^{-5} and e^5 , evenly spaced in log-space and then picking the value which maximizes the likelihood. We repeat this procedure and all subsequent steps independently in each epoch E and each barcoding interval E .

Note that if a lineage does not consist of multiple sublineages of different fitnesses and the mean fitness of the population does not change too rapidly (which will be the case for the majority of lineages and epochs), the assumption that a is constant over the course of an epoch is a good approximation for what we believe is the true model for the lineage trajectory. However, even if these criteria are not met, segments that are strongly impacted by the effects of selection will have large T statistics. Therefore, given an expectation for the distribution of T under the null hypothesis and the observed empirical distribution of T , we can identify selected lineages at a specified false discovery rate (FDR) [44].

We construct an empirical null distribution for the evolution condition from the distribution of T -statistics for all the lineages in the first 100 generations of evolution (i.e. in the first epoch) for each population separately, and the empirical null distribution for frequency changes in the barcoding condition from the distribution T -statistics of all the lineages observed over the course of the first barcoding period. These timepoints represent a good proxy for neutral evolution in our system: because the ancestral population is clonal, only new mutations that have already established should have trajectories that are significantly impacted by selection. Given the population sizes ($N_b \approx 10^6$), mutation rates ($U_b \approx 10^{-5}$), and selection coefficients ($s_b \approx 10^{-2}$) relevant for our populations, these events are extremely rare: though on the order of 10 mutations occurring within these first 100 generations are likely to establish, they are extremely unlikely to do so in significantly less than $\frac{1}{s_b} \sim 100$ generations [42].

Thus, for each of the populations separately, we compile the likelihood ratio statistics $\{T_{k,E=1}\}$ of all the trajectory segments of all the lineages k present at the outset of the experiment whose read counts do not fall below a threshold of 20 counts during this epoch. Using these data, we assign P -values for all lineage segments from this population under this empirical null distribution according to

$$P_{i,E} = \Pr [T_{k,E=1} \geq T_{i,E}], \quad (15)$$

independently for all subsequent barcoding periods and evolution epochs. In the next step of the analysis, we need to obtain joint estimates of the fitnesses of all lineage segments that we designate as “selected” based on these P -values.

Because assigning non-negative fitness to a large number of drifting lineages would bias upwards both the mean fitness of the population as well as the fitness estimates of truly selected lineages, we need to limit the total proportion of lineages falsely labeled as selected in order to remain conservative in our detection of new mutations (see also next section). Since the distribution of P -values of lineage segments can be different in different epochs and barcoding intervals, we separately assign Q -values to each of these lineage segments according to

$$Q_{i,E} = \min_{Q > P_{i,E}} \left[\frac{Q \sum_{i'} 1}{\sum_{i'} \theta(Q - P_{i',E})} \right], \quad (16)$$

where θ represents the Heaviside step function. We reject all lineage segments with $Q_{i,E} \geq 5\%$. This threshold is intentionally set to be permissive, since candidate lineages identified in this step undergo substantial further filtering.

5.2 Joint estimation of fitnesses of selected lineages

After identifying lineage trajectory segments that appear to be influenced by selection, we jointly estimate their fitnesses. At this point, the purpose of this estimation is not to obtain accurate measures of the relative fitness of all lineages, but rather to:

1. Conservatively discard lineages whose relative fitnesses are never positive in either the barcoding or the evolution condition. These lineages are unlikely to carry any new mutations, and their trajectories are likely primarily affected by drift, and possibly by negative selection once they fall substantially behind the population mean fitness. They have been picked up in the previous step only because they spent a portion of their trajectory rapidly declining in frequency.
2. Compare estimates of parent and child lineage fitness, which we will then use to infer when mutations arose.

Note that before we are able to correctly partition the population into lineages of the same fitness, we expect to systematically underestimate the fitness variance in the population: as we mentioned above, partitioning the population too finely leads to limited power to infer the fitnesses of small lineages, leading to a large fraction of the population being classified as neutral. On the other hand, partitioning too coarsely and grouping multiple lineages of different fitness into a single genotype directly leads to the underestimation of the variance in fitness within that lineage, and therefore within the entire population. Thus our estimates of the relative fitness of all lineages are likely to be conservative. In order to avoid missing a large number of truly positively selected lineages at the next filtering step, in which we require relative fitness of a lineage to be positive in at least one segment, we aim to be as permissive as possible, while still maintaining simplicity of the procedure.

To infer the fitnesses of all the segments of all trajectories that have not been rejected as neutral in the previous step, we use the following procedure. In all steps, we approximate Equation 10 with the form given in Equation 11, which allows us to quantify the effect of competition between the lineages in the population over the course of an epoch or barcoding period using a baseline common to all (parent and child) lineages, the population mean fitness relative to the beginning of the epoch. This comes at the cost that changes in the mean fitness of the population are poorly approximated during intervals in which there are lineages for which $(x_j(t) - \bar{x}(t))\Delta t \gtrsim 1$, but as we will see, this only affects a handful of barcoding intervals (see Supplementary Fig. 14). Moreover, this level of accuracy is consistent with the overall goals of the procedure. We describe the steps with reference to an evolution epoch E , but we apply an entirely equivalent procedure for all barcoding intervals.

1. Initialize the mean fitness of the population to $\bar{x}(t) = 0$ over the entire course of the epoch.
2. Start by grouping the lineages according to their first barcode only, discarding information about any further barcodes. This represents the coarsest possible way to subdivide the population. Estimate the relative fitness of each lineage $i^{(1)}$ labeled by this first barcode (BC1) by finding the $x_{i^{(1)},E}$ that maximizes the likelihood of the segment of the trajectory of lineage $i^{(1)}$ in epoch E , conditioned on the current estimate of the mean fitness of the population $\bar{x}(t)$ in that epoch (see below). We ignore changes in fitness in the lineage over the course of the epoch and consider the absolute fitness $x_{i^{(1)}}(t)$ of the lineage constant for the duration of the epoch, but allow for it to change between epochs.
3. Estimate the mean fitness of the population in this epoch according to

$$\bar{x}^{(1)}(t) = \sum_{x_{i^{(1)}}} \left[\frac{r_{i^{(1)}}(t)}{R_t} - \frac{r_{i^{(1)}}(t = E.0)}{R_{t=E.0}} \right] x_{i^{(1)},E}, \quad \text{for } E.0 \leq t \leq E.100, \quad (17)$$

with $x_{i^{(1)},E} = 0$ if the segment of this lineage was rejected as drifting in the previous step of the analysis. Because the likelihoods of the trajectories are invariant to the addition of a

constant factor to all of the $x_{i(1),E}$, we make the arbitrary choice to set the mean fitness at the beginning of the epoch to zero $\bar{x}(t = E.0) = 0$. Thus, all relative fitnesses of all lineages are measured with respect to the mean fitness at the beginning of the epoch.

4. If the lineages contain more than 1 barcode in this epoch, repeat this process by grouping the lineages according to their first and second barcodes (BC2) . Estimate the relative fitness of each of these lineages $i^{(2)}$, $x_{i(2),E}$, conditioning on the current estimate of the mean fitness of the population, $\bar{x}(t)$.
5. Re-estimate the mean fitness of the population $\bar{x}^{(2)}(t)$ using the two-barcode fitness estimates and Equation 17.
6. If the mean fitness at the end of the epoch based on this estimate is larger than the previous [$\bar{x}^{(2)}(t = E.100) > \bar{x}^{(1)}(t = E.100)$], this means that we have captured more variation in fitness in the last step than in a previous iteration. In this case, we wish to update the estimate of mean fitness $\bar{x}(t)$ to be equal to $\bar{x}^{(2)}(t)$. Though $\bar{x}^{(2)}(t)$ is still likely to be a conservative estimate of the mean fitness, it is less conservative than our previous estimate, and will lead to less conservative estimates of the fitnesses of lineage segments.
7. Repeat steps (4-6) using any other barcodes (BC3-BCE, if $E \geq 3$).
8. If the mean fitness has been updated since step (2), repeat steps (2-7) until there are no further updates. This ensures that the fitnesses of all lineages were measured relative to the same baseline mean, allowing for comparisons between child and parent lineage fitness.

To estimate the best-fit fitness of lineage i , $\hat{x}_{i,E}$ and its uncertainty, we use two different approaches depending on whether or not the frequency of the lineage is small. If the frequency of a segment of a trajectory does not exceed 10% in a given epoch (i.e. if $\frac{r_i(t)}{R_t} < 0.1$ for the entire duration of the epoch), we estimate $x_{i,E}$ by solving for the $x_{i,E}$ that maximizes the likelihood

$$0 = \frac{d}{dx_{i,E}} \mathcal{L} \left(r_{i,t} | \kappa_t, a = e^{[x_{i,E} - \bar{x}(t)] \Delta t} \right) \Big|_{x_{i,E} = \hat{x}_{i,E}}, \quad (18)$$

with the uncertainty $\Delta x_{i,E}$ given by

$$\Delta x_{i,E} = \frac{1}{\sqrt{-\frac{d^2}{dx_{i,E}^2} \mathcal{L} \left(r_{i,t} | \kappa_t, a = e^{[x_{i,E} - \bar{x}(t)] \Delta t} \right) \Big|_{x_{i,E} = \hat{x}_{i,E}}}}. \quad (19)$$

Equations 18 and 19 admit analytical solutions, which are given by

$$\hat{x}_{i,E} = \frac{2}{\Delta t} \log \left(\frac{-\beta + \sqrt{\beta^2 - 4\alpha\gamma}}{2\alpha} \right) \quad \text{and} \quad \Delta x_{i,E} = \frac{1}{2\sqrt{\left(1 - \frac{\beta}{2\alpha}\right) \left(-\beta + \sqrt{\beta^2 - 4\alpha\gamma}\right)}}, \quad (20)$$

where we have defined

$$\alpha = \sum_t \frac{r_t e^{-\bar{x}(t)\Delta t}}{\kappa_t} \frac{R_{t+\Delta t}}{R_t}, \quad \beta = - \sum_t \sqrt{r_t e^{-\bar{x}(t)\Delta t}} r_{t+\Delta t} \frac{R_{t+\Delta t}}{\kappa_t} \quad \text{and} \quad \gamma = - \sum_t \frac{1}{4}. \quad (21)$$

On the other hand, if $\frac{r_i(t)}{R_t}$ exceeds 10%, we find that convergence is improved if we slightly modify the way the expectation of the trajectory conditioned on $x_{i,E}$ is calculated according to

$$\langle r_t \rangle = r_{t_0} e^{x_{i,E}(t-t_0) - \int_{t_0}^t \bar{x}(t') dt'}, \quad (22)$$

where t_0 corresponds to the beginning of the epoch ($t_0 = E.0$). This change forces better agreement with earlier time-points in the epoch, and comparisons with simulations show that this helps avoid the iterative scheme freezing in a local maximum with smaller overall variance in fitness. Since this likelihood can no longer be maximized analytically, we find $\hat{x}_{i,E}$ by evaluating the likelihood on a grid of 2001 linearly spaced points spanning the interval $[-0.5, 0.5]$ in the evolution condition (and 2001 linearly spaced points between $[-5, 5]$ in the barcoding condition), and taking $\hat{x}_{i,E}$ to be the point at which the likelihood is the largest. The uncertainty in $\hat{x}_{i,E}$ was estimated from the 95% confidence interval, which was taken to be the equal to the range of $x_{i,E}$ within which the log-likelihood fell by no more than $\frac{T_{0.025}}{2}$ compared to the peak, where $T_{0.025}$ denotes the smallest value of the log-likelihood ratio statistic that is exceeded by no more than 2.5% of the empirical null distribution.

5.3 Identification of barcodes that carry new mutations

At this point, we use barcode frequency trajectories and fitness estimates to identify barcodes that carry new beneficial mutations that were not present in their parental genotype. As explained in the previous section, we begin by excluding from further analysis all lineages for which the best estimate of the relative fitness is not positive in any of the segments of their trajectories. This eliminates the majority of previously called lineages in the YPD and YPA environments, leaving us with a subset of a few hundred lineages in each population which all have at least one segment in either the evolution or the barcoding condition during which the lineage has a positive fitness relative to the mean.

We interpret the fact that these lineages have above average fitness at some point as evidence of a beneficial mutation establishing at some time prior to that point. Note that, because all barcodes inserted in the population are retained by the individuals, a new beneficial mutation will cause all barcodes carried by that lineage to increase in frequency (i.e. a beneficial mutation establishing in epoch 3, will cause BC1, BC2 and BC3 of that lineage to increase in frequency). Furthermore, because the population reproduces asexually and all mutations carried by an individual are passed on to all of its offspring, a beneficial mutations arising on parental backgrounds may also cause further barcodes to increase in frequency (e.g. BC4, BC5 and later barcodes arising on the background of BC3 may continue to increase in frequency until the lineage is outcompeted by the advancing mean of the population). Thus, there are several possibilities for when the beneficial mutation may have established:

1. The mutation established on the background of a parental barcode, prior to the addition of the last barcode in the identified lineage. The fitness benefit reflects a mutation established in the parent, and is shared by sibling barcodes.
2. The mutation established on the background of this barcode, prior to the addition of the next set of barcodes. This is a new mutation, its benefits are not shared by sibling barcodes.
3. The mutation established on the background of a child lineage, causing all of its earlier barcodes to increase in frequency.

In all cases, we wish to reject all lineages except the one in which the mutation established, grouping all descendant lineages that have the same fitness into a single “clone”. Occasionally, some of these descendant lineages will carry an additional mutation, causing them to increase in frequency faster (or decrease slower) than their parent. In this case, we must also find the lineage the mutation established in, and discard all intermediate and downstream lineages that were affected by it.

To infer the timing of a beneficial mutation, we validate each of the called lineages by hand, by starting from the root of the genealogy of all called lineages, and working down to the tips of the genealogy. We assign each lineage a flag according to the following criteria:

- A Mutation established on the background of this barcode, conferring a benefit in the evolution condition, and founding a new “clone” labeled by the barcode array identifying the lineage.
To be assigned this flag, a lineage must have a fitness advantage over the parental background over multiple evolution epochs, increase in frequency relative to the parent consistently while the relative fitness of the parental background is positive, until it either fixes, is outcompeted by another sibling lineage or drifts to extinction as the parental lineage drifts to extinction. In addition, we require that its direct descendants share the same benefit.
- B This flag is equivalent to flag A, but denotes a mutation conferring a benefit in the barcoding condition only.
- P Fitness of this lineage in both evolution and barcoding condition is indistinguishable from parent. This is the most common case. These lineages were excluded from further analysis.
- C Fitness difference of this lineage with respect to its parent results from a new mutation within a child lineage. These lineages were excluded from further analysis.

In about 5% of cases in each population, lineages were assigned one of these additional flags:

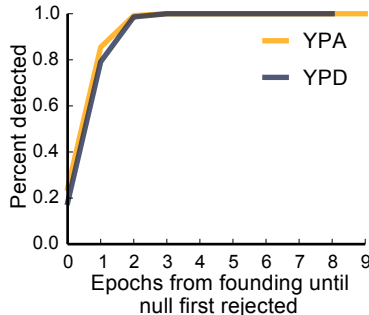
R(X) Unclear/marginal call. X denotes the alternative flag(s) considered.

R(N) Trajectory appears neutral.

+ Apparent new mutation within one of the child lineages not detected by pipeline.

Note that we sometimes find, especially when considering small lineages, that only a single child lineage is called as selected, despite apparently similar trajectories of the called lineage and its siblings. In this case, since we cannot both estimate the fitness of the segments of these sibling trajectories and maintain the same FDR, we use a heuristic to determine whether or not these direct descendants share the same benefit as the parent, motivated by the fact that genetic drift is unlikely to perturb allele frequency trajectories in exponentially expanding populations. If the relative frequencies of the sibling trajectories appear constant over multiple epochs after the parental lineage establishes and while it is expanding, we interpret this as evidence that the sibling lineages share the same benefit as the parent. If however the relative frequency of a called child lineage increases over multiple epochs by $\mathcal{O}(f)$, we interpret this as evidence that this lineage or one of its descendants carries a new mutation, since this event is unlikely to be driven by drift.

Finally, we note that the power to detect a new mutation decreases with the length of the available time-course for that lineage: early segments of lineages, during which their frequencies are low, are more strongly affected by drift, and therefore better explained by the null hypothesis. We can obtain an estimate of the length of time-course needed to detect a lineage from our data from the sequence of barcode fitness measurements of accepted lineages. As we show in Supplementary Figure 7, the majority of lineages that were determined to carry a new mutation would be rejected at the FDR of 5% if only their first epoch was considered. However, almost all of them are called within the next two epochs. Thus we conclude that we have limited power to detect mutations establishing in the last 1-2 epochs of the experiments.



Supplementary Figure 7: **Number of epochs needed to detect a positively selected lineage.** Distribution of times until a segment of a lineage carrying a new beneficial mutation is inferred to have nonzero fitness in either the evolution or the barcoding condition.

6 Analysis of clone fitness, fate, and overall population dynamics

6.1 Joint inference of clone fitnesses

Having identified barcodes that label new beneficial mutations, we split the population into clones that do not differ by any beneficial mutations that we can detect by merging all sibling lineages that have been called as equivalent in fitness. We then use these clone frequency trajectories to jointly infer clone fitnesses relative to the ancestor in both the evolution and the barcoding conditions ($x_i, x_i^{(bc)}$). As can be seen in Figures 2-4 of the Main Text, the population remains diverse throughout the experiment, with only a handful of clones reaching frequencies of order 10% or more. Thus for the vast majority of clones and times, our branching process model will represent a good approximation. For a handful of lineages that approach frequencies larger than 10%, the branching process model will overestimate the noise in the change in the number of reads mapping to a clone from time-point to time-point. However, since during these periods this (overestimated) noise is already subdominant compared to the deterministic trends caused by selection, this will have limited effect on our estimates. Thus, we consider the branching process model a reasonable approximation for all clone frequency trajectories.

We jointly infer the fitnesses of all clones in the evolution and in the barcoding conditions using cyclic coordinate descent to maximize the overall likelihood of the data with respect to the fitnesses of the clones in the two conditions. In each of the conditions independently, we begin by initializing all of the fitnesses of all clones to be equal to the fitness of the ancestor. We then cycle through each clone i , and maximize the log-likelihood of the data with respect to the fitness of that clone holding the fitnesses of all other clones constant, by performing a search for x_i on a grid of 2001 points linearly spaced between $\frac{-5}{\Delta t}$ and $\frac{5}{\Delta t}$ for the evolution condition, and -5 and 5 for the barcoding condition:

$$\hat{x}_i = \operatorname{argmax}_{x_i} \log \mathcal{L}[\text{data}] = \operatorname{argmax}_{x_i} \sum_j \log \mathcal{L}(\{r_j\}|\{x_j\}). \quad (23)$$

Here, the log-likelihood of each trajectory $\log \mathcal{L}(\{r_j\}|\{x_j\})$ is calculated by summing the log-likelihoods of each of the segments of that trajectory in the relevant environment (evolution, or barcoding), which are given by Equation 13, where we use the full form for a_i from Equation 10, relaxing all approximations we made in Section 5.2. As previously, in both the evolution and barcoding environment, we exclude all segments in which the clone read count falls below 20 counts. The 95% confidence interval for the estimate x_i is designated to be equal to the range on the grid

over which the overall likelihood of the data $\mathcal{L}[\text{data}]$ does not fall more than 2 log-likelihood units. We iterate this procedure by cycling through all the lineages until the total log-likelihood of the data changes over consecutive cycles by no more than 0.001.

We estimate a conservative confidence interval for the mean fitness trajectory according to

$$CI_{\bar{x}(t)} = \sum_j \frac{r_j(t)}{R_t} CI_{x_j}. \quad (24)$$

Clone barcode arrays, fitnesses relative to the ancestor and their confidence intervals in the evolution and in the barcoding environment, as well as the colors used to display them in Figures 2 and 3 of the Main Text are shown in Supplementary Table 3 and Supplementary Table 4.

6.2 Impact of the barcoding procedure on the evolutionary dynamics

As we explained in Section 4.2, the evolutionary dynamics during the 100-generation evolution epochs and during barcoding represent qualitatively equivalent evolutionary processes in different environmental conditions. In each condition, the population undergoes a process of growth and selection, and experiences a bottleneck.

In Section 4.4 we quantified the effective strength of the bottlenecks during each evolution and barcoding epoch, and showed that the effective bottleneck that the population experiences in the barcoding procedure is somewhat narrower than the bottleneck experienced during the daily transfer. However, there are 10 bottlenecks per evolution epoch and only one per barcoding period. Thus the net amount of drift that the population experiences over the course of each evolution epoch is roughly comparable to the amount of drift during each barcoding procedure in the YPA environment, and smaller by a factor of about 5 in the YPD environment (due to the slightly larger size that this population reaches during evolution).

This means that if the fitnesses of the lineages in the two conditions were equal, we could think of the barcoding procedure as being equivalent to a 100-generation evolutionary epoch during which the dynamics are not observed at intermediate intervals. However, our fitness inferences show that in general the effects of mutations on fitness in the evolutionary condition and in the barcoding condition are not well correlated in either the YPD or in the YPA population (see Supplementary Fig. 8). Thus, both populations evolve in what is, strictly speaking, a fluctuating environment. However, the time spent in any given condition before the environment changes is short enough that the fluctuations themselves have only a weak impact on the clone frequency trajectories. Specifically, in both the evolution and barcoding condition, the changes in the logarithms of the frequencies over the course of a single evolutionary epoch or a single barcoding period are typically small compared to 1 (see Supplementary Fig. 9).

Formally, this means that on long timescales the evolutionary dynamics can only be influenced by the average fitness between the two conditions [45]. Therefore, in all of the analysis we describe in the Main Text, we use the average fitness of a clone between the two conditions. To calculate this average, we scale the evolution fitness by the number of generations spent in the evolution environment, and report it in units of percent:

$$x_i^{(\text{avg})} = \frac{100x_i + x_i^{(\text{bc})}}{200}. \quad (25)$$

Of course, we exploit shorter-timescale information to more reliably group lineages into clones, in that lineages with the same average fitness but different fitnesses in one of the two conditions experienced in over the course of the experiment are never grouped in the same clone.

Clone Barcodes	Evolution Fitness (percent)	95% CI (per cycle)	Barcoding Fitness (percent)	95% CI (percent)	Color
CCTGGAGCAGTCTAAT	0.75	(0.50,0.95)	0.23	(-0.04,0.46)	
CCTGGAGCAGTCTAAT_CTGTGTCAGCCTCGT	1.45	(1.40,1.50)	1.06	(0.99,1.13)	
ACGAGGCGTGAATCA	-0.10	(-0.30,0.05)	1.13	(0.96,1.29)	
GTGCCTAGACCAACTT	0.00	(-0.20,0.20)	1.09	(0.90,1.27)	
AGCTATACCCCAACAA	0.65	(0.55,0.75)	0.46	(0.32,0.58)	
AGCTATACCCCAACAA_CTGATTCTGAGACTG_TACCTACAGACAATCT	1.20	(1.00,1.40)	0.77	(0.58,0.95)	
TACAGGGATGAAGCT_GTTGATTGGACGGATG	1.10	(1.05,1.15)	0.85	(0.80,0.91)	
TACAGGGATGAAGCT_GTTGATTGGACGGATG_CGGTTTGGGCTCACCT_ACCGGGATGCGATAA	1.10	(0.85,1.30)	1.10	(0.88,1.31)	
TACAGGGATGAAGCT_GTTGATTGGACGGATG_ATTACATATGCACGCC_ATTTTCGGGACGCGCC	2.45	(2.45,2.50)	1.42	(1.39,1.44)	
TACAGGGATGAAGCT_GTTGATTGGACGGATG_ATTACATATGCACGCC_ATTTTCGGGACGCGCC_TCAGGGGAGGTTTCGT	2.90	(2.80,3.00)	1.62	(1.49,1.73)	
TACAGGGATGAAGCT_GTTGATTGGACGGATG_ATTACATATGCACGCC_ATTTTCGGGACGCGCC_TCAGGGGAGGTTTCGT	4.30	(3.45,5.15)	2.19	(0.32,3.15)	
TACAGGGATGAAGCT_GTTGATTGGACGGATG_ATTACATATGCACGCC_ATTTTCGGGACGCGCC_TGGTAACCGGGTCCT	2.45	(2.30,2.55)	1.65	(1.51,1.77)	
TACAGGGATGAAGCT_GTTGATTGGACGGATG_ATTACATATGCACGCC_ATTTTCGGGACGCGCC_ACCTCTCGCATCCAAC	3.65	(2.95,4.25)	2.57	(1.48,3.27)	
TACAGGGATGAAGCT_GTTGATTGGACGGATG_ATTACATATGCACGCC_ATTTTCGGGACGCGCC_AGAGTCAGCTAGTATT	2.50	(2.35,2.60)	1.77	(1.62,1.91)	
TACAGGGATGAAGCT_GTTGATTGGACGGATG_ATTACATATGCACGCC_ATTTTCGGGACGCGCC_AGATAAAGTATTGCT	3.45	(3.30,3.60)	1.44	(1.20,1.65)	
TACAGGGATGAAGCT_GTTGATTGGACGGATG_ATTACATATGCACGCC_ATTTTCGGGACGCGCC_TGGTAACCGGGTCCT	3.30	(3.15,3.40)	2.00	(1.85,2.14)	
TACAGGGATGAAGCT_GTTGATTGGACGGATG_ATTACATATGCACGCC_ATTTTCGGGACGCGCC_CTGCTATGTTACACC	2.10	(2.00,2.25)	2.23	(2.11,2.34)	
TACAGGGATGAAGCT_GTTGATTGGACGGATG_ATTACATATGCACGCC_ATTTTCGGGACGCGCC_CTGCTATGTTACACC	2.00	(1.40,2.55)	2.98	(2.47,3.40)	
TACAGGGATGAAGCT_GTTGATTGGACGGATG_ATGTGTGCATAGAGT_ACTTTGGGCCACAGCA	1.25	(1.00,1.55)	0.97	(0.69,1.22)	
TTTGGCAACCTGGGTG_TATTGCAGTTATGCAG	1.00	(0.95,1.00)	1.02	(0.99,1.05)	
TTTGGCAACCTGGGTG_TATTGCAGTTATGCAG_TAGCAATAAACACGCT_TGACCAGTACCCGAC	0.80	(0.35,1.25)	1.56	(1.23,1.84)	
TTTGGCAACCTGGGTG_TATTGCAGTTATGCAG_TGTCGGTAAATCTAGT_ACGGGTAAAGAGTTGCG	0.75	(0.50,1.00)	1.57	(1.38,1.74)	
TTTGGCAACCTGGGTG_TATTGCAGTTATGCAG_CAACCACCCGCTATCA_TGAGGGAACCCCTTGA_ACGGTTATGTAACCGG	2.95	(2.90,3.05)	2.22	(2.14,2.29)	
TTTGGCAACCTGGGTG_TATTGCAGTTATGCAG_TCCGATGGCTTCGAGT	1.00	(0.90,1.10)	1.67	(1.56,1.75)	
TTTGGCAACCTGGGTG_TATTGCAGTTATGCAG_TCCGATGGCTTCGAGT_TAGATGGTCTTGAGC_CGGCCGATCTCGACTT	1.25	(0.90,1.60)	1.92	(1.66,2.13)	
TTTGGCAACCTGGGTG_TATTGCAGTTATGCAG_TGTTAATCAGGGGGAG	1.45	(1.45,1.50)	1.29	(1.24,1.33)	
GCATCGTACACCGGAT_GATCCACAGCCTGTAT	1.15	(1.15,1.20)	0.97	(0.92,1.03)	
GCATCGTACACCGGAT_GATCCACAGCCTGTAT_ATCAGTAAACTTTCAA_GTGGAATCGCTCTGA	2.45	(2.40,2.45)	1.38	(1.33,1.42)	
GCATCGTACACCGGAT_GATCCACAGCCTGTAT_ATCAGTAAACTTTCAA_GTGGAATCGCTCTGA_TGCCGCTTCCCCGCAC	2.00	(1.50,2.50)	2.33	(1.77,2.77)	
GCATCGTACACCGGAT_GATCCACAGCCTGTAT_ATCAGTAAACTTTCAA_GTGGAATCGCTCTGA_TGCCGCTTCCCCGCAC	4.00	(3.25,4.70)	0.89	(-0.75,1.80)	
CATCCGTCTCTGAGA_AGGTCAATCCCTCTT	1.15	(1.05,1.20)	0.55	(0.45,0.65)	
TGCTCCACACGAGTCC_CCTTTGGGGTGAAGCC	1.00	(0.95,1.05)	0.91	(0.84,0.97)	
ATAATGAGTGCACCTT_CAAGCATACTTATATG	1.95	(1.90,1.95)	0.77	(0.73,0.82)	
ATAATGAGTGCACCTT_CAAGCATACTTATATG_CTGACAGCCGAGCCCT_TTCCTGAGAAGATTGC_GACCCCTCAACATCG	3.00	(2.90,3.15)	1.89	(1.71,2.04)	
GTCTTGGCTCCGATTG_TAAAGCAGAGCACATG	0.45	(0.35,0.55)	0.61	(0.48,0.74)	
TAGTTCGCCGGAGCCA_TCTCTAAAAGCGCTAA	0.85	(0.40,1.30)	0.11	(-0.49,0.57)	
AGGGCGTCTCAGAA_TAGTGAAGCGCCGTCT	0.90	(0.75,1.00)	0.41	(0.26,0.54)	
ATCGTTGAATACACGC_CTTGCGCTCGAATGAT	1.50	(1.35,1.65)	0.33	(0.15,0.50)	
ATCGTTGAATACACGC_CTTGCGCTCGAATGAT_AATTAGGGTTAGCCCC_TTAAGTGCAGCGCTT	1.95	(1.80,2.05)	0.59	(0.43,0.74)	
GCCGCCCTATACGAA_TCCCTGTGGTCAAT	1.45	(1.05,1.85)	-1.09	(-1.95,-0.49)	
TCTAAGCGTATTGGTC_ATCCAGCGCTTGGACG	1.25	(1.20,1.30)	0.11	(0.02,0.20)	
TCTAAGCGTATTGGTC_ATCCAGCGCTTGGACG_TCTCCGCTACGCGAGT_ATTGCCGACTTTAGC	2.80	(2.65,2.90)	0.42	(0.25,0.57)	

(continued on next page)

Clone Barcodes	Evolution Fitness (percent)	95% CI (per cycle)	Barcoding Fitness (percent)	95% CI (percent)	Color
TCTAAGCGTATTGGTC_ATCCAGCGCTTGGACG_TCTCCGCTACGCGAGT_ATTGCCCGACTTTAGC_TTGAATTCGCATTGGA	3.10	(2.75,3.40)	0.47	(0.04,0.83)	Blue
TCTAAGCGTATTGGTC_ATCCAGCGCTTGGACG_TCTCCGCTACGCGAGT_ATTGCCCGACTTTAGC_AAGCCGAGGCTATTTCT_GACTAGTGGTGTAG_CTGGGGATTGTGGTAT	4.60	(4.50,4.75)	1.57	(1.31,1.79)	Blue
TCTAAGCGTATTGGTC_ATCCAGCGCTTGGACG_AGATGAACCGGCCAT	2.45	(2.40,2.50)	0.40	(0.33,0.47)	Blue
TCTAAGCGTATTGGTC_ATCCAGCGCTTGGACG_AGATGAACCGGCCAT_GGTCTCCAAAAGCGGT_TGGCGTGGCGCAGC_TTGGCCCGGGCGCAA_TTAAAAGACATCGGTAA	3.40	(3.10,3.75)	1.68	(1.19,2.07)	Blue
AATAGTTCTGGGGACC_AAGACCCGGAATCAG	0.90	(0.75,1.05)	0.58	(0.40,0.75)	Grey
TAGTGACTTAGACCTG_CTTATCAACGGTGCTA	2.05	(2.05,2.10)	0.34	(0.29,0.39)	Green
TAGTGACTTAGACCTG_CTTATCAACGGTGCTA_CAACCTGTAATTCTGC_CCCGTAAGGTAGCTCC_TGCTATAGCAAACCCG	2.80	(2.25,3.30)	0.54	(-0.13,1.04)	Green
TAGTGACTTAGACCTG_CTTATCAACGGTGCTA_CAACCTGTAATTCTGC_GGCAGGTAGACCTTCA_GAGAAGAATGGAATGA_GCAATACGCTTAAAGT	3.00	(2.45,3.50)	1.96	(1.41,2.40)	Green
TAGTGACTTAGACCTG_CTTATCAACGGTGCTA_AAGATTACGCGTATAA_TACCCTGTGATCTGTA_TGCACAGCTCCCGGAC	2.45	(2.25,2.60)	1.45	(1.25,1.63)	Green
TAGTGACTTAGACCTG_CTTATCAACGGTGCTA_TGCCTCGAAGGCCTAT_CAGCAACCGCGTTACA_CCTGCCCGTGCAGAA	2.00	(1.85,2.20)	1.65	(1.49,1.80)	Green
TAGTGACTTAGACCTG_CTTATCAACGGTGCTA_AAGATTACGCGTATAA_TACCCTGTGATCTGTA_CTGAGTAAAAGGGACT	2.55	(2.35,2.70)	1.67	(1.48,1.85)	Green
GCCGTGCCAAAGTAGG_TTGAGATCACAAAGCA	0.85	(0.45,1.20)	0.21	(-0.31,0.62)	Grey
ACTGTACCTGCGGGTT_TGCGCCTTGTGATGA	1.10	(1.00,1.20)	0.20	(0.04,0.35)	Grey
GCATAGGGGGGAGCAC_CCTGCCTGTACTGTTG	0.90	(0.40,1.40)	-0.58	(-1.49,0.06)	Grey
GGCGAGAACTTAATGG_GTGCTCGCAACTTTAC	1.00	(0.80,1.15)	0.01	(-0.24,0.24)	Grey
ATGGGGATGGCTGGCC_CCCCACTAGGTACTAT	0.75	(0.55,0.95)	0.18	(-0.08,0.42)	Grey
TACCTAGTTTACAGAC_AAAGATAGGGTCTCTA	0.75	(0.55,0.95)	0.79	(0.58,0.97)	Red
TACCTAGTTTACAGAC_AAAGATAGGGTCTCTA_AGTGCGGTCCCATTTCT_AGAGGTAACACTCTTT	0.85	(0.55,1.10)	1.49	(1.26,1.70)	Red
GGTGGCCTCATAGTAC_CATTAAGGGAGCGCTC	0.85	(0.70,1.00)	0.18	(0.01,0.35)	Grey
ACTCGATCTCGCCGCC_CGGGAAATTGCAAAC	0.95	(0.80,1.10)	1.01	(0.84,1.16)	Grey
TACGGAGCGAACAGTT_TCTTCCGAAGTTGGAC	1.15	(0.95,1.30)	0.45	(0.26,0.63)	Grey
GGGGAAGTTCGGGCA_TTTGTCAACCAAGTAGT	0.70	(0.45,1.00)	0.25	(-0.08,0.54)	Grey
TACTTTGCGTCTCAA_GCTTCTGGGGAAGTAT	1.40	(1.35,1.50)	0.27	(0.18,0.35)	Red
TACTTTGCGTCTCAA_GCTTCTGGGGAAGTAT_AAAAACCCACTTCTCC	1.20	(1.00,1.40)	0.91	(0.72,1.06)	Red
AGACCTATATCGCAAT_CGCGTCAGGGTGCTAG	0.85	(0.50,1.20)	0.29	(-0.12,0.63)	Grey
ATAGAATGGCTAAAAA_AGACCAGCTTATCAGT_CCAGCTCCCTTTCCGT	1.85	(1.55,2.20)	0.38	(0.01,0.69)	Grey
ACGCCACGCGTCTCAG_CAGGCGTGTGGGTCA_AGATACCACGTGCGTG	1.10	(0.90,1.30)	1.03	(0.85,1.21)	Grey
TAGCACTCTGTGCTA_CTCATGAGGACGAGGC_AGGAACCTACATCGTA	1.35	(0.90,1.75)	0.23	(-0.34,0.67)	Grey
CAGGACTAACTCTCCG_GCCTTCGGTCAAGCCA_GAAAGACGCAAGGATGC	0.95	(0.75,1.20)	0.77	(0.54,0.97)	Grey
AACCGGATTGCATAG_GTTACAGTGGTCTCG_TTTGAGTTAGGTGCGG	1.30	(0.75,1.85)	0.15	(-0.58,0.68)	Grey

Supplementary Table 3: **Clone fitness estimates in the YPD population.** Clones are identified by the array of barcodes that label the founding mutation. The likelihood-based confidence intervals for the quoted best-estimates of their fitnesses in the two environments are calculated as explained in Section 6.1. The colors used to display each clone in Figures 2 and 3 of the Main Text are provided for convenience.

Clone Barcodes	Evolution Fitness (percent)	95% CI (per cycle)	Barcoding Fitness (percent)	95% CI (percent)	Color
AGTAAGACCTCGGGCC	1.25	(1.15,1.30)	0.56	(0.48,0.62)	
CCACGCGCGGTACGTC	1.70	(1.65,1.75)	0.21	(0.15,0.28)	
CCACGCGCGGTACGTC_TGCCACACACGACGAT_TTCACTACCAATCTCC	2.50	(2.25,2.80)	0.51	(0.29,0.71)	
CCACGCGCGGTACGTC_TGCCACACACGACGAT_TATGTAGATCCAATGG	2.30	(2.15,2.40)	0.74	(0.65,0.84)	
CCACGCGCGGTACGTC_TGCCACACACGACGAT_TATGTAGATCCAATGG_AGACACATGACATCCA_CTGCTTAGACCGGTG_AAGACTTGCCACATTG	4.25	(3.95,4.50)	1.11	(0.77,1.41)	
CCACGCGCGGTACGTC_TGCCACACACGACGAT_ATACGCAACTCGGTTT_ATGGCTTTAACAGCCA	2.35	(1.80,2.90)	0.46	(0.01,0.83)	
AAACTTATGCACTCAC	0.40	(0.10,0.65)	0.52	(0.28,0.73)	
TACCCGATGGCGAAGC	1.10	(0.85,1.35)	0.79	(0.61,0.96)	
TCCATTGAGAACAACCT	1.70	(1.65,1.75)	0.30	(0.25,0.36)	
TCCATTGAGAACAACCT_TGCACACCCATGAGTA_CGGTCCAGTTGTATG_CCAATCGCGAGGCTT	2.80	(2.40,3.20)	0.55	(0.19,0.86)	
TCCATTGAGAACAACCT_TGCACACCCATGAGTA_CACGACATGGTAATGT	2.25	(2.00,2.45)	0.59	(0.40,0.78)	
TCCATTGAGAACAACCT_ACCTAGGTCGTGGGAG_AACGTTAAGTGTTCAT_GTCATTCTAAGGCCGG	2.40	(2.30,2.55)	1.03	(0.94,1.14)	
TCCATTGAGAACAACCT_TGCACACCCATGAGTA_GGTTGCAAGCATAGAA_GCTGTCGCGTTTCGTT_TTGCAGGGCCCGAAT_GACTGCTGCACGTTGA	4.00	(3.70,4.25)	0.84	(0.58,1.08)	
TCCATTGAGAACAACCT_TGCACACCCATGAGTA_GGTTGCAAGCATAGAA_GCTGTCGCGTTTCGTT_TTGCAGGGCCCGAAT_GACTGCTGCACGTTGA_TATCGCCCGAATGATT	7.85	(7.80,7.90)	3.33	(3.28,3.36)	
TCCATTGAGAACAACCT_TGCACACCCATGAGTA_GGTTGCAAGCATAGAA_GCTGTCGCGTTTCGTT_TTGCAGGGCCCGAAT_GACTGCTGCACGTTGA_TATCGCCCGAATGATT_CTCGAAAGTGTGCGTA_AGTTTCCCGACGGCAA	8.40	(7.40,9.30)	3.05	(1.90,3.78)	
TCCATTGAGAACAACCT_TGCACACCCATGAGTA_GGTTGCAAGCATAGAA_GCTGTCGCGTTTCGTT_TTGCAGGGCCCGAAT_GACTGCTGCACGTTGA_TATCGCCCGAATGATT_CTCGAAAGTGTGCGTA_CAAACCTCGCGTGTAC	8.65	(7.85,9.40)	3.07	(2.04,3.75)	
TCCATTGAGAACAACCT_TGCACACCCATGAGTA_GGTTGCAAGCATAGAA_GCTGTCGCGTTTCGTT_TTGCAGGGCCCGAAT_GACTGCTGCACGTTGA_TATCGCCCGAATGATT_CTCGAAAGTGTGCGTA_GTTGCACGACGAAACAT	7.50	(6.70,8.25)	4.76	(4.05,5.00)	
TCCATTGAGAACAACCT_ACCTAGGTCGTGGGAG_TTGTCCCTAGTTGTGT	2.60	(2.45,2.70)	0.41	(0.29,0.52)	
GCGTACAGTGGACCTT	0.00	(-0.45,0.40)	0.81	(0.47,1.10)	
GCAACCTGCAACTTC	1.70	(1.60,1.80)	0.20	(0.12,0.28)	
GCAACCTGCAACTTC_CAAGGAGGGTACTTCG_ACAGCGATGCCATCC	2.25	(2.00,2.45)	0.60	(0.42,0.78)	
TCTAGCGCGCCGAAT	1.35	(1.20,1.55)	-0.15	(-0.35,0.03)	
TCTAGCGCGCCGAAT_CTATGAGGTAGAAAACA	3.60	(3.60,3.60)	-0.43	(-0.46,-0.41)	
TCTAGCGCGCCGAAT_CTATGAGGTAGAAAACA_GCCTACGAATACAACC_TGAAGCTAGTTAGCGA_CGAGGACTAGAGTCAC	4.55	(4.25,4.85)	-0.29	(-0.67,0.03)	
TCTAGCGCGCCGAAT_CTATGAGGTAGAAAACA_CAGGCACTCACGTTAC_GTCAGAATTCGGAAAA	4.20	(4.05,4.35)	-0.17	(-0.34,0.00)	
TCTAGCGCGCCGAAT_CTATGAGGTAGAAAACA_GCCTACGAATACAACC_ACTGTCGTTTCGCTCGT_GAGTTAGATCCTAGAC_GAATGCTGCTTCGAC	4.75	(4.20,5.30)	-0.56	(-1.56,0.11)	
TCTAGCGCGCCGAAT_CTATGAGGTAGAAAACA_GCCTACGAATACAACC_GCCTATCACCCAGCC	3.75	(3.60,3.85)	-0.08	(-0.22,0.05)	
TCTAGCGCGCCGAAT_CTATGAGGTAGAAAACA_GATAGATATAGTAAGT_CCCGTGAAGGACATTT_TGAGACTGGCGCGCA	4.20	(4.05,4.40)	0.49	(0.29,0.68)	
TCTAGCGCGCCGAAT_CTATGAGGTAGAAAACA_TAAGTGCCTAAGTGAG_TGAGGTCAAAAGCCAT_AGACGTCGATCTATG	5.20	(5.05,5.30)	-0.13	(-0.31,0.03)	
TCTAGCGCGCCGAAT_CTATGAGGTAGAAAACA_CAGTGTGCACACGAC_TGCTTCCAAAGTGT_TCTGGCCCGGATTAT_CAGCCGGTCTGTTTT	5.15	(4.90,5.40)	-0.53	(-0.96,-0.17)	
TCTAGCGCGCCGAAT_CTATGAGGTAGAAAACA_GAAGCGGTTTGGCAT_TGCTGCATTGTGGGAG_TCCAGTAGTGCCTCC	4.50	(3.90,5.05)	-0.42	(-1.07,0.07)	
TCTAGCGCGCCGAAT_CTATGAGGTAGAAAACA_GTACAGACTGCATGCT_CTATAACAATTCGGGA	4.20	(4.10,4.25)	-0.61	(-0.71,-0.51)	
TCTAGCGCGCCGAAT_CTATGAGGTAGAAAACA_GTACAGACTGCATGCT_CTATAACAATTCGGGA_CTCGCCTGTGTTTTT_CTTTAAACGCCAAAGG_TGATAGAAAGGATCCG	8.80	(8.40,9.20)	-2.19	(-3.43,-1.42)	
TCTAGCGCGCCGAAT_CTATGAGGTAGAAAACA_GCCTACGAATACAACC_CGAACGCGGTGGCAT_AACAGAGATTTGTGCC	4.30	(3.85,4.75)	-0.45	(-0.97,-0.03)	
TCTAGCGCGCCGAAT_CTATGAGGTAGAAAACA_TTAAGAGTAAGAGAGT_ACCATCAATGCTGCGA_TTGGTTTAAATCGAGTC	5.10	(4.55,5.60)	-0.86	(-1.45,-0.40)	
TCTAGCGCGCCGAAT_CTATGAGGTAGAAAACA_GATAGATATAGTAAGT_ATTCGGCTCATAAGG_ATTCTGGTGGCTAAAC	5.80	(5.70,5.90)	-0.45	(-0.61,-0.30)	
TCTAGCGCGCCGAAT_CTATGAGGTAGAAAACA_CCGAAAGAGTACGAC	4.00	(4.00,4.05)	-0.48	(-0.51,-0.45)	
TCTAGCGCGCCGAAT_CTATGAGGTAGAAAACA_CCGAAAGAGTACGAC_CGCTGCTGACTGTCT_CGGGGTGAAGCATA_TGATACCATAAGCTGA	6.60	(6.30,6.95)	-1.62	(-2.37,-1.07)	
TCTAGCGCGCCGAAT_CTATGAGGTAGAAAACA_CCGAAAGAGTACGAC_CTAATGGGCTGATCTG_AGGGGCCGACGGTAC	5.20	(5.15,5.20)	-0.19	(-0.26,-0.12)	
TCTAGCGCGCCGAAT_CTATGAGGTAGAAAACA_CCGAAAGAGTACGAC_CTAATGGGCTGATCTG_AGGGGCCGACGGTAC_ACTCTATGTGGGATGC_CTCCTCCGTCGGGACA	6.50	(6.10,6.90)	-0.20	(-0.71,0.21)	
TCTAGCGCGCCGAAT_CTATGAGGTAGAAAACA_CCGAAAGAGTACGAC_CTAATGGGCTGATCTG_AGGGGCCGACGGTAC_ACTCTATGTGGGATGC_ACAAAATTGAGACGATA_AAGACAATAGCAGATA	7.30	(6.40,8.20)	-0.80	(-2.81,0.17)	
TCTAGCGCGCCGAAT_CTATGAGGTAGAAAACA_CCGAAAGAGTACGAC_CTAATGGGCTGATCTG_AGGGGCCGACGGTAC_GCATGAACCCGGGAAC_TAGTCCGGCTGAGAA	6.10	(5.80,6.40)	1.08	(0.70,1.41)	

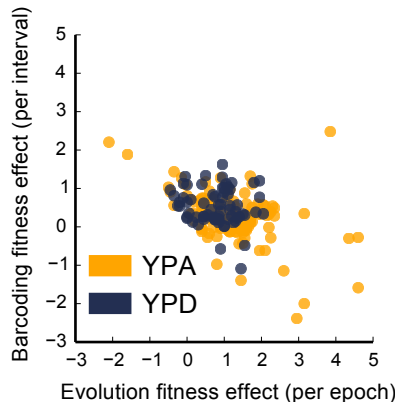
(continued on next page)

Clone Barcodes	Evolution Fitness (percent)	95% CI (per cycle)	Barcoding Fitness (percent)	95% CI (percent)	Color
TCTAGCGCGCCGCAAT_CTATGAGGTAGAAAACA_CCGGAAAGAGTACGAC_CTAATGGGCTGATCTG_AGGGGCCGACGGTAC_ACCATAAATCACACCT_CAAGGCTGCGGTGCC_TACGAGACAAGATACAC	6.70	(5.60,7.70)	-0.25	(-2.28,0.76)	
TCTAGCGCGCCGCAAT_CTATGAGGTAGAAAACA_CCGGAAAGAGTACGAC_CTAATGGGCTGATCTG_AGGGTTCCGCAACGTC	4.80	(4.60,4.95)	-0.16	(-0.38,0.06)	
TCTAGCGCGCCGCAAT_CTATGAGGTAGAAAACA_CCGGAAAGAGTACGAC_GCGTCGCGGTAGTGT_GAGAATGCTTAAATAG	4.55	(4.25,4.85)	0.21	(-0.13,0.51)	
TCTAGCGCGCCGCAAT_CTATGAGGTAGAAAACA_CCGGAAAGAGTACGAC_CTAATGGGCTGATCTG_TCGAGTAAAGGAAACG	5.35	(5.30,5.45)	-0.36	(-0.49,-0.24)	
TCTAGCGCGCCGCAAT_CTATGAGGTAGAAAACA_CCGGAAAGAGTACGAC_AGGTCCCCGACTTTAA_AGCTACAGTCTGTCCG_CCGGACGTTTATATGT_GCTCCAAATCTCTTA	5.30	(4.40,6.15)	0.03	(-1.67,0.96)	
TCTAGCGCGCCGCAAT_CTATGAGGTAGAAAACA_CCGGAAAGAGTACGAC_CTAATGGGCTGATCTG_TAGTTAAATGATGGC	4.40	(4.05,4.70)	-0.60	(-1.02,-0.26)	
TCTAGCGCGCCGCAAT_CTATGAGGTAGAAAACA_CCGGAAAGAGTACGAC_CCAACGAAGTACTT_GAGGGTCTCTTAATT_ACTGTCTATCTAGG	1.90	(1.40,2.40)	1.72	(1.43,1.98)	
TCTAGCGCGCCGCAAT_CTATGAGGTAGAAAACA_GATAGATATAGTAAGT_CCCGTGAAGGACATTT_CCTGCCGAACCTACCT_GTATCTGCTGCTGGTGC_CACAGTCTGGTATGC	5.30	(4.20,6.35)	0.89	(-0.68,1.76)	
TCTAGCGCGCCGCAAT_CTATGAGGTAGAAAACA_GCCTACGAATACAACC_GTAATCGTCAATAAA_ATACTCACACTGCTAC	4.65	(4.35,4.95)	-0.08	(-0.43,0.23)	
TCTAGCGCGCCGCAAT_CTATGAGGTAGAAAACA_GCCTACGAATACAACC_CCGGCAACAGTCC_AGAATAGGTCAGGGTT	5.00	(4.70,5.30)	-0.88	(-1.27,-0.55)	
TCTAGCGCGCCGCAAT_CTATGAGGTAGAAAACA_GAAGGCGGTTGGCAT_AAAATGGGATAGTCC_GCCCTAGTCTTGAGC	4.15	(3.55,4.65)	-0.22	(-0.84,0.25)	
TCTAGCGCGCCGCAAT_CTATGAGGTAGAAAACA_TTAAGAGTAAGAGAGT_ATCAGTGAAGGCCA_ACTGTATGCAGCGATC	4.50	(4.20,4.80)	-0.30	(-0.66,-0.00)	
TCTAGCGCGCCGCAAT_CTATGAGGTAGAAAACA_GCCTACGAATACAACC_ATCGGTGTGCCGCGC_TCGCAAAAAGGCCAAC	4.00	(3.70,4.35)	0.31	(-0.02,0.59)	
TCTAGCGCGCCGCAAT_CTATGAGGTAGAAAACA_GAAGGCGGTTGGCAT_TGCTGCATTGTGGGAG_ACGGGTAGATCGACCA_GTGGGGAGGAAAATTC	5.20	(4.85,5.55)	0.16	(-0.30,0.54)	
TCTAGCGCGCCGCAAT_CTATGAGGTAGAAAACA_GCCTACGAATACAACC_GAGTACAAGGCCAAGT_GCAGTCCGCTGGCCCT	4.95	(4.85,5.05)	-0.47	(-0.62,-0.33)	
TCTAGCGCGCCGCAAT_CTATGAGGTAGAAAACA_GCCTACGAATACAACC_GAGTACAAGGCCAAGT_GCAGTCCGCTGGCCCT_TCCCTGCGGGGTAC	5.40	(5.00,5.80)	0.28	(-0.30,0.72)	
TCTAGCGCGCCGCAAT_CTATGAGGTAGAAAACA_GCCTACGAATACAACC_CGAACGCGGTGGCAT_GGTAGTAATGGGTGTA	4.60	(4.10,5.10)	-0.36	(-1.07,0.17)	
TCTAGCGCGCCGCAAT_CTATGAGGTAGAAAACA_GATAGATATAGTAAGT_CAACGAAAAATGCGAT_GCGGAGCGACTTAGAC_TAGAGAGCTCGTGATA	2.00	(0.90,3.10)	1.45	(0.70,2.00)	
TAAGGTTCAAACCTCC	1.55	(1.25,1.85)	-0.29	(-0.64,0.01)	
TAAGGTTCAAACCTCC_TAGACCTGCCAACAT_CCTCGTGTCCGATGG	1.95	(1.60,2.25)	0.08	(-0.24,0.34)	
TTGTCCGTCTTCAAC	1.60	(1.45,1.75)	-0.03	(-0.18,0.12)	
TTGACAACCTCCGCCGT	1.35	(1.20,1.50)	0.26	(0.12,0.39)	
AAGTAAGCCACTCAC	1.55	(1.35,1.75)	0.29	(0.11,0.47)	
CGCGGTGGAACGGAGG_CGCAACATGTAACCTT	1.90	(1.90,1.90)	0.59	(0.56,0.61)	
CGCGGTGGAACGGAGG_CGCAACATGTAACCTT_TCTACGAACGAGTAGC_ATGAAGGTCTACGTTT_AAGCACCAGCGGTTT_TCGGACTGGGAGCTTG	4.25	(4.00,4.45)	0.88	(0.59,1.12)	
CGCGGTGGAACGGAGG_CGCAACATGTAACCTT_ACGGTGCGTCAAGTTT_TGAGGCAACAGGAGT_CCAAGTAGTGAGCCCT	1.75	(0.60,2.80)	1.86	(1.21,2.36)	
CGCGGTGGAACGGAGG_CGCAACATGTAACCTT_GGCCCATAGGTCATCT_GACGGGATGCTGGTAA_AGCCTACTCAATGAGC	3.15	(2.80,3.55)	1.25	(0.94,1.52)	
CGCGGTGGAACGGAGG_CGCAACATGTAACCTT_TGCCGCTCCACCCTA_GGACTTAAATCACTCG_CGTGAACATGAAAATTT	3.35	(3.10,3.65)	0.71	(0.47,0.93)	
CGCGGTGGAACGGAGG_CGCAACATGTAACCTT_TGCCGCTCCACCCTA_ATCGACTTCTGTGCG_CTCAAATGACCCGAT_TTAGGCTATGCAAAAA	2.40	(1.30,3.40)	1.76	(0.92,2.36)	
CGCGGTGGAACGGAGG_CGCAACATGTAACCTT_TGTACCGAATTATAC_GTTGTGTGCGCTCTCC_TGGCCGCTTGCCTGTT	2.75	(2.40,3.15)	1.11	(0.78,1.40)	
CGCGGTGGAACGGAGG_CGCAACATGTAACCTT_AACGATGATACGTTGTT_AGTATTATGCGACGCC	4.05	(4.05,4.05)	0.89	(0.85,0.92)	
CGCGGTGGAACGGAGG_CGCAACATGTAACCTT_AACGATGATACGTTGTT_AGTATTATGCGACGCC_TGCCAGTCAATGTTGTT_ACCGTGAGGCAAGGTC_TGAAATCCTGTTGGTT	5.35	(4.70,6.00)	0.76	(-0.12,1.38)	
CGCGGTGGAACGGAGG_CGCAACATGTAACCTT_AACGATGATACGTTGTT_AGTATTATGCGACGCC_GCACAAAAAGGAATAG_AGCGAACATTGCTGGG_CTACGCACCGTTGACT_CATGTCTGCATGGGCA	5.55	(4.55,6.50)	0.95	(-1.23,1.98)	
CGCGGTGGAACGGAGG_CGCAACATGTAACCTT_AACGATGATACGTTGTT_AGTATTATGCGACGCC_AAACGACCTCGGCATT_CTCATAGGTGTTAGTT_GTACCAGGTATGTCGC	5.55	(4.70,6.35)	0.73	(-0.66,1.55)	
CGCGGTGGAACGGAGG_CGCAACATGTAACCTT_AACGATGATACGTTGTT_AGTATTATGCGACGCC_CGCGCGGAACTGAC_TCGTCTTCAGAAAGCG_CGTGTTGGCCAAAGACC	4.95	(4.35,5.60)	0.84	(-0.06,1.47)	
CGCGGTGGAACGGAGG_CGCAACATGTAACCTT_AACGATGATACGTTGTT_AGTATTATGCGACGCC_TGCCAGTCAATGTTGTT_TAACCGAACGAGTTCC_AGGTTCGGTTTCGACC_CTCCAGACGCGCACTT	5.20	(3.85,6.50)	1.85	(0.86,2.52)	
CGCGGTGGAACGGAGG_CGCAACATGTAACCTT_AACGATGATACGTTGTT_AGTATTATGCGACGCC_TGCCAGTCAATGTTGTT_CTCCTAACCGCTTATCTAGATTGGAAATCCAC_TCCAATCATGACGGAC	6.55	(5.20,7.85)	-0.15	(-2.60,0.92)	
CGCGGTGGAACGGAGG_CGCAACATGTAACCTT_CCGTGTAGACAAAAC_AACTCTCACACACAAC	2.30	(1.95,2.65)	0.53	(0.27,0.74)	
CGCGGTGGAACGGAGG_CGCAACATGTAACCTT_CCCACTAGGATTCTC_ACGTATTGGCGCAGGC_CACAGCATTGTTGGAG_CTAACGACTGTCGTGG	3.55	(3.00,4.10)	0.59	(0.15,0.96)	
ACAATCCGAATGAAGG_CTTGGAACCTGCGTTGA	2.05	(1.95,2.15)	0.57	(0.49,0.64)	
CGAAGCGATGCGTGTGTA_CGTGTAACCTCGCAA	1.75	(1.65,1.90)	0.51	(0.40,0.60)	
CGAAGCGATGCGTGTGTA_CGTGTAACCTCGCAA_TCTGTGAGATGGAGTC	1.95	(1.75,2.20)	0.66	(0.50,0.80)	

(continued on next page)

Clone Barcodes	Evolution Fitness (percent)	95% CI (per cycle)	Barcoding Fitness (percent)	95% CI (percent)	Color
CGAAGCGATGCGTGTGTA_CGTGTAACCCCTCGCAA_TCTGTGAGATGGAGTC_CGGAAGCGTTTACCTA_AGGCACAAATTCACAA	3.60	(3.20,3.95)	0.97	(0.62,1.27)	
CGAAGCGATGCGTGTGTA_CGTGTAACCCCTCGCAA_TCTGTGAGATGGAGTC_AAGACTACACACGCTC_CGAACCGACTTACTTA	2.35	(1.75,2.90)	1.09	(0.77,1.37)	
TGCGGCATAGTCCTCG_GTAGGTTATGCGGAAG	1.85	(1.70,1.95)	0.40	(0.30,0.48)	
GTGCGGACCCCGTAT_ATCCAAAATCTGTGAC	2.20	(2.05,2.35)	0.33	(0.21,0.45)	
GTGCGGACCCCGTAT_ATCCAAAATCTGTGAC_TAGATTCATAGCTGGC	2.30	(2.10,2.50)	0.67	(0.53,0.81)	
CCGAGTCTTCCAATT_GAAACGTCTCCACCC	-0.50	(-1.10,0.05)	1.03	(0.57,1.41)	
TCGACTTGTTCAAGG_GTCGGCGAGCCAAGGA	-0.05	(-0.65,0.50)	1.10	(0.61,1.49)	
CCCGCAGAAAAGTAGC_CGCACCGAATTCATC	-0.45	(-0.90,-0.05)	0.91	(0.55,1.22)	
CCCGCAGAAAAGTAGC_CGCACCGAATTCATC_GGCGAAAACCTGTTGT	-0.80	(-1.55,-0.05)	1.47	(0.77,1.99)	
CCTTGACCTCCGACTA_AAATCTTCTACGCAA	2.10	(1.95,2.25)	0.54	(0.41,0.66)	
TGACGGGGCCCCGGTA_CAGGGTGAAGCGCTCG	0.10	(-0.40,0.55)	0.71	(0.28,1.07)	
ACCCAACAGTTCAAT_ACCAGAGGAGCGTATT	1.05	(0.75,1.35)	0.68	(0.45,0.90)	
TGCTAAAAGATCTAG_TGTGTTCAAGGGGCTG	2.35	(2.30,2.40)	0.49	(0.45,0.54)	
TGCTAAAAGATCTAG_TGTGTTCAAGGGGCTG_TTATCGGACAGCGTGC_CCAGCTGGTCCGCCAG_CATTGGAATCCGGTG	3.60	(3.30,3.90)	0.81	(0.53,1.04)	
TGCTAAAAGATCTAG_TGTGTTCAAGGGGCTG_TGAGGGGGCCGGTTTG_ATTGTCTATGCGTAAAC	3.50	(3.15,3.80)	0.36	(0.11,0.59)	
GAAGCGGTACGGGATT_ACGGCGAGGCGCAGAG	0.20	(-0.20,0.60)	0.66	(0.32,0.95)	
GAATAACGCTTCTGT_AAAGCATCCACGCGGG	1.25	(1.00,1.45)	0.35	(0.17,0.51)	
AACGAGGACGCGTTTT_CTCCTGGACACGTCTG	2.95	(2.80,3.15)	-2.38	(-2.82,-2.02)	
GAGAAAACCTAGTGTG_AGCTACCAATTAAGCA	0.80	(0.35,1.25)	-0.97	(-1.79,-0.39)	
GAAAAGAGTGACGTTT_CATTGGCTCACACAAG	1.40	(1.20,1.65)	0.21	(0.01,0.40)	
AACTTGAATAACCATG_GTATTAGGGGCACAAT	1.45	(0.90,1.95)	-1.40	(-2.62,-0.62)	
AACCGATGTTGCCCAA_TTCTCGAAGCACGTAG	1.70	(1.40,2.00)	0.30	(0.06,0.52)	
GTTAACGGCCAGTAC_CCGTGTACACGGCCAT	1.60	(1.25,1.90)	-0.08	(-0.39,0.18)	
TCAGGCCACGTAAGAG_CTGCGAATACGGTGTG	1.55	(1.40,1.65)	0.29	(0.16,0.41)	
GTTTGCAAGACAGCA_TGGAGAGGGACTGAGT	3.15	(2.80,3.50)	-2.00	(-2.77,-1.44)	
TAGTGGCTCCCGAA_CCTCTATCAATAGTTA	1.70	(1.45,1.95)	0.18	(-0.04,0.38)	
TCTGAGCCATAATAA_TAAGCAGCCACTGATC	1.10	(0.85,1.35)	-0.08	(-0.37,0.16)	
GTACCAAAGAGAGCAG_CGGGAGCGCTTGGGTT	1.75	(1.70,1.85)	0.14	(0.06,0.21)	
CAATGGTCTGATCTAT_CAGTATTGTGACGCAT	1.70	(1.40,2.00)	0.09	(-0.17,0.33)	
TTGTCTACAGCGTAA_CACGACTGGCCTTTT	1.85	(1.60,2.15)	0.40	(0.17,0.60)	
GTTTAATTTAATCGTG_TCCACTGCTTTAAGTG_TAACGAACACGTACGC	3.15	(3.05,3.25)	0.35	(0.25,0.43)	
GTTTAATTTAATCGTG_TCCACTGCTTTAAGTG_TAACGAACACGTACGC_TAGCCTGAGACTGTAT_CCGGTAGGTGGGACCT	4.60	(4.50,4.65)	0.57	(0.47,0.67)	
GTTTAATTTAATCGTG_TCCACTGCTTTAAGTG_TAACGAACACGTACGC_TAGCCTGAGACTGTAT_CCGGTAGGTGGGACCT_GATCTCITTTGACGGAG	4.95	(4.75,5.10)	1.26	(1.11,1.41)	
CGTATGAAACCTGGGG_GGCAGTTTTTGCGCGA_ATATTCTAGAAATTC	2.25	(1.90,2.55)	0.39	(0.15,0.60)	
TTATAAACCCCTTCC_CCTGGCGTGTGCAATC_GCAAACACGATACAT	1.95	(1.65,2.25)	0.33	(0.10,0.54)	
ACTTCGGGTGCTGTAG_GAACAGTCGAAGAGTA_TGGACATAATGATAGT_ACGGGCTTATCCGGAC	4.60	(4.45,4.75)	-0.27	(-0.46,-0.11)	
ACCCAGAAAAGACGCG_CGATCCGAACCTACCTA_ACCCTTATCGAGCGG_CCGTCAGCCCAAAAGG	1.55	(0.90,2.15)	0.88	(0.44,1.24)	
CTAGAGACACGTCTAC_ATGGTATATCAATGGC_CCACCTAGTCATGACT_CTGAGAGCGCGTTTGG	1.75	(1.20,2.30)	1.12	(0.69,1.47)	
TTGAGGGCAAGGCGC_ACGGACACCACTATAT_TTAGCTAACGAGTACC_GGGGAACAAATGAGC	1.95	(0.55,3.25)	-0.62	(-2.75,0.42)	
TAAATCTTGTGAGTCA_TCTGGACCCGGTAGAG_AATAAGCAAAGTCAGT_AAGGACATACTGTCCC	4.35	(4.05,4.60)	-0.30	(-0.62,-0.03)	

Supplementary Table 4: **Clone fitness estimates in the YPA population.** Clones are identified by the array of barcodes that label the founding mutation. The likelihood-based confidence intervals for the quoted best-estimates of their fitnesses in the two environments are calculated as explained in Section 6.1. The colors used to display each clone in Figures 2 and 3 of the Main Text are provided for convenience.



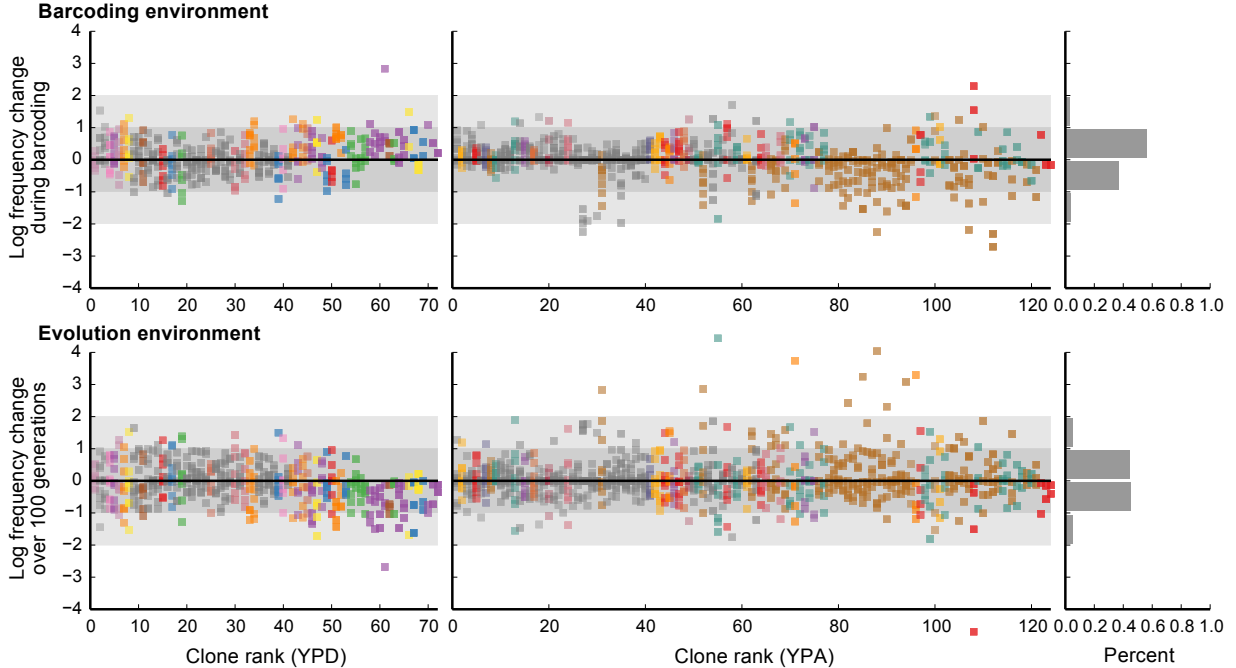
Supplementary Figure 8: **Comparison of the effect sizes of individual mutations on fitness in the evolutionary and barcoding conditions** Each point represents a mutation founding a called clonal lineage. Its effect on fitness in the two conditions was calculated by subtracting the fitness of the clone from the fitness of the clonal background on which it arose.

We note that the overall effect of selection in the evolutionary environment is several-fold stronger than the overall effect of selection in the barcoding environment (Supplementary Fig. 8), so the average fitness in each population is more strongly influenced by the evolution environment than the barcoding environment (Supplementary Fig. 10). As a result, the traveling-wave fitness distributions in both environments (and the patterns of jumps within them) are qualitatively similar regardless of whether we use the average fitness of each clone, or simply use the evolution-environment fitness and neglect the barcoding fitness effects (see Supplementary Fig. 11 and Supplementary Fig. 12).

6.3 Replication of dynamics without barcoding

One particularly striking aspect of the evolutionary dynamics we observe is the rapid expansion of a single clonal lineage (denoted in red) in the YPA population, which represents a significant leapfrogging event. Our analysis of the barcode frequency trajectories of this lineage prior to the large expansion observed during the 8th barcoding interval suggests that this lineage acquired a large-effect mutation in epoch 7, two epochs prior to the point at which the large expansion is seen in the barcoding interval. The inference of fitness from the barcode frequency data finds that this lineage acquired mutations that provide a significant selective advantage in both the evolution and barcoding conditions (see e.g. Supplementary Fig. 10). To verify these inferences, we re-evolved twelve replicates of the YPA population for 150 generations starting at timepoint 8.90, just prior to the barcoding process, but with no further re-barcoding.

After 150 generations, we found that all twelve populations had indistinguishable Sanger sequencing traces of the barcode locus, that all suggested a dramatic expansion of the same clonal lineage. To confirm this, we conducted Illumina sequencing of the barcode locus for two of these re-evolved populations (Supplementary Fig. 13). In both cases, we observed nearly identical lineage dynamics. In addition, in both replicate populations the sublineages of the red lineage show large and consistent increases in frequency over the course of the 150-generation period, confirming our prediction that the beneficial mutation arose and established prior to the addition of the 8th barcode. Thus these replay experiments confirm that the expansion of this red lineage is not an artifact of our barcoding procedure.



Supplementary Figure 9: **Changes in the log-frequencies of each called clone over the course of (top) a single barcoding interval or (bottom) a single evolution epoch.** (Left) Each point represents a clonal lineage. Lineages that appear in multiple epochs (or multiple barcoding intervals) are represented using multiple points, each corresponding to the change in a single epoch (or barcoding interval). To avoid taking the logarithm of a null quantity, the frequency at both the beginning and end of an epoch (or interval) have been increased by a small quantity, $\varepsilon = 10^{-4}$. Colors are consistent with Main Text Figure 2 and Supplementary Table 3 and Supplementary Table 4. Clones are ranked by the epoch in which they establish, with random ordering of clones establishing in the same epoch. (Right) Histograms of the log-frequency changes in the panels on the left.

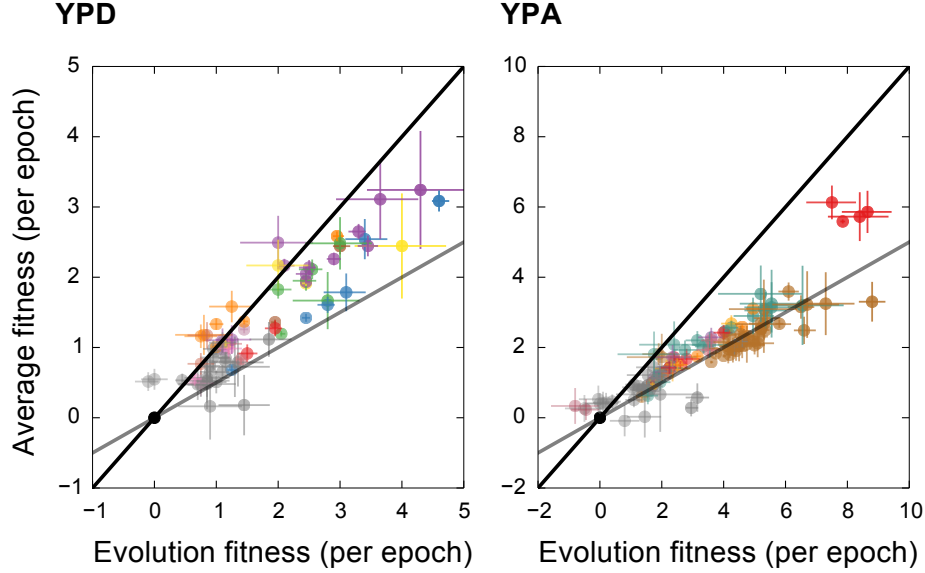
6.4 Genetic and fitness diversity

As can be seen in Figure 3 of the Main Text, the populations contain a large number of diverse genotypes over the entirety of the experiment. An evolutionarily important measure of this diversity is the variance in fitness in a population, which we denote $\sigma^2(t)$. This quantity controls the rate of adaptation, $\frac{d\bar{x}}{dt}$, which in the absence of recombination obeys

$$\frac{d\bar{x}(t)}{dt} = \sigma^2(t) + \mu \langle s \rangle, \quad (26)$$

where $\langle s \rangle$ denotes the average effect size of new mutations and μ denotes the per-generation per-individual mutation rate. In the absence of the mutation term, $\mu \langle s \rangle$, Equation 26 is known as R. A. Fisher’s “fundamental theorem of natural selection” [46]. We will see that in our case, $\sigma^2(t)$ is of order 10^{-4} , while $\mu \langle s \rangle \approx 10^{-5} \cdot 10^{-2} \approx 10^{-7}$, making this term subdominant in our populations. We calculate $\sigma^2(t)$ directly from our clone frequency trajectories directly according to

$$\sigma^2(t) = \sum_i (x_i - \bar{x}(t))^2 \frac{r_i(t)}{R_t}. \quad (27)$$



Supplementary Figure 10: **Average fitness compared to the evolution environment fitness for each called clone.** Each point represents a called clonal lineage, and the error bars denote 95% confidence intervals. The likelihood-based confidence intervals for the quoted best-estimates of their fitnesses in the two environments are calculated as explained in Section 6.1. The ancestor is shown in black. On the black line, the per epoch average fitness equals the per-epoch evolution fitness (expected if barcoding fitness equals evolution fitness), and on the gray line the per-epoch average fitness is equal to half the per-epoch evolution fitness (expected if there are no barcoding fitness effects). Note the strong positive correlation between the average fitness and the evolution fitness in both populations, indicating that selection during barcoding is a small perturbation. Colors are consistent with Main Text Figure 2 and Supplementary Table 3 and Supplementary Table 4.

Similarly, we can obtain the rate of adaptation according to

$$\frac{d\bar{x}(t)}{dt} = \frac{[\bar{x}(t) - \bar{x}(t - \Delta t)]}{\Delta t}. \quad (28)$$

In the evolution condition only, we further convolve the instantaneous rate of adaptation with a Gaussian smoother with a kernel of $\Delta t = 10$ generations before plotting on Supplementary Fig. 14.

To quantify how the diversity in the population is distributed among the lineages, we use the Shannon entropy,

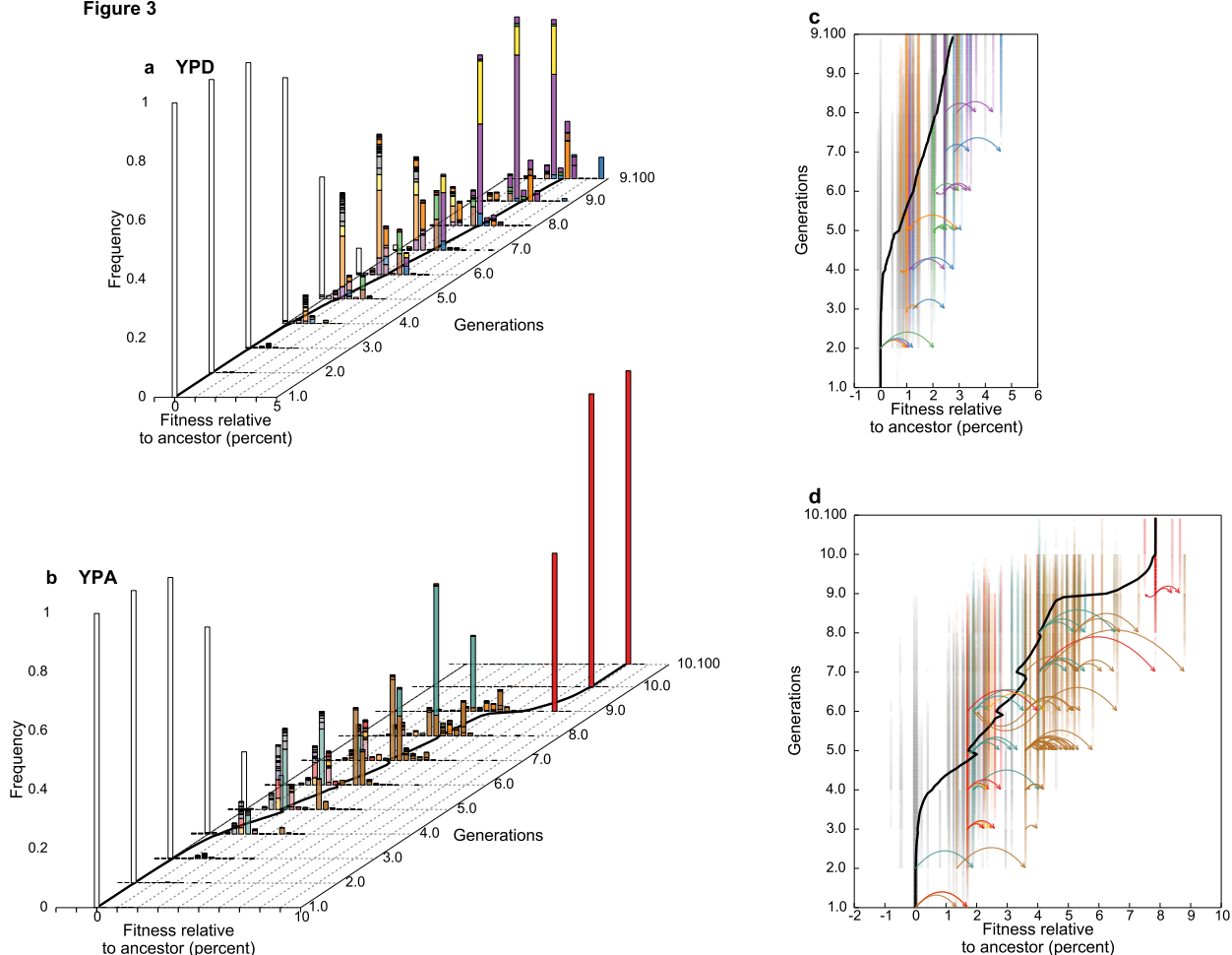
$$S(t) = \left\langle -\log \frac{r_i(t)}{R_t} \right\rangle_i, \quad (29)$$

where the expectation is taken over all clones i including the ancestor. In analogy to this measure for the diversity of lineage frequencies, we also define “fitness entropy” according to

$$\Sigma(t) = \langle -\log q_i(t) \rangle_i, \quad (30)$$

where $q_i(t) = \frac{r_i(t)(x_i - \bar{x}(t))^2}{R_t \sigma^2(t)}$ denotes the proportion of the fitness variance contributed by lineage i . We expect the fitness entropy to be low whenever there is a large outlier in the fitness distribution, but then to rapidly recover as further mutations establish on its background.

Figure 3



Supplementary Figure 11: **Traveling wave dynamics in the evolution environment.** This is a replot of Main Text Figure 3, but with each lineage shown at its evolution fitness.

6.5 Probability of acquiring further established beneficial mutations

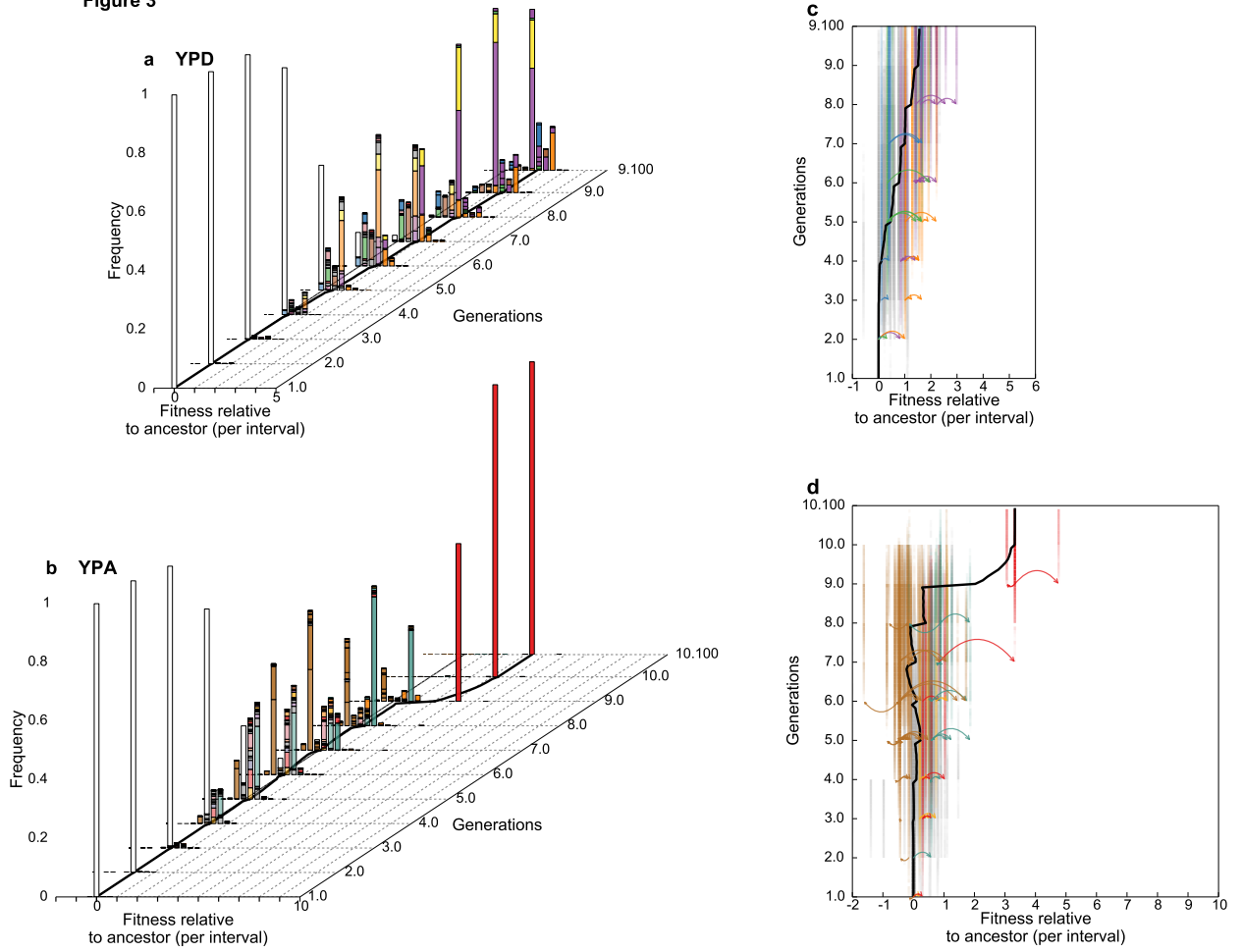
As we discuss in the Main Text, whether or not a lineage acquires further beneficial mutations is a major factor in determining its long-term success. In this section, we consider two simple models of the probability of acquiring further beneficial mutations, and test to what extent these models explain the variability in the numbers of mutations detected in each epoch and clone.

The simplest null hypothesis is that, on average, each lineage acquires a fraction of all the beneficial mutations that arise in the experiment that is proportional to the integral of its frequency over time. To calculate the expected distribution of mutations among the lineages and epochs under this model, we redistributed the mutations observed in the experiment among the different clonal lineages. This was done by resampling the background of each mutation, so that the probability that each mutation established on the background of clone i in epoch E is equal to

$$p_{i,E} = \frac{\sum_{t=E,0}^{t=E,100} f_i(t)}{\sum_j \sum_{E'} \sum_{t=E',0}^{t=E',100} f_j(t)}, \quad (31)$$

where $f_i(t)$ denotes the frequency of clone i at time t , the summation in the denominator runs over

Figure 3



Supplementary Figure 12: **Traveling wave dynamics in the barcoding environment.** This is a replot of Main Text Figure 3, but with each lineage shown at its barcoding condition fitness.

all lineages j , and all epochs E' between which mutations are being permuted.

We exclude from this resampling procedure the last epoch of the experiment in each population, because mutations establishing in this epoch are unlikely to be detected by our inference pipeline (see Supplementary Fig. 7). Indeed, in both populations, all detected mutations established prior to this epoch (see Main Text Figure 2). We also exclude from this procedure the first two epochs of the experiment in each population, for reasons we describe in our description of the second model below. In each of 10^4 redistributions of all the mutations that were detected in other epochs (3-8 in YPD, 3-9 in YPA), we recorded the total number of mutations $m_{i,E,\text{trial}}$ that established on the background of clone i during epoch E . We calculated a P -value for each clone i and epoch E under this model according to,

$$P_{i,E} = \Pr[m_{i,E,\text{model}} \geq m_{i,E,\text{exp}}], \quad (32)$$

as well as a P -value for the overall number of mutations acquired by each clone as

$$P_i = \Pr \left[\sum_{E=1}^{K-1} m_{i,E,\text{model}} \geq \sum_{E=1}^{K-1} m_{i,E,\text{exp}} \right], \quad (33)$$

where $m_{i,E,\text{exp}}$ denotes the total number of mutations establishing on the background of i in epoch E in the experiment.

We also constructed a second model in which the probability that a mutation established on the background of a clone was modulated by its own average fitness effect and the relative fitness of the background. As we explain in the Main Text, in a rapidly adapting population, variation in fitness between clones in the population creates variation in the probability that beneficial mutations establish once they arise. To establish, a lineage must escape the effects of drift, and the probability of this event increases with the difference between its fitness and the mean fitness of the population. To account for this, for each mutation k , we calculated the probability that it establishes on the background of clone i in epoch E according to

$$p_{i,E} = \frac{\sum_{t=E.0}^{t=E.100} f_i(t) p_{\text{est}} \left(x_i^{(\text{avg})} + s_k^{(\text{avg})}, \bar{x}^{\text{avg}}(t), t \right)}{\sum_j \sum_{E'} \sum_{t=E'.0}^{t=E'.100} f_j(t) p_{\text{est}} \left(x_i^{(\text{avg})} + s_k^{(\text{avg})}, \bar{x}^{\text{avg}}(t), t \right)}, \quad (34)$$

where $x_i^{(\text{avg})}$ is the average fitness of clone i (Equation 25), $s_k^{(\text{avg})}$ is the average fitness effect of mutation k , $\bar{x}^{(\text{avg})}(t)$ is the mean average fitness of the population, and $p_{\text{est}}(x, \bar{x}(t), t)$ denotes the establishment probability of a mutation with fitness x , conditioned on the average fitness of a mutation of the population being $\bar{x}(t)$.

In rapidly adapting populations, the general form of $p_{\text{est}}(x, \bar{x}(t), t)$ is not known, and we expect it to subtly depend on the mean fitness of the population, $\bar{x}(t)$, the fitness of the created mutant x , and also to depend explicitly on time [47]. This is because establishment is an extended process, and the chance of a mutation escaping drift depends not only on its relative fitness at the time of founding, but also on how its relative fitness changes over the course of establishment as the population adapts further, and also explicitly on time. Specifically, in a population with an approximately constant $\bar{x}(t) \approx \bar{x}$, a lineage at relative fitness $x - \bar{x}$ should take about $\frac{1}{x - \bar{x}}$ generations to establish, which is why we do not expect many lineages to establish during the first epoch.

However, our main purpose here is not to precisely quantify the establishment probability at every timepoint and for every background, but rather to estimate whether or not any of the ‘‘bursts’’ in the numbers of mutations observed in an epoch are larger than we would expect based on the frequency of the mutation, and to see whether conservatively accounting for the fitnesses in the experiment changes our expectations. To do this, we use an approximate form for the establishment probability:

$$p_{\text{est}}(x_i^{(\text{avg})} + s_k^{(\text{avg})}, \bar{x}^{\text{avg}}(t), t) = \begin{cases} 0, & \text{if } x_i^{(\text{avg})} + s_k^{(\text{avg})} - \langle \bar{x}^{\text{avg}}(t) \rangle \leq X_c, \\ x_i^{(\text{avg})} + s_k^{(\text{avg})} - \langle \bar{x}^{\text{avg}}(t) \rangle, & \text{otherwise,} \end{cases} \quad (35)$$

where $\langle \bar{x}^{(\text{avg})}(t) \rangle$ denotes the mean average fitness of the population in the relevant epoch. In both populations, we choose $X_c = 5 \cdot 10^{-5}$, which is on the order of the strength of genetic drift, $\frac{2b}{N_b}$. With this choice, Equation 35 is expected to be accurate for sufficiently high relative fitnesses and sufficiently low relative fitnesses, and to slightly overestimate the establishment probabilities on backgrounds of intermediate relative fitness [47]. This means that our choice represents a conservative favoring of backgrounds of higher fitness that is consistent with the effects of genetic drift. Note Equation 35 still does not account for the amount of time it takes for a mutation to establish, leading to an overestimation of the expected number of established mutations in the first epoch, which is why we exclude that epoch from consideration. It also does not account for the changing rate of adaptation in the experiment (see Supplementary Fig. 14), which leads to the underestimation of the establishment probability in the second epoch relative to the establishment probabilities in other epochs. For that reason, we also exclude this epoch from consideration.

By repeating a procedure analogous to the one that we described with reference to the frequency-only null model, we can obtain a P -value for each clone and epoch under this updated model. As we can see from the P -values for the observed number of mutations for each clone and epoch under the frequency-only model (Supplementary Fig. 15), a small number of epochs are found to have a significant excess of mutations in the experiment under the frequency only model, corresponding to the “bursts” visible on Main Text Figures 2 and 3. However, under the model that also takes fitness into account, the majority of these segments are assigned substantially higher P -values, and only a single segment in the YPA population remains marginally significant at an FDR of 5% (Supplementary Fig. 16). Thus, the apparent “bursts” identified assuming a frequency-only model are no longer significant under a model that conservatively takes into account fitness differences between the backgrounds, and are therefore not strong evidence of mutator phenotypes.

This lack of evidence for mutator phenotypes is further underscored by the P -values for the overall number of mutations observed for each clone. If a burst was the result of a mutator phenotype, we would expect that a clone acquires more mutations than expected by chance over multiple epochs. However, as we can see on Supplementary Fig. 17, there is limited evidence that any clone is accumulating more mutations than expected by chance. Under both models, no clones have a significant enough P -value to be detected at an FDR of 5%. Furthermore, we see that when mutations are aggregated over many epochs, the P -values calculated under the two models are roughly concordant.

6.6 Comparison with fitness measurements using competitive fitness assays

As described in Section 3, we use competitive fitness assays to obtain independent measurements of the population mean fitness in the evolution condition, which we can compare to the inferred mean fitness trajectory $\bar{x}(t)$ in the same condition. As can be seen in Extended Data Figure 2, the inferred mean fitness trajectory reproduces the measured trajectory of the population mean fitness in both shape and magnitude, suggesting that we have detected the majority of established beneficial mutations. However, we note that there are aspects of the inferred mean fitness trajectory that are inconsistent with the results of our competitive fitness assays. For instance, in the YPA population, the $\sim 3\%$ increase in fitness of the population over the course of Epoch 3 is not consistent with our barcode sequence data. In addition to this, we infer a $\sim 2\%$ increase in fitness over the course of the barcoding period during which the 9th barcode is inserted that is not seen in fitness assays. Similar effects can be seen in the YPD population, where we find that the measured mean fitness declines systematically during population barcoding.

These inconsistencies could arise either from inaccuracies in our inferences or from errors in our competitive fitness assays. While it is likely that there is some contribution from both effects, the discrete nature of the inconsistencies (which are concentrated in a few epochs) and the fact that these are not reflected in barcoding data from these timepoints suggest that the dominant contribution is due to errors in competitive fitness assays. These could arise from factors such as intransitive effects (in which case our inferred fitnesses are a better description of the relevant instantaneous within-population selective differences), or from technical issues such as unfreezing artifacts or other complications due to measuring fitness immediately after barcoding interval recovery steps. The fact that the inferred rate of change in mean fitness is well explained by the inferred variance in fitness (Supplementary Fig. 14), as expected from the fundamental theorem of natural selection, provides further evidence in support of the accuracy of our inferred fitnesses.

7 Metagenomic sequencing and sequence data analysis

Genomic DNA was sequenced at the end of each epoch using previously described methods with no modifications [48]. We trimmed the raw sequencing reads using `trimmomatic v0.35` [49] and then used `breseq v0.27.1b` [50] to align the reads to the reference BY4742 genome. The sequence of BY4742 was obtained from SGD. Since this SGD reference was generated using short-read sequencing, three genes were modified to better agree with recent Pacbio sequence data [51]: *FLO9*, *FLO1*, and *YMR317W*. The *WHI2* gene in the Toronto reference sequence contains a nonsense mutation, but sequencing of our strain revealed no mutation when compared with S288C. Finally, a small section of the gene *YDR544C* was also changed to agree with the sequence of our strain. We found that these changes prevent mapping errors that sometimes show up when the coverage is low. The changes in *FLO1* and *FLO9* are particularly important because they are in repeat regions.

Candidate SNVs and small indels were identified from the `breseq` output using the `breseq-lite` scripts built to identify these variants in metagenomic sequence time-series data in *E. coli* [52]. This pipeline merges all alternate alleles at a site at which an indel was detected into a single alternative read, but distinguishes between different SNVs at the same site. Following Ref. [52], we keep for further analysis only candidate variants that satisfy the basic quality standard of having at least two time-points in which the number of alternative reads exceeds 2, and require that at least at one of those time-points the overall read depth at that site is at least 10 and that the frequency of the alternate variant is at least 5%. We also exclude all variants occurring at sites within 100bp of repetitive regions based on the BY4742 reference genome annotation.

To distinguish between true mutations and errors, we use a similar approach to Ref. [52], where we require the trajectory to have a strong autocorrelation, start with low frequency and have a large area under the curve compared to a null model based on a permutation of the observed trajectory. However, we impose more stringent quality controls on individual time-courses and time-points. In addition to the basic quality criterion above, we exclude time-points for which the median depth was lower than 10, and require for each individual trajectory

1. There are at least two time-points in which the alternative allele is supported by 5 reads, the overall read depth is at least 10, and the frequency of the alternative allele is at least 5%.
2. The alternative allele frequency in the first time-point does not exceed 5%, and its combined frequency in the first two time-points does not exceed 20%.
3. The difference between the maximum frequency and the capped average, $\bar{f}^* = \min\{\bar{f}, 0.5\}$, is at least 0.1.

Furthermore, due to the limited length of our time-course, we do not attempt to model changes in the depth of the trajectory to infer insertion and deletion events.

For each trajectory, we calculate the composite P -value, based on the autocorrelation, derived allele sojourn weight and average frequency relaxation time statistics compared to a permutation-based null model as in Ref. [52], with small modifications to the average frequency relaxation time statistic, which we calculate according to

$$T = \max \left\{ T : \frac{\sum_{t \leq t'} A_{pmt}}{\sum_{t \leq t'} D_{pmt}} \leq 0.5 \frac{\sum_t A_{pmt}}{\sum_t D_{pmt}} \quad \forall \quad 0 \leq t' \leq T \right\}. \quad (36)$$

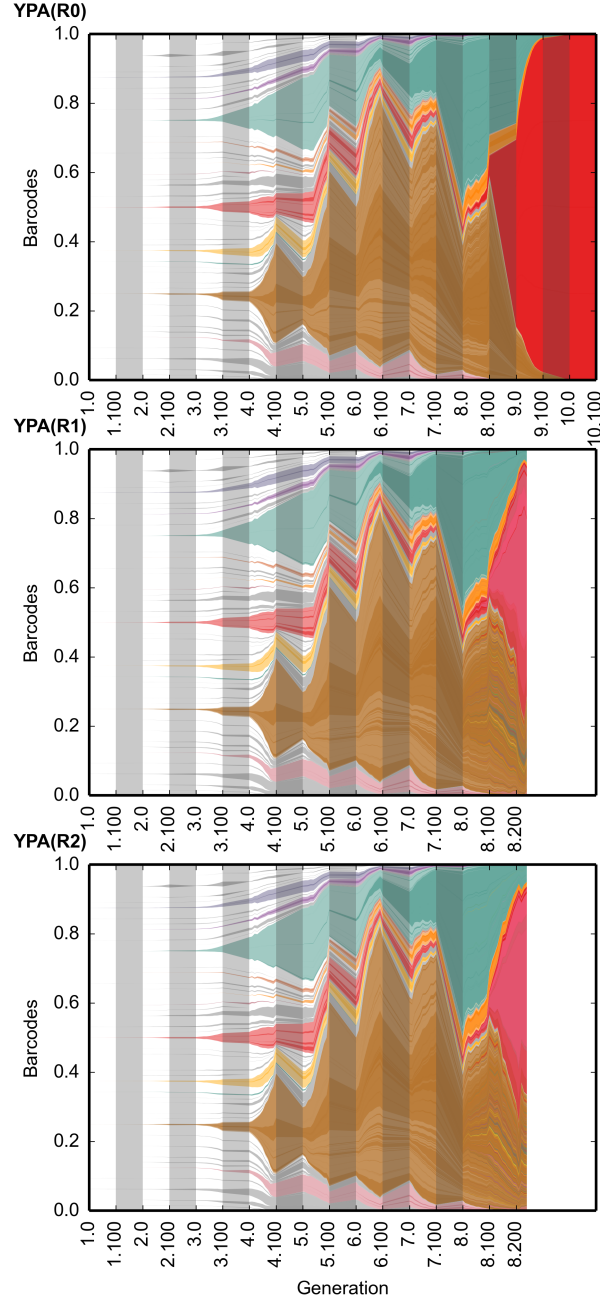
Here A_{pmt} and D_{pmt} refer to the alternative read count and the overall sequencing depth of locus m in population p at time t . We reject all trajectories with composite $P \geq 0.05$. Trajectories with $P < 0.05$ are shown in Extended Data Figure 1.

We note that several of the mutations called in our metagenomic sequencing data have been identified in previous studies or have known functional effects relevant to our experimental conditions. For example, among mutations observed in our YPD population, *HST4* is *SIR2* homolog, which is a frequent target of selection in earlier experiments [53], and *CCR4* has been hypothesized as a target of selection in low glucose [54]. Among mutations observed in our YPA population, *BNA3* is known to increase acetic acid tolerance [55], *ASG1* and *ASC1* are previously observed targets of selection for acetic acid tolerance [56, 57], and the role of *OAF1* in acetic acid tolerance has been previously implicated in Ref [58].

7.1 Consistency between metagenomic and barcode sequencing data

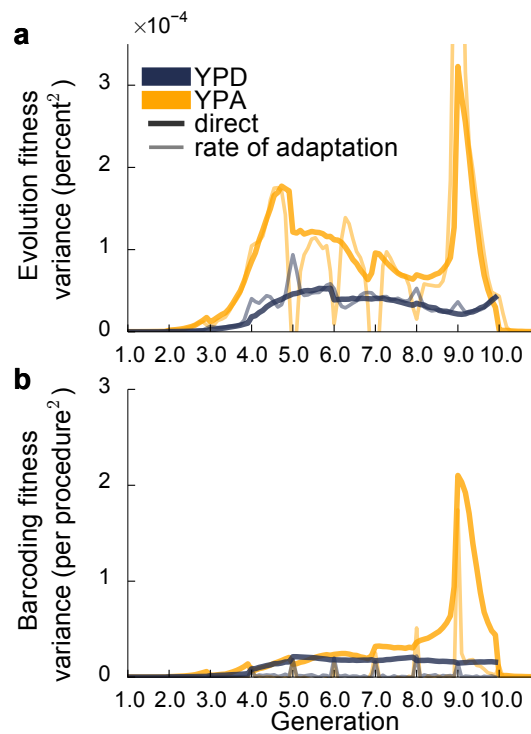
The number of mutations called in each of the populations from metagenomic sequence data is vastly lower than the number of lineages observed during evolution. Note however that these two datasets are broadly consistent when the difference in resolution between the two is taken into account. To demonstrate this, we simulated the process of whole genome sequencing and mutation calling from metagenomic data. Specifically, we treated the clone frequency trajectories called from the barcode sequencing data as the true frequencies of clones segregating in the population and assigned each of these clones a unique “driver” mutation and a number of unique “passenger” mutations relative to the background that it arose on. All driver and passenger mutations are inherited by all children of a given clone. For each clone, the number of passenger mutations was sampled from a Poisson distribution with a mean of 5. This mean was chosen to produce rough agreement with the cohort sizes observed in Ref [25], by doubling the mean cohort size observed in the haploid populations sequenced in that study. Note that this may still represent an underestimate of the typical number of passenger mutations, because Ref [25] also employed a conservative method for distinguishing true mutations from sequencing errors, though we emphasize that the frequency resolution of metagenomic sequencing of that study is about an order of magnitude higher than the frequency resolution of our study.

To simulate the allele frequency trajectories of all of these alleles, we simulated the sequencing separately at each of the driver and passenger loci, m . At each timepoint t , we first sampled the overall sequencing depth, D_{pmt} at the locus of the mutation from a Poisson distribution with a mean equal to the median sequencing depth at that timepoint, $\langle D_{pmt} \rangle_m$. To account for the fact that our asexually reproducing population is diploid, we randomly apportioned the reads to the chromosome carrying the mutation and to the wild type sister chromosome by sampling from a binomial distribution with $p = 0.5$ and $n = D_{pmt}$, denoting the number that map to the sister chromosome with d_{pmt} . At each locus, we also allowed for gene conversion shortly after a mutation arose, with an overall probability of 0.01, which would lead to the mutation being present on both alleles. Finally, we sampled the number of reads that map to the alternate allele from a binomial distribution with $p = f_{pmt}$ and $n = (D_{pmt} - d_{pmt}) + \mathcal{I}_{\text{conversion}}d_{pmt}$, where $\mathcal{I}_{\text{conversion}}$ denotes a gene conversion event. Finally, all of the mutational frequency trajectories generated in this way were subjected to the same filters as the data. In Supplementary Figure 18, we show 4 simulated datasets generated using this procedure. The colors of these mutational trajectories are consistent with the colors of the clonal backgrounds they founded, as shown in Main Text Figures 2 and 3. The number of mutations observed in these simulated datasets is broadly consistent with the number of mutations called from our metagenomic sequencing data.

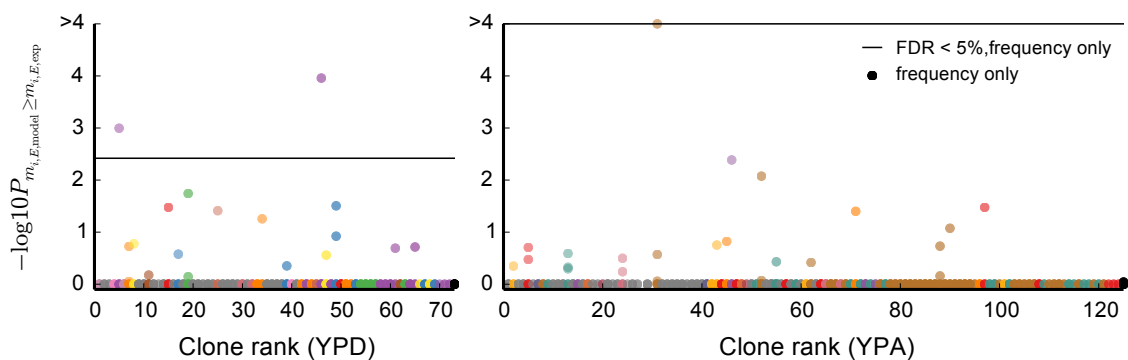


Supplementary Figure 13: **Replication of evolutionary dynamics in the YPA population.**

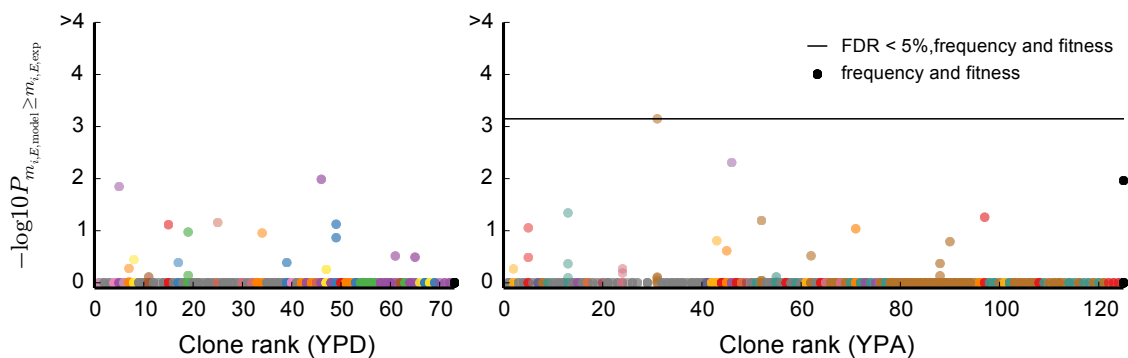
Top panel shows the clonal dynamics in our original evolved population. Bottom two panels show the corresponding lineage dynamics in the two Illumina sequenced replicate populations in which evolution is conducted without barcoding. Note that during the replay portion of these experiments, from timepoint 8.0 onwards, we do not group barcoded lineages into clones, but instead show all sublineages that exceed frequency of 0.1%. The colors of the sublineages are chosen to be consistent between the two replicates.



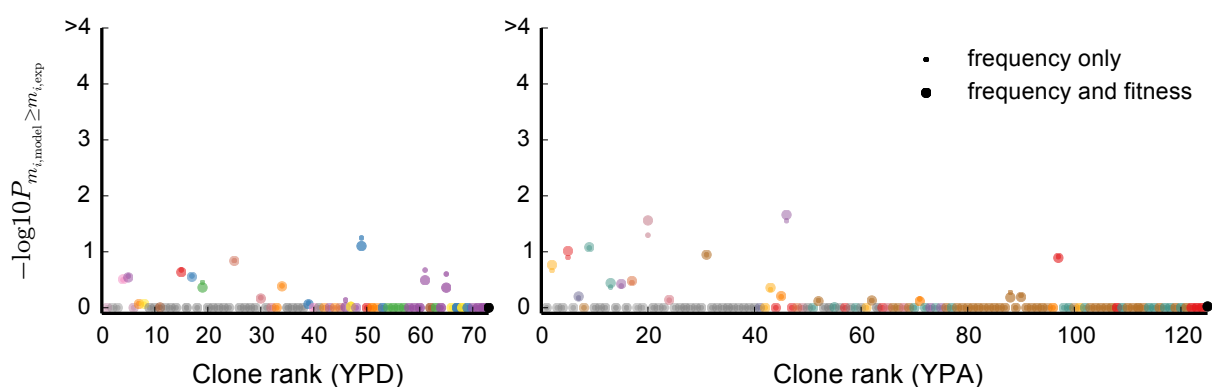
Supplementary Figure 14: **Variance in fitness through time** Thick lines show direct measurements of fitness variance based on the inferred fitness and frequency of each clonal lineage in (a) the evolution environment and (b) the barcoding environment. Thin lines show the rate of change in the mean fitness; note the good agreement between this rate of change and the variance in fitness in the current environment, as expected from the fundamental theorem of natural selection.



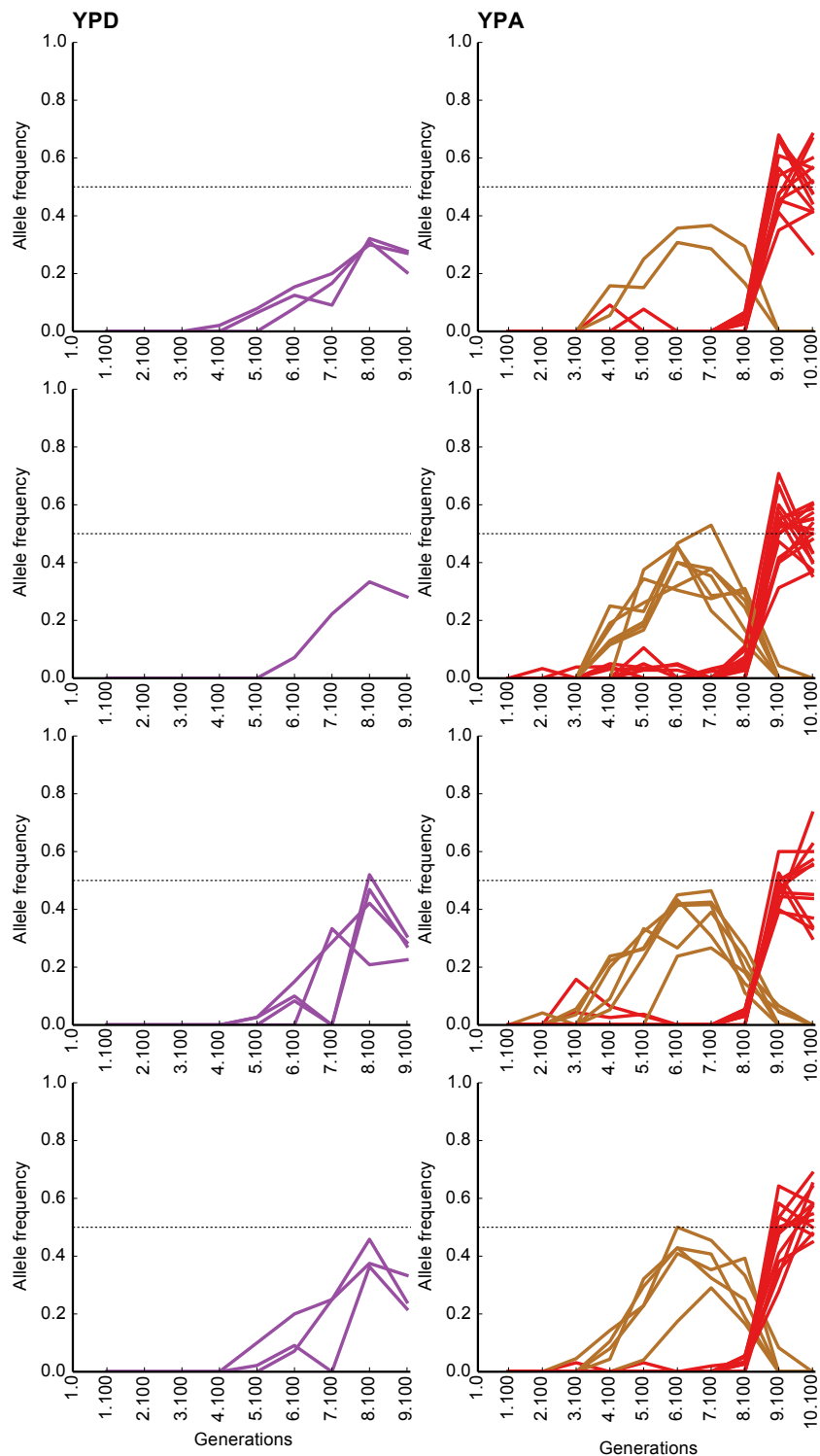
Supplementary Figure 15: **Probabilities of accumulating an equal or larger number of mutations in each epoch based on a clone’s frequency.** Each clonal lineage is represented by a separate point for each epoch in which it is observed. The P -values were calculated from 10^4 redistributions of all mutations detected in epochs 3-8 for the YPD population, or epochs 3-9 for the YPA population under the frequency-only null model. This null model is described in detail in Section 6.5 and is defined by Equation 31. The ancestor is colored black, and all other colors are consistent with Main Text Figure 2 and Supplementary Table 3 and Supplementary Table 4. Clones are ranked by the epoch in which they establish, with random ordering of clones establishing in the same epoch.



Supplementary Figure 16: **Probabilities of accumulating an equal or larger number of mutations in each epoch based on a clone’s frequency and fitness.** Each clonal lineage is represented by a separate point for each epoch in which it is observed. The P -values were calculated from 10^4 redistributions of all mutations detected in epochs 3-8 for the YPD population, or epochs 3-9 for the YPA population under the frequency plus fitness null model. This null model is described in detail in Section 6.5 and is defined by Equation 34 and Equation 35. The ancestor is colored black, and all other colors are consistent with Main Text Figure 2 and Supplementary Table 3 and Supplementary Table 4. Clones are ranked by the epoch in which they establish, with random ordering of clones establishing in the same epoch. Note that no line appears in the YPD panel, because no segments are significant at an FDR of 5%.



Supplementary Figure 17: **Probabilities of accumulating an equal or larger number of mutations over the course of the experiment.** Each clonal lineage is represented by two points: small points denote P -values under the frequency-only mode (defined by Equation 31) and circles denote P -values under the frequency plus fitness model (defined by Equation 34 and Equation 35). For each of the null models, the P -values were calculated from 10^4 redistributions of all the mutations detected in epochs 3-8 for the YPD population, or epochs 3-9 for the YPA population. The ancestor is colored black, and all other colors are consistent with Main Text Figure 2 and Supplementary Table 3 and Supplementary Table 4. Clones are ranked by the epoch in which they establish, with random ordering of clones establishing in the same epoch. Note that in either population, no clones accumulate a significant number of mutations at an FDR of 5%.



Supplementary Figure 18: **Simulated metagenomic sequencing datasets.** Each panel represents an independent simulated metagenomic sequencing dataset. In each panel, lines represent frequency trajectories of individual mutations. The colors of these trajectories are consistent with the colors of the clonal backgrounds that these mutations correspond to, as used in Main Text Figures 2 and 3.

Supplementary Data Tables

Supplementary Table 5: **List of all barcoded lineages and their frequencies in each sequencing time-point.** The table contains the population barcode and all other lineage-specific barcodes for each lineage that passed basic quality filters outlined in Section 2, along with the time-course of the fraction of reads mapping to that lineage in each sequencing time-point.

Supplementary Table 6: **List of total lineage read counts in each sequencing time-point.** The table contains the generation marker for each sequencing time-point, as well as the total number of reads recovered after error correction.

Supplementary Table 7: **List of all mutations called from metagenomic sequence data.** The table contains the genomic position and identity of each mutation that had a trajectory with composite $P < 0.05$, along with its alternate allele read trajectory, overall depth trajectory, auto-correlation, and P -value.

References

- [1] James Hose, Chris Mun Yong, Maria Sardi, Zhishi Wang, Michael A Newton, and Audrey P Gasch. Dosage compensation can buffer copy-number variation in wild yeast. *eLife*, 4:e05462, may 2015.
- [2] Anna M. Selmecki, Yosef E. Maruvka, Phillip A. Richmond, Marie Guillet, Noam Shoresh, Amber L. Sorenson, Subhajyoti De, Roy Kishony, Franziska Michor, Robin Dowell, and David Pellman. Polyploidy can drive rapid adaptation in yeast. *Nature*, 519:349–352, 2015.
- [3] Laura M. Wingler and Virginia W. Cornish. Reiterative recombination for the in vivo assembly of libraries of multigene pathways. *Proceedings of the National Academy of Sciences*, 108(37):15135–15140, 2011.
- [4] Soumen Nandy, Shan Zhao, Bhuvan P. Pathak, Muthusamy Manoharan, and Vibha Srivastava. Gene stacking in plant cell using recombinases for gene integration and nucleases for marker gene deletion. *BMC Biotechnology*, 15(1):93, Oct 2015.
- [5] Li Lin, Yao-Guang Liu, Xinping Xu, and Baojian Li. Efficient linking and transfer of multiple genes by a multigene assembly and transformation vector system. *Proceedings of the National Academy of Sciences*, 100(10):5962–5967, 2003.
- [6] E-Chiang Lee, Qi Liang, Hanif Ali, Luke Bayliss, Alastair Beasley, Tara Bloomfield-Gerdes, Laura Bonoli, Richard Brown, Jamie Campbell, Adam Carpenter, Sara Chalk, Alison Davis, Nick England, Alla Fane-Dremucheva, Bettina Franz, Volker Germaschewski, Helen Holmes, Steve Holmes, Ian Kirby, Miha Kosmac, Anais Legent, Hui Lui, Anais Manin, Siobhan O’Leary, Jemima Paterson, Rocco Sciarillo, Anneliese Speak, Dominik Spensberger, Laura Tuffery, Nikole Waddell, Wei Wang, Sophie Wells, Vivian Wong, Andrew Wood, Michael J Owen, Glenn A Friedrich, and Allan Bradley. Complete humanization of the mouse immunoglobulin loci enables efficient therapeutic antibody discovery. *Nature Biotechnology*, 32:356—363, 2014.
- [7] Lili Hou, Yuan-Yeu Yau, Junjie Wei, Zhiguo Han, Zhicheng Dong, and David W. Ow. An open-source system for in planta gene stacking by bxb1 and cre recombinases. *Molecular Plant*, 7:1756—1765, 2014.
- [8] Weiqiang Chen and David W Ow. Protocol for in vitro stacked molecules compatible with in vivo recombinase-mediated gene stacking. *Methods in molecular biology (Clifton, N.J.)*, 1469:31—47, 2016.
- [9] Gwang Lee and Izumu Saito. Role of nucleotide sequences of loxp spacer region in cre-mediated recombination. *Gene*, 216(1):55 – 65, 1998.
- [10] Stephen J. Langer, A. Paiman Ghafoori, Marshall Byrd, and Leslie Leinwand. A genetic screen identifies novel noncompatible loxp sites. *Nucleic Acids Research*, 30(14):3067–3077, 2002.
- [11] Henrik Albert, Emily C. Dale, Elsa Lee, and David W. Ow. Site-specific integration of dna into wild-type and mutant lox sites placed in the plant genome. *The Plant Journal*, 7(4):649–659, 1995.
- [12] S F Levy, J R Blundell, S Venkataram, D A Petrov, D S Fisher, and G Sherlock. Quantitative evolutionary dynamics using high-resolution lineage tracking. *Nature*, 519:181–186, 2015.

- [13] Kimi Araki, Yuka Okada, Masatake Araki, and Ken-ichi Yamamura. Comparative analysis of right element mutant lox sites on recombination efficiency in embryonic stem cells. *BMC Biotechnology*, 10(1):29, Mar 2010.
- [14] James G. Thomson, Edmund B. Rucker, and Jorge A. Piedrahita. Mutational analysis of loxp sites for efficient cre-mediated insertion into genomic dna. *genesis*, 36(3):162–167, 2003.
- [15] Jamie Sheren, Stephen J. Langer, and Leslie A. Leinwand. A randomized library approach to identifying functional lox site domains for the cre recombinase. *Nucleic Acids Research*, 35(16):5464–5473, 2007.
- [16] Jean Livet, Tamily A. Weissman, Hyuno Kang, Ryan W. Draft, Ju Lu, Robyn A. Bennis, Joshua R. Sanes, and Jeff W. Lichtman. Transgenic strategies for combinatorial expression of fluorescent proteins in the nervous system. *Nature*, 450:56–62, 2007.
- [17] Carrie Baker Brachmann, Adrian Davies, Gregory J. Cost, Emerita Caputo, Joachim Li, Philip Hieter, and Jef D. Boeke. Designer deletion strains derived from *saccharomyces cerevisiae* s288c: A useful set of strains and plasmids for pcr-mediated gene disruption and other applications. *Yeast*, 14(2):115–132, 1998.
- [18] Adam M Deutschbauer and Ronald W Davis. Quantitative trait loci mapped to single-nucleotide resolution in yeast. *Nature Genetics*, 37:1333–1340, 2005.
- [19] Francesca Storici, L. Kevin Lewis, and Michael A. Resnick. In vivo site-directed mutagenesis using oligonucleotides. *Nature Biotechnology*, 19:773–776, 2001.
- [20] Alan L. Goldstein and John H. McCusker. Three new dominant drug resistance cassettes for gene disruption in *saccharomyces cerevisiae*. *Yeast*, 15(14):1541–1553, 1999.
- [21] Kathleen A. Curran, Nicholas J. Morse, Kelly A. Markham, Allison M. Wagman, Akash Gupta, and Hal S. Alper. Short synthetic terminators for improved heterologous gene expression in yeast. *ACS Synthetic Biology*, 4(7):824–832, 2015. PMID: 25686303.
- [22] A H Tong and C Boone. Synthetic genetic array analysis in *saccharomyces cerevisiae*. *Methods Mol Biol.*, 313:171–92, 2006.
- [23] Carola Engler, Romy Kandzia, and Sylvestre Marillonnet. A one pot, one step, precision cloning method with high throughput capability. *PLOS ONE*, 3(11):1–7, 11 2008.
- [24] Michael Hanscho, David E. Ruckerbauer, Neha Chauhan, Harald F. Hofbauer, Stefan Krahulec, Bernd Nidetzky, Sepp D. Kohlwein, Juergen Zanghellini, and Klaus Natter. Nutritional requirements of the by series of *saccharomyces cerevisiae* strains for optimum growth. *FEMS Yeast Research*, 12(7):796–808, 2012.
- [25] Gregory I Lang, Daniel P Rice, Mark J Hickman, Erica Sodergren, George M Weinstock, David Botstein, and Michael M Desai. Pervasive Genetic Hitchhiking and Clonal Interference in 40 Evolving Yeast Populations. *Nature*, 500(7464):571–574, August 2013.
- [26] Isaac Kinde, Jian Wu, Nick Papadopoulos, Kenneth W. Kinzler, and Bert Vogelstein. Detection and quantification of rare mutations with massively parallel sequencing. *Proceedings of the National Academy of Sciences*, 108(23):9530–9535, 2011.

- [27] Derek S Lundberg, Scott Yourstone, Piotr Mieczkowski, Corbin D Jones, and Jeffery L Dangel. Practical innovations for high-throughput amplicon sequencing. *Nature Methods*, 10:999–1002, 2013.
- [28] Gregory I. Lang, David Botstein, and Michael M. Desai. Genetic variation and the fate of beneficial mutations in asexual populations. *Genetics*, 188(3):647–661, 2011.
- [29] J P Bollback, T L York, and R Nielsen. Estimation of $2n_e s$ from temporal allele frequency data. *Genetics*, 179:497–502, 2008.
- [30] A-S Malaspinas, O Malaspinas, S N Evans, and M Slatkin. Estimating allele age and selection coefficient from time-serial data. *Genetics*, 192:599–607, 2012.
- [31] I Mathieson and G McVean. Estimating selection coefficients in spatially structured populations from time series data of allele frequencies. *Genetics*, 193:973–984, 2013.
- [32] Matthieu Foll, Yu-Ping Poh, Nicholas Renzette, Anna Ferrer-Admetlla, Claudia Bank, Hyunjin Shim, Anna-Sapfo Malaspinas, Gregory Ewing, Ping Liu, Daniel Wegmann, Daniel R. Caffrey, Konstantin B. Zeldovich, Daniel N. Bolon, Jennifer P. Wang, Timothy F. Kowalik, Celia A. Schiffer, Robert W. Finberg, and Jeffrey D. Jensen. Influenza virus drug resistance: A time-sampled population genetics perspective. *PLOS Genetics*, 10(2):1–17, 02 2014.
- [33] A F Feder, S Kryazhimskiy, and J B Plotkin. Identifying signatures of selection in genetic time series. *Genetics*, 196:509–522, 2014.
- [34] M Lacerda and C Seoighe. Population genetics inference for longitudinally-sampled mutants under strong selection. *Genetics*, 198:1237–1250, 2014.
- [35] Matthias Steinrücken, Anand Bhaskar, and Yun S. Song. A novel spectral method for inferring general diploid selection from time series genetic data. *Ann. Appl. Stat.*, 8(4):2203–2222, 12 2014.
- [36] Hande Topa, Ágnes Jónás, Robert Kofler, Carolin Kosiol, and Antti Honkela. Gaussian process test for high-throughput sequencing time series: application to experimental evolution. *Bioinformatics*, 31(11):1762–1770, 2015.
- [37] A Ferrer-Admetlla, C Leuenberger, J D Jensen, and D Wegmann. An approximate markov model for the wright-fisher diffusion and its application to time series data. *Genetics*, 2016.
- [38] B S Khatri. Quantifying evolutionary dynamics from variant-frequency time series. *Sci. Rep.*, 6:32497, 2016.
- [39] J G Schraiber, S N Evans, and M Slatkin. Bayesian inference of natural selection from allele frequency time series. *Genetics*, 203:493–511, 2016.
- [40] I B Krukov, B deSanctis, and A P J de Koning. Wright–fisher exact solver (wfes): scalable analysis of population genetic models without simulation or diffusion theory. *Bioinformatics*, 33:1416–1417, 2017.
- [41] J H Gillespie. *The Causes of Molecular Evolution*. OUP, 1991.
- [42] Daniel S. Fisher. Course 11 evolutionary dynamics. In Marc Mézard Jean-Philippe Bouchaud and Jean Dalibard, editors, *Complex Systems*, volume 85 of *Les Houches*, pages 395 – 446. Elsevier, 2007.

- [43] Michael M. Desai and Daniel S. Fisher. Beneficial mutation–selection balance and the effect of linkage on positive selection. *Genetics*, 176(3):1759–1798, 2007.
- [44] John D Storey and Robert Tibshirani. Statistical significance for genomewide studies. *PNAS*, 100(6):9440–9445, 2003.
- [45] Ivana Cvijović, Benjamin H. Good, Elizabeth R. Jerison, and Michael M. Desai. The fate of a mutation in a fluctuating environment. *Proceedings of the National Academy of Sciences*, 112(36):E5201–E5208, 2015.
- [46] R.A. Fisher. *The Genetical Theory of Natural Selection*. Clarendon, Oxford, 1930.
- [47] D S Fisher. Asexual evolution waves: fluctuations and universality. *Journal of Statistical Mechanics*, 2013.
- [48] Michael Baym, Sergey Kryazhimskiy, Tami D. Lieberman, Hattie Chung, Michael M. Desai, and Roy Kishony. Inexpensive multiplexed library preparation for megabase-sized genomes. *PLOS ONE*, 10(5):1–15, 05 2015.
- [49] A. M. Bolger, M. Lohse, and B. Usadel. Trimmomatic: A flexible trimmer for illumina sequence data. *Bioinformatics*, btu170, 2014.
- [50] D.E. Deatherage and J.E. Barrick. Identification of mutations in laboratory-evolved microbes from next-generation sequencing data using *breseq*. *Methods Mol. Biol.*, 1151:165–188, 2014.
- [51] Louise Aigrain Johan Hallin Karl Persson Karen Oliver Anders Bergström Paul Coupland Jonas Warringer Marco C. Lagomarsino Gilles Fischer Richard Durbin Jia-Xing Yue, Jing Li and Gianni Liti. Contrasting evolutionary genome dynamics between domesticated and wild yeasts. *Nat Genet.*, 49(6):913–924, 2017.
- [52] Benjamin H. Good, Michael M. McDonald, Jeffrey E. Barrick, Richard E. Lenski, and Michael M. Desai. The dynamics of molecular evolution over 60,000 generations. *Nature*, 551:45–50, 2017.
- [53] Elizabeth R Jerison, Sergey Kryazhimskiy, James Kameron Mitchell, Joshua S Bloom, Leonid Kruglyak, and Michael M Desai. Genetic variation in adaptability and pleiotropy in budding yeast. *Elife*, 6:e27167, 2017.
- [54] David Gresham, Michael M Desai, Cheryl M Tucker, Harry T Jenq, Dave A Pai, Alexandra Ward, Christopher G DeSevo, David Botstein, and Maitreya J Dunham. The repertoire and dynamics of evolutionary adaptations to controlled nutrient-limited environments in yeast. *PLoS Genetics*, 4(12):e1000303, 2008.
- [55] Marlene Sousa, Ana Marta Duarte, Tânia R Fernandes, Susana R Chaves, Andreia Pacheco, Cecília Leão, Manuela Côrte-Real, and Maria João Sousa. Genome-wide identification of genes involved in the positive and negative regulation of acetic acid-induced programmed cell death in *saccharomyces cerevisiae*. *BMC Genomics*, 14(1):838, 2013.
- [56] Daniel González-Ramos, Arthur R Gorter De Vries, Sietske S Grijseels, Margo C Van Berkum, Steve Swinnen, Marcel Van Den Broek, Elke Nevoigt, Jean-Marc G Daran, Jack T Pronk, and Antonius JA Van Maris. A new laboratory evolution approach to select for constitutive acetic acid tolerance in *saccharomyces cerevisiae* and identification of causal mutations. *Biotechnology for Biofuels*, 9(1):173, 2016.

- [57] Yeji Lee, Olviyani Nasution, Eunyong Choi, In-Geol Choi, Wankee Kim, and Wonja Choi. Transcriptome analysis of acetic-acid-treated yeast cells identifies a large set of genes whose overexpression or deletion enhances acetic acid tolerance. *Applied Microbiology and Biotechnology*, 99(15):6391–6403, August 2015.
- [58] Miho Kawahata, Kazuo Masaki, Tsutomu Fujii, and Haruyuki Iefuji. Yeast genes involved in response to lactic acid and acetic acid: acidic conditions caused by the organic acids in *saccharomyces cerevisiae* cultures induce expression of intracellular metal metabolism genes regulated by *aft1p*. *FEMS yeast research*, 6(6):924–936, 2006.