Cell Stem Cell, Volume *28*

Supplemental Information

Reconstructing the Lineage Histories and Differentiation Trajectories of Individual Cancer Cells in Myeloproliferative Neoplasms

Debra Van Egeren, Javier Escabi, Maximilian Nguyen, Shichen Liu, Christopher R. Reilly, Sachin Patel, Baransel Kamaz, Maria Kalyva, Daniel J. DeAngelo, Ilene Galinsky, Martha Wadleigh, Eric S. Winer, Marlise R. Luskin, Richard M. Stone, Jacqueline S. Garcia, Gabriela S. Hobbs, Fernando D. Camargo, Franziska Michor, Ann Mullally, Isidro Cortes-Ciriano, and Sahand Hormoz

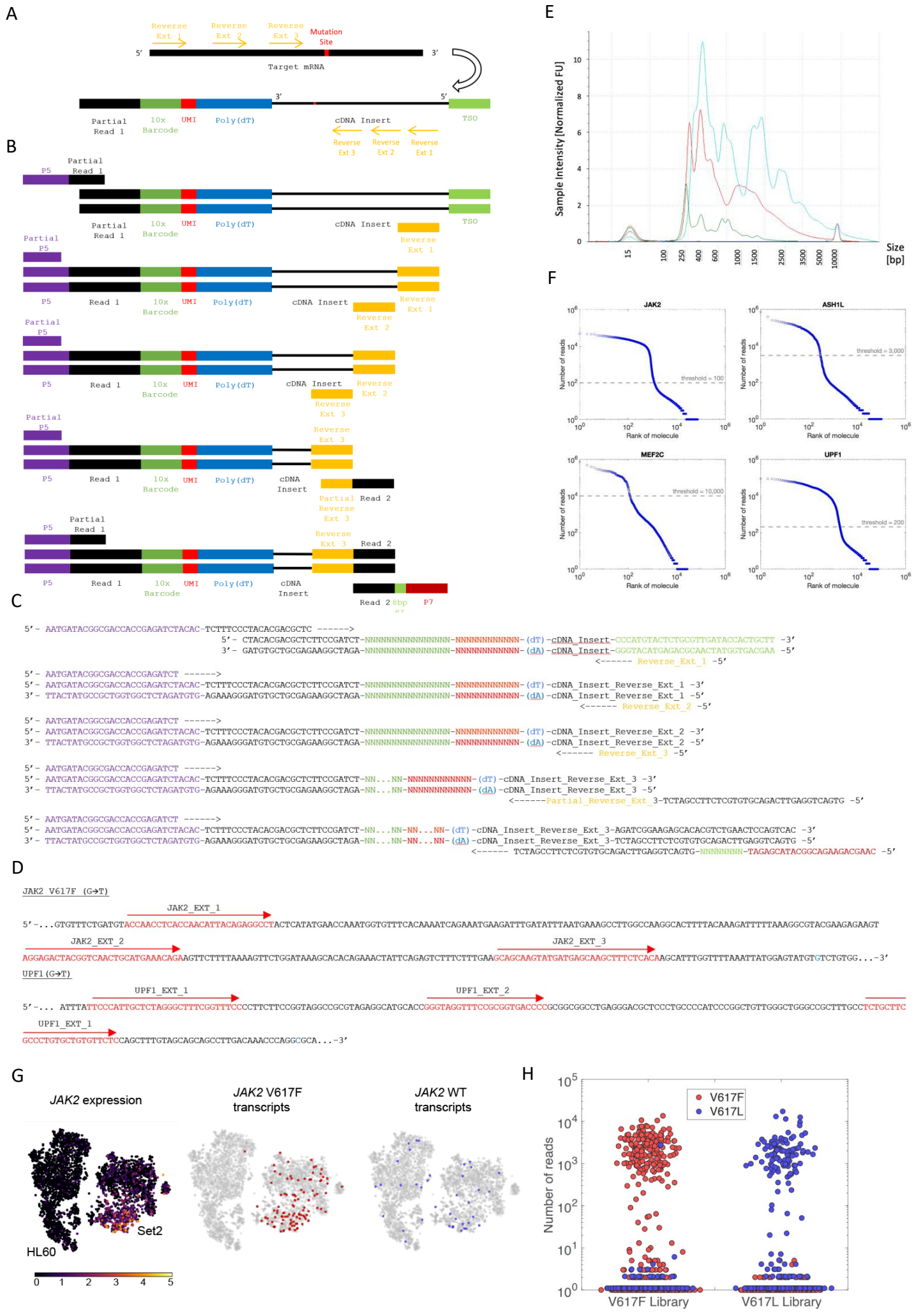

A
Reverse Ext 1
Reverse Ext 2
Reverse Ext 3
Mutation Site
5′
3′
Target mRNA
Partial Read 1
10x Barcode
UMI
Poly(dT)
cDNA Insert
TSO
B
P5
Read 1
Read 2
Read 2 8bp
P7
C
5′- AATGATACGGCGACCACCGAGATCTACAC-TCTTTCCCTACACGACGCTC ------>
5′- CTACACGACGCTCTTCCGATCT-NNNNNNNNNNNNNNNN-NNNNNNNNNNNN-(dT)-cDNA_Insert-CCCATGTACTCTGCGTTGATACCACTGCTT -3′
3′- GATGTGCTGCGAGAAGGCTAGA-NNNNNNNNNNNNNNNN-NNNNNNNNNNNN-(dA)-cDNA_Insert-GGGTACATGAGACGCAACTATGGTGACGAA -5′
<------ Reverse_Ext_1 -5′
<------ Reverse_Ext_2 -5′
<------ Reverse_Ext_3 -5′
<------Partial_Reverse_Ext_3-TCTAGCCTTCTCGTGTGCAGACTTGAGGTCAGTG -5′
<------ TCTAGCCTTCTCGTGTGCAGACTTGAGGTCAGTG-NNNNNNNN-TAGAGCATACGGCAGAAGACGAAC -5′
D
JAK2 V617F (G→T)
JAK2_EXT_1
JAK2_EXT_2
JAK2_EXT_3
UPF1 (G→T)
UPF1_EXT_1
UPF1_EXT_2
E
Sample Intensity [Normalized FU]
Size [bp]
F
JAK2
ASH1L
MEF2C
UPF1
Number of reads
Rank of molecule
threshold = 100
threshold = 3,000
threshold = 10,000
threshold = 200
G
JAK2 expression
JAK2 V617F transcripts
JAK2 WT transcripts
HL60
Set2
H
V617F
V617L
Number of reads
V617F Library
V617L Library

Figure S1. Primer designs, directions, locations, sequences of target mutation amplification and troubleshooting; Accurate identification of the mutated cells from the amplicon libraries. Related to Figure 1. A. Schematic illustration of primer direction against target mRNA and the change of primer directionality during amplification of sc-cDNA. **B.** Schematic diagrams of the nested PCR from step 1 to step 5, respectively. **C.** Oligonucleotide sequences and localization of common primers and adaptors. **D.** Example primer positions and sequences of targeted mutation. **E.** Example TapeStation trace for QC and optimization from step 1 to 5. **F.** Number of reads vs rank of molecule and threshold of cell calling. **G.** In a control experiment HL60 (WT cells) were mixed with SET2 cells (heterozygous *JAK2*-V617F mutation) and ran through the experimental and analysis pipeline. The two cell populations could be distinguished based on their transcriptional profiles: two distinct clusters were seen when transcriptomes of the cells were visualized using tSNE. Marker genes were used to identify the clusters as either HL60 or SET2 cells. Cells in which a mutated *JAK2* transcript (middle panel) or a WT *JAK2* transcript (right panel) were detected in the amplicon libraries are shown as colored points. All other cells are shown in gray. *JAK2* mutation site was detected in ~4% of cells. The detection limit is primarily determined by the JAK2 expression levels (shown in the left panel). The false positive rate of detecting a mutated transcript in a WT HL60 cell is less than 1%. **H.** In another control experiment we combined the single-cell libraries of a *JAK2*-V617F patient (ET 1) and that of the *JAK2*-V617L patient before the libraries were fragmented and indexed. We then ran the combined library through the experimental and analysis pipeline. The *JAK2* amplicon sequences could be mapped back to the library from which they originated based on their single-cell barcode. The plot shows the number of reads of each *JAK2* transcript detected for the V617F library (left) and V617L library (right). The colors of the points denote whether the transcript sequence contained a V617F or V617L mutation. Blue dots on the left side and red dots on the right side correspond to incorrect mapping of a mutation to a single-cell barcode, most likely due to PCR crossover events during amplification. Above the threshold of 100 reads, the false positive rate is negligible.

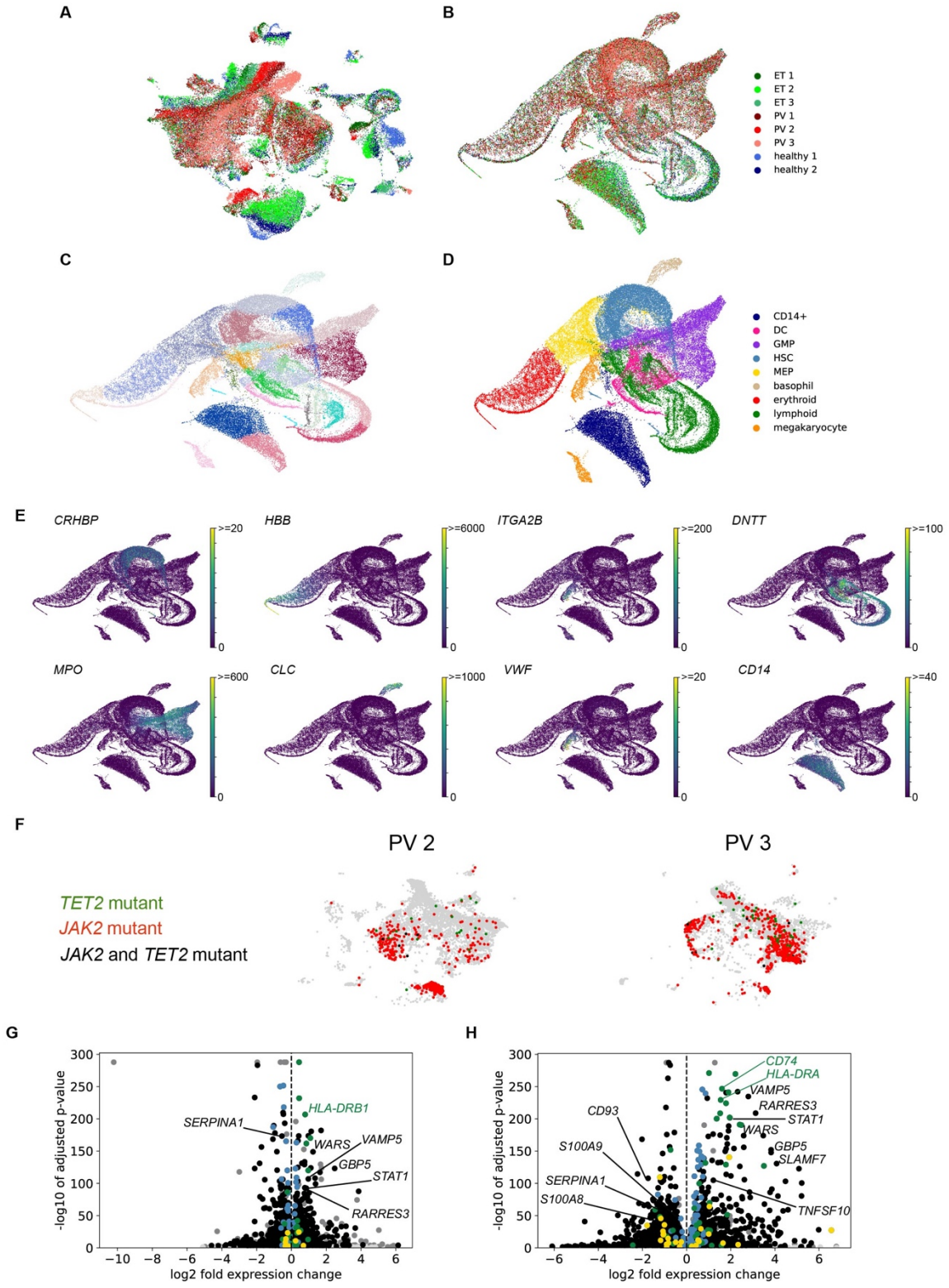


Figure S2. scRNA-seq analysis and genotyping of *JAK2*-mutant MPNs. Related to Figure 2. A. UMAP of CD34-enriched bone marrow scRNA-seq data from all patients before batch correction, colored by donor as in **B**. **B-D.** UMAP of CD34-enriched bone marrow scRNA-seq data from all patients after Seurat batch correction, colored by donor (**B**), Louvain cluster (**C**), and final cell type identification (**D**). **F.** UMAPs for patient PV 2 (left) and PV 3 (right) with cells with *JAK2* and/or *TET2* mutant transcripts highlighted. G-H. Volcano plots of differential expression for all CD14+ cells between ET patients and healthy controls (**G**) or PV patients and

healthy controls (**H**). Genes found to be differentially expressed in all pairwise comparisons between different patient subsets are highlighted and colored by KEGG 2019 biological process group (gold: hematopoiesis-related, green: antigen presentation, blue: ribosomal, black: other).

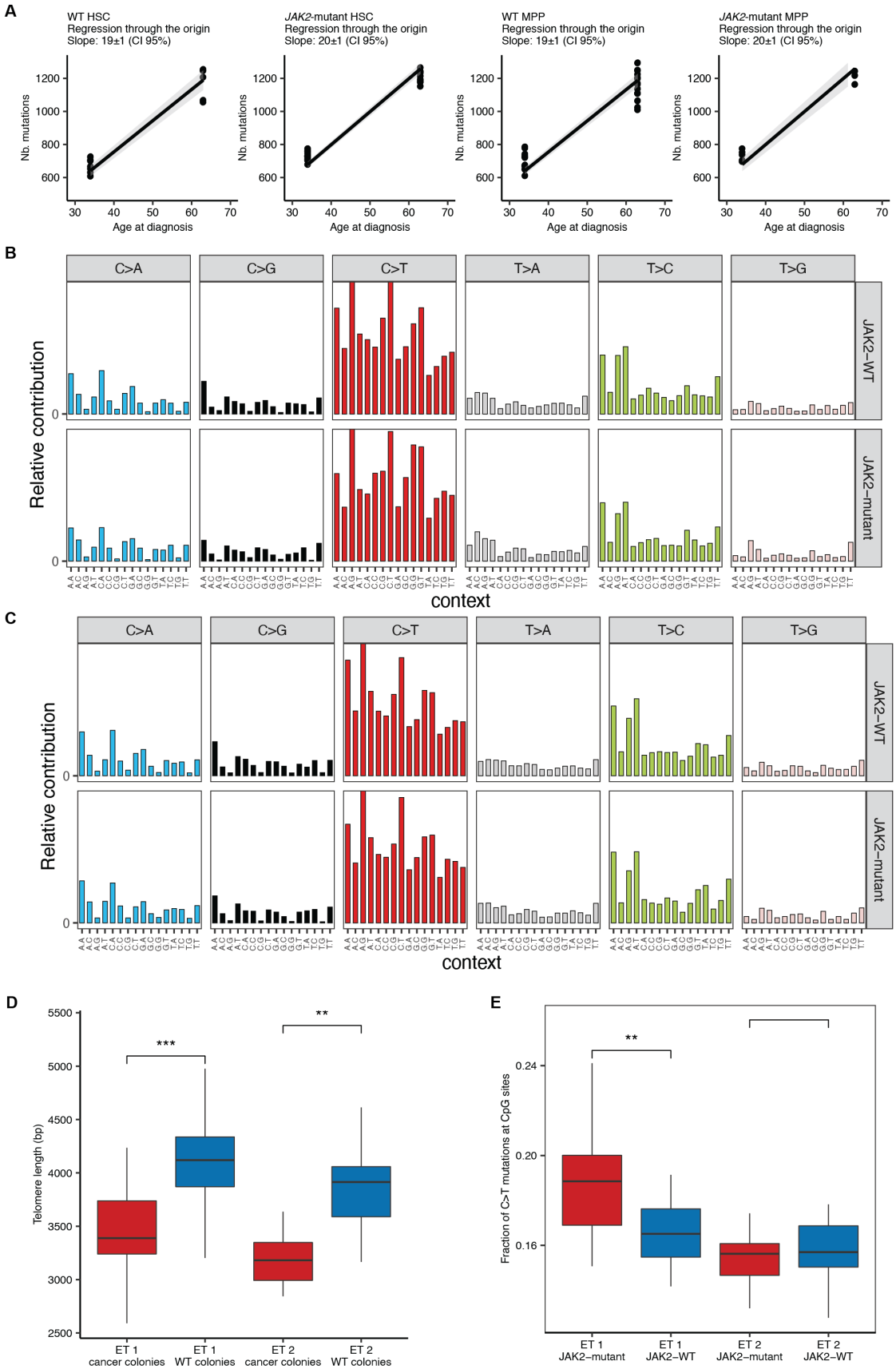

A
WT HSC
Regression through the origin
Slope: 19±1 (CI 95%)
JAK2-mutant HSC
Regression through the origin
Slope: 20±1 (CI 95%)
WT MPP
Regression through the origin
Slope: 19±1 (CI 95%)
JAK2-mutant MPP
Regression through the origin
Slope: 20±1 (CI 95%)
Nb. mutations
Age at diagnosis
B
C>A
C>G
C>T
T>A
T>C
T>G
JAK2−WT
JAK2−mutant
Relative contribution
context
C
D
Telomere length (bp)
***
**
ET 1 cancer colonies
ET 1 WT colonies
ET 2 cancer colonies
ET 2 WT colonies
E
Fraction of C>T mutations at CpG sites
ET 1 JAK2−mutant
ET 1 JAK2−WT
ET 2 JAK2−mutant
ET 2 JAK2−WT

Figure S3. Mutational analysis of individual HSCs and MPPs from MPN patients. Related to Figure 3. A. Correlation between age at diagnosis (years; x-axis) and the number of somatic SNVs (y-axis) detected in WT HSCs, *JAK2*-mutant HSCs, WT MPPs, and *JAK2*-mutant MPPs. Each dot corresponds to a single-derived colony, and the lines represent the regression through the origin. The estimated values for the slope and the 95% confidence intervals (CI) are shown. **B-C.** Patterns of somatic mutations for *JAK2*-mutant or *JAK2*-WT colonies from patients ET 1 (**B**) and ET 2 (**C**). The relative fraction of each mutation type in the catalogue of point mutations detected in each colony is reported. Base substitutions are further stratified into categories based on the trinucleotide context in which the mutation occurs. **D.** Distribution of telomere lengths estimated using the whole-genome sequencing data for *JAK2*-mutant and *JAK2*-WT HSPCs from patients ET 1 and ET 2. **E.** Distribution of the number of C>T mutations at CpG dinucleotides in *JAK2*-mutant and *JAK2*-WT HSPCs from patients ET 1 and ET 2. The box plots in **D-E** show the median, first and third quartiles (boxes), and the whiskers encompass observations within a distance of 1.5× the interquartile range from the first and third quartiles. The box plots in **D-E** show the median, first and third quartiles (boxes), and the whiskers encompass observations within a distance of 1.5× the interquartile range from the first and third quartiles. The single, double, and triple asterisks indicate statistical significance at $P < 0.05$, $P < 0.01$, and $P < 0.001$, respectively (Wilcoxon rank sum test).

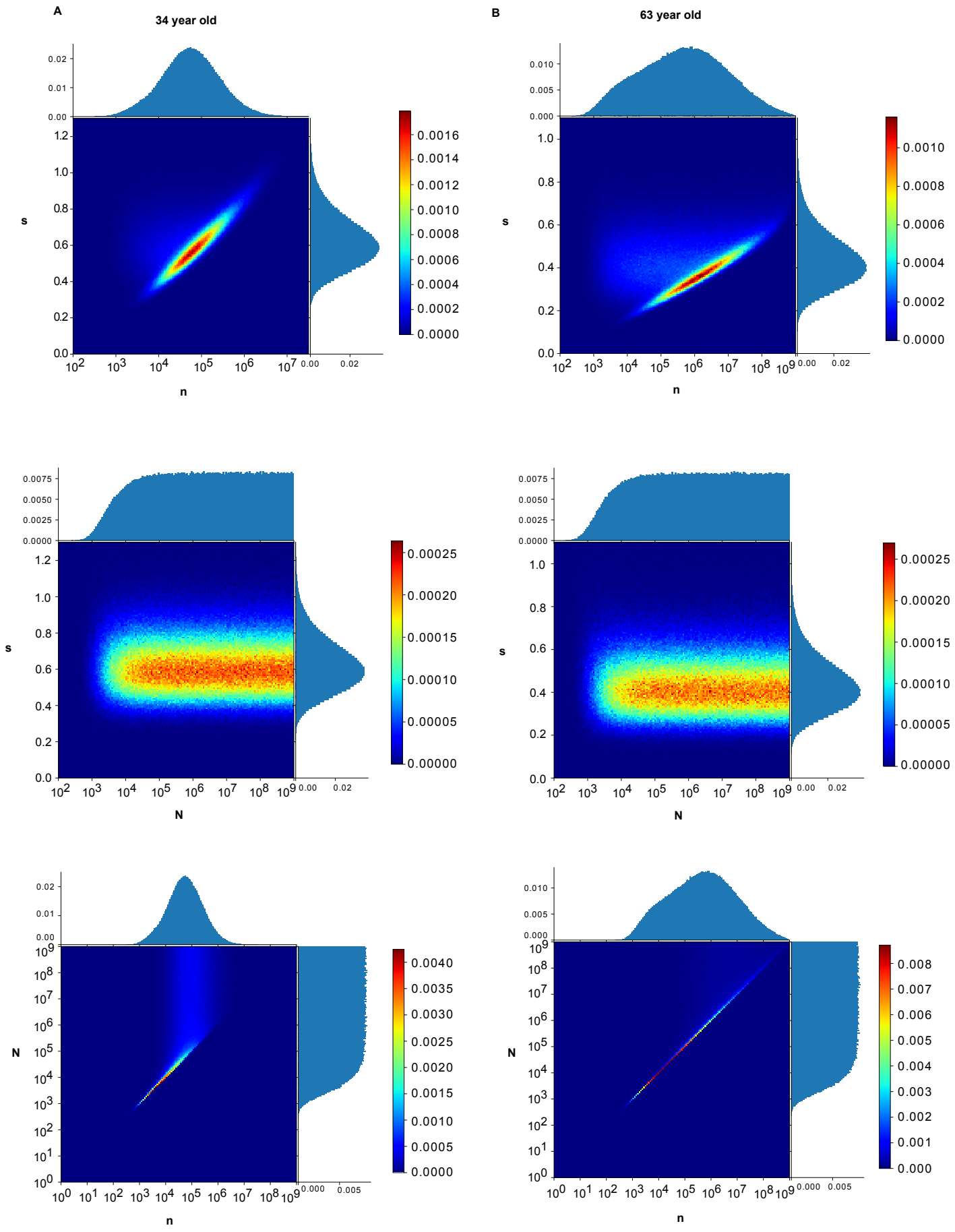

A
34 year old
B
63 year old
s
n
N

Figure S4. Inference on patient data. Related to Figure 4. ABC was run on the patient data, and the model parameters were inferred. Joint distributions of the parameter values were plotted along with their marginals. *s* is in growth per year, and age of onset of the disease is in years. **A.** Distributions for inference on 34-year-old patient data. **B.** Distributions for inference on 63-year- old patient data. As observed, inference of *n* can indicate saturation when the number of cancer cells approaches *N*. In this case, the cancer expansion slows down and begins to exhibit neutral dynamics. This changes the coalescent structure, allowing ABC to detect the saturation and infer a saturation parameter value of $n = N$. When *N* is too large to affect the exponential growth dynamics of the cancer cells, ABC can only put a bound on *N*, namely that *N* must be larger than the number of cancer cells at the final time-point.

Table S1. Primers and sequences for mutation-specific single-cell amplicon libraries (5′→3′). Related to STAR Methods.

INTERNAL_FORWARD	AATGATACGGCGACCACCGAGATCTACACTCTTTCCCTACACGACGCTC
SHORT_INT_FOR	AATGATACGGCGACCACCGAGATCT
JAK2_EXT1	ACCAACCTCACCAACATTACAGAGGCCT
JAK2_EXT2	AGGAGACTACGGTCAACTGCATGAAACAGA
JAK2_EXT3	GCAGCAAGTATGATGAGCAAGCTTTCTCACA
JAK2_EXT4	GTGACTGGAGTTCAGACGTGTGCTCTTCCGATCTGCAGCAAGTATGATGAGCAA
ASH1L_Ext1	GCATCTCACTCCTATCTGAAAAGTTGACAAGC
ASH1L_Ext2	TGGCCACAAAGAAAAACCTAGACCATGTCA
ASH1L_Ext3	GGAAATGTCCCCTTCAGGCTGTCGTATCAA
ASH1L_Ext4	GTGACTGGAGTTCAGACGTGTGCTCTTCCGATCTGGAAATGTCCCCTTCAGGCT
HSPA9_EXT1	ACCTGACAAGAGTCTTAAGCAACCAAAGCA
HSPA9_EXT2	GTGGGTCATGCCTGTAATCCCAACACTTG
HSPA9_EXT3	GTGTGGGAGTTGAAGATCACCCTAGGCAA
HSPA9_EXT4	GTGACTGGAGTTCAGACGTGTGCTCTTCCGATCTGTGTGGGAGTTGAAGATCAC
NRROS_EXT1	GAATCCATCTGTCTCCTTTCCTCAGCTTTGCCT
NRROS_EXT2	AGTCCCGGAGCTGGTGGCAAAGA
NRROS_EXT3	TCTCACGGGCCCAGCCTTACTCA
NRROS_EXT4	GTGACTGGAGTTCAGACGTGTGCTCTTCCGATCTTCTCACGGGCCCAGCCTTAC
UPF1_EXT1	TTCCCATTGCTCTAGGGCTTTCGGTTTCC
UPF1_EXT2	GGGTAGGTTTCCGCGGTGACCCC
UPF1_EXT3	TCTGCTTCGCCCTGTGCTGTGTTCTC
UPF1_EXT4	GTGACTGGAGTTCAGACGTGTGCTCTTCCGATCTTCTGCTTCGCCCTGTGCTGT
TET2B_EXT1	CACATAACTGCAGTGGGCCTGAAAATCCAG
TET2B_EXT2	TAATGGTGCTACAGTTTCTGCCTCTTCCGT
TET2B_EXT3	ACATCTCACATAAATGCCATTAACAGTCAGGC
TET2B_EXT4	GTGACTGGAGTTCAGACGTGTGCTCTTCCGATCTACATCTCACATAAATGCCAT

Methods S1, related to STAR Methods.

Outline. In Section 1, we first define our mathematical model for mutated HSC growth dynamics, followed by the model with feedback where the fitness of the mutant cells decreases over time, and then describe an efficient way to simulate the mutant growth dynamics for inference. We then go on to derive analytical mathematical expressions for the average growth rate of mutant cells in our model, and use these to show that the fitness in terms of growth per year can be inferred from lineage trees without knowledge of the mutation rate. In Section 2, we include a detailed description of the simulations, analyses, and figures to verify our theoretical calculations and to demonstrate the robustness of our inference with respect to the model assumptions. Section 2 can be read before Section 1.

1. Mathematical Description of Growth Dynamics of Mutated Cells

WF model with selection

We chose to model mutated HSC growth (or the growth of a population of mutant cells within a population of wild-type cells) as a variation of the Wright-Fisher stochastic process (Fisher, 1923; Wright, 1931) with selection included. We begin with an initial population of N stem cells at $t = 0$. The stochastic process arises by iterating the following rule on the initial population: the cells at generation t give birth to N stem cells which correspond to generation $t + 1$. Each cell in generation $t + 1$ then selects a cell at random from generation t as its parent, and the cells at generation t die off. At generation t', a mutation arises in one of the stem cells that gives it a selective advantage. From then on, the iterative rule changes in that instead of the cells from the parent generation being selected at random, each wild-type cell has probability p of being selected as a parent, and each mutant cell has probability $(1 + s)p$ of being selected as a parent. The cells always inherit the phenotypic state of their parents (mutant or wild-type). After L iterations, we produce an evolving population of stem cells for $t = 0,1, \dots, t', t' + 1, \dots, L$ along with a set of genealogical relationships.

Note that here there are a total of $L + 1$ generations of cells, since the first generation corresponds to $t = 0$. In the computational section (Section 2), we instead define L to be the total number of generations. The way we have defined L here is more convenient for the mathematical derivations, but it should be noted that any expression we have derived here in the mathematical section (Section 1) that uses L will be replaced with $L - 1$ in the computational section (Section 2).

Computing the number of mutant cells as a function of time

Here, we derive the distributions for the number of mutant cells as a function of time. Given the current generation of mutant cells, the number of mutant cells in the generation that follows is binomially distributed with parameter values that depend on the number of mutant cells in the current generation. This fact is used in the subsequent subsections to compute the mean growth of mutant cells and for efficient simulation of clonal expansions.

Suppose that there are n mutant cells at generation t (Note that in Section 2, we define n to be the number of mutant cells at the final time-point. This n is not the same). Since each wild-type cells has probability p of being selected as a parent, and each mutant cell has probability $(1+s)p$ of being selected as a parent, then the probability that a cell at t + 1 choses a wild-type cell is $(N-n)p$, and the probability it chooses a mutant cell is (1 + s)np. Since probabilities must sum to one, p can be derived from the condition that $(N-n)p + (1+s)np = 1$, and so we obtain $p = \frac{1}{N+ns}$. After substituting, we obtain the probability a cell selects a mutant cell as its parent as:

$$\frac{(1+s)n}{N+ns}$$

Since each of the N cells at t + 1 either chooses a mutant cell or it does not, and since the choices are independent, then the number of mutant cells at generation t + 1 is binomially distributed with parameters N and $\frac{(1+s)n}{N+ns}$. It then follows that we can compute the number of mutant cells as a function of time by beginning with an initial condition of $n = 1$ mutant cell, and then carrying out a series of binomial draws where we update n before each draw to equal the current number of mutant cells.

Wright-Fisher model with feedback

The Wright-Fisher model is an idealized model that ignores a wide range of biologically plausible scenarios. For example, as the number of mutant cells increases, it is conceivable that there are underlying biological mechanisms that slow the growth of the mutant cells. To simulate such scenarios, we decided to incorporate feedback into the Wright-Fisher model. This is accomplished by letting the value of s change depending on the current number of mutant cells. In particular, if the number of mutant cells at generation t is n, then the selection parameter at t is $s(1-\frac{n-1}{N-1})^x$. Notice that when $x = 0$, the selection parameter remains constant and we recover the usual Wright-Fisher model with selection.

To simulate clonal expansions for mutant cells with feedback, we simply draw a series of numbers from binomial distributions as described in the previous section, except that instead of just updating n before each draw, we first update n and then $s(1-\frac{n-1}{N-1})^x$. Clonal expansions with feedback in Section 2 are simulated in this way.

Dynamics of average population size

Define n(t) as the number of mutant cells as a function of time. We now compute the expectation of $n(t)$, which we will call the mean trajectory. We consider the expected value of $n(t)$, conditioned on the mutant clone consisting of n cells at time $t-1$. Upon conditioning, $n(t)$ reduces to a binomially distributed random variable as shown before, and so its mean is given by:

$$E_N[n(t) \mid n(t-1) = n] \;=\; N\frac{(1+s)\,n}{N+ns}$$

Subscript N is used to emphasize the dependence on population size. We then assume $N >> ns$, and drop subscript $N$to obtain:

$$E[\,n(t) \mid n(t-1) = n\,] \;=\; (1+s)n$$

Rewriting the expected value as a conditional expectation gives us

$$E[\,n(t) \mid n(t-1)\,] \;=\; (1+s)\,n(\,t-1\,)$$

We then take the expectation of both sides to generate the following recursion

$$E[n(t)] \;=\; (1+s)\,E[n(t-1)]$$

initial condition: $n(0) = 1$

where without loss of generality we have let the time at which the mutation arrives be $t = 0$.

The recursion is then easily solved to obtain

$$E[n(t)] \;=\; (1+s)^t$$

Dynamics of average population size conditioned on survival.

When the number of mutant cells is small, they are susceptible to stochastic fluctuations and extinction. After growing to a sufficiently large size, their growth dynamics become deterministic and fluctuations can be safely ignored.

When using ABC to infer our model's parameters, we only consider trees where mutant cells have not gone extinct. We are thus interested in the growth dynamics conditioned on no stochastic extinction.

In the previous section, we computed the expectation value of the number of mutant cells as a function time across all trajectories. Here, we will constrain the expectation value to trajectories that do not go extinct. As would be expected, the average population size is larger when extinction is not allowed.

We begin by defining the conditioning event for our trajectories as $F = \{fixation\ will\ occur\}$. Then we use Bayes' theorem to compute

$$E_N[\, n(t) \mid F \,] \;=\; \sum_{n\,=\,1}^{N} n\, P(\, n(t) = n \mid F \,) \;=\; \sum_{n\,=\,0}^{N} \frac{n\, P(\, n(t) = n \,)\, P(\, F \mid n(t) = n \,)}{P(\, F \,)}$$

The probability of fixation of a clone of size n within a sufficiently large population of size N and with fitness $1 + s$ is given by Kimura's diffusion approximation (Kimura, 1962)

$$\frac{1 - e^{-2sn}}{1 - e^{-2sN}}$$

We therefore put

$$P(\, F \mid n(t) = n \,) = \frac{1 - e^{-2sn}}{1 - e^{-2sN}}$$

$$P(\, F \,) \;=\; \frac{1 - e^{-2s}}{1 - e^{-2sN}}$$

The probability of fixation $P(\, F \,)$ independent of the clone size is simply the probability that a clone of size $n = 1$ will eventually fix. Substituting both probabilities back into the sum and cancelling terms gives us

$$E_N[\, n(t) \mid F \,] \;=\; \frac{1}{1 - e^{-2s}} \sum_{n\,=\,0}^{N} \quad n\, P(\, n(t) = n \,)\, (1 - e^{-2sn})$$

Next, we make a key biologically motivated assumption: we assume that t is sufficiently large so that the population of mutant cells is:

1) either large enough to exhibit deterministic dynamics, or
2) has gone extinct.

Therefore, $P(\, n(t) = n \,)$ vanishes except for when n is large, or $n = 0$. Since the only nonzero terms in the expectation are those for large n, and since $1 - e^{-2sn}$~1 when n is large, then we may replace $1 - e^{-2sn}$ with 1 in the expectation as an approximation. After replacing $1 - e^{-2sn}$ with 1 and observing that the sum is now the unconditional expectation, we obtain

$$E_N[\, n(t) \mid F \,] \;=\; \frac{1}{1 - e^{-2s}}\; E_N[\, n(t) \,]$$

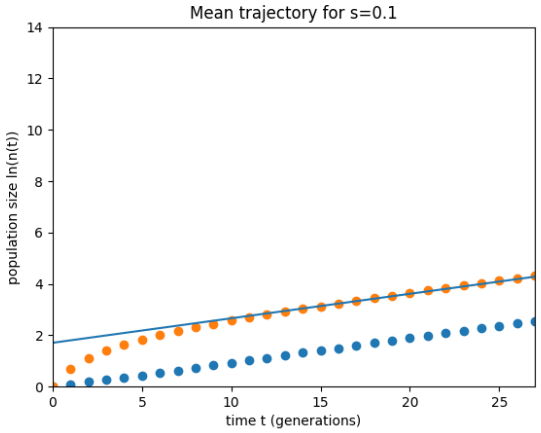
If we let $N >> ns$ and drop the subscripts, the expectation on the right-hand side becomes the mean trajectory $E[n(t)] \;=\; (1+s)^t$ that we derived in the previous section. Substituting then gives us

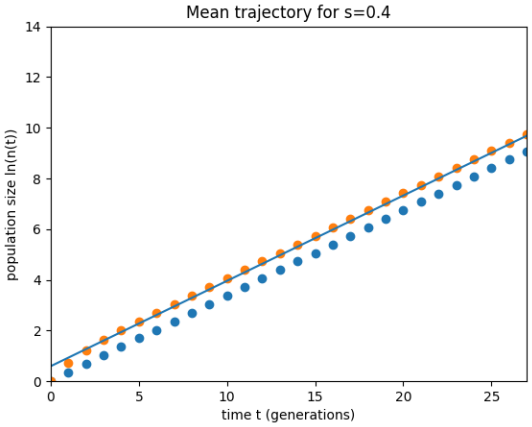
$$E[\, n(t) \mid F \,] \;=\; \frac{1}{1 - e^{-2s}}\; (1+s)^t$$

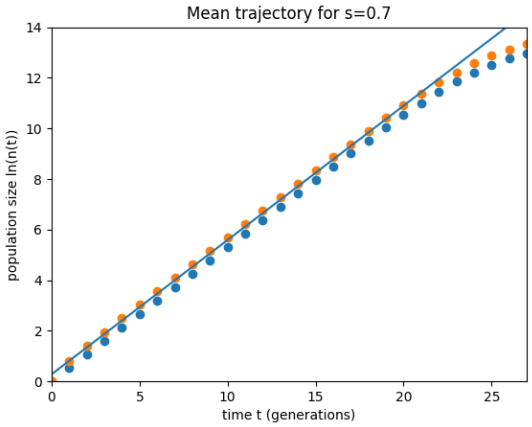
Note that the above approximation is not valid for small values of t, for example, evaluating at $t = 0$ does not give an average population size of 1, because we assumed that t must be sufficiently large.

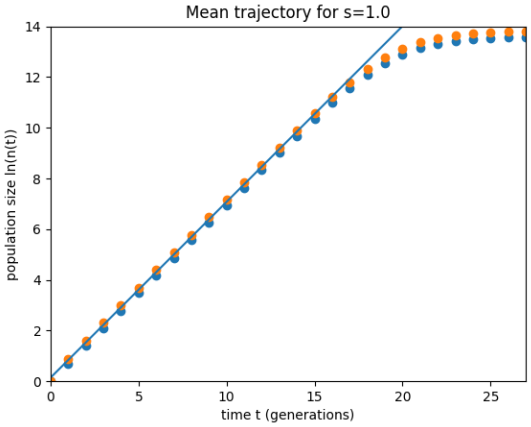
Importantly, the above approximation has an interesting biological interpretation. The growth dynamics excluding extinction events is functionally equivalent to the growth dynamics of unconditional trajectories that begin with a clone size $\frac{1}{1-e^{-2s}}$. Later, we will use this observation to show that s could be inferred without prior knowledge of mutation rate.

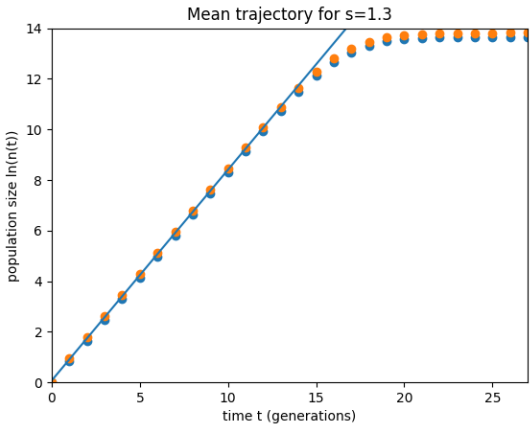
To validate the approximation derived above, we simulated the average growth dynamics of the mutant cells. Figures below show the simulated mean trajectory (blue dots), the simulated mean trajectories conditioned on no stochastic extinction (orange dots), and then our approximation of the mean trajectories conditioned on no stochastic extinction (blue curve) for different values of s. The simulated mean trajectories were collected by taking the average number of mutant cells at each time slice over a large number of simulated clonal expansions. The simulated mean trajectories conditioned on no stochastic extinction were generated similarly by first letting the simulated clones expand until they either went extinct or fixed, discarding the clones that went extinct, and then taking the average number of mutant cells at each time slice over the remaining clonal expansions. All simulated expansions were run with $N = 10^6$ and are shown for $g = 28$ generations.











Decoupling of fitness and mutation rate for weakly expanding clones

Under the neutral Wright-Fisher model, it is not possible to separately infer population size without knowledge of the mutation rate per generation. If we underestimate the mutation rate, we will

overestimate the number of generations between coalescent events in the tree, and thereby overestimate the population size. Conversely, if we overestimate the mutation rate, we will underestimate the number of generations between coalescent events in the tree, and thereby underestimate the population size. It is therefore necessary to have a priori information about one parameter, for example mutation rate, to extract any information about the other, for example population size, from the tree.

Fitness, or growth rate per year s_y, can be inferred from the shape of the reconstructed lineage tree of a small number of cells randomly sampled from the population at the final time point. If s_y is large, the population of mutated stem cells grows rapidly, therefore the coalescent events on the lineage tree will be confined to the first few generations, when the population size was small. Conversely, if s_y is small, coalescent events are more likely to occur in the last few generations. Critically, unlike population size, s_y can be inferred without any knowledge of the mutation rate, or equivalently, the total number of generations along the lineage tree. To intuitively understand this, note that rescaling the number of generations by a given factor scales the inferred population size by the same factor. Because at the onset of disease there is only one mutant cell, it might be expected that the growth rate, or s, must also be changed to achieve the scaled population size at the final time point. However, rescaling the number of generations also scales the minimum population size required before the mutated cells can escape stochastic extinction and grow exponentially. This is because more generations implies that the fitness advantage per generation is smaller and therefore the population is more susceptible to going extinct from random birth and death events. Taken together, these two competing effects precisely cancel and thus s_y can be inferred directly from the observed lineage trees without knowledge of the mutation rate or the number of generations.

We will make the above intuition precise by deriving the analytical expression for coalescent statistics as a function of s_y and showing that when expressed as growth per unit time, s_y does not vary with the mutation rate.

Mathematical analysis of mutation rate per generation and rate of population growth per unit time

Here, we will show that it is possible to infer the population growth rate per unit time without knowledge of the mutation rate per generation.

First, we will define the population growth rate per unit time. Then, we will derive an expression for the expected coalescent times of a random sample of mutant cells, and use it to estimate the impact of mutation rate per generation on the inferred population growth rate.

Define L to be the total number of divisions that an HSC would have undergone averaged across all HSCs, or equivalently the total number of generations in our trees. Note that knowing L is equivalent to knowing the mutation rate per generation, since we know the number of mutations accrued throughout the patient's life.

Definition of population growth rate per unit time

Note that $1 + s$, where s is the selection parameter, is the average growth per generation. We can also define a related quantity s_y as the average percent growth per unit time (for example percent growth rate per year). If we let a be the age of the patient, the number of generations per unit time is $\frac{L}{a}$. Hence, per unit time, the mutant clone is expected to grow by a factor of $(1 + s)^{\frac{L}{a}}$, and so we arrive at the expression

$$s_y(s, L) = (1 + s)^{\frac{L}{a}} - 1$$

Estimating the coalescent times

We now derive an expression for the expected time for coalescence of k randomly sampled mutant cells given that the clone has expanded for g generations.

Let t denote time in number of generations measured from the leaves of the tree towards the root. If there are $n(t)$ mutants at generation t, then the amount of coalescence time that passes from generation t to $t + 1$is $\frac{1}{n(t)}$, whereby coalescence time refers to the timescale in the Kingman Coalescent model (Kingman, 1982a, 1982b, 1982c). To understand what we are doing intuitively, note that for the standard Wright-Fisher model without selection, where the population size N is constant over time, the average time for coalescence of k randomly sampled lineages is $\frac{N}{\binom{k}{2}}$generations. This is generally computed by scaling time so that N generations correspond to 1 unit of time, and then letting N become large. In doing so, the times of coalescence of k randomly

sampled lineages converge to the Kingman Coalescent where the coalescence times are known to be $\frac{1}{\binom{k}{2}}$. The coalescence times in generations can then be recovered through an inverse time-scale transformation. Note that this is equivalent to scaling time so that the time between two neighboring generations is $\frac{1}{N}$. To account for a variable population size, we let the time between two neighboring generations t and $t+1$ be the inverse of the population size at t and assume the population size is always large. In doing so, the times until coalescence also converge to the Kingman Coalescent model. The statistics of coalescence times are then recovered by transforming back to time in generations.

Since the expected coalescence time of k lineages is $\frac{1}{\binom{k}{2}}$ in units of coalescence, and since the population size in our simulations grows approximately as $n(t) = \frac{1}{1-e^{-2s}}(1+s)^t$, the expected coalescence time of k lineages in units of generations is the t satisfying:

$$\frac{1}{\binom{k}{2}} = \sum_{k=0}^{t-1} \frac{1}{\frac{1}{1-e^{-2s}}(1+s)^{g-k}}$$

We then notice the R.H.S. is a geometric sum and rewrite to obtain:

$$1 + \frac{s}{1-e^{-2s}} \frac{(1+s)^g}{\binom{k}{2}} = (1+s)^t$$

Solving for t we obtain:

$$t = \frac{log\left(1 + \frac{s}{1-e^{-2s}} \frac{(1+s)^g}{\binom{k}{2}}\right)}{log\,(1+s)}$$

We use $\frac{L}{a}$ to convert 3) from generational time to real time (such as years), thereby obtaining

$$t_k(s,g,L) = \frac{a}{L}\frac{log\left(1+\frac{s}{1-e^{-2s}}\frac{(1+s)^g}{\left(\frac{k}{2}\right)}\right)}{log\,(1+s)}$$

Invariance theorem for weakly expanding clones

We now show that for a weakly expanding clone, we always infer the correct percent growth rate per unit time (for example per year) independent of our assumption of L.

We begin by assuming that $s_p, g_p,$ and L_p are the parameter values associated with our patient's tree, and that $|s_p| \ll 1$ so that selection is weak.

We then assume that we have incorrectly estimated our g and L parameters (i.e., our mutation rate) so that we erroneously believe they are $g_c = cg_p$ and $L_c = cL_p$ respectively. Note that we have kept the ratio $\frac{g_p}{L_p} = \frac{g_c}{L_c}$, and so we have treated the arrival time of the first mutated cell in real time as known.

Then we show that if we incorrectly assume our parameter values to be g_c and L_c, we then infer $s_c = (1+s_p)^{\frac{1}{c}} - 1$ for our s parameter, where

$$s_y(s_c, L_c) = s_y(s_p, L_p)$$

That is, we always infer the same percent growth per year. The way we show our inferred s is s_c is by plugging in g_c, L_c and s_c into the expected coalescent time expression we derived, and then showing that the coalescent times are identical to having plugged in $s_p, g_p,$ and L_p. In other words, we show that when we erroneously assume g_c and L_c are our parameter values, the s value that generates trees that match our patient's is s_c, and that s_c we infer combined with the L_c we've

assumed give us the same inference for yearly percent growth as the correct parameter values. We first show that s_c is our inferred s:

Begin by recalling that our coalescent time expression is given by

$$t_k(s,g,L) = \frac{a}{L}\frac{log\left(1+\frac{s}{1-e^{-2s}}\frac{(1+s)^g}{\left(\frac{k}{2}\right)}\right)}{log\,(1+s)}$$

Since $|s| \ll 1$, a Taylor expansion lets us make the following approximation:

$1-e^{-2s}$ ~$2s$

where we have let the 2nd order terms vanish. Substituting above and cancelling gives us

$$t_k(s,g,L) = \frac{a}{L}\frac{log\left(1+\frac{1}{2}\frac{(1+s)^g}{\left(\frac{k}{2}\right)}\right)}{log\,(1+s)}$$

But then using $s_c = \left(1+s_p\right)^{\frac{1}{c}} - 1$, $g_c = cg_p$ and $L_c = cL_p$ we can show:

$$t_k(s_c,g_c,L_c) = \frac{a}{L_c}\frac{log\left(1+\frac{1}{2}\frac{(1+s_c)^{g_c}}{\left(\frac{k}{2}\right)}\right)}{log\,(1+s_c)} = \frac{a}{cL_p}\frac{log\left(1+\frac{1}{2}\frac{\left(1+\left[(1+\,s_p)^{\frac{1}{c}}-1\right]\right)^{cg_p}}{\left(\frac{k}{2}\right)}\right)}{log\left(1+\,[(1+\,s_p)^{\frac{1}{c}}-1]\right)}$$

$$= \frac{a}{L_p} \frac{log\left(1 + \frac{1}{2}\frac{(1 + s_p)^{g_p}}{\left(\frac{k}{2}\right)}\right)}{log\,(1 + s_p)} = t_k(s_p, g_p, L_p)$$

so that s_c is our inferred s.

We then show our inference of percent growth is identical using $s_c = (1 + s_p)^{\frac{1}{c}} - 1$ and $L_c = cL_p$:

$$s_y(s_c, L_c) = (1 + s_c)^{\frac{L_c}{a}} - 1 = \left(1 + \left[\left[(1 + s_p)^{\frac{1}{c}} - 1\right]\right)^{\frac{cL_p}{a}} - 1$$

$$= (1 + s_p)^{\frac{L_p}{a}} - 1 = s_y(s_p, L_p)$$

2. Description of Simulations for Inference and Validation

Description of ABC

Approximate Bayesian Computation, or ABC, is an algorithm used to infer the parameter values of a stochastic model. ABC works by simulating data with the model using parameter values drawn from a prior distribution, and then computing a metric distance between the simulated data and the observed data. If their distance is smaller than a predetermined threshold, the parameter values are retained, otherwise, they are discarded. This procedure is iterated until a sufficient number of parameter values are retained to construct the posterior distribution.

ABC implementation

To perform ABC, it is first necessary to define a model. We briefly describe our model (see Section 1 for more details), and then give a detailed description of our ABC inference algorithm.

The model we used to infer the population dynamics of mutant cells is a variation of the Wright-Fisher model with selection (Fisher, 1923; Wright, 1931). Briefly, we consider a population of N stem cells that exists in discrete generations. There are L generations in total. At each generation, each cell chooses a parent cell at random from the previous generation. After t' generations, a cancerous mutation is acquired by one of the cells. Critically, the mutant cells are $1 + s$ times as likely to be chosen as a parent than the wild type cells. As a result, the number of mutant cells grows as $\sim (1 + s)^i$, for $i = 1, \ldots , g$, where $g = L - t$' corresponds to the disease duration. For convenience we use g as opposed to t' as a parameter in the following sections. However, provided L is given, if we know the value of g, we also know the value of t' and *vice versa*, and so the two parameters are equivalent. To summarize, the parameters of our model are:

N = saturation parameter (the total number of stem cells)

L = total number of generations

g = disease duration ($g <= L$)

s = selection parameter

We also define n as the number of mutant cells at the final time-point.

We now outline the steps of the ABC algorithm, and then elaborate on the details.

1. Draw s from its prior distribution.
2. Draw N, L, and g from their prior distributions.
3. Simulate a clonal expansion with our model for g generations.

4. If the final number of mutant cells is $n < k$, where k is the number of mutant cells we sample from the final population, back to 2). Else, move on to 5).
5. Sample k mutant cells from the final population and simulate their lineage history.
6. Simulate the number of mutations along the branches of the tree with the given mutation rate.
7. Perform the averaging algorithm on the tree so that the number of mutations from any leaf to the root of the tree is the same.
8. Convert the tree to an LTT plot.
9. If the area between the LTT plot of our simulated tree and the LTT plot of our patient tree is smaller than epsilon, retain the parameter values, otherwise discard them.
10. If a sufficient number of parameters to construct a distribution has been retained, finish. Else, back to 1).

Note that to perform the ABC, we must first specify the prior distributions on s, N, L and g (to test the robustness of our simulation *in silico*, we will sometimes fix parameter values rather than drawing from a distribution), the number of cells we will sample from the final population k, the mutation rate, and the epsilon threshold for retaining or discarding parameter values.
We now elaborate on the details of each step. We begin with 3) since 1) and 2) simply involve assigning a distribution, which will be specified when the simulations are described below.

After drawing the parameter values in 1) and 2), we simulate a clonal expansion for g generations. By a clonal expansion, we mean the number of mutant cells as a function of time. This can be attained in linear time complexity through a series of binomial draws. We begin with an initial condition of one mutant cell, since the number of mutant cells is always one when the mutation first arises. Then, assuming there are $n(i)$ cells in the ith generation, the number of mutant cells in the $(i + 1)$th generation is drawn from a binomial distribution with parameters N and $p = n(i)*(1+s)/(N + n(i)*s)$ (see Section 1 for the derivation). After iterating the binomial draw g times beginning with the initial condition, we recover the number of mutant cells as a function of time (see figure below that shows the ABC schematic). If the mutant clone does not grow to at least k cells, we redraw the parameters in 2) and re-simulate 3), iterating until we have acquired an expansion that does.
It is important to note that the clonal expansion is conditioned on $n >= k$, which is equivalent to conditioning on the mutant cells escaping stochastic extinction. In general, if the clonal expansion is simulated for a sufficient number of generations, that is for a sufficiently large g, then the mutant

clone will either go extinct or grow to a large size and exhibit deterministic dynamics. Therefore, when g is sufficiently large, if the mutant clone has grown to more than *k* cells, its size will be much larger than *k* and will have escaped stochastic extinction. Conditioning on escaping stochastic extinction has the important consequence of allowing us to infer the fitness in percent growth per year from the lineage trees (see Section 1 for details).

After obtaining a clonal expansion, we randomly sample *k* cells from the population of mutant cells at the final timepoint and simulate the lineage history of only the random sample, while ignoring the lineage history of all other cells. The lineage history is constructed by letting each sampled cell choose a mutant cell at random from the previous generation to be its parent. Each mutant cell chosen from the previous generation then chooses its parent at random from the generation of mutants before, etc. This is repeated until all lineages have coalesced (figure below).

It is worth noting that simulating the number of mutant cells as a function of time and thus initially ignoring all genealogical relationships, and then simulating only the genealogical history of the random sample backwards in time is statistically equivalent to simulating the genealogical process forward in time, and then producing a tree by following the lineages of the random sample back to common ancestry. This equivalence follows from the fact that each mutant cell can be descended from any of the mutant cells in the previous generation with equal probability. Simulating the lineage history for only the subset of the k chosen cells significantly increases the speed of the simulations without the loss of any information.

Once we have simulated the lineage tree, we then simulate the mutational process. To each edge of the tree (edge refers to a single line connecting two nodes on the tree, see figure below), we assign the number of mutations drawn from a Poisson distribution, where the mean is equal to the mutation rate (in units of mutations per generation). The mutation rate is generally computed empirically from the patient tree by dividing the total length of the patient tree in mutations by the value of *L* - 1, where *L* was drawn in step 2). It is very important to note that in this case, when we refer to the length of the tree, we mean the total number of mutations from the very bottom of the tree, which corresponds to the present time, to the very top of the tree which corresponds to the birth of the patient (not to the common ancestor of the mutant cells).

Before computing the mutation rate empirically from the data tree, we need to rescale the branches of the tree so that the distance (in mutations) from any leaf to the root of the tree is the same. If we don't do this, the number of mutations from each leaf to the root of the tree would not be the same, resulting in a tree length and mutation rate that is not well-defined. We accomplished

this by applying an averaging algorithm described below. The same averaging algorithm is also applied to simulated trees immediately after they are constructed before computing the metric distance between the simulated tree and the data tree. Therefore, any information loss from the algorithm will be expressed as uncertainty in the error bars of our inference.

The averaging algorithm we designed is based on the principle that the best estimate of time to common ancestry between two lineages is the average number of mutations between the two. In particular, let's define a tree as well-averaged if the distance from any leaf to the root is the same. In pseudocode, the algorithm works by calling the following function on the parent of any two sisters:

```
Average( currentNode )
{
```

If the left subtree of currentNode is not well-averaged:

```
    Average(left child of currentNode)
```

If the right subtree of currentNode is not well-averaged:

Average(right child of currentNode)

If both the left and right subtrees are well-averaged:

Compute the average length of the left and right subtrees. Then, for both the left and right subtree, rescale the branches of the subtree proportionally so that the length of the subtree equals the average.

```
    if currentNode != root:
        Average( parent of currentNode)
    else:
        break
}
```

We begin at the parent of two sisters, where the subtrees are single branches connecting a parent node to two leaf nodes. Note that we may start at the parent of any two sisters (or even more generally, at any node) and produce the same averaged tree since averaging the two subtrees of any node produces a unique value. For implementation of this algorithm refer to our GitHub repository. See figure below for a schematic of the averaging algorithm.

After producing a well-averaged tree, we construct its LTT (Lineages Through Time) plot. The LTT plot of a tree shows the number of lineages as a function of time in mutations (figure below). The LTT plot of a tree loses all information about its topology (the way the branches are connected). However, since the mutant cells in each generation pick their parents at random from the mutant cells in the previous generation, any topology on the tree is equally likely, and thus the tree topology contains no information about the parameter values that gave rise to the tree. Therefore, LTT contains all possible information about the parameter values.

After converting the simulated tree to an LTT plot, we compute the distance between the LTT plot of the simulated tree to the LTT plot of the data tree, defined as the area between the two LTT curves. The LTT plot of the data tree is always constructed before the ABC begins by first applying the averaging algorithm we previously described so that the leaf nodes line up side by side, and then converting it to an LTT plot. If the area between the two plots is smaller than the epsilon threshold, we retain the parameter values s, N, L, and g drawn from the priors, as well as the cancer trajectories $n(t)$ and the LTT curve produced, and if the area is >= epsilon we discard them. This process is iterated until a sufficient number of parameter values (along with the trajectories and LTT plots) to construct a convergent posterior distribution is retained.

The LTT curves start at zero but may end at different values because of different tree lengths. The area between two LTT curves that do not end at the same point on the x axis is undefined. To address this, we extend the end points of LTT curves, which corresponds to a value of 1, to infinity.

Since the lengths of LTT curves tended to vary, we decided to divide the area by k^*(the length of the data tree) before checking if epsilon was smaller than the threshold. This allowed us to run ABC without having to choose a new epsilon for each tree, since a smaller epsilon would be required for a tree of smaller length, and a larger epsilon for a tree with a larger length. Intuitively, this is equivalent to taking the percent difference between the data tree and the simulated trees.

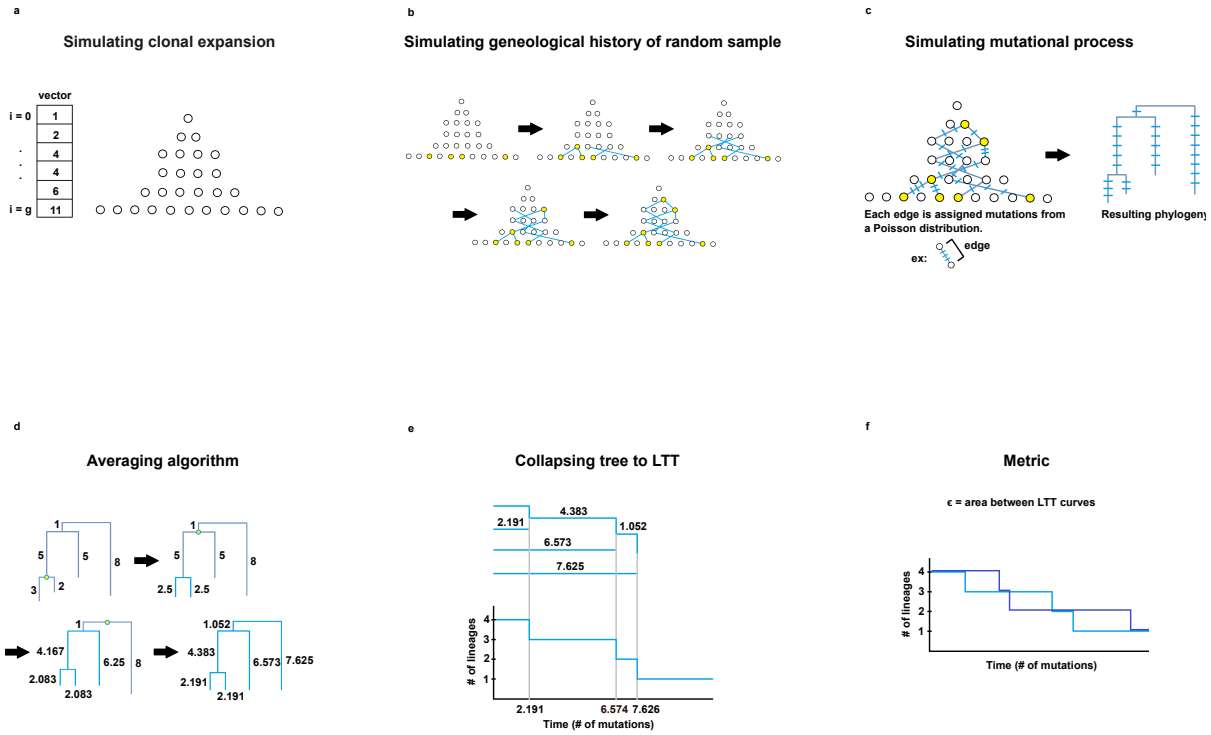


Figure. Schematic of the Approximate Bayesian Computation. First, the parameters that determine the growth dynamics (i.e. fitness, population size, total number of stem cells, and age of onset) are randomly drawn from a prior distribution. a. The clonal expansion of the cancer cells given the selected parameters is simulated. b. A specified number of cells is randomly chosen from the final population. The lineage tree is reconstructed for these cells. c. We assign a number of mutations accrued along each branch by drawing from a Poisson distribution at each edge with the mutation rate. Those mutations are shown pictorially as blue dashes on the tree. The resulting tree is a phylogeny with branch lengths in mutations as opposed to generations. d. Next, the branch lengths are scaled so that the total number of mutations from the root of the tree to each leaf node is the same. e. The rescaled trees are then plotted as LTT curves. f. A distance is computed between the LTT curve of the simulated tree and the observed tree. If the distance is below a threshold, the initial set of parameters is retained, otherwise they are discarded. This process is iterated.

Simulating data

To test the robustness of our ABC inference, we were interested in inferring the model parameters from simulated data where the ground truth is known. The way we simulated data as follows:

1. Draw *s* from its prior distribution.
2. Draw *N*, *L*, and *g* from their prior distributions.

3. Simulate a clonal expansion for g generations.
4. If the final number of mutant cells $n < k$, where k is the number of mutant cells, we sample from the final population, back to 2). Else, move on to 5).
5. Sample k mutant cells from the final population and simulate their lineage history.
6. Simulate the mutational process on the tree with the given mutation rate.
7. Perform the averaging algorithm on the tree.

The resulting tree is used as the data tree in ABC, and its parameters are inferred. Note that these steps are simply the first 7 steps of ABC.

In silico validation of the inference algorithm

To validate our inference, we decided to test the inference on simulated data over a wide range of parameter values. We began by simulating 30 trees as data with our model, where the underlying parameter values were known. Each tree was constructed using the following specifications:

1. s was drawn from a uniform distribution on (0, 1.2).
2. N was drawn from 10^X, where X is uniformly distributed on (1, 9).
3. L was drawn from round(Y), where Y is a Gaussian with mean 35 and std 5. If we drew $L < 2$, we redrew L until L >= 2 since at least 2 generations are necessary to produce a tree.
4. g was drawn uniformly on 2, … , L.
5. k = 22 mutant cells were randomly sampled. (22 is the number of mutant stem cells sampled for ET 1 patient data)
6. The mutation rate was 723/(L - 1). 723 was the number of mutations observed in ET 1 patient data.

We then inferred the parameters for each tree using ABC. For the ABC, we used the exact same specifications as the data to generate trees for comparison, except that we instead drew s from a uniform distribution on (0, 5) in Step 1, and the mutation rate was instead estimated empirically as (total length of tree in data in # of mutations)/(L - 1) in Step 6. Epsilon was set to 0.03, since this threshold was sufficient to obtain an inferred distribution that converges to the posterior distribution for most inferences, while also allowing a large number of points to be retained for the inferred distribution. The simulations were run until they accrued ~10,000 or more points for the posterior.

In the figure called “In silico validation of the inference algorithm without feedback,” we show a representative set of 10 inferences out of the 30 inferences we ran.

To quantify the accuracy of our ABC inference, we then simulated 200 trees in precisely the same way as above, except that the value of *s* for the tree data was drawn uniformly on (0, 2) instead to obtain data across a much wider range of fitness values. We then carried out ABC inferences on each tree with epsilon = 0.0225 until most of the posterior distributions had accrued ~400 or more points. Tree data where the inference accrued less than 30 points for the posterior were excluded.

We then applied the following filters to the data:

1. We excluded data trees where the ratio of the standard deviation to the mean of the posterior of *s* (in percent growth per year) was greater than 0.425
2. We excluded data trees where the std of *n* was larger than 1.15

We arrived at the first filtering criterion by noting that inferences for small s tended to have large error bars relative to their inferred means (or large coefficient of variation), and their inferred means were generally inaccurate and much larger than the true values. ABC inference cannot determine whether a small number of cells at the final time-point is due to a small growth rate *s* or small saturation limit (see figure “In silico validation of the inference algorithm with feedback,” row 2 column 2). In both scenarios the population size is small and coalescence events occur rapidly, producing similar trees. We reasoned that if expansions produced by small s produce ABC inferences characterized by large coefficient of variation, then by eliminating ABC inferences exhibiting this characteristic we could exclude inaccurate inferences without any knowledge of the ground truth.

Similarly, we arrived at the second filtering criterion because simulations that expanded to sufficiently large population sizes generated inferred log(*n*) distributions with large standard deviations and mean values distributed around 10^6. This suggested that ABC was extracting little information from the data and that the log(*n*) distributions were almost identical to the prior. We reasoned that by filtering out inferences with large log(*n*) standard deviations we could exclude inaccurate inferences without any knowledge of the ground truth value. We emphasize that this filtering procedure does not use the ground truth values in any way. The inference is deemed inaccurate if the posterior distribution width is too large regardless of the ground truth value. Therefore, the filtering procedure can also be applied to actual data where the ground truth is not known. Finally, when devising the filtering criteria, we were conservative with our choices. As such, the interpretation of these data was not sensitively dependent on the filters we chose.

The inferred vs true values of the inferences is plotted in the figure called “ABC accurately infers model parameters”.

Next, we quantified the accuracy of our ABC inference for a 63-year-old patient in a similar manner by simulating 200 trees, applying filters to the data, and then plotting the inferred vs true values (figure “ABC accurately infers model parameters”). The data were produced in a similar manner as for the 34-year-old patient using the following criteria:

We simulated 200 trees as data for a 63-year-old patient using the following specifications:

1. *s* was drawn from a uniform distribution on (0, 2).
2. *N* was drawn from 10^X, where *X* is uniformly distributed on (1, 9).
3. *L* was drawn from round(*Y*), where *Y* is a Gaussian with mean 64 and std 10. If we drew *L* < 2, we redrew *L* until L >= 2 since at least 2 generations are necessary to produce a tree.
4. *g* was drawn uniformly on 2, … , *L*.
5. we sampled *k* = 13 cancer cells, which is the number of sampled cells in the ET 2 patient data.
6. The mutation rate was 1205/*L*. 1205 was the number of mutations observed in ET 2 patient data.
7. No feedback was included.

We then carried out ABC inferences on each tree using the same specifications as the data, except that we drew s from (0, 5) uniformly in Step 1, and the mutation rate was estimated empirically as (total length of tree in data)/(*L*-1) for 6). Epsilon was set to 0.0125. The inferences were left running until about half of them (many ABC inferences accrued little to no points for the inferred distributions) had accrued ~100 or more points for the posterior distribution. Many simulations accrued little or no points, and so we excluded trees with less than 30 points.

We then applied the following filters to the data:

1. We excluded data trees where the mean of the posterior of s (in percent growth per generation) was larger than 1.5.
2. We excluded data trees where the ratio of the std to the mean of the posterior of s (in percent growth per year) was greater than 1.5.

Similar reasoning was applied to devise the above filtering criteria. Mainly, values outside of above criteria contain little information beyond the prior distributions.

ABC on simulated data with feedback

Our model of growth dynamics of the mutant cells only approximates the actual growth dynamics. In particular, the population of mutant cells seems to saturate when it has expanded to a certain fraction of the total population of stem cells. Therefore, it is conceivable that the mutants lose their fitness advantage as their population size increases. Here, we set out to test whether the inference of the parameters of the simple model of growth dynamics remains accurate if the actual dynamics is simulated using a different model. To do so, we constructed a model with feedback, whereby the fitness advantage of the mutant cells decreases compared to wild-type cells as their population size increases. The model with feedback is described in detail in Section 1.

We then repeated the simulations carried out for 34-year-old and 63-year-old patients, except that we used feedback with x = 30 when generating the simulated data (see Section 1 for definition of x parameter). The ABC did not incorporate feedback in the model, since we were interested in how well we could infer the parameter values if the ground truth incorporated feedback. For the 10 example trees shown in the figure called “In silico validation of the inference algorithm with feedback,” we ran ABC until most of the inferences had accrued ~5,000 or more points for the posterior. For the 200 trees for the 34-year-old (figure “ABC accurately infers model parameters”), we ran ABC until most of the inferences had accrued ~300 or more points for the posterior. For the 200 trees for the 63-year-old (figure “ABC accurately infers model parameters”), we ran ABC until about half of the inferences had accrued ~50 or more points for the posterior. Inferences that accrued less than 30 points were always excluded. The filters were applied in an identical fashion as for the inferences without feedback.

Taken together, the inferences suggest that the ABC inference can infer model parameters over a wide range of parameter values, regardless of whether or not feedback is incorporated in the underlying model. In particular, *s* in percent per year and the age of onset of the disease can be inferred from lineage trees, even if feedback is incorporated. However, it appears n can only be inferred for the 34-year-old patient. The inferred *n* vs true *n* plots for the 63-year-old patient indicate that the trees have no information about the number of mutant cells at the final time point. This is likely due to the fact that we have sampled only 13 lineages (as opposed to 22 lineages for the 34-year-old patient), and that the clone has expanded for much longer. Because of this, most coalescent events occur in the early history of the expansion, and information about the dynamics of the later history are lost.

Age = 34 and x = 0 (no feedback)

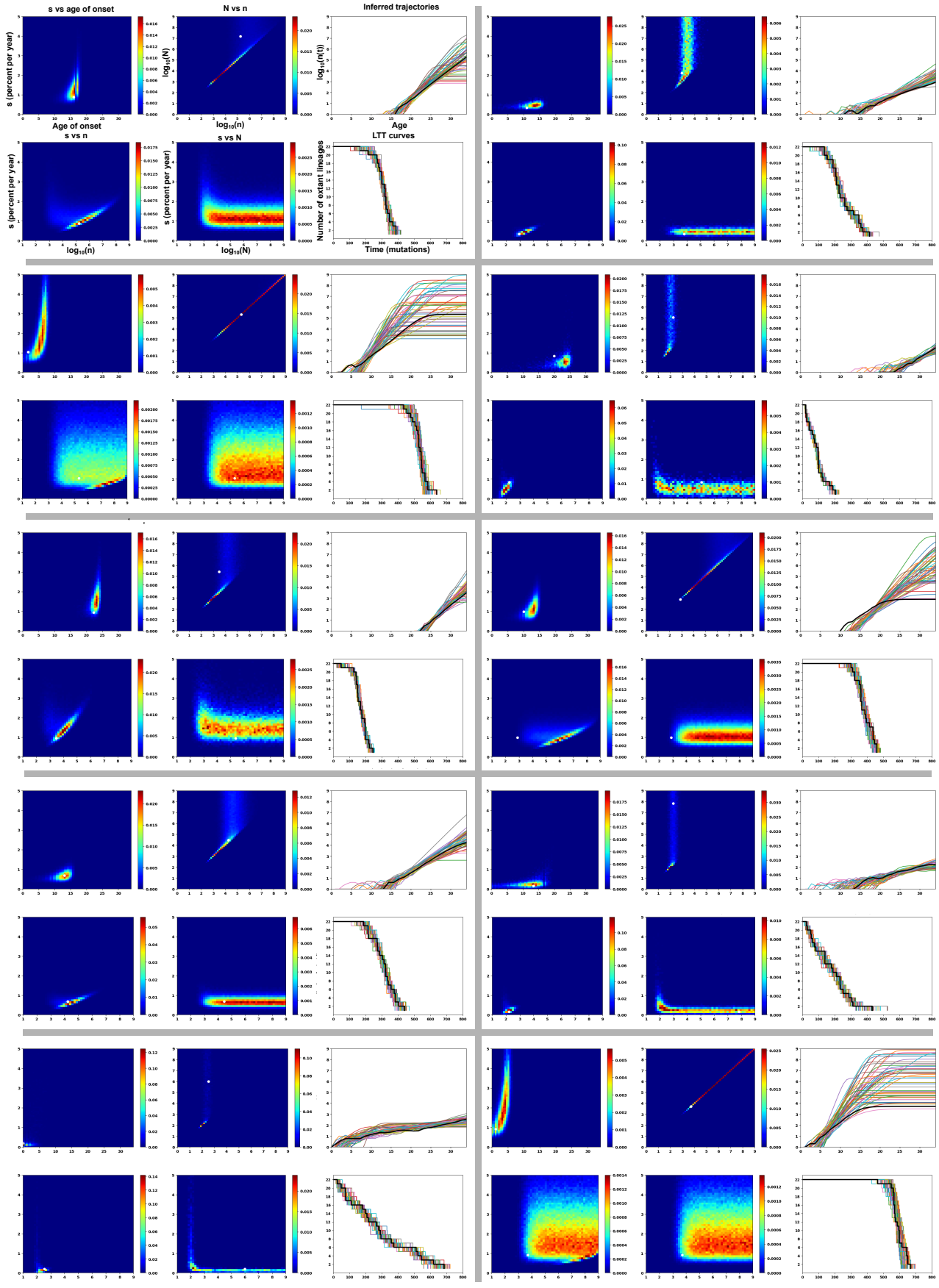


Figure. In silico validation of the inference algorithm without feedback. To validate the inference algorithm, we generated simulated ground truth growth dynamics and then inferred the parameters using ABC. In these simulations, the ground truth dynamics were simulated using the same model as in the ABC. In all the heatmaps, the ground truth parameters are shown as white dots. In the traces, the ground truth is shown in black. 10 illustrative examples are shown. For each example, the heatmaps of inference of s (fitness parameter), n (number of cancer cells), N (total populations size), and g (age of onset) are shown alongside the LTT curves that were retained and the inferred trajectories of population growth.

Age = 34 and x = 30 (with feedback)

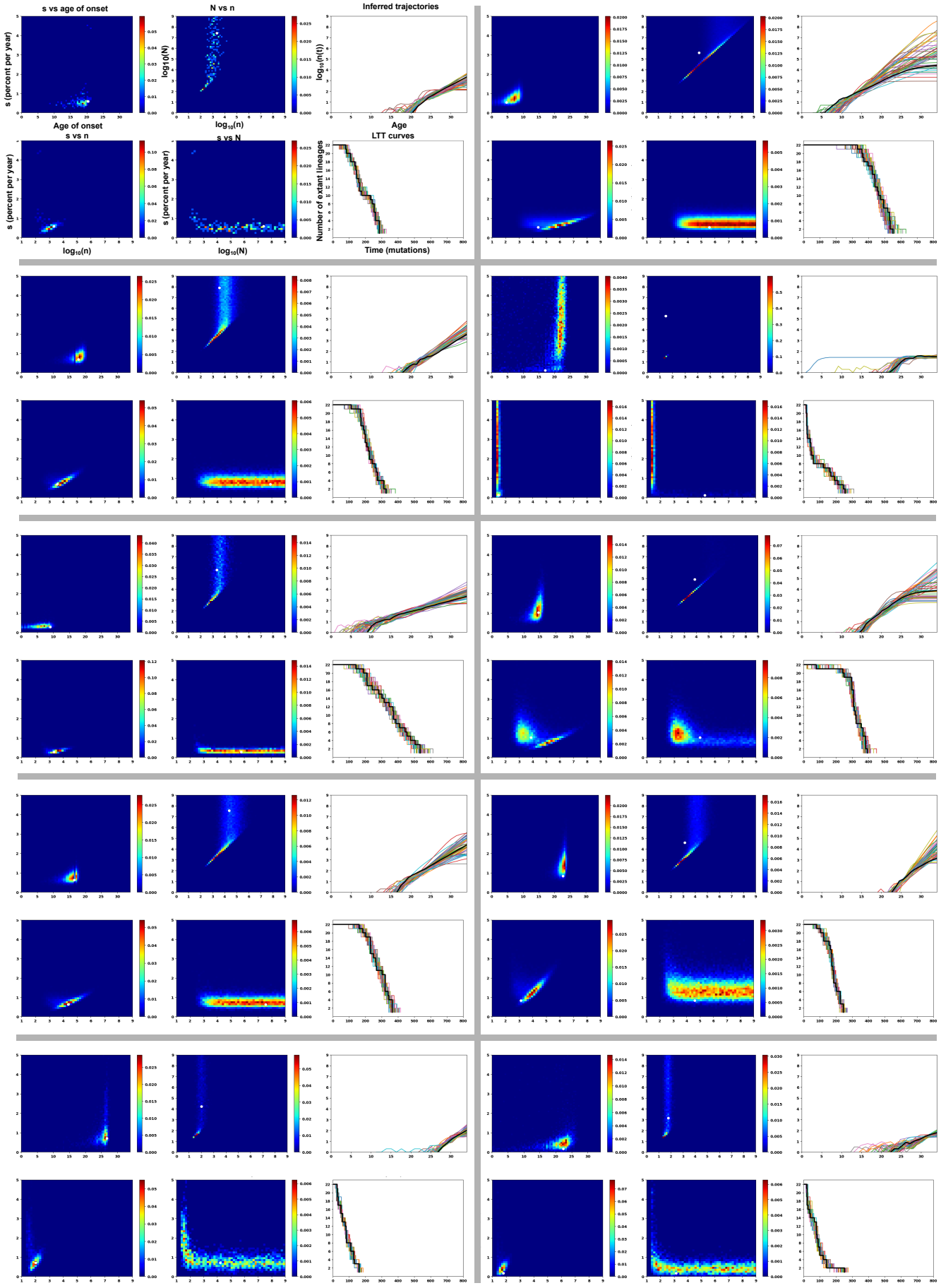


Figure. In silico validation of the inference algorithm with feedback. To validate the inference algorithm, we generated simulated ground truth growth dynamics and then inferred the parameters using ABC. In these simulations, the ground truth dynamics were simulated with a feedback where the fitness of cancer cells decreased as the population size increased. The growth dynamics for generating trajectories for ABC did not incorporate the feedback. In all the heatmaps, the ground truth parameters are shown as white dots. In the traces, the ground truth is shown in black. 10 illustrative examples are shown. For each example, the heatmaps of inference of s (fitness parameter), n (number of cancer cells), N (total populations size), and g (age of onset) are shown alongside the LTT curves that were retained and the inferred trajectories of population growth. Taken together, accurate inference is possible even if additional features, such as feedback, are not incorporated in the ABC dynamics.

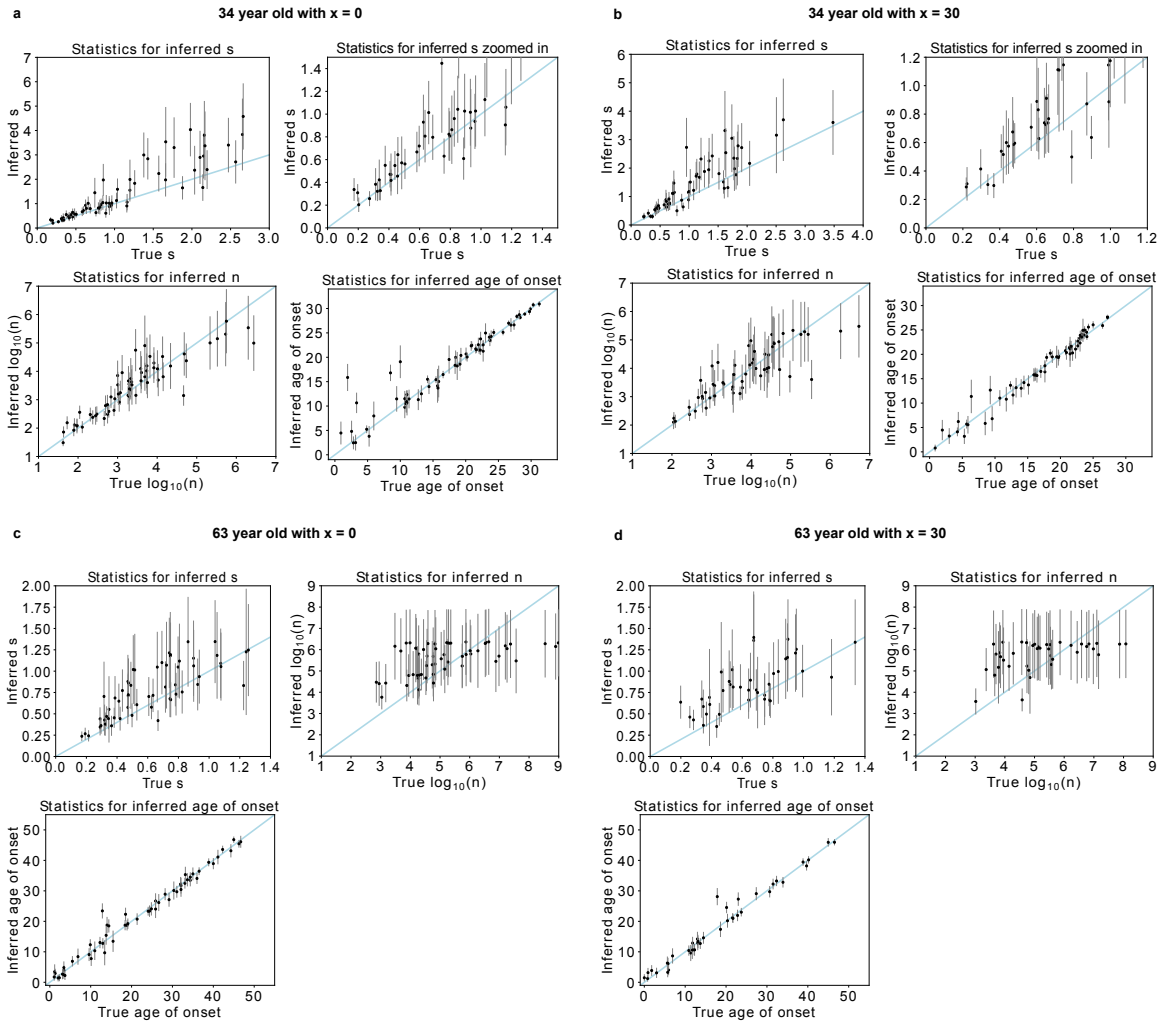


Figure. ABC accurately infers model parameters. To verify that the inference framework can accurately infer model parameters, we simulated growth dynamics across a range of scenarios (corresponding to different parameter values) and inferred the parameters using ABC. The inferred parameter values were plotted against their true values for s (percent per year), n, and age of onset of the disease (years), and error bars were included to denote 1std in the inference. Filters were applied to exclude inferences with large error margins, and were done completely agnostic of the ground truths. a-b Many iterations of ground truths were simulated for a 34-year-old patient. In a., the model used to simulate ground truths did not incorporate feedback, while in b. the model did. In both cases, the model used for the ABC inference did not incorporate feedback but was able to infer the parameters correctly within the statistical error. c. same as a. and d. same as b., except the ground truth simulations and ABC inferences were carried out for a 63-year-old patient. In both c. and d., the model used for the ABC inference also did not incorporate feedback, but was able to infer the parameter values of s and the age of onset of disease within the statistical error. However, for c. and d., n could not be inferred since there were not enough coalescent events in the later history to extract information about its dynamics after the initial expansion. For the inferred n vs true n plots in c. and d., we decided to show them with no filter so the reader could see that the inferences contain no information.

Fitness can be inferred without knowing the number of generations

So far, we have shown that our inference is robust to feedback but have not shown how well we can infer the parameter values if our assumption about the total number of generations is incorrect. Surprisingly, fitness, when converted to percent growth per year, can always be inferred without knowing the number of generations (Section 1). To validate this, we simulated ~10 data trees for a 34-year-old patient and carried out ABC on them. We then selected a data tree where the ABC inference precisely inferred the parameter values. The data tree had been simulated with the following specifications:

1. Parameter values were fixed at $s = 0.264911$, $N = 10^9$, $g = 50$, $L = 70$, arbitrary values for which the inference was accurate.
2. $k = 22$ cells were sampled
3. The mutation rate was 723/69 per generation

We then inferred *s*, *n*, and the age of onset of the disease having kept all other parameter values fixed, but assuming that the number of generations *L* was *c* times 70 (the ground truth *L*) in the

ABC model. More precisely, for each c = 0.5, 1, 2, 4, 8, 16, 32, 64 we ran an ABC inference on the data tree with the following specifications:

1. s was drawn uniformly on (0, 10/c)
2. We fixed $N = 10^9$ and $L = c*70$
3. g was drawn uniformly on 2, … , L
4. A mutation rate of 723/(70*c - 1) was used
5. An epsilon distance of 0.02 was used

The inferences were run until the ABC had accrued ~15,000 points for the posterior distribution. We then plotted inferred joint distributions for s in growth per generation, s in growth per year, age of onset of the disease, and n (figure below).

As expected, when increasing the number of generations assumed by the model for ABC inference, the inferred s in percent growth per generation decreased while the inferred percent growth per year remained invariant. In theory, the decrease in growth per generation will increase the rate of stochastic extinction, and so the number of mutant cells must fluctuate to a larger population size to escape stochastic extinction. As expected, the number of mutant cells increased at the final time point.

Taken together, our simulation results are consistent with our theoretical calculations (Section 1) in that fitness, in percent growth per year, and the age of onset in years can be inferred from lineage trees without prior knowledge of L, while prior knowledge of L is necessary to infer n.

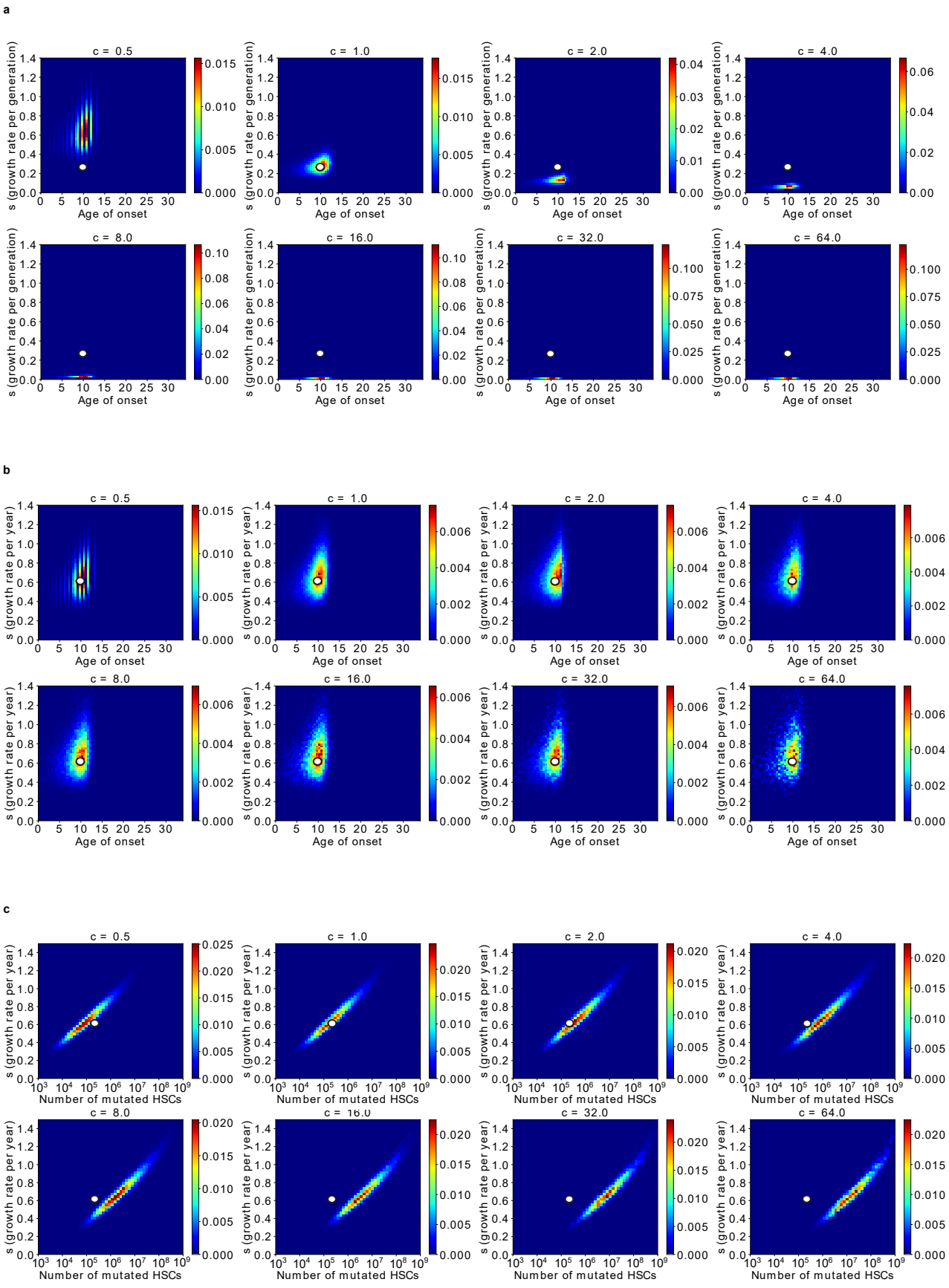

a
c = 0.5
c = 1.0
c = 2.0
c = 4.0
c = 8.0
c = 16.0
c = 32.0
c = 64.0
s (growth rate per generation)
Age of onset
b
c = 0.5
c = 1.0
c = 2.0
c = 4.0
c = 8.0
c = 16.0
c = 32.0
c = 64.0
s (growth rate per year)
Age of onset
c
c = 0.5
c = 1.0
c = 2.0
c = 4.0
c = 8.0
c = 16.0
c = 32.0
c = 64.0
s (growth rate per year)
Number of mutated HSCs

Figure. Fitness can be inferred without knowing the number of generations. Ground truth growth dynamics was simulated, and multiple ABC inferences were carried out under different c values. For each value of c, the number of generations used in ABC was multiplied by c for the ABC inference. a. inferred s (percent per generation) vs inferred age of onset of disease (years) are plotted for different values of c. As expected, increasing c, and hence the number of generations, decreases the growth per generation to keep the percent growth per year invariant. b. Inferred s was then converted to growth per year. As suggested by our derivation, the inferred s in growth per year remains invariant as the number of generations is scaled. c. The inferred s (percent per year) vs the inferred number of mutated HSCs at the final time-point is plotted for different values of c. As expected, increasing c, and hence the number of generations, decreases the growth per generation and thereby the rate of stochastic extinction. The number of mutated HSCs must then fluctuate to a larger population size early on to escape extinction. This results in a larger number of mutated stem cells at the final time-point.

The analytical calculation of coalescent times matches the simulations results

In Section 1, we provide an analytical calculation of the average coalescent times of our model, and show that the average coalescence times do not change if we scale the number of generations while keeping the percent growth the same (suggesting that fitness can be inferred without prior knowledge of the number of generations).

To validate our analytical calculation of coalescent times, we performed the following simulations. For each s = 0.1, 0.3, … 1.5 (growth per generation), we constructed thousands of data trees using the following specifications:

1. g = 25, L = 35, $N = 10^9$ (with the corresponding s)
2. k = 22 cells were randomly sampled
3. A mutation rate of 723/34 was used

We then converted the branches of each data tree to years, assuming the tree was for a 34-year-old patient, by multiplying the branch lengths by 34 / 723. Then, for each value of s separately, we constructed a distribution for each of the i = 1, …, 21 coalescence times using the corresponding trees. We computed the means and standard deviations of those distributions and plotted them (figure below).

The analytical derivation of coalescence times therefore matches the simulated coalescence times within a standard deviation. Our derivations also predict that the coalescence times of a tree should not change when scaling the number of generations while keeping the percent growth per year fixed. To verify this occurs in our simulated trees, we did the following:

For each s' = 0.1, 0.6, 1.1 and for each c = 0.5,1, 2, 10, 100, we simulated thousands of data trees using the following specifications:

1. $s = (1 + s')^{1/c} - 1$ (this s, in growth per generation, along with the L in specification 2), keep the percent growth per year invariant (See supplemental).
2. $L = c*35$, $g = c*25$, $N = 10^9$,
3. 22 cells were randomly sampled
4. A mutation rate of $723/(c*35 - 1)$ was used

We then converted the branches of each data tree to years, assuming the tree was for a 34-year-old patient, by multiplying the branch lengths by $(c*35 - 1)/723$. For each combination of s and c, we constructed distributions for each of the $i = 1, \ldots, 21$ coalescence times of the corresponding trees. We then computed the means and standard deviations of the distributions and plotted them (figure below).

Consistent with our theoretical predictions, scaling the number of generations while keeping the percent growth per year invariant appears to not significantly change the average times until coalescence. This implies that trees are indistinguishable when the percent growth per year is the same, even if the number of generations is different, showing that percent growth per year can be inferred from lineage trees without knowing L.

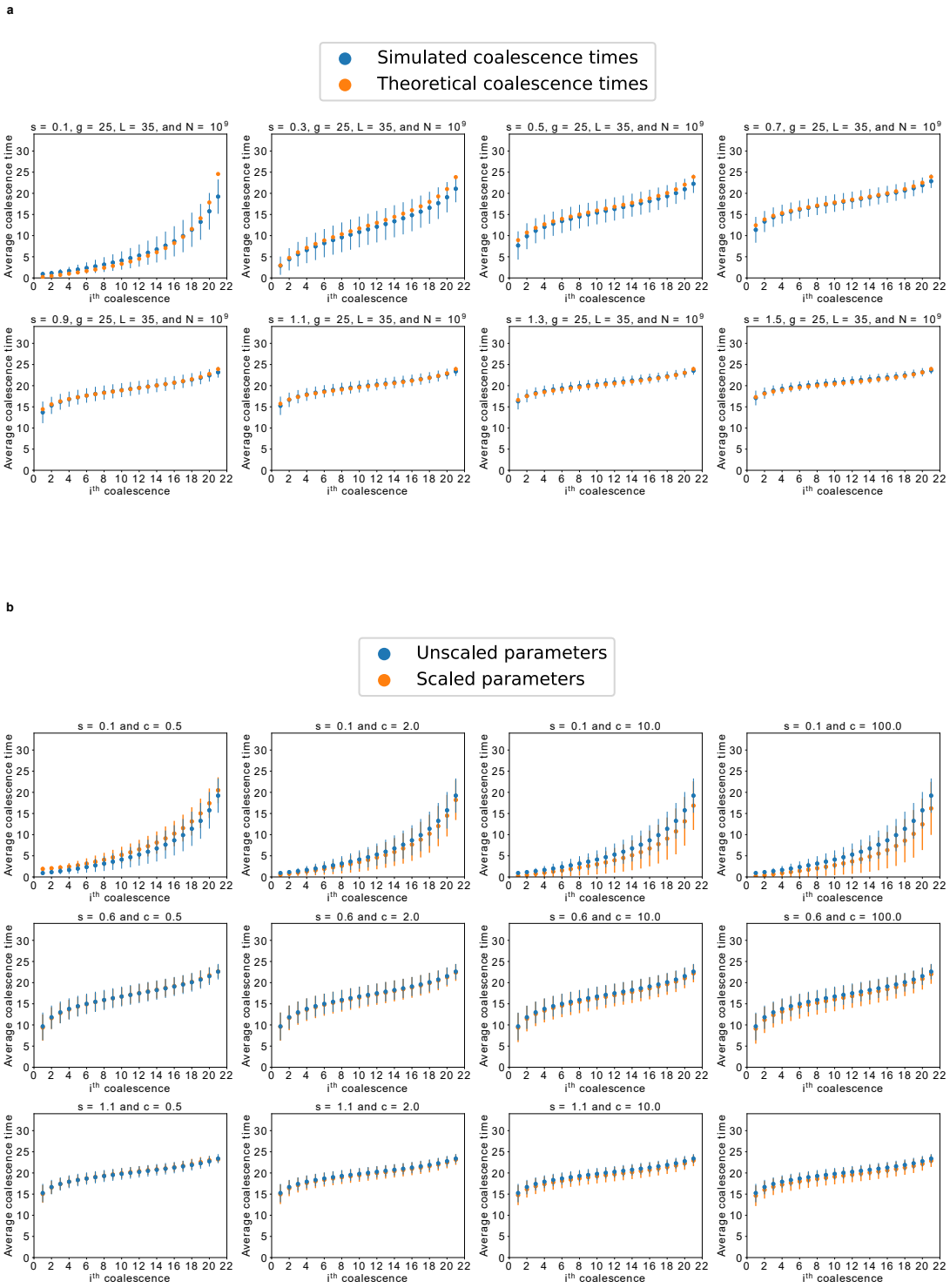


Figure. The analytical calculation of coalescent times matches the simulations results. a. We verify that our analytical calculation of the average coalescence times of our model matches the empirically observed coalescence times. Blue dots show the average coalescence times computed empirically for different s by simulating a large number of cancer expansions,

constructing trees, and then averaging over the times until coalescence. Error bars denote 1 std in the coalescence times. Orange dots show the average coalescence times derived analytically for comparison. b. Our mathematical derivation shows that scaling the number of generations by a factor of c while maintaining the percent growth per year, s, fixed does not change the average coalescence times of trees. We take s = 0.1, 0.6, and 1.1 in percent growth per year corresponding to rows 1, 2, and 3 respectively, scale the number of generations by different factors of c, and compute the average coalescent times empirically. Blue dots are the empirically computed coalescent times before scaling the number of generations by c, and the orange dots are the empirically computed coalescent times after scaling the number of generations by c. The average coalescent times using scaled parameter values always lie within 1 std of the average coalescent times using unscaled parameters. The small deviations are due to the fact that s in growth per year is not sufficiently small when the number of generations is not scaled. As c increases, increasing the number of generations, the growth per generation decreases to maintain constant percent growth per year. As the growth per generation decreases, the coalescent times converge according to our mathematical derivation.

ABC on simulated data with fitness heterogeneity

The Wright-Fisher model assumes that each cell at each generation has the same fitness value. However, it is conceivable that there is heterogeneity in the fitness of cells, or more precisely that the fitness of each cell comes from a distribution. To test the robustness of our inference to the assumption of homogeneous s, we simulated 20 data trees using the following specifications:

1. We fixed parameter values $N = 10^7$, $g = 25$, $L = 35$.
2. Clonal expansions were simulated for g generations, but with a slight modification to the Wright-Fisher process with selection that we use for ABC. In particular, each generation of cells creates the next generation of cells according to the following rule: Non-cancer cells have probability p of being chosen as a parent by a cell in the next generation, and the i^{th} cancer cell has probability $(1 + s_i)$*p of being chosen, where the s_i are i.i.d. Gaussians with mean 0.6 and std 0.2. That is, a fitness value is assigned to each cell in the current generation, and then each cell from the next generation picks a parent cell randomly according to the probabilities determined by the fitness values drawn from a Gaussian.
3. $k = 22$ mutant cells were randomly sampled
4. The mutation rate was $723/(L - 1)$

We then inferred the parameter values for each tree using the same specifications as the ABC model used for the 34-year-old patient data (see Inference on patient data at the end of the document). Results were plotted in the figure below and show that the mean growth per year can be accurately inferred, even when incorporating Gaussian noise.

To test how sensitive our results are to the assumed fitness distribution, we decided to simulate 20 data trees in an identical fashion as before, except that instead of drawing from a Gaussian we chose to draw from a uniform distribution on (0, 1.2). Results were plotted in the figure below and suggest that the mean growth per year can be accurately inferred, even when the fitness distribution is highly variable.

The Wright-Fisher model assumes a specific amount of genetic drift, an assumption that is often neglected. To test the impact of this assumption on our inference, we simulated 20 data trees using the following specifications:

1. We fixed parameter values $s = 0.6$, $N = 10^7$, $g = 25$, $L = 35$.
2. Clonal expansions were simulated for g generations. At each generation, 25% of the total population of stem cells were randomly “inactivated”, so that when the lineage trees were simulated, the cells could not select an “inactivated” cell. This forces a smaller number of stem cells to contribute to the exponential growth, so that genetic drift occurs more rapidly. If all the cells in a generation were randomly inactivated by chance, the clonal expansion was re-simulated since this represents an extinction event.
3. $k = 22$ mutant cells were randomly sampled
4. The mutation rate was $723/(L - 1)$

Similar to the previous simulations, we used the ABC model specifications that were applied to the data tree for the 34-year-old. The results are plotted in the figure below, and show that the inference is robust to a different amount of genetic drift.

We then decided to simulate a more realistic version of the previous scenario, where cells are continuously killed at a constant rate. 20 simulated trees were constructed in an identical fashion as above, except that instead of simulating the entire clonal expansion first and then inactivating

cells at each generation, the cells were killed off simultaneously with the exponential growth. In particular, before a new generation was produced, 25% of the total number of stem cells were randomly killed, and this new smaller pool of stem cells was used to create the next generation of cells. Note that in this case, the p-value of the binomial parameter would be computed using *N**0.75 instead of *N*, before *N* cells are chosen from current generation to create the next generation of cells. This is subtlety distinct from the drift scenario, where the parameter values for the binomial draws do not factor the temporary reduction in population size due to the random killing. The results are plotted in the figure below and show that the inference is also robust to randomly killing cells.

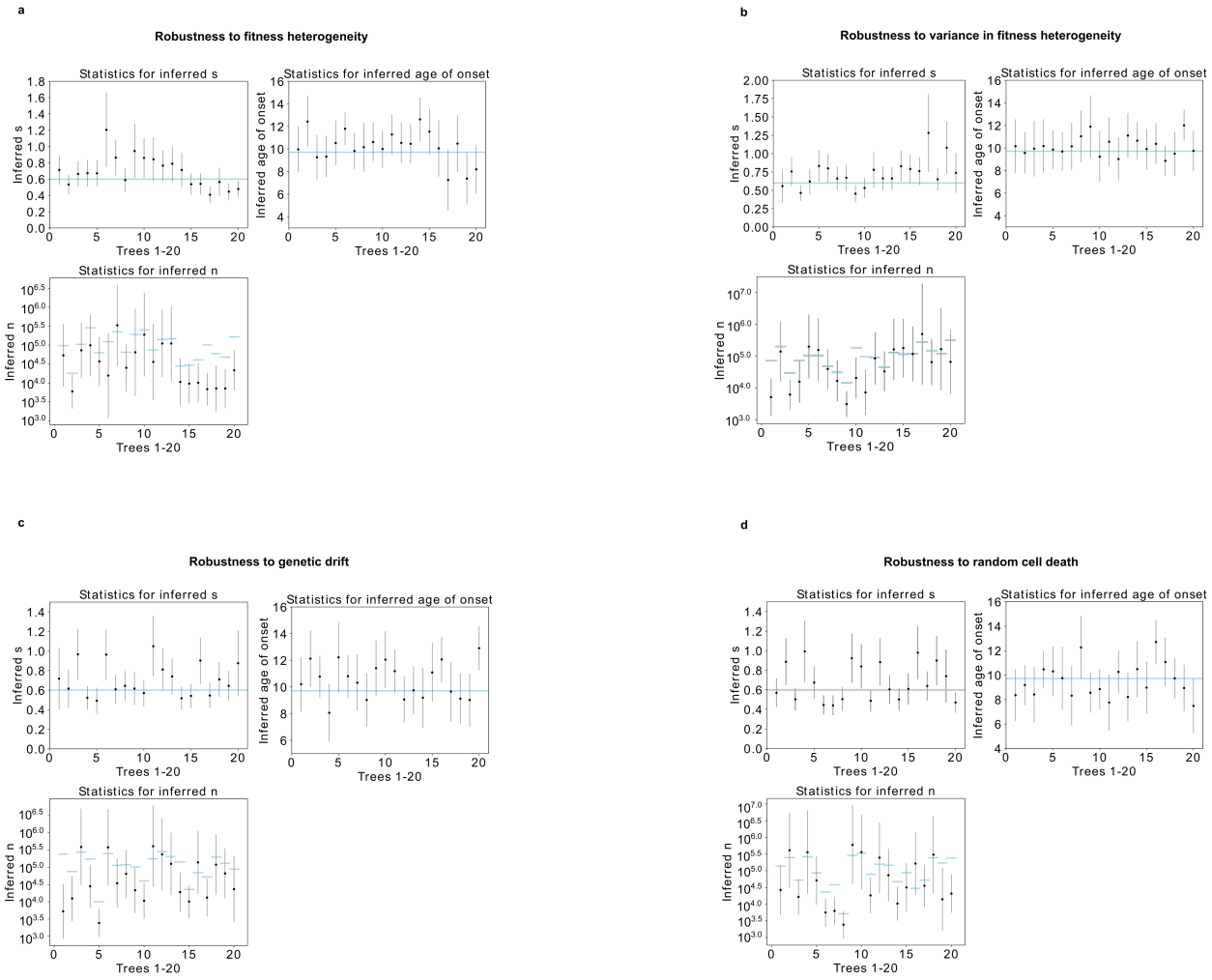


Figure. Robustness to cell heterogeneity. a. 20 data trees were simulated for a 34-year-old patient using parameter values g = 25, L = 35, and N = 10^7, and where the fitness of each cell was drawn from a Gaussin with mean 0.6 and std 0.2. ABC inferences were then carried out on each data tree using the usual model that was applied to data, which assumes s is the same for

each cell. The inferred mean of the posterior distributions for each tree were plotted with error bars to denote 1 std. The blue lines represent the ground truth parameter values. For the inferred vs true s figure, the blue line is just the mean growth per year of the clonal expansions. b. These figures were constructed in an identical fashion to figures 1b, except that s was instead drawn from a uniform distribution on (0, 1.2). c. 20 data trees were simulated for a 34-year-old patient by running clonal expansions with parameter values s = 0.6, g = 25, L = 35, and N = 10^7. After each clonal expansion, genetic drift was increased by inactivating 25% of the total number of stem cells at each generation so that ~75% of the cells were contributing to the exponential growth at each time step. ABC inferences were then carried out on each data tree using the usual model that was applied to data, which did not include the additional genetic drift. The inferred mean of the posterior distributions were plotted for each tree with error bars to denote 1 std. The blue lines represent the ground truth parameter values. d. 20 data trees were simulated for a 34-year-old patient by running clonal expansions with parameter values s = 0.6, g = 25, L = 35, and N = 10^7, except that 75% of the total population of stem cells was randomly killed at each generation before the next generation of cells was birthed. ABC inferences were then carried out on each data tree using the usual model that was applied to data, which does not include random killing. The inferred mean of the posterior distributions were plotted for each tree with error bars to denote 1 std. The blue lines represent the ground truth parameter values.