BIOINFORMATICS

Vol. 00 no. 00 2005
Pages 1–9

Fast stochastic algorithm for simulating evolutionary population dynamics

William H. Mather[1,2,3], Jeff Hasty[1,2,3,4], Lev S. Tsimring[2,3*]

[1]Department of Bioengineering, University of California, San Diego, CA, USA
[2]BioCircuits Institute, University of California, San Diego, CA, USA
[3]San Diego Center for Systems Biology, San Diego, CA, USA
[4]Molecular Biology Section, Division of Biological Sciences, University of California, San Diego, CA
, USA

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT
Motivation: Many important aspects of evolutionary dynamics can only be addressed through simulations. However, accurate simulations of realistically large populations over long periods of time needed for evolution to proceed are computationally expensive. Mutant can be present in very small numbers and yet (if they are more fit than others) be the key part of the evolutionary process. This leads to significant stochasticity that needs to be accounted for. Different evolutionary events occur at very different time scales: mutations are typically much rarer than reproduction and deaths.
Results: We introduce a new exact algorithm for fast fully stochastic simulations of evolutionary dynamics that include birth, death, and mutation events. It produces a significant speedup compared to the direct stochastic simulations in a typical case when the population size is large and the mutation rates are much smaller than birth and death rates. The algorithm performance is illustrated by several examples that include evolution on a smooth and rugged fitness landscape. We also show how this algorithm can be adapted for approximate simulations of more complex evolutionary problems and illustrate it by simulations of a stochastic competitive growth model.
Contact: ltsimring@ucsd.edu

1 INTRODUCTION

Natural evolution is an inherently stochastic process of population dynamics driven by mutations and selection, and the details of such evolutionary dynamics are increasingly becoming accessible via experimental investigation (Finkel and Kolter, 1999; Ruiz-Jarabo *et al.*, 2003; Barrick *et al.*, 2009; Pena *et al.*, 2010; Chou *et al.*, 2011). The importance of stochasticity comes from the fact that populations are always finite, mutations are random and rare, and at least initially, new mutants are present in small numbers. This realization prompted intensive studies of stochastic effects in evolutionary dynamics (Gillespie, 1984; Baake and Gabriel, 2000; Jain and Krug, 2007; Desai *et al.*, 2007; Brunet *et al.*, 2008; Hallatschek, 2011). Most of the models in these studies consider a reproducing population of individuals which are endowed with genomes that can mutate and thus affect either reproduction or death rate, as with the classical Wright-Fisher (Fisher, 1930; Wright, 1931) and Moran models (Moran, 1958) which describe a fixed population of replicating individuals. Specific models vary in the details of fitness calculation and mutation rules, but recent theoretical studies of even relatively simple models lead to non-trivial predictions on the rate of evolution as a function of the population size and the details of the fitness landscape (Tsimring *et al.*, 1996; Kessler *et al.*, 1997; Rouzine *et al.*, 2003; Desai *et al.*, 2007; Brunet *et al.*, 2008; Hallatschek, 2011). However, the complexity of more realistic evolutionary models makes them analytically intractable and requires researchers to resort to direct numerical simulations in order to gain quantitative understanding of underlying dynamics.

On the most basic level, an evolutionary process is a Markov chain of discrete reactions of birth, deaths, and mutations within a population of individuals. A direct and exact way of computing individual evolutionary "trajectories" is to use the Stochastic Simulation Algorithm (SSA) (Gillespie, 1977) or its variants (Gillespie, 1976; Gibson and Bruck, 2000; Lu *et al.*, 2004), in which birth, death, and mutation events are treated as Markovian "reactions." Unfortunately, for realistically large population sizes, direct stochastic simulation of even simple models becomes prohibitively expensive. Hence, there is an acute need for developing accelerated methods of stochastic simulations of evolutionary processes. Such methods usually involve approximations to the exact stochastic process based on certain small or large parameters that characterize the problem (for example, population size or mutation rates). Several approximate methods have been developed in recent years in the context of stochastic biochemical kinetics (Gillespie, 2001; Rathinam *et al.*, 2003; Cao *et al.*, 2005; Rathinam and El Samad, 2007; Jahnke and Altintan, 2010). Recently, Zhu et al. (Zhu *et al.*, 2011) proposed an approximate hybrid algorithm suitable for simulation of evolutionary dynamics by combining the τ-leap algorithm (Gillespie, 2001) appropriate for abundant sub-populations that do not change their sizes much between individual events, and the direct SSA algorithm for small sub-populations. This method allows one to use large time steps during which multiple birth and death reactions may have occurred. However it slows down dramatically after a new mutant has been produced, since the algorithm resorts to

[*]to whom correspondence should be addressed

the direct SSA for all events in which the new mutants are involved until the population of the new mutant class reaches a pre-defined threshold.

Here we develop a novel *exact* algorithm for simulation of evolutionary dynamics of a multi-species population undergoing asexual reproduction, death, and mutation. Unlike the direct SSA, it only samples the evolutionary process at the times of mutations. Stochastic contributions from mutation, birth, and death are included exactly, which is especially important for new species that initially contain a population size of one. We call this algorithm BNB ("Binomial-Negative Binomial"), since as the name indicates, a population update requires sampling binomial and negative binomial pseudorandom variables with specific weights. This can be done efficiently using techniques similar to those used in the next reaction method (Gibson and Bruck, 2000).

If the mutations are rare compared with other (birth and death) events, this algorithm offers a significant speed advantage with respect to the SSA. Indeed, in most organisms, the mutation rate is much smaller than the birth and death rates, e.g. the probability of mutation per division for the genome in bacteria is $\mu_g \sim 10^{-3}$ (Drake *et al.*, 1998). Thus, only a small (compared to the population size) number of new mutants appear in each generation. Even in viruses that generally are characterized by a high mutation rate $\mu_g \sim 1$, most mutations are neutral and thus do not strongly influence the population dynamics.

In the following, we begin with a general approach to the stochastic simulation of a system of reactions that are arbitrarily divided into "fast" and "slow" reactions. We then specialize to the evolutionary model in which the mutation rate is assumed to be much smaller than the birth and death rates. We present examples that illustrate the accuracy and power of the proposed algorithm for models describing evolution of a population regulated by serial dilution. Then we discuss a modification of the algorithm that allows for its use in more complex situations when the exact algorithm is not applicable. Finally, we illustrate the approximate method by a simple example of co-evolving species competing for a common nutrient source.

2 ALGORITHM

The BNB algorithm is a stochastic updating rule for the state of an evolving set of species, which are defined by their internal state ("genotype") that in turn determines the birth, death, and mutation rates for each species. This algorithm is exact when the birth, death, and mutation rates (not the propensities!) remain constant between consecutive mutations. A single iteration of the BNB algorithm updates the state of the system to the time just after the next mutation has occurred. By applying this updating rule multiple times, the dynamics of the evolving system can be sampled by "jumping" from one mutation to the next. In case when the rates are changing slowly between mutations, an approximate variant of the BNB can be applied (see below).

The core of the BNB algorithm is based on an exact solution for a stochastic model of dividing, dying, and mutating discrete populations of cells. A single iteration of the BNB algorithm uses this solution to rapidly perform the following steps: (1) determine from which species and at what time a new mutant cell is generated, (2) update the populations of all species to the time just prior to this mutation, (3) generate a new mutant cell that either establishes a new species in the simulation or is added to a species already contained in the simulation, and (4) update the time of the simulation to the time of this mutation.

This section contains the derivation and the detailed description of the BNB algorithm.

Stochastic simulation of a two-scale stochastic process

We consider the general case of a continuous time and discrete state stochastic system that is subject to a set of reactions among which some are "fast" and some are "slow." We designate them as fast and slow operationally, for a given state of the system (e.g., abundances of each species) at a given time. Typically, the mean time interval between two consecutive fast reactions will be much smaller than the mean time interval between two consecutive slow reactions. Our goal is to jump directly from one slow reaction event to the next and exactly sample the state of the system at the time of slow reaction.

Let us lump all slow reactions into one that we call "mutation" and consider the dynamics of the system between two consecutive mutations. For simplicity, we assume that the propensities for each possible mutation are proportional to each other for a given system state, such that we can select the type of mutation independently of when a mutation occurs. If the probability of mutations were zero, the probability $p_i(t)$ for being at state i at time t satisfies the master equation that only includes fast reactions

$$\frac{dp_i}{dt} = \sum_j R_{ij}\, p_j \ , \ \text{ with } \ R_{ii} = -\sum_{j\neq i} R_{ij} \text{ and other } R_{ij} \geq 0 \ . \tag{1}$$

Now, suppose that mutations occur with rate μ_i at state i. We can introduce the probability $P_i(t)$ that the system is at state i at time t **and** a mutation has not yet occurred. It is easy to see that $P_i(t)$ satisfies the "leaky" master equation

$$\frac{dP_i}{dt} = \sum_j R_{ij}\, P_j - \mu_i P_i \ . \tag{2}$$

The probability $Y_i(t)$ that at least one mutation has occurred while the system was at state i before time t satisfies the following equation

$$\frac{dY_i}{dt} = \mu_i\, P_i \ . \tag{3}$$

Note that $Y_i(t) = 0$ at initial time $t = 0$. Define the probability $P(t) \equiv \sum_i P_i(t)$ for no mutation to have occurred by time t and $Y(t) \equiv \sum_i Y_i(t)$ for some mutation to have occurred at least once at any state by the time t. By construction, $Y(t) + P(t) = 1$, and therefore

$$\frac{dP}{dt} \quad = \quad -\frac{dY}{dt} = -\sum_i \mu_i\, P_i \ . \tag{4}$$

Thus, $P(t)$ is strictly non-increasing in time, as expected. Knowledge of $P(t)$ allows us to sample time to the next mutation t_m. We also need to know which state of the system is mutated. It is easy to show that the probability $\rho_i(t)$ that the system is at state i at the time of a mutation is

$$\rho_i(t_m) = \frac{\mu_i\, P_i(t_m)}{\sum_i \mu_i\, P_i(t_m)} \ . \tag{5}$$

Thus, assuming we can solve for $P_i(t)$, we can formulate the following algorithm for updating the stochastic system at mutation times:

Algorithm 1

1. Define the initial state of the system i_0, i.e. define $P_i(0) = \delta_{ii_0}$ (where δ_{ij} is the Kronecker symbol).
2. Solve for $P_i(t)$, which provides functions $P(t)$ and $\rho_i(t)$ (Eqs. 4–5).
3. Sample the next mutation time according to the cumulative probability $P(t)$. This can be done via the inversion method, such that the next time $t_m = P^{-1}(r)$, where r is a uniform random variable between 0 and 1.
4. Add t_m to the current time.
5. Sample the distribution $\rho_i(t_m)$ to generate the new state i_m just before the mutation (slow reaction).
6. Choose the specific mutation according to their relative propensities and update the state of the system *after* the state update in Step 5.
7. Return to Step 1 until finished.

Of course, to complete this algorithm, we should be able to solve for or otherwise compute the dynamics of the probabilities P_i according to Eq. 2. While this may be difficult in general to do analytically, it may still be much simpler that solving the full system. In particular, as we discuss in the following Section, the problem can be solved exactly when the fast reactions include only birth and death while the slow reactions include only mutations.

Generating function solution for a single-species birth/death/mutation model

There exists a vast literature on the analysis of statistical properties of the so-called linear birth-death processes. The analytical treatments usually involve solving the corresponding master equation via the generating function method (see Bartlett (1955); Cox and Miller (1965)). Exact solutions have been found for several models including pure birth-death systems as well as systems with immigration and emigration (see, e.g., Karlin and Mcgregor (1958); Ismail *et al.* (1988); Novozhilov *et al.* (2006); Crawford and Suchard (2011)). Here we will follow the same general approach, but since we are interested in the statistics of mutating species, we will add the mutation "reaction" in the model which manifests itself through leakage of probability. We begin with the case of a single class of species. The number of individuals n can fluctuate due to statistically independent birth, death and mutation reactions. Birth has propensity gn, death has propensity γn, and mutation has propensity μn. As before, we are only interested in the interval of time between two subsequent mutations, so the resultant state of the mutated individual is irrelevant. Thus the mutation is simply defined as the creation and subsequent departure of a single individual from the class.

Define $P_n(t)$ to be the probability that the system is at state n at time t and that a mutation has not yet occurred. The generating function $G(s,t) = \sum_{n=0}^{\infty} P_n(t)\, e^{sn}$ can be computed for an initial population n_0 at time $t = 0$ by (see SI for details)

$$G(s,t) \quad = \quad [(p_M(t) - p_E(t))\, e^s\, G_1(s,t) + p_E(t)]^{n_0} \qquad (6)$$

with

$$G_1(s,t) \quad = \quad \frac{1 - p_B(t)}{1 - p_B(t)e^s}, \qquad (7)$$

$$p_M(t) \quad \equiv \quad \frac{RC(t) + 2\gamma S(t) - WS(t)}{RC(t) - 2gS(t) + WS(t)}\,, \qquad (8)$$

$$p_E(t) \quad \equiv \quad \frac{\gamma\,(1 - p_M(t))}{W - \gamma - g\,p_M(t)}\,, \qquad (9)$$

$$p_B(t) \quad \equiv \quad \frac{g p_E(t)}{\gamma}\,, \qquad (10)$$

$R \equiv \sqrt{(g-\gamma)^2 + (2g + 2\gamma + \mu)\mu}$ and $W = g + \gamma + \mu$. Using a uniform random number r distributed between 0 and 1, the next mutation time is then

$$t_m \quad = \quad \frac{1}{R} \ln\left[\frac{r^{1/n_0}\,(R - W + 2g) - W - R + 2\gamma}{r^{1/n_0}\,(-R - W + 2g) - W + R + 2\gamma}\right] \qquad (11)$$

which exists for

$$\left(\frac{R - W + 2\gamma}{R + W - 2g}\right)^{n_0} < r \leq 1\,. \qquad (12)$$

When Eq. 12 is not satisfied, this indicates that the population will go extinct before a mutation occurs if the population is unperturbed for infinite time. The time to extinction, t_x, can then be sampled by

$$t_x = P_0^{-1}(r) = \frac{1}{R} \ln\left[\frac{W - R - 2\gamma r^{-1/n_0}}{W + R - 2\gamma r^{-1/n_0}}\right]\,. \qquad (13)$$

which depends on inversion of the extinct state probability $P_0(t)$.

Binomial – Negative Binomial expansion

After computing the time to the next mutation, we need to generate a sample number of individuals at the time of mutation. The number of individuals conditional on no mutation at time t is distributed according to the generating function $G(s,t)$ given by Eq. 6. Here we show that this seemingly complicated distribution can be exactly sampled by drawing two random numbers - one binomial, and one negative binomial. Many popular software packages, e.g. (Press *et al.*, 2007), contain fast algorithms for generating these random numbers (note that negative binomials can be generated by Poisson random variates with a Gamma-distributed parameter).

Equation 6 can be recast via a binomial expansion

$$G(s,t) = p_M(t)^{n_0} \sum_{m=0}^{n_0} \frac{n_0!}{m!\,(n_0 - m)!} G_1(s,t)^m e^{ms} \cdot \left(1 - \frac{p_E(t)}{p_M(t)}\right)^m \left(\frac{p_E(t)}{p_M(t)}\right)^{n_0 - m}\,. \qquad (14)$$

Since an integer power of a geometric generating function corresponds to a negative binomial generating function, Eq. 14 can be interpreted as a generating function of a process in which the system either has mutated by time t with probability $1 - p_M(t)^{n_0}$, or if the system hasn't yet mutated, then it is in a state $\tilde{n}$ whose distribution is a binomial superposition of n_0 negative binomial distributions. While Eq. 14 does not directly provide the probability to be in a particular state at the time of a mutation, it provides the probability $P_n(t)$ at an arbitrary time t conditional on no mutation. We can then generate a sample of the population $\tilde{n}$ conditional on no mutation at time t by the following procedure.

Algorithm 2

1. Generate a binomial random number $\tilde{m}$, with success probability $1 - (p_E(t)/p_M(t))$ and n_0 terms.
2. If $\tilde{m} = 0$, then the system at time t is in the extinct state $\tilde{n} = 0$.
3. Otherwise, generate the new state variable $\tilde{n}$: $\tilde{n} = \tilde{m} + \tilde{NB}(\tilde{m}, p_B(t))$, where $\tilde{NB}(\tilde{m}, p_B(t))$ is a negative binomial number of order $\tilde{m}$ and probability of success $p_B(t)$.

We are also interested in the probability $\rho_n(t)$ for a system to be in the state n *at the mutation time*. It is easy to see that $\rho_n(t) \propto \mu_n P_n(t) \propto n P_n(t)$ (see Eq. 5). To compute these probabilities, we introduce the corresponding generating function $G_\rho(s,t) = \sum_{n=0}^{\infty} \rho_n(t)\, e^{sn}$. After straightforward algebra, we obtain from Eq. 6

$$G_\rho(s,t) = \left(\frac{(p_M(t) - p_E(t))\, e^s\, G_1(s,t) + p_E(t)}{p_M(t)}\right)^{n_0-1} \cdot e^s\, G_1(s,t)^2\ . \quad (15)$$

which has the binomial expansion

$$G_\rho(s,t) = \sum_{m=0}^{n_0-1} \frac{n_0!}{m!\,(n_0-m)!}\, G_1(s,t)^{m+2} \left(1 - \frac{p_E(t)}{p_M(t)}\right)^m \cdot e^{(m+1)s} \left(\frac{p_E(t)}{p_M(t)}\right)^{n_0-1-m}\ . \quad (16)$$

Equation 16 has the same form as Eq. 14, and thus, ρ_n can be also sampled. Specifically, the algorithm for computing the state of the system just before the next mutation (at time t_m) for the single species reads as follows.

Algorithm 3

1. Generate a binomial random number $\tilde{m}$, with success probability $1 - (p_E(t_m)/p_M(t_m))$ and $n_0 - 1$ terms.
2. Generate the updated state $\tilde{n}$ at the mutation time: $\tilde{n} = \tilde{m} + 1 + \tilde{NB}(\tilde{m} + 2, p_B(t_m))$, where $\tilde{NB}(\tilde{m}, p_B(t))$ is a negative binomial number of order $\tilde{m}$ and probability of success $p_B(t)$.

Note that the system will never be in the extinct state, which reflects that an extinct population cannot mutate.

Simulating multiple co-evolving species: first mutation method

In this section we return to the original problem of an evolving population of multiple species. We enumerate species by index i, with $n_i(t)$ individuals in each species. We are interested in sampling the set $\{n_i(t_m)\}$ at mutation times t_m. We assume that the system parameters (birth, death, and mutation rates) do not change between mutations unless the algorithm is ended early between two mutations. At the time of mutation, one individual is created from mutating class i_m and, depending on the type of mutation, is either added to one of the other existing classes (if such a class already exists) or becomes the founding member of a new class.

The algorithm for generating a sample stochastic evolution trajectory, which we call First Mutation BNB, is as follows.

Algorithm 4

1. Initialize the system with N classes of species at time $t = 0$. Specify populations of all classes $n_i, i = 1, ..., N$. Each class has its own set of birth, death, and mutation rates g_i, γ_i, μ_i.
2. For each class, generate N random numbers r_i uniformly distributed between 0 and 1. For each $i = 1, ..., N$, generate a time t_i to the next mutation by Eq. 11. When Eq. 12 is not satisfied, set $t_i = \infty$.
3. Find the minimum mutation time $t_m = \min(t_i)$ and the corresponding class i_m. Update the time $t \to t + t_m$.
4. Update the population for the mutated class i_m using two random numbers (one binomial and another negative binomial) according to the Algorithm 3.
5. Update the populations of all other classes according to Algorithm 2.
6. Select the specific mutation that occurs. If the mutation generates a member of a nonexistent class, create a new class $N + 1$ with $n_{N+1} = 1$ and its own set of parameters $g_{N+1}, \gamma_{N+1}, \mu_{N+1}$. Otherwise, add 1 to the corresponding existing class.
7. One or several of the non-mutated classes may have zero population and are thus extinct. Remove extinct classes from the list and reduce the number N of classes accordingly.
8. Return to Step 2 until the algorithm has completed.

To end the algorithm at a specific time rather than at a mutation event, all populations can be updated according to Algorithm 2 with the time duration $t^* - t$, where t is the current time, and t^* is the prescribed end time. This update would be done just after Step 2 when $t^* < t + \min(t_i)$ first occurs. Ending at a specific time is useful for a number of purposes, such as if the population is reported or modified at fixed time intervals, or if rates are adjusted at fixed time intervals.

The Algorithm 4 is analogous to the First Reaction Method used for stochastic simulation of reaction networks (Gillespie, 1976), in that the simulation of a system with N classes of co-evolving species generates $3N$ random numbers in order to step to the next mutation. This algorithm can thus become inefficient as the number of classes becomes large. To remedy this shortcoming, an optimized and only slightly more complex version of this algorithm is presented in the next section.

Simulating multiple co-evolving species: next mutation method

In fact, the number of random variables generated for each mutation in Algorithm 4 is excessive. Different species evolve independently between mutations, and even at the mutation time, only two classes are coupled, due to the mutating population generating and then contributing a single member to another species class. If this mutational coupling did not exist, the dynamics of species would be statistically independent at all times, and we could simulate all species independently using only 3 random numbers per mutation event.

This line of reasoning leads to a similar but optimized algorithm (see SI for the algorithm and further justification), where the populations and next mutation times of species are resampled only for the two species that are coupled via a mutation event, while population sizes and next mutation times of all other classes are *not* re-sampled. Validity of the algorithm hinges on the statistical independence of species that are uncoupled by a mutation. The method is analogous to the Next Reaction Method (Gibson and Bruck, 2000), so we label the algorithm Next Mutation BNB.

The optimized scheme reduces the typical number of new random variables required per mutation to only 6 after the first iteration, independently of the total number of classes N. Only initialization and finalization of the algorithm have a computational cost of order N, so efficiency of the algorithm primarily depends on how frequently the algorithm is restarted, as is the case whenever the whole population is sampled for observation.

The Next Mutation BNB algorithm is always as fast or faster than the First Mutation BNB. We thus use Next Mutation BNB (or just BNB) exclusively for the simulation examples of this paper.

Approximate simulation method using BNB

One major benefit of the BNB algorithm is that binomials and negative binomials rapidly generate an update for the evolving system with linear propensities for birth, death, and mutation in a non-interacting population. While this situation is typically the case for cells kept in log-phase growth, the cases when species are interacting or when propensities deviate from a linear law are also of interest. Because of this, we outline how the BNB algorithm can be adapted to approximately, but accurately, simulate more complicated systems.

The basis of the BNB algorithm is the generating function solution Eq. 6, and it is straightforward to show from the short time form of this generating function that the BNB algorithm applied for sufficiently short time increments, during which birth, death, and mutation rates are considered constant, can simulate systems with population-dependent rates. Between BNB updates, all of these rates can be updated in a state-dependent manner. This approach is similar to the τ-leap approximation to stochastic systems, which is often used to accelerate simulations of chemical reaction networks (Gillespie, 2001). The basis of τ-leap is that the propensities for reactions can be considered approximately constant during some time interval, such that the update scheme for τ-leap assumes each reaction occurs a Poisson-distributed number of times. Simulation error magnitude in τ-leap is closely associated with how well propensities are kept constant during a given time interval, and based on this connection, a few prescriptions for the step size have been suggested (Gillespie and Petzold, 2003; Cao *et al.*, 2006, 2007). In contrast, BNB as an approximate updating scheme assumes that the propensities are approximately linear with respect to population, i.e. having constant rates. Deviation from the linear law is the primary factor influencing simulation error in BNB updating.

An important aspect of an approximate BNB updating method is that large and small species populations are treated uniformly, such that the same updating scheme applies to both situations with equal speed and relative accuracy. This may be contrasted to τ-leap methods, which due to large relative fluctuations of the propensity for small populations are no longer valid except for very short time steps. Zhu *et al.* (2011) introduced a hybrid τ-leap method which simulates species lower than a given population (the "cutoff") using direct Gillespie algorithm. The tradeoff for the increased accuracy is a much-increased workload, since Gillespie algorithm simulates each reaction event individually. New species, which start as single cells, or species that naturally exist in low abundances are especially susceptible to an increase in workload for finite cutoff.

3 RESULTS

Exact simulations

In this section we will apply the BNB algorithm to examples that can be exactly simulated using BNB. These examples deal with modeling the evolution of heterogenous cell populations in a hypothetical bioreactor designed to maintain exponentially growing cultures. We illustrate several phenomena that have been explored previously in analogous situations, e.g. for populations of fixed size, though we pursue these phenomena in the regime where large fluctuations in total population size (10-fold in most of our simulations) are routine.

The following models assume that cells are kept sufficiently dilute in culture such that limiting nutrients and other cell-cell interactions are not a factor. These cells thus grow and divide freely. The bioreactor prevents cell cultures from growing too dense by measuring the population size periodically (after every time duration Δt) and diluting the culture by binomial sampling to the mean population size n_{min} once the population has exceeded the population size n_{max}. In the simulations, we advance time directly from one mutation to the next or until the system has evolved longer than the maximal time duration Δt, at which point cells may be diluted if the population has exceeded n_{max}. It is also straightforward to simulate a bioreactor that continuously dilutes cultures to stem population growth, where the rate of media turnover and, correspondingly, cell "death" is controlled, but we do not consider such an case here. An analysis in the SI demonstrates that performance of BNB for these situations can far exceed that for direct Gillespie and τ-leap methods.

Abrupt dilution events can greatly enhance the effect of stochasticity, since there is a corresponding reduction in genetic diversity associated with each subsampling of the population. The smaller population after a dilution event will be heavily influenced by the particular individuals retained, leading to a form of the founder effect (Templeton, 1980). Even in light of this fact, we show that many phenomena found for fixed population sizes, e.g. wave behavior for population fitness, also occur using a dilution protocol that might occur experimentally.

Linear fitness model Suppose that species are characterized by a positive integer index m that is a measure of fitness. Birth rate g_m is a linear function of m, $g_m = 1 + \epsilon\,(m-1)$. Death rate γ_m is constant across species. Mutation rate is proportional to growth rate (faster growing species also mutate faster), $\mu_m = \eta g_m$. During a mutation of species with index m, a new member of species with index $m-1$ or $m+1$ is created, as chosen uniformly at random. If

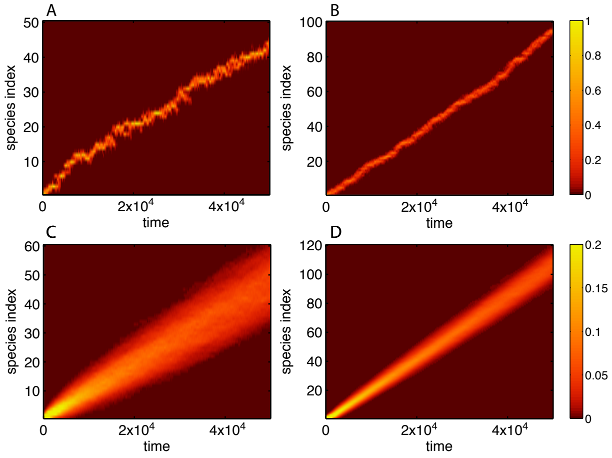

Fig. 1. Simulations of the linear fitness model with $\Delta t = 0.1$, $\epsilon = 10^{-3}$, $\eta = 10^{-3}$, $n_{min} = n_{max}/10$. The instantaneous distributions of the populations over the species index normalized by n_{max} as a function of time are shown for $n_{max} = 10^4$ (A) and $n_{max} = 10^5$ (B). Wave-like behavior is evident in both cases, though the smaller population leads to a noisier and slower wave. Panels C and d show the corresponding probabilities averaged over 800 realizations. The wave velocity, by a least squares linear fit to the ensemble mean fitness, is 0.93×10^{-3} and 2.1×10^{-3} indices per unit time for (C) and (D), respectively.

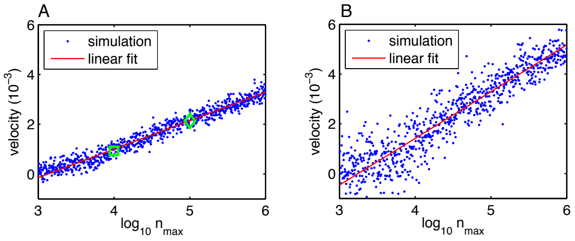

Fig. 2. (A) The wave velocity (indices per unit time) of the linear fitness system has a slow (logarithmic) dependence on the population size set by n_{max}, in agreement with theoretical results (Tsimring *et al.*, 1996; Kessler *et al.*, 1997; Rouzine *et al.*, 2003; Desai *et al.*, 2007; Brunet *et al.*, 2008; Hallatschek, 2011) (parameters are the same as in Fig. 1). Blue dots represent individual velocity measurements based on least squares fitting of a line to the last half of the mean index trajectory. Red line shows the least squares fit of the velocity as a linear function of $\ln n_{\max}$ over the range $n_{\max} > 10^4$. The velocities from Fig. 1C and Fig. 1D are plotted as green squares and diamonds, respectively. (B) Same as (A) for $\epsilon = 10^{-4}$ and $\eta = 10^{-2}$. The weaker fitness gradient leads to a noisier distribution of velocities.

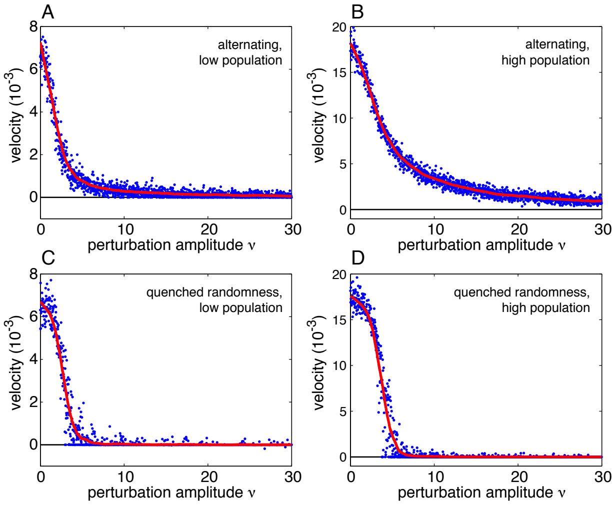

Fig. 3. Ruggedness of the fitness landscape impacts speed of evolution in a linear fitness model. Shown are apparent wave velocities (blue dots) derived by least-square fitting of the mean index $\langle m \rangle$ across species as a function of time. The model with deterministic alternating fitness and $n_{max} = 10^4$ (A) or $n_{max} = 10^5$ (B) leads to a smooth decay of wave velocity with respect to the perturbation amplitude ν. In contrast, a model with quenched disorder in fitness and $n_{max} = 10^4$ (C) or $n_{max} = 10^5$ (D) exhibits an abrupt decrease in wave velocity suggesting a phase transition. In all cases, $n_{min} = n_{max}/10$, $\eta = 10^{-3}$, $\epsilon = 10^{-2}$, $\gamma_m = 0.1$, and $\Delta t = 0.02$. The red curve indicates trend lines generated by a Savitzky-Golay filter.

a species with index $m = 1$ mutates, a new member of the species with index 2 is always created.

It has been demonstrated for $\epsilon > 0$ in the case of a constant total population that evolution on a linear fitness landscape leads to traveling population waves (Tsimring *et al.*, 1996; Kessler *et al.*, 1997; Rouzine *et al.*, 2003; Desai *et al.*, 2007; Brunet *et al.*, 2008; Hallatschek, 2011), such that the mean fitness of the population linearly grows in time. However, the finite-size stochastic system can only be treated heuristically (Tsimring *et al.*, 1996; Kessler *et al.*, 1997), asymptotically (Rouzine *et al.*, 2003; Desai *et al.*, 2007; Brunet *et al.*, 2008), or under certain specific modeling assumptions (Hallatschek, 2011). Thus, exact numerical simulations of large evolving populations in linear fitness landscapes are useful for testing the existing theories. Simulations indeed produce wave-like behavior (see Fig. 1). The wave velocity scales linearly with the logarithm of the population size, as predicted (see Fig. 2).

We used similar simulations to study the effects of quenched fitness fluctuations on the propagation of traveling evolution waves. This problem is qualitatively analogous to the models of transport in systems with quenched disorder that are known to exhibit phase transitions (Bouchaud *et al.*, 1990; Monthus and Bouchaud, 1996), and we expect similar behavior for evolution in a linear model with quenched disorder in the growth rate law. We assumed that the fitness as a function of the species index m has a fluctuating piece in addition to the linear dependence. Specifically, we consider growth rates that vary as $g_m^{(q)} = 1 + \epsilon\,(m - 1 + \nu \tilde{R}_m)$, where $\nu \geq 0$ provides the scaling of noise, and $\tilde{R}_m \in [-0.5, 0.5]$ are independent uniform random numbers. In the case when $\nu < 1$, an increase in m always leads to an increase in growth rate, and wave propagation should proceed but with moderately reduced velocity. The case with $\nu > 1$ is qualitatively different, since an increase in m need not imply an increase in fitness. In this regime, it is possible to form rare but wide barriers due to fluctuations in the fitness, and these barriers when they exist can trap the system for an exponentially large time. This case can be contrasted against a

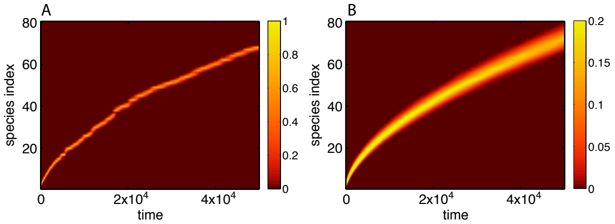

Fig. 4. Wave behavior for the evolution in a model with competition, simulated using BNB as an approximate algorithm with time step $\tau = 1$. (A) A single realization of the species distribution as a function of time for initial population 100. (B) The mean population distribution for an ensemble of 800 simulations.

potential with similar but deterministic variation $g_m^{(a)} = 1 + \epsilon\,(m - 1 + \nu((m \bmod 2) - 0.5))$, which for $\nu > 1$ has fitness barriers only a single species wide. Figure 3 shows that quenched disorder exhibits substantially different behavior than the case when fitness contains regular variation. The system with quenched disorder in particular exhibits a sharp decrease in wave velocity as disorder is increased to $\nu > 1$, akin to a phase transition.

NK model simulations Due to the general way the BNB algorithm treats mutations, it can be applied to more complicated evolutionary models. We used a variant of the NK model (Kauffman and Levin, 1987) to simulate evolution on fitness landscapes with various degrees of ruggedness. Despite large fluctuations in population, we could reproduce classical results for NK models, including state-dependent wave speed for smooth fitness landscapes, and punctuated evolution for rugged landscapes. Results and analysis of this model are found in the SI.

BNB as an approximate algorithm: evolution in nutrient-limited environments

BNB can also be applied as an approximate algorithm for systems with state-dependent growth rates. Propensities may deviate from the linear law assumed in the BNB algorithm, but the BNB algorithm may still approximate a system with non-constant birth, death, and mutation rates by evolving the system with a BNB step restricted to a short duration τ. Rates are then updated using the new populations before integrating the system with another BNB step, and so on. Validity of this process depends on self-consistency of the assumptions in the BNB algorithm, especially that propensities for reactions are independent of other species and proportional to population. See SI for details.

We checked performance of this approximate algorithm for a system in which several species compete for a common nutrient that is supplied at a constant rate. Different species can consume this nutrient with different effectiveness, which provides selective pressure. Specifically, we suppose a linear fitness model for species, $g_m = a_m \left(1 + \sum_\ell a_\ell n_\ell / K_0\right)^{-1}, \gamma_m = 0.1, \mu_m = \eta\, g_m, a_m = 1 + \epsilon(m-1)$, with species index m, $\eta = 10^{-3}$, and a scaling factor $\epsilon = 1$. In contrast to the other simulations in this text, birth rates are coupled in such a way that the total population in the system autonomously relaxes on average to a fixed value $\bar{n} \approx 10\,K_0$ without the need of dilution events triggered by the population.

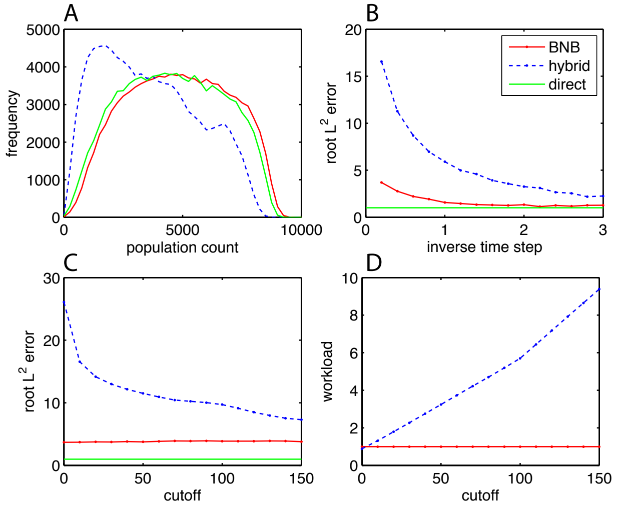

Fig. 5. Simulation accuracy for the model with competition. Using BNB (red), hybrid τ-leap (dashed blue), or direct SSA(light green), the model with $K_0 = 1000$ and initial population $=$ 100) was simulated over 10^5 realizations. As a measure of error, statistics of the population of the first mutant (index=2) were examined at $t = 50$. (A) The histogram (bin width= 250) of this population for simulations using step size $\tau = 5$. BNB matches direct simulation closely, while hybrid τ-leap with cutoff 10 suffers from major inaccuracies. (B) L^2 error between the histogram of direct SSA simulation and that of either BNB or the hybrid τ-leap normalized by the minimal expected statistical deviation, see SI for details. (C) same as (B), but as a function of the cutoff value for the hybrid τ-leap algorithm with $\tau = 5$. (D) Mean workload of the hybrid τ-leap and the approximate BNB algorithms, normalized by the workload for the BNB algorithm, as a function of the cutoff value.

The evolution of the system is linked to the ratio of growth rates $g_{m1}/g_{m2} = a_{m1}/a_{m2}$, which indicates that species with a higher index m tend to grow faster than those with lower index. Due to this effect, the system exhibits wave-like behavior (see Fig. 4).

The recurrent creation and subsequent growth of new species in the competition model suggests that BNB could maintain better accuracy than τ-leaping schemes, since BNB faithfully simulates arbitrarily small populations and also exponential growth. We tested this for short-time simulations, and we found that in this context that BNB can provide consistently increased accuracy when compared to a hybrid τ-leap algorithm (see Fig. 5).

4 DISCUSSION

In this paper, we have proposed an algorithm, which can be used to sample *exactly* co-evolving species that do not interact between mutations, and faithfully approximate the evolution of weakly-interacting species. BNB algorithm not only accounts for the stochastic fluctuations that arise due to the random nature of genetic mutations, but it also accounts for the small-number fluctuations due to the growth of new species that are spawned as single cells. Each iteration of the BNB algorithm generates the time of the next mutation and the abundances of all species just after the mutation.

This algorithm is exact when the birth, death, and mutation rates do not change between consecutive mutations. Although similar in spirit to approximate leaping schemes developed for modeling stiff stochastic chemical kinetics (Gillespie, 2001; Rathinam *et al.*, 2003; Cao *et al.*, 2005; Rathinam and El Samad, 2007; Jahnke and Altintan, 2010; Zhu *et al.*, 2011), it differs significantly by providing an exact sampling at (irregular) intervals corresponding to mutational events. The method yields a substantial speed advantage over a straightforward stochastic simulation algorithm when the mutations are rare compared with birth and death events. The method is accessible, since the central part in implementing BNB is constructing fast methods that generate binomial and negative binomial pseudorandom numbers, both of which are available in standard code libraries (Press *et al.*, 2007). More generally, the BNB algorithm is applicable to the simulations of systems in which underlying reactions are all first order and their rates remain unchanged between coarse-grained simulation steps.

Using the exact BNB algorithm, we simulated several evolution models for a hypothetical bioreactor that performs abrupt dilutions of cell culture when the total cell population exceeds a prescribed value. An analogous experimental bioreactor would periodically reduce the total number of cells, replenish nutrients, and remove wastes in order to maintain log-phase growth of bacterial populations. In contrast to the classical theoretical setting, where the total number of cells is often kept constant, our model bioreactor maintained periodic 10-fold variations in the total number of cells. Despite these wild fluctuations in total population size, most phenomena and population size scaling were preserved. We found the classical scaling laws of adaptation velocity with the population size, as well as the evidence of a phase transition in the case of rugged linear models.

Real cell cultures almost always exhibit some degree of interaction within and among species, and so we showed how the BNB algorithm can also be extended to an approximate algorithm that is competitive with τ-leap and hybrid schemes adapted for evolutionary dynamics simulations (see, for example, (Zhu *et al.*, 2011)). A practical advantage of the approximate BNB algorithm is its uniformity; a BNB step is implemented with identical code for all population sizes. A specific model for species competing for common nutrients was introduced to test BNB, and BNB was found to readily provide good accuracy with minimal workload when compared to analogous τ-leap simulations. We anticipate the advantage of BNB to be maintained in the case where simulations require accurate and fast simulation of exponential growth of species that routinely are found at low population counts, as is the case when new fitter species grow to overtake the population. It should be noted, however, that even though the BNB algorithm can be used to simulate rather general systems, there are systems where BNB performs comparably to or even worse than τ-leap.

The present work presents the foundation for the BNB algorithm, but there exist several immediate directions for future refinement. We anticipate that simple modification of the BNB algorithm should enhance the accuracy for a wide variety of models with interacting species, analogously to a proposed midpoint method for τ-leaping (Anderson *et al.*, 2010). Similarly straightforward modifications may also lead to a BNB formalism that approximates time-dependent birth, death, and mutation rates, as needed for externally driven metabolic networks, e.g. the GAL network (Bennett *et al.*, 2008). A less trivial extension would be to remove the assumption that birth, death, and mutation rates are constant across species. Experimentally, cells within a common species exhibit variability in their cellular state (Elowitz *et al.*, 2002), which could lead to a distribution of growth rates within a single species. Such a modified BNB could then be useful for answering questions concerning how species evolution couples to cellular state.

ACKNOWLEDGEMENT

Funding: This work was supported by the National Institutes of Health, grants P50GM085764 [WM]; RO1GM069811 [JH]; R01GM089976 [LT].

REFERENCES

Anderson, D. F., Ganguly, A., and Kurtz, T. G. (2010). Error analysis of tau-leap simulation methods. *arXiv:0909.4790v2*.

Baake, E. and Gabriel, W. (2000). Biological evolution through mutation, selection, and drift: An introductory review. *Ann. Rev. Comp. Phys*, **7**, 203–264.

Barrick, J. E., Yu, D. S., Yoon, S. H., Jeong, H., Oh, T. K., Schneider, D., Lenski, R. E., and Kim, J. F. (2009). Genome evolution and adaptation in a long-term experiment with escherichia coli. *Nature*, **461**(7268), 1243–1247.

Bartlett, M. (1955). *An Introduction Stochastic Processes with Special Reference to Methods and Applications*. Cambridge University Press, Cambridge, U.K.

Bennett, M. R., Pang, W. L., Ostroff, N. A., Baumgartner, B. L., Nayak, S., Tsimring, L. S., and Hasty, J. (2008). Metabolic gene regulation in a dynamically changing environment. *Nature*, **454**(7208), 1119–1122.

Bouchaud, J. P., Comtet, A., Georges, A., and Ledoussal, P. (1990). Classical diffusion of a particle in a one-dimensional random force-field. *Annals Of Physics*, **201**(2), 285–341.

Brunet, E., Rouzine, I. M., and Wilke, C. O. (2008). The stochastic edge in adaptive evolution. *Genetics*, **179**(1), 603–620.

Cao, Y., Gillespie, D. T., and Petzold, L. R. (2005). The slow-scale stochastic simulation algorithm. *Journal Of Chemical Physics*, **122**(1), 014116.

Cao, Y., Gillespie, D. T., and Petzold, L. R. (2006). Efficient step size selection for the tau-leaping simulation methods. *Journal Of Chemical Physics*, **124**(4), 044109.

Cao, Y., Gillespie, D. T., and Petzold, L. R. (2007). Adaptive explicit-implicit tau-leaping method with automatic tau selection. *Journal Of Chemical Physics*, **126**(22), 224101.

Chou, H.-H., Chiu, H.-C., Delaney, N. F., Segre, D., and Marx, C. J. (2011). Diminishing returns epistasis among beneficial mutations decelerates adaptation. *Science*, **332**(6034), 1190–1192.

Cox, D. and Miller, H. (1965). *The Theory of Stochastic Processes*. Wiley, New York.

Crawford, F. W. and Suchard, M. A. (2011). Transition probabilities for general birth–death processes with applications in ecology, genetics, and evolution. *J. Math. Biol.*, pages DOI: 10.1007/s00285–011–0471–z.

Desai, M., Fisher, D., and Murray, A. (2007). The speed of evolution and maintenance of variation in asexual populations. *Current biology*, **17**(5), 385–394.

Drake, J. W., Charlesworth, B., Charlesworth, D., and Crow, J. F. (1998). Rates of spontaneous mutation. *Genetics*, **148**(4), 1667–1686.

Elowitz, M. B., Levine, A. J., Siggia, E. D., and Swain, P. S. (2002). Stochastic gene expression in a single cell. *Science*, **297**(5584), 1183–1186.

Finkel, S. E. and Kolter, R. (1999). Evolution of microbial diversity during prolonged starvation. *Proceedings of the National Academy of Sciences of the United States of America*, **96**(7), 4023–4027.

Fisher, R. (1930). *The genetical theory of natural selection.* Clarendon Press.

Gibson, M. A. and Bruck, J. (2000). Efficient exact stochastic simulation of chemical systems with many species and many channels. *Journal Of Physical Chemistry A*, **104**(9), 1876–1889.

Gillespie, D. T. (1976). General method for numerically simulating stochastic time evolution of coupled chemical-reactions. *Journal of Computational Physics*, **22**(4), 403–434.

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical-reactions. *Journal Of Physical Chemistry*, **81**(25), 2340–2361.

Gillespie, D. T. (2001). Approximate accelerated stochastic simulation of chemically reacting systems. *Journal Of Chemical Physics*, **115**(4), 1716–1733.

Gillespie, D. T. and Petzold, L. R. (2003). Improved leap-size selection for accelerated stochastic simulation. *Journal Of Chemical Physics*, **119**(16), 8229–8234.

Gillespie, J. (1984). Molecular evolution over the mutational landscape. *Evolution*, pages 1116–1129.

Hallatschek, O. (2011). The noisy edge of traveling waves. *Proceedings of the National Academy of Sciences*, **108**(5), 1783.

Ismail, M. E. H., Letessier, J., and Valent, G. (1988). Linear birth and death models and associated Laguerre and Meixner polynomials. *Journal Of Approximation Theory*, **55**(3), 337–348.

Jahnke, T. and Altintan, D. (2010). Efficient simulation of discrete stochastic reaction systems with a splitting method. *Bit Numerical Mathematics*, **50**(4), 797–822.

Jain, K. and Krug, J. (2007). Deterministic and stochastic regimes of asexual evolution on rugged fitness landscapes. *Genetics*, **175**(3), 1275–1288.

Karlin, S. and Mcgregor, J. (1958). Linear growth, birth and death processes. *Journal Of Mathematics And Mechanics*, **7**(4), 643–662.

Kauffman, S. and Levin, S. (1987). Towards a general-theory of adaptive walks on rugged landscapes. *Journal Of Theoretical Biology*, **128**(1), 11–45.

Kessler, D. A., Levine, H., Ridgway, D., and Tsimring, L. (1997). Evolution on a smooth landscape. *Journal Of Statistical Physics*, **87**(3-4), 519–544.

Lu, T., Volfson, D., Tsimring, L., and Hasty, J. (2004). Cellular growth and division in the Gillespie algorithm. *Systems Biology*, **1**(1), 121–128.

Monthus, C. and Bouchaud, J. P. (1996). Models of traps and glass phenomenology. *Journal Of Physics A-Mathematical And General*, **29**(14), 3847–3869.

Moran, P. (1958). Random processes in genetics. *Math. Proc. of the Cambridge Phil. So.*, **54**(01), 60–71.

Novozhilov, A. S., Karev, G. P., and Koonin, E. V. (2006). Biological applications of the theory of birth-and-death processes. *Briefings In Bioinformatics*, **7**(1), 70–85.

Pena, M. I., Davlieva, M., Bennett, M. R., Olson, J. S., and Shamoo, Y. (2010). Evolutionary fates within a microbial population highlight an essential role for protein folding during natural selection. *Molecular Systems Biology*, **6**, 387.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007). *Numerical Recipes: The Art of Scientific Computing, Third Edition*. Cambridge University Press, New York.

Rathinam, M. and El Samad, H. (2007). Reversible-equivalent-monomolecular tau: A leaping method for "small number and stiff" stochastic chemical systems. *Journal of Computational Physics*, **224**(2), 897–923.

Rathinam, M., Petzold, L. R., Cao, Y., and Gillespie, D. T. (2003). Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method. *Journal Of Chemical Physics*, **119**(24), 12784–12794.

Rouzine, I. M., Wakeley, J., and Coffin, J. M. (2003). The solitary wave of asexual evolution. *Proceedings Of The National Academy Of Sciences Of The United States Of America*, **100**(2), 587–592.

Ruiz-Jarabo, C. M., Miller, E., Gomez-Mariano, G., and Domingo, E. (2003). Synchronous loss of quasispecies memory in parallel viral lineages: A deterministic feature of viral quasispecies. *Journal Of Molecular Biology*, **333**(3), 553–563.

Templeton, A. R. (1980). The theory of speciation via the founder principle. *Genetics*, **94**(4), 1011–1038.

Tsimring, L. S., Levine, H., and Kessler, D. A. (1996). RNA virus evolution via a fitness-space model. *Physical Review Letters*, **76**(23), 4440–4443.

Wright, S. (1931). Evolution in Mendelian populations. *Genetics*, **16**(2), 97.

Zhu, T., Hu, Y., Ma, Z.-M., Zhang, D.-X., Li, T., and Yang, Z. (2011). Efficient simulation under a population genetics model of carcinogenesis. *Bioinformatics*, **27**(6), 837–843.

Supporting Information: Fast stochastic algorithm for simulating evolutionary population dynamics

William H. Mather[1,2,3], Jeff Hasty[1,2,3,4], Lev S. Tsimring[2,3]*
1 Department of Bioengineering, University of California, San Diego, CA, USA
2 BioCircuits Institute, University of California, San Diego, CA, USA
3 San Diego Center for Systems Biology, San Diego, CA, USA
4 Molecular Biology Section, Division of Biological Sciences, University of California, San Diego, CA, USA
∗ Corresponding author. E-mail: ltsimring@ucsd.edu

Contents

A Details for the generating function solution for a single-species birth/death/mutation model

Define $P_n(t)$ to be the probability that the system is at state n at time t and that a mutation has not yet occurred. This probability obeys the master equation

$$\frac{dP_n}{dt} = g[(n-1)P_{n-1} - nP_n] + \gamma[(n+1)P_{n+1} - nP_n] - \mu n P_n \ . \tag{S1}$$

The generating function

$$G(s,t) = \sum_{n=0}^{\infty} P_n(t)\, e^{sn} \tag{S2}$$

satisfies the following first-order equation

$$\frac{\partial G}{\partial t} = \left[(e^s - 1)g + (e^{-s} - 1)\gamma - \mu\right] \frac{\partial G}{\partial s}\ . \tag{S3}$$

By the method of characteristics, the general solution to Eq. S3 is

$$\begin{aligned} G(s,t) &= F(z(s,t))\ , & \text{(S4)} \\ z(s,t) &= t + \frac{2}{R}\,\mathrm{arctanh}\left(\frac{W - 2ge^s}{R}\right)\ , & \text{(S5)} \end{aligned}$$

where $F(\cdot)$ is an arbitrary function, $R \equiv \sqrt{(g-\gamma)^2 + (2g + 2\gamma + \mu)\mu}$ and $W = g + \gamma + \mu$. Since we are interested in generating sample stochastic trajectories, we assume that at time $t = 0$ the number of species n_0 is given, i.e.

$$G(s,0) = e^{sn_0}\ . \tag{S6}$$

The exact solution for this initial condition, as can be checked by direct substitution, is given by

$$G(s,t) = \left[\frac{(WS(t) - RC(t))e^s - 2\gamma S(t)}{2gS(t)e^s - WS(t) - RC(t)}\right]^{n_0}\ , \tag{S7}$$

where $C(t) \equiv \cosh(Rt/2)$ and $S(t) \equiv \sinh(Rt/2)$. Note that due to linearity of the problem all other solutions can be written as a superposition of such solutions. This solution can be simplified into a form that is easier to interpret

$$G(s,t) = \left[(p_M(t) - p_E(t))\, e^s\, G_1(s,t) + p_E(t)\right]^{n_0} \tag{S8}$$

with

$$\begin{aligned} G_1(s,t) &= \frac{1 - p_B(t)}{1 - p_B(t)e^s}, & \text{(S9)} \\ p_M(t) &\equiv \frac{RC(t) + 2\gamma S(t) - WS(t)}{RC(t) - 2gS(t) + WS(t)}\ , & \text{(S10)} \\ p_E(t) &\equiv \frac{\gamma\,(1 - p_M(t))}{W - \gamma - g\, p_M(t)}\ , & \text{(S11)} \\ p_B(t) &\equiv \frac{g p_E(t)}{\gamma}\ . & \text{(S12)} \end{aligned}$$

Here $p_M(t)^{n_0} = G(0,t)$ is the probability that a mutation has not yet occurred, $G_1(s,t)$ is the generating function for a geometric distribution with probability parameter $p_B(t)$, and $p_E(t)^{n_0}$ is the probability of extinction. The factor e^s in front of $G_1(s,t)$ shifts the geometric distribution upward by 1 unit of population.

In order to obtain the time of the next mutation t_m, we need to solve the equation $P(t) = r$, where r is a random number uniformly distributed between 0 and 1, and $P(t)$ is obtained from $G(s,t)$ as

$$P(t) = G(0,t) = p_M(t)^{n_0} = \left(\frac{RC(t) + 2\gamma S(t) - WS(t)}{RC(t) - 2gS(t) + WS(t)}\right)^{n_0}\ . \tag{S13}$$

However, since $P(t)$ does not reach 0 as $t \to \infty$ due to possible extinction, we have to distinguish two cases. The solution to $P(t) = r$,

$$\begin{aligned} t_m &= P^{-1}(r) \\ &= \frac{1}{R} \ln \left[\frac{r^{1/n_0} (R - W + 2g) - W - R + 2\gamma}{r^{1/n_0} (-R - W + 2g) - W + R + 2\gamma} \right] \end{aligned} \tag{S14}$$

exists for

$$\left(\frac{R - W + 2\gamma}{R + W - 2g} \right)^{n_0} < r \leq 1 \ . \tag{S15}$$

If r is less than this bound, then the system goes extinct before mutating, and the time to extinction can be computed by inverting the probability of extinction before time t,

$$P_0(t) = G(-\infty, t) = \left[\frac{2\gamma S(t)}{RC(t) + WS(t)} \right]^{n_0} \ . \tag{S16}$$

Inversion of formula (S16) yields the time of extinction

$$t_x = P_0^{-1}(r) = \frac{1}{R} \ln \left[\frac{W - R - 2\gamma r^{-1/n_0}}{W + R - 2\gamma r^{-1/n_0}} \right] \ . \tag{S17}$$

As stated above, this solution only exists for

$$0 \leq r < P_0(\infty) \tag{S18}$$

where

$$P_0(\infty) = \left(\frac{R - W + 2\gamma}{R + W - 2g} \right)^{n_0} \tag{S19}$$

is the asymptotic extinction probability.

B Speedup for exact simulations: BNB relative to SSA and τ-leap algorithms

Two of the most common methods to simulate chemical reaction networks are exact SSA [1] and approximate τ-leap methods [2]. We find that when BNB holds exactly, it far outcompetes these alternative algorithms. The case when BNB does not hold exactly is treated in the next section.

The direct SSA algorithm [1] can generate exacts realizations of birth, death, and mutation reactions. However, when mutations are rare, direct algorithm spends most of the time implementing birth and death reactions. This can be contrasted to BNB, where each iteration of the algorithm jumps from one mutation to the next. The number of steps ("workload") Υ is defined as the total number of simulated reactions in the system plus the total number of times the population is tested whether it exceeds n_{max} (periodically with period Δt). The workloads Υ_{SSA} and Υ_{BNB} corresponding to the direct SSA and BNB algorithms, respectively, can be approximated with

$$\Upsilon_{\mathrm{SSA}} = \left(\frac{T}{\Delta t} \right) + (g + \gamma + \mu) \langle n \rangle T \ , \tag{S20}$$

$$\Upsilon_{\mathrm{BNB}} = \left(\frac{T}{\Delta t} \right) + \mu \langle n \rangle \ T \ , \tag{S21}$$

with Δt the sampling period, $\langle n \rangle$ a typical population size, T the time of simulation, and the typical rates g, γ and $\mu = \eta g$ for birth, death, and mutation, respectively. Considerable speedup relative to

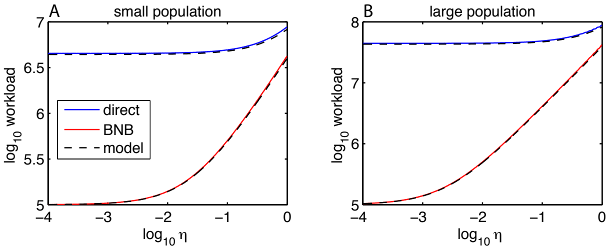

Figure S1. Comparison between the workload of the direct SSA (blue) and the BNB algorithm (red). Dashed lines correspond to the theoretical estimate, Eqs. S20-S21. The corresponding simulations were done for a linear fitness model with $n_{max} = 10^3$ (A) and $n_{max} = 10^4$ (B). Other parameters were $n_{min} = n_{max}/10$, $g = 1$, $\gamma = 0.1$, and the total time of simulation $T = 10^5$. Mutations were counted towards the workload when they occurred, but no new species were created for simplicity. For comparison with the theoretical formulas Eqs. S20-S21 we used the average population $\langle n \rangle \approx 0.391\, n_{max}$, as derived for a deterministic exponential growth between $n_{min} = n_{max}/10$ and n_{max}.

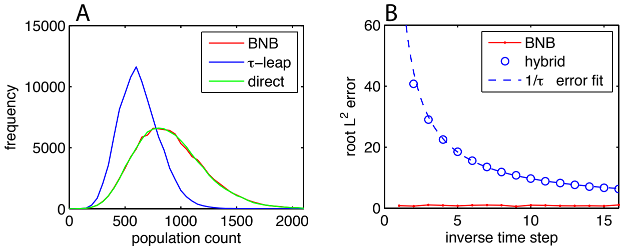

Figure S2. Comparison between the accuracy of τ-leap relative to the exact BNB algorithm. (A) Using either BNB (red), hybrid τ-leap (blue), or direct SSA(green), pure exponential growth ($n_{max} = \infty$) of a single species without mutation was simulated with initial population $= 1$ and parameters $g = 1$, $\gamma = 0.1$. A population histogram with bin width 50 and ensemble size 10^5 is shown for simulations with time step $\tau = 0.5$, cutoff 10, and total time of simulation $T = 5$. (B) We quantified the error for different time steps (inverse of the time step is plotted) by computing the root mean square difference between the histogram of direct SSA simulation and that of either BNB or the hybrid τ-leap. We normalized this error by the minimal expected statistical deviation (see last Section in this SI). The difference between τ-leap error and the minimal expected error is well-approximated by best-fit linear power law, error $\propto \tau$ (blue dashed line).

direct simulation can then be expected for the BNB algorithm when $g + \gamma \gg \mu$. Fig. S1 illustrates typical results, where the BNB algorithm significantly accelerates simulation relative to SSA when the mutation rate is low. Note that the "cost" of each step in BNB is somewhat higher than in SSA since it requires generation of several random numbers as compared to only two uniform random numbers for SSA. However this cost increase is small compared with significant benefits of jumping over birth and death reactions for the case of rare mutations.

The workload (number of iterations) $\Upsilon_{\tau\text{-leap}}$ for the hybrid τ-leap algorithm [3] can far exceed that of the BNB algorithm in the case when birth, death, and mutation rates are constant. Ignoring dilution events, BNB only requires a single iteration to integrate the system between mutations, as assumed in Eq. S21. The hybrid τ-leap algorithm instead requires that time steps are sufficiently small if a given accuracy (with scale $\mathcal{E}$) is to be assured. For a large population size, the error of the τ-leap algorithm tends to be linked to that of a first order Euler method for integration, and the workload then scales as $\Upsilon_{\tau\text{-leap}} \propto 1/\mathcal{E}$. Accuracy of the method is then proportional to the time step. We indeed find that simulation accuracy for typical parameters is largely determined by this linear law (see Fig. S2). The hybrid τ-leap algorithm also assumes that species with population sizes below a certain cutoff are simulated using the direct method, which can increase the workload.

C Next mutation BNB algorithm

C.1 Statement of algorithm

The optimized Next Mutation BNB algorithm as follows.

Algorithm 5

1. Initialize the system with N classes of species at time $t = 0$. Specify populations of all classes $n_i, i = 1, ..., N$. Each class has its own set of birth, death, and mutation rates g_i, γ_i, μ_i. Flag each class to have its next mutation time updated.

2. For flagged classes, generate random numbers r_i uniformly distributed between 0 and 1. For each $i = 1, ..., N$, generate a next mutation time $\tau_i = t_i + t$ (t is the current time, t_i is the duration of time to the next mutation) using Eq. S14. When Eq. S15 is not satisfied, set $\tau_i = \infty$. Store the current time (time of last sampling) for each such flagged class as T_i. Unflag all classes.

3. Find the minimum next mutation time $\tau_m = \min(\tau_i)$ and the corresponding class i_m. Update the time $t \to \tau_m$.

4. Update the population for the mutated class i_m according to the Algorithm 3, using the duration since last update $(t - T_i)$= $(\tau_i - T_i)$ for the variable t_m.

5. Select the specific mutation that occurs.

6. If the mutation generates a member of a nonexistent class, create a new class with index $N+1$ with $n_{N+1} = 1$ and its own set of parameters $g_{N+1}, \gamma_{N+1}, \mu_{N+1}$. Store the index $j = N + 1$. Otherwise for the appropriate existing class (index j), update the population according to Algorithm 2 using the duration since last update $(t - T_j)$. Add 1 to the population of class j.

7. Flag the classes i_m and j from Steps 3 and 6, respectively.

8. Return to Step 2 until the algorithm has completed.

9. Finalize the algorithm by sampling the population of all unflagged classes according to Algorithm 2 with duration $(t - T_i)$ for class index i.

As in Algorithm 4 of the main text, the algorithm can be ended at any specific time t^*. This is done by jumping to Step 9 after Step 2 when $t^* < \min(\tau_i)$ first occurs.

C.2 Comments on the validity of the Next Mutation BNB algorithm

Since the different classes of species are uncoupled between mutations, the Next Mutation BNB algorithm supposes that a species class only needs to be sampled just prior to the point when it is influenced by a mutation. It is relatively straightforward to show that species not involved in this mutation, as either the origin or the destination of a mutant species, do not need to be sampled.

We are interested in a species class (index i) that starts with a population $\tilde{n}_i(T_0) = x_0$ at time T_0 and is known to mutate at a time $\tilde{\tau}_i > T_1$, where $T_1 > T_0$. This is the case in the BNB algorithm when the next mutation time T_1 belonging to some other class is earlier than the next mutation time $\tilde{\tau}_i$ of class index i. It is important to understand how the information $\tilde{\tau}_i > T_1$ influences the probability distribution of the process described by population $\tilde{n}_i(t)$ and next mutation time $\tilde{\tau}_i$. A particular function of interest is the conditional probability ($T_2 > T_1 > T_0$)

$$\Pr(\tilde{n}_i(T_2) = x_2, \tilde{\tau}_i > T_2 \,|\, \tilde{n}_i(T_0) = x_0, \tilde{\tau}_i > T_1). \tag{S22}$$

This function encodes how the information $\tilde{\tau}_i > T_1$ influences statistics at a later time T_2 (see [4] for use of this function in a related context). Straightforward manipulation reveals

$$\begin{aligned}
&\Pr(\tilde{n}_i(T_2) = x_2, \tilde{\tau}_i > T_2 \,|\, \tilde{n}_i(T_0) = x_0, \tilde{\tau}_i > T_1) \\
&\quad = \Pr(\tilde{n}_i(T_2) = x_2, \tilde{\tau}_i > T_2, \tilde{n}_i(T_0) = x_0, \tilde{\tau}_i > T_1) \;/\; \Pr(\tilde{n}_i(T_0) = x_0, \tilde{\tau}_i > T_1), \\
&\quad = \Pr(\tilde{n}_i(T_2) = x_2, \tilde{\tau}_i > T_2 \,|\, \tilde{n}_i(T_0) = x_0) \;\cdot\; \Pr(\tilde{n}_i(T_0) = x_0) \;/\; \Pr(\tilde{n}_i(T_0) = x_0, \tilde{\tau}_i > T_1)
\end{aligned}$$

by definition of a conditional probability. Thus

$$\Pr(\tilde{n}_i(T_2) = x_2, \tilde{\tau}_i > T_2 \,|\, \tilde{n}_i(T_0) = x_0, \tilde{\tau}_i > T_1) = \frac{\Pr(\tilde{n}_i(T_2) = x_2, \tilde{\tau}_i > T_2 \,|\, \tilde{n}_i(T_0) = x_0)}{\Pr(\tilde{\tau}_i > T_1 \,|\, \tilde{n}_i(T_0) = x_0)}. \tag{S23}$$

Equation S23 demonstrates that the conditional probability in Eq. S22 can be derived by two simpler probabilities that do explicitly condition on the information $\tilde{\tau}_i > T_1$.

Nothing has so far been assumed concerning the statistical independence of species i from other quantities, e.g. T_1. We now assume statistical independence, such that the conditional probabilities on the right hand side of Eq. S23 can be expressed using the results of the single species statistical formalism used to derive the BNB algorithm. The resulting probabilities can be used to show that the Next Mutation BNB algorithm generates the correct probabilities between mutations. For example, Eq. S23 in the form

$$\begin{aligned}
&\Pr(\tilde{n}_i(T_2) = x_2, \tilde{\tau}_i > T_2 \,|\, \tilde{n}_i(T_0) = x_0) = \\
&\qquad \Pr(\tilde{n}_i(T_2) = x_2, \tilde{\tau}_i > T_2 \,|\, \tilde{n}_i(T_0) = x_0, \tilde{\tau}_i > T_1) \cdot \Pr(\tilde{\tau}_i > T_1 \,|\, \tilde{n}_i(T_0) = x_0)
\end{aligned} \tag{S24}$$

indicates that sampling the population at a time $T_2 < \tilde{\tau}_i$ without knowing $\tilde{\tau}_i > T_1$ generates the correct probabilities adjusted by the fixed scale $\Pr(\tilde{\tau}_i > T_1 \,|\, \tilde{n}_i(T_0) = x_0)$, which is precisely the chance that we should observe $\tilde{\tau}_i > T_1$ in the BNB algorithm.

The remaining difficulty in demonstrating validity of the Next Mutation BNB algorithm is in ensuring that statistical independence holds among species in the time between mutations. Since the only coupling is at discrete mutation events, it is intuitively clear that all species that are initially independent (in the sense of Eq. S23) remain independent if the species are not involved in the next mutation. We then sample only the mutating species and the species receiving the new mutant cell, such that the next mutation dynamics of the two species again become independent *conditional* on knowing the outcome of the last mutation.

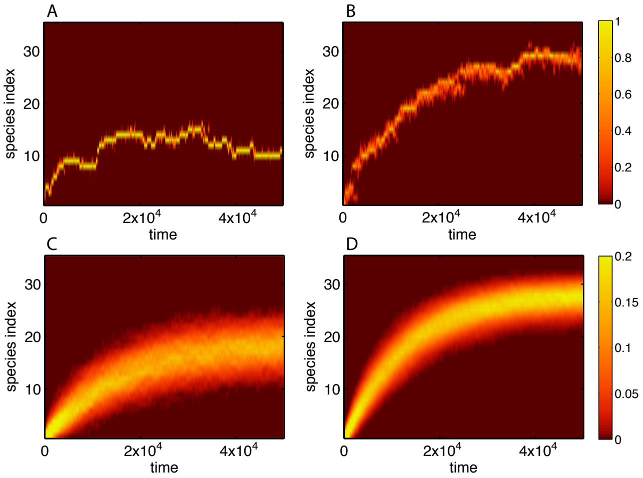

Figure S3. Results for the NK model with a smooth fitness landscape ($N = 32, K = 0$, $\Delta t = 0.1$, $\epsilon = 10^{-3}$, $\eta = 10^{-3}$, $n_{min} = n_{max}/10$). Single trajectories are shown for $n_{max} = 10^3$ (A) and $n_{max} = 10^4$ (B), where species with the same fitness (of the same class) have been identified for purposes of visualization. Species index is equal to one plus the sum of bits. Ensemble mean probabilities from 800 realizations corresponding to (A) and (B) are shown in (C) and (D), respectively.

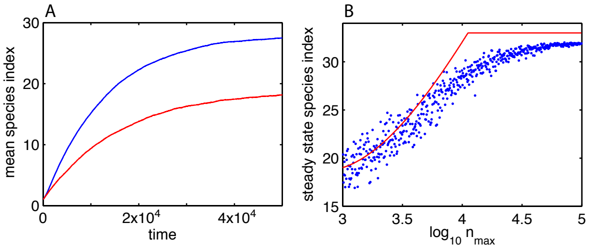

Figure S4. Evolution velocity substantially slows down and stalls as the system approaches the fitness maximum. (A) Plot of the ensemble mean index from Fig. S3C (red) and Fig. S3D (blue). The long-time ensemble mean species index for each population apparently tends towards a value less than that of the fitness optimum, indicating a balance between the negative entropic velocity and the positive fitness gradient velocity. (B) Estimate of the steady state species index for different population sizes. Trajectories of duration 2×10^6 were recorded, and the time-averaged mean species index was reported (blue dots). A simple model (red) was compared by first fitting the velocity data in Fig. 2 of the main text to a smooth cubic polynomial of $\log_{10} n_{max}$ via least squares minimization on the range $n_{max} \leq 10^5$, and then using this estimate for velocity in a heuristic model (see Eq. S29).

D Exact simulations: NK model

A similar investigation as for the linear model in the main text was done for an NK model [5]. In the model, a species m is characterized by its "genome" that is a binary string of length N, with bits $b_{m,j} \in \{0, 1\}, j = 1, ..., N$. The fitness is determined by the state of all N bits as a sum of contributions from all N loci. However, the contribution from each locus in general may depend on the states of K other loci. If $K = 0$, each bit contributes to the fitness independently, which corresponds to the smooth

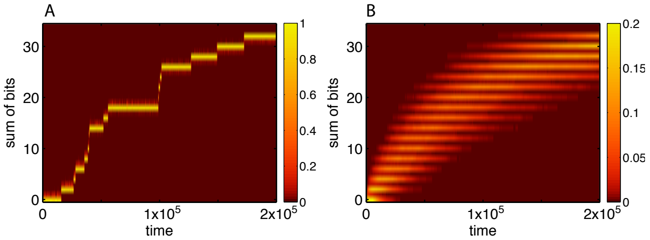

Figure S5. Simulation of a NK model with a rugged landscape $N = 32, K = 1$ (see Eq. S30). (A) Fraction of the population with a given bit sum ($\sum_{j=1}^{N} b_{m,j}$) for a single realization of the model with $A = -1$, $B = 2$. These parameters provide a mildly rugged landscape that requires evolving populations to exhibit suboptimal fitness fluctuations in order to evolve to optimal fitness. It is clear that the sum of bits defining the state tend to be even, while odd sums typically indicate a suboptimal fluctuation. In contrast to smooth models, long residence times for a species tend to be observed. (B) The punctuated nature of single trajectories is less apparent for the bit sum probability distribution for an ensemble of 800 realizations. Other parameters for the model are $\epsilon = 10^{-2}$, $\eta = 10^{-3}$, $n_{min} = n_{max}/10$, $n_{max} = 10^4$, and $\Delta t = 0.1$.

landscape with a unique maximum, whereas $K = N - 1$ corresponds to the extremely rugged landscape, such that every N-bit string corresponds to a different, often chosen at random, fitness value. At each mutation event, one of the bits (chosen uniformly at random) in the mutated species is flipped, i.e. with one mapped to zero or zero mapped to one.

We simulated evolution of a finite cell population in the NK model on a smooth landscape ($K = 0$), where the growth rate only depends on the sum $X_m = \sum_{j=1}^{N} b_{m,j}$ of the N bits. The evolutionary model is defined by the following birth, death, and mutation rates

$$g_m = 1 + \epsilon \sum_{j=1}^{N} b_{m,j} \ , \tag{S25}$$

$$\gamma_m = 0.1 \ , \tag{S26}$$

$$\mu_m = \eta \, g_m \ . \tag{S27}$$

As before, we randomly sample the population once it reaches the pre-defined size n_{max} to reduce it to n_{min}.

Representative results for $N = 32$ in Fig. S3 show waves of evolving species approaching the fittest state (all bits set to one). In contrast to the linear fitness model, the wave velocity is seen to depend on the mean fitness of the system (mean number of ones in the genome) due to an entropic effect (see Fig. S4). This is linked to the fact that a newly mutated species that arises from a progenitor species with a total X_m of 1-bits has a probability $(N - X_m)/N$ to increase to a total $X_m + 1$ of 1-bits and a probability X_m/N to decrease to a total $X_m - 1$ of 1-bits. This bias tends to prevent species from approaching the fitness maximum $X = N$.

In the absence of a fitness gradient ($\epsilon = 0$), it can be shown that this mutational bias leads to a state-dependent velocity $v_\mu(X_m)$ that is the rate of change for the average total of 1-bits $\langle X_m \rangle$ [6]

$$v_\mu(X_m) = \mu \left(1 - \frac{2 \langle X_m \rangle}{N}\right) \ . \tag{S28}$$

This can be compared to the velocity v_0 for populations satisfying $X_m \approx N/2$ in the presence of a fitness gradient, where the mutational bias velocity is small relative to the velocity due to selection. The velocity

v_0 then should be predictable from a corresponding linear fitness model. It may happen then that the velocity due to selection can be counteracted by the velocity due to mutational bias, allowing the sum of bits to relax to an asymptotic mean value $\langle X_m \rangle_\infty$. A simple *ad hoc* model for the steady state condition supposes that v_0 balances v_μ at steady state, leading to the condition (assuming $v_0 > 0$)

$$\langle X_m \rangle_\infty = \min \left[\frac{N}{2} \left(1 + \frac{v_0}{\mu} \right) \ , \ N \right] \ . \tag{S29}$$

We show in Fig. S4B that our results for wave velocity in linear fitness can be used in Eq. S29 to make nontrivial predictions for the apparent steady state value $\langle X_m \rangle_\infty$.

A more rugged landscape can be created when the fitness depends on interactions between bits ($K > 0$). We introduce a mildly rugged $K = 1$ model in which the birth rate depends on the state of bits with index separated by $N/2$

$$g_m = 1 + \epsilon \left(A \sum_{j=1}^{N} b_{m,j} + B \sum_{j=1}^{N} b_{m,j} b_{m,j^*} \right) \ , \tag{S30}$$

where $j^* = [(j - 1 + (N/2)) \bmod N] + 1$. It can be shown for negative A and $B > |A|$ that there may be up to $N/2$ barriers separating the fitness optimum from an initial condition. At each of these barriers, the evolving population must transiently contain one suboptimal species before a fitter species can be created. An evolving population can be stuck in these local fitness optima for long times, significantly slowing the speed of evolution. Results for such a model are presented in Fig. S5.

E Use of BNB as an approximate algorithm

Propensities may deviate from the linear law assumed in the BNB algorithm, but the BNB algorithm may still approximate a system with non-constant birth, death, and mutation rates by evolving the system with a BNB step restricted to a short duration t. Rates are then updated using the new populations before integrating the system with another BNB step, and so on. Validity of this process depends on self-consistency of the assumptions in the BNB algorithm, which we discuss presently.

A major assumption in the BNB algorithm is that propensities for reactions are independent of other species and proportional to population. Suppose for example that the birth rate $g(\vec{n})$ is some function of the set of populations $\vec{n}$. Then, for some starting population $\vec{n}_0$ and some deviation from this population $\vec{\Delta n}$, the propensity $a_{g,i}(\vec{n}) = g_i(\vec{n}) n_i$ for the birth rate of population i has the deviation from the BNB assumption

$$\begin{aligned} \Delta a_{g,i}(\vec{n}_0 + \vec{\Delta n}) &= g_i(\vec{n}_0 + \vec{\Delta n}) \, (n_{0,i} + \Delta n_i) - g(\vec{n}_0) \, (n_{0,i} + \Delta n_i), \\ &= \sum_j \frac{\partial g_i}{\partial n_j} (\vec{n}_0) \, n_{0,i} \, \Delta n_j + O(\Delta n^2). \end{aligned} \tag{S31}$$

We argue that the birth propensity error in Eq. S31 (the argument extends to death and mutation propensities) can be made small if BNB is applied for short durations.

E.1 BNB algorithm as an approximation: negligible mutation

The approximate short-time BNB algorithm can be most readily analyzed in the case that mutation is a negligible effect. The general case with mutations is considered in the next section. The approach to evaluation of the error follows that in Ref. [7].

If population starts in the state $\vec{n}_0$, and if the mutation rate is temporarily considered negligible, then it can be shown that Eq. S31 give the statistics for the deviation of the propensity after some time t

$$\begin{aligned}
\left\langle \Delta a_{g,i} \right\rangle_t &\approx \sum_j \frac{\partial g_i}{\partial n_j}(\vec{n}_0)\, n_{0,i} \left\langle \Delta n_j \right\rangle_t , && \text{(S32)}\\
\left\langle \Delta a_{g,i}^2 \right\rangle_t &\approx \sum_j \left(\frac{\partial g_i}{\partial n_j}(\vec{n}_0) \right)^2 n_{0,i}^2 \left\langle \Delta n_j^2 \right\rangle_t \\
&= \sum_j \left(\frac{\partial g_i}{\partial n_j}(\vec{n}_0) \right)^2 n_{0,i} \left(\frac{g_j + \gamma_j}{g_j - \gamma_j} \right) \left(\left\langle \Delta n_j \right\rangle_t + n_{0,j} \right) \left\langle \Delta n_j \right\rangle_t , && \text{(S33)}
\end{aligned}$$

where

$$\left\langle \Delta n_i \right\rangle_t = n_{0,i}(e^{(g_i(\vec{n}_0) - \gamma_i(\vec{n}_0))t} - 1), \qquad \text{(S34)}$$

and $\langle \cdot \rangle_t$ indicates an average over a BNB step with time step t. Equation S33 depends on $\left\langle \Delta n_i \Delta n_j \right\rangle_t = \delta_{i,j} \left\langle \Delta n_i^2 \right\rangle_t$ for a BNB step, where $\delta_{i,j}$ is the Kronecker delta. Analogous expressions can be straightforwardly written for changes in propensity for the death reactions.

An appropriate step size may be chosen by ensuring that a relative change in the propensity, given by Eqs. S32–S33, is much smaller than the zeroth order mean total propensity, which we define as the sum of birth and death mean propensities

$$\langle a_{0,i}(t) \rangle = n_{0,i} \left(g_i(\vec{n}_0) + \gamma_i(\vec{n}_0) \right) e^{(g_i(\vec{n}_0) - \gamma_i(\vec{n}_0))t}. \qquad \text{(S35)}$$

A step size t may be chosen to ensure that the relative error in the propensity is kept of order ϵ

$$\begin{aligned}
\epsilon \left\langle a_{0,i} \right\rangle_t &\geq \left| \left\langle \Delta a_{g,i} \right\rangle_t \right|, && \text{(S36)}\\
\epsilon^2 \left\langle a_{0,i} \right\rangle_t^2 &\geq \left\langle \Delta a_{g,i}^2 \right\rangle_t . && \text{(S37)}
\end{aligned}$$

If the rates change sufficiently slowly with the population, it can be checked that Eqs. S36–S37 predict large times t for an iteration.

It is instructive to examine the case when the relative change in the population is small, i.e. $(g_i + \gamma_i)t \ll 1$. In this limit, a step size $t = \tau$ can be readily expressed that satisfies Eqs. S36–S37 for each propensity. The infinitesimal t versions of Eqs. S32–S33 are

$$\begin{aligned}
\left\langle \Delta a_{g,i} \right\rangle_t &\approx \mu_{g,i} t, && \text{(S38)}\\
\left\langle \Delta a_{g,i}^2 \right\rangle_t &\approx \sigma_{\gamma,i}^2 t, && \text{(S39)}
\end{aligned}$$

where

$$\begin{aligned}
\mu_{g,i} &\equiv \sum_j \frac{\partial g_i}{\partial n_j}(\vec{n}_0)\, n_{0,i}^2 \left(g_i(\vec{n}_0) - \gamma_i(\vec{n}_0) \right), && \text{(S40)}\\
\sigma_{g,i}^2 &\equiv \sum_j \left(\frac{\partial g_i}{\partial n_j}(\vec{n}_0) \right)^2 n_{0,i}^3 \left(g_i(\vec{n}_0) + \gamma_i(\vec{n}_0) \right) && \text{(S41)}
\end{aligned}$$

set the scale of variation by mean drift terms and diffusive terms, respectively. Analogous expressions exist for death. Application of Eqs. S36–S37 leads to a choice of sufficiently small time steps that keeps the variation small for propensities that use the rates g_i and γ_i

$$\begin{aligned}
\tau_{g,i} &= \min \left[\frac{\epsilon n_{0,i}(g_i + \gamma_i)}{|\mu_{g,i}|} , \frac{\epsilon^2 n_{0,i}^2 (g_i + \gamma_i)^2}{\sigma_{g,i}^2} \right], && \text{(S42)}\\
\tau_{\gamma,i} &= \min \left[\frac{\epsilon n_{0,i}(g_i + \gamma_i)}{|\mu_{\gamma,i}|} , \frac{\epsilon^2 n_{0,i}^2 (g_i + \gamma_i)^2}{\sigma_{\gamma,i}^2} \right]. && \text{(S43)}
\end{aligned}$$

A good overall step size is the minimum of all such step sizes

$$\tau = \min_i \left[\tau_{g_i}, \tau_{\gamma_i}\right]. \tag{S44}$$

The approximate step size in Eq. S44 is useful when comparing to other algorithms that require small times t for an integration step. It should be emphasized that the full expressions Eqs. S32–S33 should provide a better estimate of the error outside of the small t regime.

E.2 BNB algorithm as an approximation: with mutation

The analysis of error and step sizes above can be generalized to the case with mutation, where the BNB algorithm may spawn one or more mutants during the time step of size t. The primary difference in this case is that the expression for deviation of propensities should be made conditional on a mutation event not yet occurring, which is tied to the single species generating function ($G(s,t)/G(0,t)$, with $G(s,t)$ from the main text) of the conditional distribution. For simplicity, we assume that the rate of mutation μ_i of species i satisfies $\mu_i = \eta_i g_i$, where $0 < \eta_i < 1$. Thus, fixing the relative error of the birth propensity to be ϵ is sufficient to keep the relative error in mutation propensity to be less than this value.

It can then be shown that the the updated equations for Eqs. S40–S41 are the same to lowest order in t. If $\eta_i \ll 1$ (often the case), then Eqs. S42–S44 also hold. The discussion in the case with negligible mutation in this way provides a good estimate for the time step with slow mutation, as expected.

Outside of the small t limit, i.e. when $(g+\gamma)t \ll 1$ does not hold, full expressions analogous to those in Eqs. S32–S33 should be used. These depend on the full form of

$$\langle n \rangle_t = \left[\frac{\partial}{\partial s}\frac{G(s,t)}{G(0,t)}\right]_{s=0}, \tag{S45}$$

$$\left\langle \Delta n^2 \right\rangle_t + \langle n \rangle_t^2 = \left[\frac{\partial^2}{\partial s^2}\frac{G(s,t)}{G(0,t)}\right]_{s=0}, \tag{S46}$$

which can be given analytically.

E.3 Performance of approximate BNB relative to τ-leap algorithms

The approximate τ-leap algorithm is often used to accelerate stochastic simulations of chemical reaction networks [2]. It is based on the assumption that propensities for birth, death, and mutation are roughly constant for some time duration t, providing an corresponding generating function solution $G_{\tau\text{-leap},i}$ for species i

$$G_{\tau\text{-leap},i}(s,t) = \exp\left[-n_0\mu_i t + n_0 g_i t\left(e^s - 1\right) + n_0\gamma_i t\left(e^{-s} - 1\right)\right], \tag{S47}$$

which can be used to construct a τ-leap update

1. Generate the time $\tilde{t} = -(\sum_i n_{0,i}\mu_i)^{-1}\ln(\tilde{r})$, with uniform random number $\tilde{r} \in [0,1]$. If $\tilde{t} < \tau$, where τ is a time step small enough to keep a certain relative error, then flag that a mutation occurs after a time duration $\tilde{t}$. Set $t = \min(\tilde{t}, \tau)$.
2. Increment the time and update the populations to $n_i = n_{0,i} + \mathbf{P}_{1,i}(n_{0,i}g_i t) - \mathbf{P}_{2,i}(n_{0,i}\gamma_i t)$, where all of $\mathbf{P}_{1,i}$ and $\mathbf{P}_{2,i}$ are independent Poisson processes. For self-consistency, n_i should be nonnegative.
3. If a mutation has been flagged, create a new mutant from species i with probability $n_{0,i}\mu_i/(\sum_j n_{0,j}\mu_j)$.

Accuracy of the τ-leap algorithm depends sensitively on the step size of the algorithm. One prescription is to bound the relative error as in Eqs. S36–S37, as has been explored previously [7]. This leads to the modified definitions to be used in Eqs. S42–S44

$$\mu_{g,i} \equiv \sum_j \left(\frac{\partial g_i}{\partial n_j}(\vec{n}_0)\, n_{0,i} + g_i(\vec{n}_0)\delta_{i,j} \right) n_{0,i}\left(g_i(\vec{n}_0) - \gamma_i(\vec{n}_0)\right) \tag{S48}$$

$$\sigma^2_{g,i} \equiv \sum_j \left(\frac{\partial g_i}{\partial n_j}(\vec{n}_0)\, n_{0,i} + g_i(\vec{n}_0)\delta_{i,j} \right)^2 n_{0,i}\left(g_i(\vec{n}_0) + \gamma_i(\vec{n}_0)\right) \tag{S49}$$

and analogous definitions for death and mutation. Equations S48–S49 and corresponding equations can be substituted into Eqs. S42–S44 to provide the appropriate τ-leap time step with relative error ϵ.

In cases when elementary reactions are close to first order (as in population dynamics), the approximate BNB algorithm can perform much better than τ-leap and its analogs. Comparing the small time limits Eqs. S40–S41 to Eqs. S48–S49 allows some insight into how BNB compares to τ-leaping. For instance, it can be shown from these equations that BNB will have larger time step candidates $\tau_{g,i}$ than those for τ-leap if both

$$\left| \sum_j \frac{\partial g_i}{\partial n_j}(\vec{n}_0) n_{0,i} + g_i(\vec{n}_0) \right| > \left| \sum_j \frac{\partial g_i}{\partial n_j}(\vec{n}_0) n_{0,i} \right| \quad , \text{ and} \tag{S50}$$

$$\left| \frac{\partial g_i}{\partial n_i}(\vec{n}_0) n_{0,i} + g_i(\vec{n}_0) \right| > \left| \frac{\partial g_i}{\partial n_i}(\vec{n}_0) n_{0,i} \right| \tag{S51}$$

which imply

$$\sum_j \frac{\partial g_i}{\partial n_j}(\vec{n}_0) > -\frac{g_i(\vec{n}_0)}{2n_{0,i}} \quad , \text{ and} \tag{S52}$$

$$\frac{\partial g_i}{\partial n_i}(\vec{n}_0) > -\frac{g_i(\vec{n}_0)}{2n_{0,i}} \tag{S53}$$

Similar relations hold for death and mutation. Thus, weakly interacting species are better approximated by BNB for sufficiently slowly changing rates. For a single species, it can be shown that the boundary where BNB has similar accuracy as τ-leap implies a rate $\propto 1/\sqrt{n}$ and propensity $\propto \sqrt{n}$.

F Difference of histograms for independently sampled processes

In the main text, we used the difference of histograms of a population count as a measure of error for the process. We demonstrate how these statistics behave in the following.

A realization of a histogram for our purposes is defined by a total of N independent observations that are distributed to bins with probability ρ_i for the bin with index i. This realization provides bin counts $\tilde{a}_i$ for bin i, with $\sum_i \tilde{a}_i = N$. It follows that the $\tilde{a}_i$'s are distributed according to a multinomial distribution with total number N and probabilities ρ_i for bin i. The moments of the multinomial distribution can be approached via its generating function

$$G(\{s_i\}) = \left(\sum_i \rho_i e^{s_i} \right)^N \tag{S54}$$

which leads to the averages

$$\begin{aligned}\mu_i &\equiv \langle \tilde{a}_i \rangle = \lim_{s_k\to 0,\forall k} \frac{\partial}{\partial s_i} G = \rho_i\, N & \text{(S55)}\\ \mathcal{C}_{ij} &\equiv \langle \tilde{a}_i \tilde{a}_j \rangle - \langle \tilde{a}_i \rangle \langle \tilde{a}_j \rangle = \lim_{s_k\to 0,\forall k} \frac{\partial}{\partial s_i}\frac{\partial}{\partial s_j} G - \langle \tilde{a}_i \rangle \langle \tilde{a}_j \rangle \\ &= \left(\delta_{ij}\rho_i - \rho_i\rho_j\right) N & \text{(S56)}\end{aligned}$$

with δ_{ij} the Kronecker delta. For small ρ_i, $\mathcal{C}_{ij}$ is well approximated by

$$\mathcal{C}_{ij} \approx \delta_{ij} N \rho_i \ , \quad \text{small } \rho_i, \rho_j, \tag{S57}$$

which is the same as if each $\tilde{a}_i$ was independently Poisson distributed with mean value $N\rho_i$.

Consider two independent histograms with bin counts $\tilde{a}_{1,i}$ and $\tilde{a}_{2,i}$ that have respective bin probabilities $\rho_{1,i}$ and $\rho_{2,i}$. Furthermore, the total number of observations for each histogram is N, i.e. $\sum_i \tilde{a}_{1,i} = N$ and $\sum_i \tilde{a}_{2,i} = N$. We are interested in statistics of the error

$$\tilde{\mathcal{E}} = \sum_i \left(\tilde{a}_{1,i} - \tilde{a}_{2,i}\right)^2 . \tag{S58}$$

In particular, the mean of the error is

$$\begin{aligned}\left\langle \tilde{\mathcal{E}} \right\rangle &= \mathcal{E}_0 + N^2 \sum_i (\rho_{1,i} - \rho_{2,i})^2, & \text{(S59)}\\ \mathcal{E}_0 &= \sum_i \left[N\rho_{1,i}(1-\rho_{1,i}) + N\rho_{2,i}(1-\rho_{2,i})\right], & \text{(S60)}\end{aligned}$$

where $\mathcal{E}_0$ is the baseline expected error due to random fluctuations alone. Note that for small probabilities ρ_i, this is approximately

$$\mathcal{E}_0 \approx \sum_i \left[N\rho_{1,i} + N\rho_{2,i}\right] = 2N \ , \quad (\text{small } \rho_i), \tag{S61}$$

which is independent of the distributions.

For purposes of normalization in the main text, we divide $\mathcal{E}$ by the baseline $\mathcal{E}_0$ to get an error

$$\Delta = \frac{1}{\mathcal{E}_0} \sum_i \left(\tilde{a}_{1,i} - \tilde{a}_{2,i}\right)^2 , \tag{S62}$$

which has the average value

$$\langle \Delta \rangle = \frac{\langle \mathcal{E} \rangle}{\mathcal{E}_0} = 1 + \frac{N^2 \sum_i (\rho_{1,i} - \rho_{2,i})^2}{\mathcal{E}_0}. \tag{S63}$$

For small probabilities ρ_i, the expected value for Δ becomes

$$\langle \Delta \rangle \approx 1 + \frac{N}{2} \sum_i (\rho_{1,i} - \rho_{2,i})^2 \ , \quad (\text{small } \rho_i) \tag{S64}$$

which only depends on the square difference between bin probabilities.

The measurement of Δ depends on an estimate for $\mathcal{E}_0$. We do this by setting $\rho_{1,i} \approx \tilde{a}_{1,i}/N$ and $\rho_{2,i} \approx \tilde{a}_{2,i}/N$ and then calculating $\mathcal{E}_0$ by Eq. S60.

References

1. Gillespie DT (1977) Exact stochastic simulation of coupled chemical-reactions. Journal Of Physical Chemistry 81: 2340–2361.

2. Gillespie DT (2001) Approximate accelerated stochastic simulation of chemically reacting systems. Journal Of Chemical Physics 115: 1716–1733.

3. Zhu T, Hu Y, Ma ZM, Zhang DX, Li T, et al. (2011) Efficient simulation under a population genetics model of carcinogenesis. Bioinformatics 27: 837–843.

4. Gibson MA, Bruck J (2000) Efficient exact stochastic simulation of chemical systems with many species and many channels. Journal Of Physical Chemistry A 104: 1876–1889.

5. Kauffman S, Levin S (1987) Towards a general-theory of adaptive walks on rugged landscapes. Journal Of Theoretical Biology 128: 11–45.

6. Tsimring LS, Levine H, Kessler DA (1996) RNA virus evolution via a fitness-space model. Physical Review Letters 76: 4440–4443.

7. Gillespie DT, Petzold LR (2003) Improved leap-size selection for accelerated stochastic simulation. Journal Of Chemical Physics 119: 8229–8234.