Supporting Information
e3SIM: epidemiological-ecological-evolutionary simulation framework for genomic epidemiology

Peiyu Xu[1†], Shenni Liang[2, 3†], Andrew Hahn[2], Vivian Zhao[2], Wai Tung 'Jack' Lo[4], Benjamin C. Haller[4], Benjamin Sobkowiak[5], Melanie H. Chitwood[5], Caroline Colijn[6], Ted Cohen[5], Kyu Y. Rhee[7], Philipp W. Messer[4], Martin T. Wells[8], Andrew G. Clark[1, 4‡], and Jaehee Kim[4*‡]

[1]Department of Molecular Biology & Genetics, Cornell University, Ithaca, NY, USA
[2]Department of Computer Science, Cornell University, Ithaca, NY, USA
[3]Tri-Institutional Computational Biology & Medicine PhD Program, Weill Cornell Medicine, New York, NY, USA
[4]Department of Computational Biology, Cornell University, Ithaca, NY, USA
[5]Department of Epidemiology of Microbial Disease, Yale School of Public Health, New Haven, CT, USA
[6]Department of Mathematics, Simon Fraser University, Burnaby, BC, Canada
[7]Department of Medicine, Weill Cornell Medicine, New York, NY, USA
[8]Department of Statistics and Data Science, Cornell University, Ithaca, NY, USA
[†]These authors contributed equally.
[‡]These authors jointly supervised this work.
[*]Corresponding author. Email: jaehee.kim@cornell.edu

SI Text

S1 Details of genetic architecture and event probabilities

e3SIM explicitly links pathogen genotype to epidemiological outcomes by defining additive quantitative traits on the generalized linear model (GLM) link scale—e.g., log-odds for a logit link; log cumulative hazard, $\text{cloglog}(p) = \log[-\log(1-p)]$, for a complementary log–log discrete-time proportional-hazards parameterization—and mapping them to corresponding event probabilities via the inverse link [1]. We provide two built-in traits by default: transmissibility (per-contact transmission probability) and drug resistance (per-tick survival under treatment). Additional quantitative traits can be incorporated through straightforward source code customization.

S1.1 Additive genetic architecture

We model each trait using a standard additive quantitative-genetic architecture [2, 3]. For trait i, let $S^{(i)}$ denote the set of causal sites. We define its additive genetic value as:

$$G_i = \sum_{j \in S^{(i)}} q_{ij} x_j, \tag{S1}$$

where $x_j \in \{0, 1\}$ indicates the presence ($x_j = 1$) or absence ($x_j = 0$) of the effect allele at site j in the haploid pathogen genome, and q_{ij} is the additive effect (on the link scale) of site j on trait i [4]. Pleiotropy can be modeled when the same site j has nonzero effects q_{ij} on multiple traits.

S1.2 Link functions

We map the genetic value G_i of trait i to the probability of the corresponding epidemiological event using standard GLM links. We support the logit link and the complementary log-log (cloglog) link. Here, we illustrate two traits, transmissibility and drug resistance, but e3SIM supports any user-defined pathogen genetic traits through source code modification.

Logit link (log-odds scale). Let $\beta \in (0, 1)$ be the baseline per-contact transmission probability at $G_{\text{trans}} = 0$, and $s \in (0, 1)$ the baseline per-tick pathogen survival probability under treatment at $G_{\text{drug}} = 0$. We map the additive genetic values to probabilities via:

$$\begin{aligned} P(\text{transmission}) &= \text{logit}^{-1}\left(\text{logit}(\beta) + \alpha_{\text{trans}} G_{\text{trans}}\right), \\ P(\text{survival} \mid \text{treatment}) &= \text{logit}^{-1}\left(\text{logit}(s) + \alpha_{\text{drug}} G_{\text{drug}}\right), \end{aligned} \tag{S2}$$

where $\text{logit}(p) = \log(p/(1-p))$, $\text{logit}^{-1}(z) = 1/(1+e^{-z})$, and $\alpha_{\text{trans}}, \alpha_{\text{drug}}$ are slope parameters. Thus, under this link, a unit increase in G_{trans} multiplies the odds of transmission by $e^{\alpha_{\text{trans}}}$, and a unit increase in G_{drug} multiplies the odds of survival by $e^{\alpha_{\text{drug}}}$.

Complementary log–log (cloglog) link (discrete-time proportional hazards). Let $\lambda = -\log(1-\beta) > 0$ be the baseline transmission cumulative hazard per contact and $\kappa = -\log(s) > 0$ the baseline pathogen clearance cumulative hazard per tick under treatment (so baseline pathogen survival equals $e^{-\kappa}$). Then,

$$\begin{aligned} P(\text{transmission}) &= 1 - \exp\left(-\lambda e^{\alpha_{\text{trans}} G_{\text{trans}}}\right), \\ P(\text{survival} \mid \text{treatment}) &= \exp\left(-\kappa e^{-\alpha_{\text{drug}} G_{\text{drug}}}\right). \end{aligned} \tag{S3}$$

Equivalently, the per-tick clearance probability under treatment is $1 - P(\text{survival} \mid \text{treatment}) = 1 - \exp\left(-\kappa e^{-\alpha_{\text{drug}} G_{\text{drug}}}\right)$. Thus, $e^{\alpha_{\text{trans}}}$ is the transmission hazard ratio per $+1$ unit of G_{trans}, and $e^{-\alpha_{\text{drug}}}$ is the clearance hazard ratio per $+1$ unit of G_{drug} (so larger G_{drug} increases survival by reducing clearance hazard). These expressions correspond to standard discrete-time proportional hazards models [5, 6].

S2 Details of pre-simulation modules

e3SIM consists of four pre-simulation modules that configure the main simulation (Section 2.1). Here, we describe each pre-simulation module in detail.

S2.1 `NetworkGenerator`: Configure host population contact network

The host contact network is modeled as an undirected, unweighted graph, with nodes representing hosts and edges indicating potential epidemiological contacts. Users may either provide a custom contact network in a tab-delimited adjacency-list format or use the `NetworkGenerator` module to generate random contact networks using one of several supported, commonly used random network models (Figure 2A). In e3SIM, transmission events occur only between directly connected hosts, making the contact network crucial for shaping outbreak dynamics [7]. The resulting network topology remains static throughout the simulation.

e3SIM uses NetworkX [8] for random network generation, supporting Erdős–Rényi [9], Barabási–Albert [10], and random partition [11] networks. Given user-specified parameters and a chosen network model, `NetworkGenerator` generates a contact network and saves it as an adjacency list in the designated working directory. The Erdős–Rényi model approximates a homogeneous contact structure, with node degrees following a binomial distribution. The Barabási–Albert network generates a scale-free, preferential-attachment network with a highly skewed degree distribution, capturing heterogeneity in host contacts, and thus effectively modeling scenarios involving "superspreaders", individuals with substantially higher connectivity than the rest of the population [12]. The random partition model produces networks characterized by community structure, making it suitable for simulating epidemiological dynamics within spatially or socially structured populations (e.g., urban-rural differentiation).

Because topological characteristics of contact networks strongly affect epidemic dynamics and outcomes [13], the specific attributes of the contact network are critical. Thus, the graphical user interface (GUI) of the pre-simulation module provides a visualization of the generated network's degree distribution (Figure 2A, bottom), enabling users to fine-tune network parameters (e.g., connectivity and degree heterogeneity) to match the intended epidemiological scenario. Although the GUI explicitly visualizes only the degree distribution, other induced network properties (e.g., assortativity, clustering, and community structure) are determined by the underlying network model. Users may also directly input custom networks, including structured topologies like age-assorted contact networks, thereby offering flexibility in modeling diverse epidemiological contexts.

S2.2 `SeedGenerator`: Generate seed pathogen sequences

To initiate an epidemic simulation, e3SIM requires initial pathogen sequences ("seed sequences") infecting the host population. The `SeedGenerator` module generates these initial sequences by constructing their mutation profiles, represented as `VCF` files, along with their associated ancestral relationships. Users may provide custom seed sequences directly in `VCF` format or employ `SeedGenerator` to randomly generate a specified number of sequences. In the latter case, e3SIM uses `SLiM` 4.3 [14], which supports two modes: the Wright–Fisher mode and the network-based epidemiological mode.

In the Wright–Fisher mode (Figure 2B, top left), `SeedGenerator` performs a classical Wright–Fisher evolutionary simulation [15, 16]. The simulation initializes pathogen genomes identical to the provided reference genome and evolves them over a user-specified number of generations, using a substitution model defined by a transition probability matrix (Section S3.2) and an effective population size, both specified by the user. In the network-based epidemiological mode (Figure 2B, top right), `SeedGenerator` simulates an initial outbreak on the host contact network provided by `NetworkGenerator`, again initializing pathogen genomes identical to the reference genome. In both modes, the desired number of seed sequences for the main simulation module is randomly sampled from the pathogen population resulting from the `SeedGenerator` simulation (Figure 2B, bottom right) and stored as `VCF` files. Additionally, the ancestral relationships (genealogy) of the generated seed sequences are recorded (Figure 2B, bottom left) and saved in Newick (`NWK`) format.

S2.3 `GeneticEffectGenerator`: Define genetic architectures of traits

e3SIM explicitly integrates epidemiological, ecological, and evolutionary (epi-eco-evo) processes by letting pathogen genetic variation dynamically modulate key epidemiological traits. Users may (i) specify a genetic architecture explicitly (optionally with pleiotropy) or (ii) generate one at random (see Section S2.3.1 below for details).

We note that, while e3SIM provides a random genetic architecture generator for convenience, it is primarily suited for exploratory analyses and is not intended to be a comprehensive quantitative genetics simulator. Because e3SIM's main focus is on explicit epi-eco-evo dynamics, users who require biologically calibrated genetic architectures should either derive them from empirical studies or generate them using dedicated architecture/phenotype simulators that sample causal variants and effect sizes. The resulting architecture can then be imported via the user-input mode of `GeneticEffectGenerator` as a `CSV` file. Suitable options include PhenotypeSimulator [17], MultiTraitGWAS [18], `tstrait` [19], and `GWASBrewer` [20].

To accommodate dynamic epidemiological environments, such as temporal shifts in drug treatment strategies, genetic architectures can be specified in an epoch-specific manner (Section S3.1). This approach enables distinct genetic architectures to be applied for each epoch and trait (Figure 3A). `GeneticEffectGenerator` calculates and stores trait values for all seed sequences based on the defined genetic architecture, outputting the results to a `CSV` file in the working directory (Figure 2C). Users may inspect these trait values via the `HostSeedMatcher` tab within the GUI.

S2.3.1 Random genetic architecture generation

In the random generation mode, the module samples causal sites from a user-supplied candidate set in a `CSV` format. Within selected causal sites, mutations contribute additively to the trait, with individual effects drawn independently from a user-specified parametric family. We provide a principled two-step generator: (i) sparse causal site selection and (ii) flexible effect-size sampling.

Step 1: Choose causal sites $S^{(i)}$. For trait i, let $S_0^{(i)}$ be the trait-specific candidate set of genomic sites (the union of causal regions for trait i), and $n_i = |S_0^{(i)}|$. Let $c_{ij} \in \{0, 1\}$ indicate whether site $j \in S_0^{(i)}$ is causal for trait i, and define the number of causal sites for trait i as $K_i = \sum_{j \in S_0^{(i)}} c_{ij}$.

We model heterogeneity in polygenicity across traits by placing a Beta–Binomial prior on K_i:

$$\pi_i \sim \text{Beta}(a_i, b_i), \quad K_i \mid \pi_i \sim \text{Binomial}(n_i, \pi_i).$$

To make hyperparameters interpretable, we re-parameterize the Beta prior for per-trait polygenicity by its mean $\mu_i = \mathbb{E}[\pi_i]$ and concentration ξ, such that $a_i = \mu_i \xi$ and $b_i = (1 - \mu_i)\xi$. Under this parameterization,

$$\mathbb{E}[K_i] = n_i \mu_i, \quad \text{Var}(K_i) = n_i \mu_i (1 - \mu_i) \frac{\xi + n_i}{\xi + 1}.$$

Users specify an expected fraction of causal sites μ_i (default 0.003 [21]) and an optional prior strength ξ controlling dispersion around μ_i. We default to a moderately informative choice $\xi = 100$.

Given K_i, causal sites are sampled uniformly without replacement from the available candidate set for that trait. When more than one trait is simulated, to avoid unintended pleiotropy, we enforce trait exclusivity so that each site can be causal for at most one trait by requiring $\sum_i c_{ij} \leq 1$ for all genomic sites j. Traits are processed in a random order to reduce order effects. For trait i, define the available candidate site set $A^{(i)} = S_0^{(i)} \setminus \bigcup_{\ell < i} S^{(\ell)}$ and draw $S^{(i)} \subseteq A^{(i)}$ uniformly without replacement with $|S^{(i)}| = K_i$. We then set $c_{ij} = 1$ for $j \in S^{(i)}$ and 0 otherwise. If $|A^{(i)}| < K_i$, we set $S^{(i)} = A^{(i)}$, effectively capping the realized number of causal sites at $|A^{(i)}|$.

Step 2: Draw non-zero effect sizes. Conditional on $c_{ij} = 1$, we draw q_{ij} on the link scale from one of the following parametric families based on user input:

1. **Normal (point-normal)**: $q_{ij} \sim \mathcal{N}(0, \tau_i^2)$ with sparsity handled in Step 1 [22].
2. **Laplace (Bayesian lasso)**: $q_{ij} \sim \text{Laplace}(0, \eta_i)$ (heavy-tailed with shrinkage near zero) [23].
3. **Student-t**: $q_{ij} \sim t_{v_i}(0, \sigma_i)$ (heavy-tailed to allow a few large effects) [24].

We require $\eta_i > 0$, $\sigma_i > 0$, and recommend $v_i > 2$ for a finite variance $\text{Var}(q_{ij}) = \sigma_i^2 v_i / (v_i - 2)$. Default hyperparameter values are shown in Table S4.

S2.3.2 Standardize genetic values and calibrate link slopes

Given additive link-scale genetic values G_i (Eq. S1) for seeds and trait i, we (i) standardize G_i to mean zero and a user-specified variance on the link scale, and (ii) calibrate the link slope α_i so that probabilities implied by Eqs. S2 and S3 match interpretable per-SD (standard deviation) odds ratios (logit) or hazard ratios (cloglog). This calibration follows standard GLM parameterization for the logit link and the grouped-time proportional-hazards interpretation of the cloglog link [25].

Step 1: Mean-centering by allele frequency and variance standardization. To make α_i interpretable and identifiable, we first standardize the seeds' genetic values to mean zero and a user-chosen variance. We adopt allele-frequency centering, which is standard in quantitative genetics [4].

Let m be the number of seed genomes, and let $G_{i,k}$ be the raw link-scale genetic value for trait i in seed genome $k = 1, \ldots, m$. Let $x_{j,k} \in \{0, 1\}$ denote the derived allele indicator at site j in seed k. Define the allele frequency of mutation j in the seeding population and the centered genotype as:

$$\hat{p}_j = \frac{1}{m} \sum_{k=1}^{m} x_{j,k}, \quad z_{j,k} = x_{j,k} - \hat{p}_j.$$

The centered genetic value is then

$$G'_{i,k} = \sum_{j \in S^{(i)}} q_{ij} z_{j,k}.$$

By construction, $\frac{1}{m} \sum_{k=1}^{m} z_{j,k} = 0$ for each j, and thus $\frac{1}{m} \sum_{k=1}^{m} G'_{i,k} = 0$ for any fixed effect vector (q_{ij}).

The across-seed dispersion of genetic values is then fixed by moment matching [26, 27]. We compute the empirical mean and variance of $G'_{i,k}$ across seeds:

$$\hat{\mu}_i = \frac{1}{m}\sum_{k=1}^{m} G'_{i,k} = 0, \quad \hat{V}_i = \frac{1}{m-1}\sum_{k=1}^{m}(G'_{i,k} - \hat{\mu}_i)^2.$$

Given a user-chosen target variance $V_{i,\text{target}} > 0$, define the standardized trait value and scaled effect sizes as:

$$\widetilde{G}_{i,k} = r_i(G'_{i,k} - \hat{\mu}_i), \quad \widetilde{q}_{ij} = r_i q_{ij}, \quad \text{where } r_i = \sqrt{\frac{V_{i,\text{target}}}{\hat{V}_i}},$$

By construction, $\frac{1}{m}\sum_{k=1}^{m}\widetilde{G}_{i,k} = 0$ and $\frac{1}{m-1}\sum_{k=1}^{m}\widetilde{G}^2_{i,k} = V_{i,\text{target}}$. This fixes the scale of genetic values, so that slopes α_i are interpretable per $\text{SD}_i = \sqrt{V_{i,\text{target}}}$ units [28, 29]. We set the default $V_{i,\text{target}} = 1$ (hence $\text{SD}_i = 1$).

If seeds are monomorphic at all causal sites for trait i, $\hat{V}_i = 0$, so empirical per-SD calibration is undefined. We therefore standardize using the expectation of additive genetic variance [30] under a Beta prior on allele frequencies p_j [31] rather than the empirical seed variance:

$$V_i^* = \sum_{j \in S^{(i)}} q_{ij}^2 \mathbb{E}\left[p_j(1-p_j)\right].$$

By default, we assume loci are in linkage equilibrium and use a uniform Beta prior to set a reference scale, for which $\mathbb{E}\left[p_j(1-p_j)\right] = 1/6$. We then rescale effects by $\widetilde{q}_{ij} = r_i q_{ij}$ with:

$$r_i = \sqrt{\frac{V_{i,\text{target}}}{V_i^*}},$$

giving a fixed reference unit $\text{SD}_i = \sqrt{V_{i,\text{target}}}$.

Step 2: Calibrating the link-scale slope. Users specify per-SD effect multipliers R_i for trait i, which are translated to slopes α_i as:

- **Logit link (Eq. S2)**: user supplies per-SD odds ratio $R_{i,\text{OR}} > 0$ for trait i; set:

$$\alpha_i = \frac{\log R_{i,\text{OR}}}{\text{SD}_i}, \quad \text{which ensures} \quad \frac{\text{odds}(\text{SD}_i)}{\text{odds}(0)} = R_{i,\text{OR}}.$$

- **Cloglog link (Eq. S3), transmissibility**: user supplies per-SD transmission hazard ratio $R_{\text{trans,HR}} > 0$; set:

$$\alpha_{\text{trans}} = \frac{\log R_{\text{trans,HR}}}{\text{SD}_{\text{trans}}}, \quad \text{which ensures} \; \frac{h_{\text{trans}}(\text{SD}_{\text{trans}})}{h_{\text{trans}}(0)} = R_{\text{trans,HR}}, \; \text{where } h_{\text{trans}}(G) = \lambda e^{\alpha_{\text{trans}} G}.$$

- **Cloglog link (Eq. S3), drug resistance**: user supplies per-SD clearance hazard ratio $R_{\text{drug,clr}} > 0$; set:

$$\alpha_{\text{drug}} = -\frac{\log R_{\text{drug,clr}}}{\text{SD}_{\text{drug}}}, \quad \text{which ensures} \; \frac{h_{\text{clear}}(\text{SD}_{\text{drug}})}{h_{\text{clear}}(0)} = R_{\text{drug,clr}}, \; \text{where } h_{\text{clear}}(G) = \kappa e^{-\alpha_{\text{drug}} G}.$$

By default, we set $R_{i,\text{OR}} = 1.5$, $R_{i,\text{HR}} = 1.5$, and $R_{i,\text{clr}} = 0.67$ ($\approx 1/1.5$) to impose moderate per-SD effects and keep link-scale coefficients in a numerically stable range. Under the logit link, $\log(1.5) \approx 0.405$ (odds-ratio interpretation). Under the cloglog link, the exponentiated coefficient is a discrete-time hazard ratio; thus $R_{i,\text{HR}} = 1.5$ and $R_{i,\text{clr}} = 0.67$ correspond to hazard ratios of 1.5 and 0.67, respectively, with 0.67 representing an approximately 33% reduction in hazard (a moderate protective effect).

S2.4 `HostSeedMatcher`: Match seed sequences to initial hosts

To initiate a simulation in e3SIM, pathogen seed sequences must be assigned to one or more initial hosts (Figure 2D). This process requires providing the main simulator module with a `CSV` file that specifies the mapping between seed sequences and hosts. Users may either generate this mapping via the `HostSeedMatcher` module or supply it manually.

`HostSeedMatcher` supports three assignment schemes. The "Uniform Selection" scheme assigns each seed to a host uniformly at random from all available hosts. The "Contact Ranking" scheme assigns seeds based on the hosts' contact degree ranking. For example, the host with the highest contact degree receives the first seed sequence, the second highest receives the second seed, and so forth. Each seed can be assigned to a host of any rank. The "Percentile-Based Selection" scheme assigns seeds to hosts within a user-specified percentile range of the contact-degree distribution (e.g., selecting hosts from the top 10% of contact degrees). Each seed sequence may use a distinct matching scheme. When configuring via the GUI, users can interactively view the host contact-degree distribution, inspect calculated seed trait values, select matching schemes per seed, and inspect the resulting host–seed assignments interactively on the displayed distributions. By default, all seeds are matched with the "Uniform Selection" scheme. When multiple seeds use different schemes, seeds are processed in the order of (i) "Contact Ranking", (ii) "Percentile-Based Selection", and (iii) "Uniform Selection"; hosts matched under an earlier scheme are excluded from subsequent schemes.

S3 Details of main module `OutbreakSimulator`

The main simulation module of e3SIM uses `SLiM` as a backend to conduct a discrete-time epi-eco-evo simulation (Figure 3). Here, we describe this module in detail.

S3.1 Ticks and epochs

`OutbreakSimulator` employs a discrete-time simulation framework, where each time step (*tick*) represents a unit interval in which epidemiological, ecological, and evolutionary events occur for hosts and pathogens. The simulation runs for a predefined number of ticks. Within each tick, pathogen transmission events and compartmental transitions among hosts are evaluated. Ticks are grouped into *epochs*, each consisting of a sequence of consecutive ticks. Within an epoch, base compartmental transition probabilities remain constant, and trait-specific genetic architectures are either activated or deactivated. Compartmental probabilities and the activation states of genetic architectures may vary across epochs (Figure 3A), enabling dynamic modeling of temporal environmental changes—such as modifications in public health interventions or treatment protocols—at specified times during an outbreak.

S3.2 Evolutionary model

Mutations accumulate stochastically in all pathogen genomes at every tick (Figure 3B(i)), according to a user-specified substitution model. This substitution model is represented by a nucleotide substitution transition probability matrix $\mathbf{P}$, a 4×4 matrix in which P_{ij} specifies the probability that a nucleotide in state i transitions to state j within a single tick ($i, j \in \{\mathrm{A, C, G, T}\}$). For $i \neq j$, P_{ij} gives the per-tick substitution probability from i to j, and $P_{ii} = 1 - \sum_{j \neq i} P_{ij}$ is the probability of no change. The total probability of mutation per site per tick from state i is thus $\sum_{j \neq i} P_{ij}$. If users do not explicitly provide a transition probability matrix, a mutation rate must be specified, and the substitution model defaults to the Jukes–Cantor model [32], which assumes equal substitution rates among nucleotides.

Mutations occur probabilistically at each tick as dictated by the substitution model, regardless of whether pathogens undergo reproduction (Section S3.3) in that tick. Pathogen trait values, determined by their current mutation profiles (Section S2.3), are recalculated prior to evaluating epidemiological events (Figure 3B(ii)). These recalculated trait values subsequently modify the probabilities of relevant epidemiological events, such as pathogen transmission (Section S3.3) and host recovery (Section S3.4). This dynamic coupling addresses crucial feedback mechanisms frequently overlooked in classical epidemiological simulators (Table 1 and Figure 1A). Users may independently specify substitution models for the `SeedGenerator` and `OutbreakSimulator` modules.

S3.3 Reproduction: transmission, superinfection, and within-host replication

In e3SIM, pathogen reproduction occurs through two primary mechanisms: host-to-host transmission and within-host replication. Pathogen genomes are represented as discrete agents within each host. When both within-host replication and superinfection are disabled, each infected host carries exactly one genome; the single agent plays the role of a representative genotype for that infection. When either feature is enabled, a host may carry multiple genomes up to a user-specified within-host capacity. This discrete-agent representation can be viewed as a coarse genotype-class approximation where abundances are represented by the counts of genomes per genotype present.

Transmission

Transmission involves a pathogen infecting a new host through the production of offspring genomes in that host and occurs exclusively between directly connected hosts. During each tick, all effective host contacts are evaluated for potential transmissions (Figure 3B(i)). An effective contact is defined as a directly connected host pair comprising an infectious individual (infector) and another host capable of acquiring infection (infectee); infectees may include already infected hosts if superinfection is enabled. e3SIM assumes a strong transmission bottleneck [33–35]: for each effective contact, one genome is sampled uniformly at random from the infector, and transmission success is then modeled by a Bernoulli trial with success probability mapped from the sampled genome's transmissibility trait (Section S1). Successful trials result in the transmission of the sampled pathogen genome to the infectee (subject to within-host capacity and superinfection rules).

Superinfection

"Superinfection" refers to a scenario in which a host can be infected by multiple pathogens originating from different infectors, either simultaneously within a single tick or sequentially across multiple ticks (Figure 3B(iii)). When superinfection is disabled, infection is restricted to hosts in the susceptible state. If multiple successful transmission events target the same susceptible host within a tick, a single infecting pathogen is randomly selected from all successful transmission events to that host within that tick. When superinfection is enabled, multiple transmitted genomes can enter the same host in the same tick and coexist provided the total does not exceed the user-defined maximum within-host capacity. In this mode, successful transmissions into a host within the same tick are retained up to the within-host capacity.

Within-host replication

Within-host replication occurs when pathogens reproduce within an individual host (Figure 3B(iv)). This process is activated only when within-host replication is enabled and the within-host pathogen load (census number of genomes) is below the user-specified maximum capacity; once this hard cap is reached, replication is suspended and no additional offspring are generated. Under these conditions, each pathogen within the host independently produces at most one offspring per tick with a user-defined probability. Mutations accumulate independently of reproduction events, as detailed in Section S3.2. Therefore, all pathogen genomes, regardless of their replication status, have the same per-tick mutation probability (i.e., sequence evolution follows a per-tick molecular clock rather than a replication-coupled mutation model).

This within-host representation tracks genetic diversity within hosts and defines the pool from which transmitted founders are sampled (under the transmission bottleneck), while allowing genotype-dependent survival under treatment (Section S3.4). Additional within-host interactions such as density-dependent growth or explicit strain competition are not modeled in the current implementation to keep within-host dynamics lightweight and emphasize epi–eco–evo coupling at the transmission scale. Complementary frameworks place greater emphasis on within-host evolutionary structure, for example: simulators that explicitly parameterize transmission bottlenecks and within-host effective population size with associated dynamics (e.g., SEEDY [36]) and frameworks such as Opqua [37] that incorporate coinfection with explicit fitness-weighted within-host competition and selection, potentially allowing transmission of multiple genomes.

S3.4 State changes: activation, recovery, and loss of immunity

Host state transitions (Figure 3C) are evaluated after reproduction events. At each tick, each host undergoes at most one state transition. For each host, the next state is sampled independently via a categorical (multinoulli) draw over all admissible outcomes from its current state (including no change). In treatment epochs, infectious hosts first undergo a recovery trial; conditional on not recovering, the remaining state transitions are then evaluated. All major epidemiological events—including infection, activation, recovery, and sampling—are recorded in `CSV` files stored in the working directory.

Activation and deactivation

Following successful pathogen transmission to a susceptible host, the host transitions to either the exposed or infectious state. The exposed state represents a latent infection period, during which pathogens are present (and potentially replicating within the host) but not yet transmissible. A newly infected host enters the exposed state with probability ζ or directly transitions to the infectious state with probability $1 - \zeta$. An exposed host transitions to the infectious state (*activation*) with probability ν. Conversely, an infectious host may revert to the exposed state (*deactivation*) with probability ϕ, representing a return to latency.

Recovery and treatment

Recovery in e3SIM is the transition of a host from the exposed or infectious state to the recovered state. Recovery may also be induced by sampling, as detailed in Section S3.5. At each tick, an exposed host recovers with probability τ and an infectious host recovers with probability γ via an immune-mediated host-level recovery trial representing host-intrinsic (innate) processes independent of treatment.

When treatment is active, an infectious host additionally undergoes drug-induced clearance within the same tick. Because latent infections are typically asymptomatic and undetected, and treatment targets clinically apparent/diagnosed cases, we apply treatment only to infectious hosts and omit treatment for exposed hosts, consistent with clinical practice and standard compartmental models [38, 39]. We first evaluate the innate recovery event with probability γ; on success, the host is set to recovered, and all within-host pathogens are cleared. If the immune trial fails, we then evaluate drug-induced clearance at the pathogen level: each pathogen k present in the host at the time of treatment evaluation survives treatment independently with probability $p_k = P(\text{survival} \mid \text{treatment})$, mapped from its drug-resistance genetic value (Eq. S1) via the specified link (Eqs. S2 and S3) and baseline survival s at $G_{\text{drug}} = 0$. The host recovers if and only if all within-host pathogens fail their survival trials in that tick; otherwise, the host remains in its current infectious state with the surviving pathogens.

Assuming independence between the innate immune process and drug-induced clearance within a tick, and independence across pathogens, recovery occurs if either innate immunity succeeds (probability γ) or, conditional on failure (probability $1 - \gamma$), all pathogens are cleared by treatment. The per-tick host recovery probability under treatment is thus:

$$P_{\text{rec}} = \gamma + (1 - \gamma) \prod_{k=1}^{n_h} (1 - p_k),$$

where n_h is the number of pathogens in the host when treatment is evaluated in that tick. When there is no treatment or all pathogens are completely drug-resistant ($p_k = 1$ for all k), then $P_{\text{rec}} = \gamma$. When the drug is perfectly effective ($p_k = 0$ for all k), $P_{\text{rec}} = 1$. At each tick, the host transitions to the recovered state with probability P_{rec}; otherwise, it remains in its current state.

Immunity

e3SIM assumes recovered hosts have temporary immunity, preventing immediate reinfection. However, recovered hosts may lose this immunity with probability ω per tick, transitioning back to the susceptible state and becoming vulnerable to reinfection.

S3.5 Sampling

`OutbreakSimulator` supports two sampling schemes: sequential sampling and concerted sampling, consistent with standard birth-death-sampling frameworks [40]. In sequential sampling, infectious hosts are sampled independently at each tick, with each sampling event modeled as a Bernoulli trial; under a constant per-tick sampling probability, waiting times between sampling events follow a geometric distribution for each infectious host [41, 42]. In contrast, concerted sampling samples infectious hosts at predefined ticks [43].

Sequential (heterochronous) sampling occurs independently for each infectious host at each tick with probability ϵ_s, concurrent with the host state transitions (Section S3.4). For each sampled host, post-sampling recovery occurs with probability δ_s (Figure 3C) [42, 44, 45]. This post-sampling recovery is distinct from the pre-sampling recovery trial and is executed at the host level without pathogen-level survival evaluation, regardless of treatment status. Concerted (isochronous) sampling occurs at user-defined ticks immediately after host state transitions. At each concerted event, every currently infectious host is independently sampled with probability ϵ_c. Sampled hosts then undergo immediate host-level post-sampling recovery with probability δ_c. This post-sampling recovery is again separate from the pre-sampling recovery trial and does not invoke pathogen-level survival evaluation, regardless of treatment status. Both sequential and concerted sampling can be enabled concurrently within the same simulation. When both methods are enabled, sequential sampling events are executed first at each tick, followed by concerted sampling at user-specified ticks.

S4 Methods for simulation studies

S4.1 Epi-eco-evo dynamics in fast-evolving pathogens (Section 3.1.1)

We simulated a SARS-CoV-2 outbreak in a population of 10,000 individuals over 730 ticks (1 tick = 1 day), partitioned into three epochs. The host contact network was generated using the Barabási–Albert model in `NetworkGenerator` (`-model BA -m 2`). We ran 50 replicates; 43 were successful, with at least one pathogen genome sampled at the end of the simulation. Ten replicates were randomly selected for visualization in Figures S2–S4, and one was selected for visualization in Figure 4.

We defined genetic architectures for transmissibility and drug resistance using the random mode of `GeneticEffectGenerator` by specifying candidate causal regions and sampling causal sites within those regions. Candidate regions on the SARS-CoV-2 reference genome hCoV-19/Wuhan/WIV04/2019 (EPI_ISL_402124) [46] from GISAID [47] were ORF1a (266–13468, length: 13202 nt) for transmissibility (motivated by the roles of ORF1a-encoded nonstructural proteins in host–virus interactions [48]) and S (spike; 21563–25384, length: 3821 nt) for both transmissibility and drug resistance (consistent with spike as a major therapeutic/vaccine target in SARS-related coronaviruses [49]). Drug resistance was modeled using two disjoint sets of causal sites, one for each drug activated sequentially in epochs 2 and 3, representing resistance to temporally distinct treatment pressures. Effect sizes were sampled from a normal distribution and scaled to match target variances of the (link-scale) genetic values in the seeding population. The genetic architectures were generated using the following `GeneticEffectGenerator` arguments:

```
-method randomly_generate -num_init_seq 1 -csv candidate_regions.csv
-site_frac 0.05 0.1 0.1 -trait_n ’{"transmissibility": 1, "drug_resistance": 2}’
-var_target 1 10 10  -func n
```

The host population was seeded with a single SARS-CoV-2 reference genome and assigned to the host with median contact degree using the “Contact Ranking” mode of `HostSeedMatcher`:

```
-method randomly_generate -num_init_seq 1
-match_scheme ’{"0": "ranking"}’ -match_scheme_param ’{"0": 5000}’
```

Treatment was activated at ticks 250 and 500 (epochs 2 and 3, respectively). Sequential samples were collected with probability 0.0002 per tick during epoch 1 and 0.0003 per tick during epochs 2 and 3, reflecting increased screening during treatment. Full `OutbreakSimulator` settings are provided in Configuration File S1.

S4.2 Epidemic dynamics with superspreaders (Section 3.1.2)

We simulated a *Mtb* outbreak in a host population of 10,000 individuals over a 10-year period using e3SIM. Two contact network structures were considered: (i) an Erdős–Rényi network (`-p_ER 0.001`) with comparatively homogeneous mixing, and (ii) a Barabási–Albert network (`-m 2`) with degree heterogeneity and superspreaders. Seed sequences were generated with `SeedGenerator` under the Wright–Fisher model (`-seed_size 5 -method SLiM_burnin_WF -Ne 1000`) using the *Mtb* H37Rv reference genome (GCF_000195955.2) [50]. We defined a genetic architecture for transmissibility based on standing genetic variation in the seed sequences generated by `SeedGenerator`, such that the seed set spans a range of transmissibility values (Figure 5A). The genetic architecture includes three causal sites with effect sizes 0.4, 0.2, and 0.2, respectively (see Code Availability). We calibrated the logit-link slope using a per-SD odds ratio of 2 (Section S2.3.2). The arguments to run `GeneticEffectGenerator` are:

```
-method user_input -num_init_seq 5 -csv causal_gene_info.csv
-trait_n ’{"transmissibility": 1, "drug_resistance": 0}’ -calibration_link T -Rs 2
```

Using the two network structures generated by `NetworkGenerator`, we simulated three scenarios to demonstrate the impact of superspreaders and network structure on epidemic dynamics. For each scenario, samples were collected sequentially with probability 0.00001 per tick across 50 replicates. All 50 simulation replicates were successful (i.e., at least one pathogen genome was sampled).

In scenario 1, we assigned the five seed sequences uniformly at random to hosts in the Erdős–Rényi network (Figure 5A) using the “Uniform Selection” mode of `HostSeedMatcher`. In scenario 2, high-transmissibility seed sequences were assigned to highly connected hosts and lower-transmissibility seeds to less connected hosts in the Barabási–Albert network (Figure 5B), using the “Contact Ranking” mode. The `HostSeedMatcher` arguments were:

```
-match_scheme ’{"0": "ranking", "1": "ranking", "2": "ranking", "3":"ranking", "4":"ranking"}’
-match_scheme_param ’{"0": 1000, "1": 2000, "2": 500, "3": 5000, "4":6000}’ -num_init_seq 5
```

In scenario 3, we reversed the matching scheme, assigning lower-transmissibility seeds to higher-degree hosts and vice versa, again using the “Contact Ranking” mode. The `HostSeedMatcher` arguments were:

```
-match_scheme ’{"0": "ranking", "1": "ranking", "2": "ranking", "3":"ranking", "4":"ranking"}’
-match_scheme_param ’{"0": 3000, "1": 5000, "2": 6000, "3": 500, "4":550}’ -num_init_seq 5
```

The complete configuration details for `OutbreakSimulator` are provided in Configuration File S2.

S5 Methods for runtime profiling (Section 3.2)

To benchmark the performance of `OutbreakSimulator`, we ran tests on a MacBook Pro (Apple M2 Pro, 32 GB RAM; macOS Sequoia 15.7.1). Erdős–Rényi networks were generated using `NetworkGenerator` with an average contact degree of 10, varying the host population size $N \in \{10000, 25000, 50000, 75000, 100000\}$ and setting `-p_ER` from 0.001 to 0.0001. For each simulation, the host with the median contact degree was seeded with the SARS-CoV-2 reference genome, and sequential sampling was performed throughout the simulation. Sampling probabilities were adjusted so that the maximum expected number of samples per tick was 1, 5, or 10 for each population size, and runtimes were recorded (Figure 6). The same benchmarks were repeated on a Linux system (Figure S7A); for each parameter set, we ran 10 replicates on macOS and 50 replicates on Linux.

We conducted additional performance benchmarks on a Linux system to evaluate runtime across varying outbreak sizes at a fixed host population size (100,000) under an Erdős–Rényi contact network. Simulation parameters matched those in the 100,000-host scenario in Figure S7B, except that the base transmission probability (β) was varied from 0.01 to 0.06. Sequential sampling was performed with a probability of 0.0001 per tick. These benchmarks were run on a single CPU core with memory limited to 6 GB.

SI Tables

Table S1. Overview of random generation modes in e3SIM modules. For each module, users can either directly provide custom input files or use built-in random generation options. This table summarizes inputs, outputs, and primary functions for each module in random generation mode. Detailed documentation and usage examples are provided in the software manual (see Code Availability).

	Module	**Inputs**	**Outputs**	**Function**
Pre-simulation	**`NetworkGenerator`** **(Section S2.1)**	• Random network model and associated parameters.	• Adjacency list of the contact network.	Generate a host contact network.
	`SeedGenerator` **(Section S2.2)**	• Pathogen reference genome (`FASTA`). • Simulation model and parameters. • Optional: Contact network from `NetworkGenerator`.	• `VCF` files for each seed sequence. • `NWK` file of the seed genealogy.	Generate seed pathogen sequences for the main simulation.
	`GeneticEffectGenerator` **(Section S2.3)**	• Candidate causal regions (`CSV`). • Parameters for genetic architecture generation. • Optional: Seed sequences (`VCF`) from `SeedGenerator`.	• `CSV` file specifying the genetic architecture. • `CSV` file containing computed trait values for each seed sequence.	Define genetic architectures and compute pathogen trait values (e.g., transmissibility, drug resistance).
	`HostSeedMatcher` **(Section S2.4)**	• Host contact network from `NetworkGenerator`. • Matching scheme and parameters.	• `CSV` file specifying the host–seed mapping.	Assign seed sequences to initial hosts based on user-defined matching criteria.
Main & Post-simulation	**`OutbreakSimulator`** **(Section 2.2)**	• Contact network from `NetworkGenerator`. • Seed sequences from `SeedGenerator`. • Genetic architecture from `GeneticEffectGenerator`. • Host–seed mapping from `HostSeedMatcher`. • Pathogen reference genome. • Simulation configuration (`JSON`).	• `VCF` files of sampled pathogen genomes. • `CSV` files logging all epidemiological events. • `NWK` file and node metadata file for sampled pathogen genealogies. • Visualization of sampled pathogen genealogies. • SEIR compartment trajectory plot. • Lineage frequency trajectory plot.	Execute the full simulation using provided inputs and the configuration file, generating output files documenting epidemiological events and sampled pathogen sequences.

Table S2. Main simulation components defining epidemiological events in `OutbreakSimulator`. Reproduction, compartmental state transitions, and sampling events occur probabilistically in each tick. These processes are configured within specific blocks in the simulation's `JSON` configuration file, with corresponding block names listed in the second column.

Component	**Relevant configuration blocks (in the configuration file)**	**Function**

Reproduction (Section S3.3)	`genetic_architecture,`
	`EvolutionModel`
		Defines between-host transmission and within-host reproduction events. For each trait, the `genetic_architecture` block contains a list whose length equals the number of epochs; the k-th element gives the index of the genetic architecture activated in epoch k. These indices map to effect-size columns in the genetic architecture `CSV` from `GeneticEffectGenerator`.

Compartmental model (Section S3.4)	`transition_prob, model,`
	`genetic_architecture`
		Defines the compartmental epidemiological model that specifies transition probabilities among compartments S (Susceptible), E (Exposed), I (Infectious), and R (Recovered). Each transition probability is a list whose length equals the number of epochs defined in `epoch_changing`. Transmission and recovery probabilities are further modulated dynamically by pathogen genetic architectures affecting transmissibility and drug resistance traits (Section S2.3).

Sampling (Section S3.5)	`transition_prob,`
	`massive_sampling`
		Defines the sampling scheme used in the simulation. Sequential sampling probabilities and associated recovery probabilities upon sampling are specified within the `transition_prob` block, whereas concerted sampling events are defined separately in the `massive_sampling` block.

Table S3. Symbols for epidemiological parameters (Figure 3C). Unless otherwise noted, probabilities are evaluated per tick during host state updates. In treatment epochs, infectious hosts first undergo host-level immune recovery with probability γ; conditional on failure, pathogen-level treatment survival trials use baseline survival s. Sequential sampling (ϵ_s) is evaluated each tick, whereas concerted sampling (ϵ_c) is evaluated only at user-defined concerted sampling ticks. Post-sampling recovery probabilities (δ_s, δ_c) are conditional on sampling and are applied at the host level without pathogen-level survival evaluation, independent of treatment status.

Symbol	Description
ω	Probability of immunity loss for a recovered host ($R \rightarrow S$).
β	Baseline probability of successful infection per effective host contact.
ζ	Probability of entering the exposed state upon successful infection.
ϕ	Probability of transitioning back to the exposed state for an infectious host (deactivation; $I \rightarrow E$).
ν	Probability of transitioning from exposed to infectious state (activation; $E \rightarrow I$).
τ	Probability of direct transition from exposed to recovered state ($E \rightarrow R$).
γ	Probability of immune-mediated, host-level recovery for an infectious host; during treatment epochs, evaluated before drug-induced clearance.
s	Baseline probability of pathogen survival under treatment, conditional on failure of the immune-mediated recovery trial (baseline at $G_{\text{drug}} = 0$).
ϵ_s	Probability of an infectious host being sampled in sequential sampling.
δ_s	Probability of post-sampling recovery following sequential sampling of an infectious host.
ϵ_c	Probability of an infectious host being sampled in concerted sampling.
δ_c	Probability of post-sampling recovery following concerted sampling of an infectious host.

Table S4. Symbols and default values for the genetic architecture model of e3SIM.

	Symbols	Descriptions	Default values	Note
Additive genetic architecture & link mapping	$S^{(i)}$	Set of causal sites for trait i.	–	User-specified (CSV) or drawn in random mode.
	x_j	Indicator of the effect allele at site j (haploid pathogen genome).	–	$x_j \in \{0,1\}$; for seed genome k, use $x_{j,k}$. • $x_j = 1$ if the effect allele is present; • $x_j = 0$ if the effect allele is absent.
	q_{ij}	Additive effect of site j on trait i.	–	• User-specified or sampled if $j \in S^{(i)}$. • User-defined pleiotropy allowed when a site has nonzero q_{ij} for multiple traits.
	G_i	Genetic value for trait i.	derived	$G_i = \sum_{j\in S^{(i)}} q_{ij}x_j$; for seed genome k, use $G_{i,k}$.
	β	Baseline per-contact transmission probability at $G_{\text{trans}} = 0$.	–	Set in the epidemiological model (Table S3).
	s	Baseline per-tick survival under treatment at $G_{\text{drug}} = 0$.	–	Set in the epidemiological model (Table S3).
	λ	Baseline transmission cumulative hazard per contact (cloglog link).	derived	$\lambda = -\log(1-\beta)$.
	κ	Baseline clearance cumulative hazard per tick under treatment (cloglog link).	derived	$\kappa = -\log(s)$.
Random genetic architecture generation	$S_0^{(i)}$	Trait-specific candidate site set (union of candidate regions for trait i).	–	User-supplied candidate set (CSV).
	n_i	Candidate site count for trait i.	derived	$n_i = \lvert S_0^{(i)} \rvert$.
	π_i	Per-site causal probability for trait i.	–	$\pi_i \sim \text{Beta}(a_i, b_i)$ with $a_i = \mu_i\xi, b_i = (1-\mu_i)\xi$.
	μ_i	Prior mean causal fraction for trait i; $\mu_i = E[\pi_i]$.	0.003	Default expected fraction of causal sites.
	ξ	Beta prior concentration (precision) for π_i.	100	Controls dispersion around μ_i (moderately informative).
	K_i	Number of causal sites for trait i.	–	$K_i \mid \pi_i \sim \text{Binomial}(n_i, \pi_i)$.
	c_{ij}	Causal indicator for site j on trait i.	–	$c_{ij} = 1$ if $j \in S^{(i)}$, else 0. Random mode enforces trait exclusivity $\sum_i c_{ij} \le 1$ by default.
	τ_i^2	Variance for Normal effects (if Normal chosen): $q_{ij} \sim \mathcal{N}(0, \tau_i^2)$.	1	$\tau_i^2 > 0$ (pre-standardization scale). (not to be confused with epi-model τ in Table S3)
	η_i	Scale for Laplace effects (if Laplace chosen): $q_{ij} \sim \text{Laplace}(0, \eta_i)$.	0.75	$\eta_i > 0$ (pre-standardization scale).
	(σ_i, v_i)	Scale and df for Student-t effects (if Student-t chosen): $q_{ij} \sim t_{v_i}(0, \sigma_i)$.	$\sigma_i = 1$, $v_i = 3$	$v_i > 2$ for finite variance (pre-standardization scale); $\text{Var}(q_{ij}) = \sigma_i^2 v_i/(v_i - 2)$.
Seed-based standardization	m	Number of seed genomes.	–	User input.
	$\hat{p}_j$	Empirical allele frequency at site j in seeds.	derived	$\hat{p}_j = \frac{1}{m}\sum_{k=1}^{m} x_{j,k}$.
	$z_{j,k}$	Centered genotype at site j for seed k.	derived	$z_{j,k} = x_{j,k} - \hat{p}_j$.
	$G'_{i,k}$	Centered genetic value for seed k and trait i.	derived	$G'_{i,k} = \sum_{j\in S^{(i)}} q_{ij} z_{j,k}$. $\hat{\mu}_i = \frac{1}{m}\sum_{k=1}^{m} G'_{i,k} = 0$ by construction.
	$\hat{V}_i$	Empirical variance of $G'_{i,k}$ across seeds.	derived	$\hat{V}_i = \frac{1}{m-1}\sum_{k=1}^{m}(G'_{i,k} - \hat{\mu}_i)^2$ (with $\hat{\mu}_i = 0$ under allele-frequency centering).
	$V_{i,\text{target}}$	Target variance of standardized genetic values across seeds.	1	Sets the reference scale on the link scale. $\text{SD}_i = \sqrt{V_{i,\text{target}}} = 1$ by default.
	V_i^*	Expected additive genetic variance when seeds are monomorphic ($\hat{V}_i = 0$).	derived	$V_i^* = \sum_{j\in S^{(i)}} q_{ij}^2 \mathbb{E}[p_j(1-p_j)]$ with $p_j \sim \text{Beta}(1,1), \mathbb{E}[p_j(1-p_j)] = 1/6$.
	r_i	Standardizing coefficient for trait i.	derived	$r_i = \sqrt{V_{i,\text{target}}/\hat{V}_i}$ (if $\hat{V}_i > 0$) or $r_i = \sqrt{V_{i,\text{target}}/V_i^*}$ (if $\hat{V}_i = 0$).
	$\tilde{G}_{i,k}$	Standardized genetic value for seed k and trait i.	derived	$\tilde{G}_{i,k} = r_i\, G'_{i,k}$ (and standardized effect sizes $\tilde{q}_{ij} = r_i q_{ij}$).
Link-slope calibration	$R_{i,\text{OR}}$	Per-SD_i odds ratio for trait i (logit link).	1.5	Used to calibrate α_i under logit.
	$R_{\text{trans,HR}}$	Per-SD_{trans} transmission hazard ratio (cloglog link).	1.5	Used to calibrate α_{trans} under cloglog.
	$R_{\text{drug,clr}}$	Per-SD_{drug} clearance hazard ratio under treatment (cloglog link).	0.67	Used to calibrate α_{drug} under cloglog (< 1 implies reduced clearance / higher survival).
	α_i	GLM link slope for trait $i \in \{\text{trans}, \text{drug}\}$.	derived	Calibrated from per-SD multipliers: • Logit: $\alpha_i = \log(R_{i,\text{OR}})/\text{SD}_i$; • Cloglog: $\alpha_{\text{trans}} = \log(R_{\text{trans,HR}})/\text{SD}_{\text{trans}}$, $\alpha_{\text{drug}} = -\log(R_{\text{drug,clr}})/\text{SD}_{\text{drug}}$.

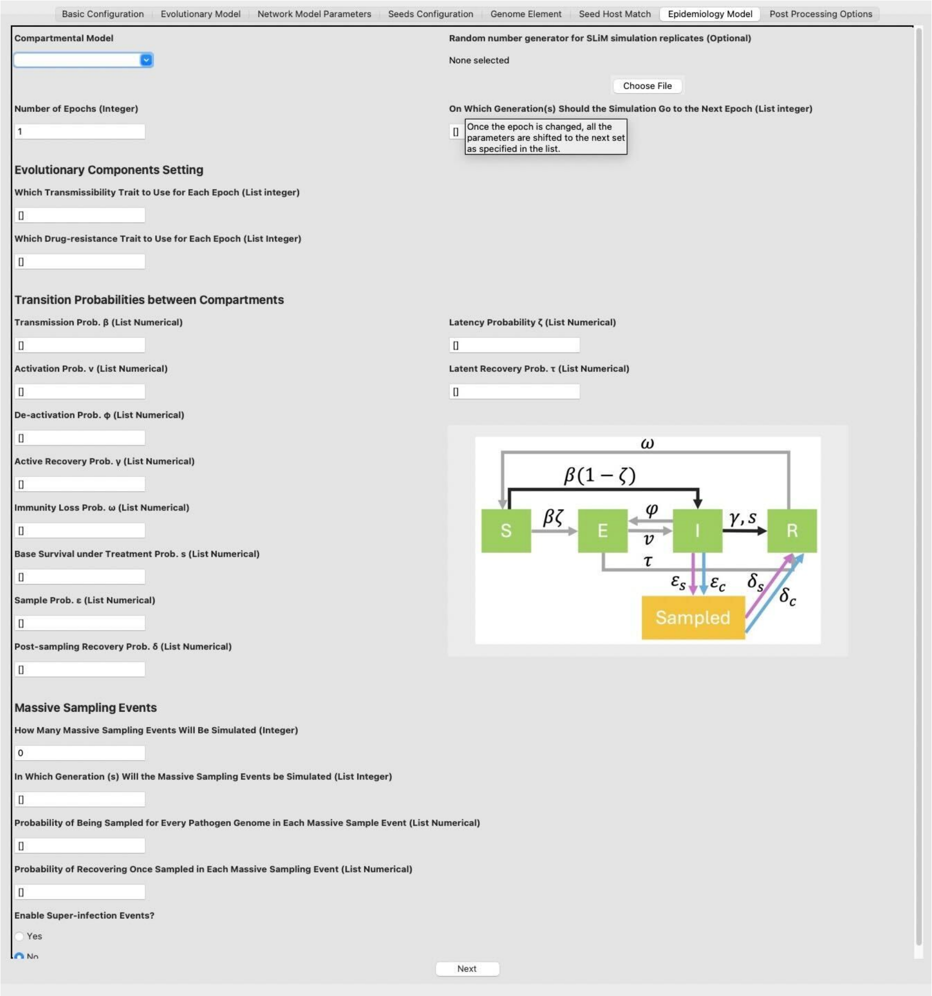
Table S5. Fixed parameters used in runtime profiling simulations across host population sizes (Section 3.2) using the SARS-CoV-2 reference genome. Sampling probabilities are listed in Table S6.

Simulation Parameters	**Values**
Genome length	29,891 nt
Number of causal sites for transmissibility trait	525
Mutation rate per site per tick (μ)	1.8×10^{-6}
Host contact network	Erdős–Rényi network
Number of ticks	730
Average contact degree	10
Baseline per-contact transmission probability (β)	0.04 (host-size benchmarks); 0.01–0.06 (β-sweep; Figure S7B)
Baseline immune-mediated recovery probability (γ)	0.03
Activation probability (ν)	0.30

Table S6. Sequential sampling probabilities (ϵ_s) per tick for infectious hosts in runtime profiling experiments (Section 3.2). Each table entry reports the per-tick sequential sampling probability ϵ_s for the corresponding host population size N and target maximum expected sample size per tick $N\epsilon_s \in \{1, 5, 10\}$. This quantity serves as a proxy for sampling-related computational cost and provides an upper bound on sampled pathogen genomes per tick, attained when all N hosts are infectious and each infectious host carries a single genome. Because sequential sampling is applied only to infectious hosts and the number of infectious hosts is at most N, realized sample counts per tick are typically much lower and depend on epidemic dynamics and outbreak size.

		Expected (maximum) sample size per tick		
		1	**5**	**10**
Host population size	**10000**	0.0001	0.0005	0.001
	25000	0.00004	0.0002	0.0004
	50000	0.00002	0.0001	0.0002
	75000	0.000013	0.000067	0.00013
	100000	0.00001	0.00005	0.0001

SI Figures



Compartmental Model

Random number generator for SLiM simulation replicates (Optional)

None selected

Number of Epochs (Integer)

1

On Which Generation(s) Should the Simulation Go to the Next Epoch (List integer)

[]

Once the epoch is changed, all the parameters are shifted to the next set as specified in the list.

Evolutionary Components Setting

Which Transmissibility Trait to Use for Each Epoch (List integer)

[]

Which Drug-resistance Trait to Use for Each Epoch (List Integer)

[]

Transition Probabilities between Compartments

Transmission Prob. β (List Numerical)

[]

Latency Probability ζ (List Numerical)

[]

Activation Prob. v (List Numerical)

[]

Latent Recovery Prob. τ (List Numerical)

[]

De-activation Prob. ϕ (List Numerical)

[]

Active Recovery Prob. γ (List Numerical)

[]

Immunity Loss Prob. ω (List Numerical)

[]

Base Survival under Treatment Prob. s (List Numerical)

[]

Sample Prob. ε (List Numerical)

[]

Post-sampling Recovery Prob. δ (List Numerical)

[]

Massive Sampling Events

How Many Massive Sampling Events Will Be Simulated (Integer)

0

In Which Generation (s) Will the Massive Sampling Events be Simulated (List Integer)

[]

Probability of Being Sampled for Every Pathogen Genome in Each Massive Sample Event (List Numerical)

[]

Probability of Recovering Once Sampled in Each Massive Sampling Event (List Numerical)

[]

Enable Super-infection Events?

○ Yes

● No

Figure S1. Example GUI for epidemiological model configuration. The GUI contains eight tabs for executing pre-simulation modules and configuring the main simulation module. The “Epidemiology Model” tab, shown here, allows users to specify the compartmental model structure and inter-compartment transition probabilities. Specified parameters are saved in a configuration file used by `OutbreakSimulator`. Interactive tooltips provide concise parameter definitions when users hover the cursor over each label.

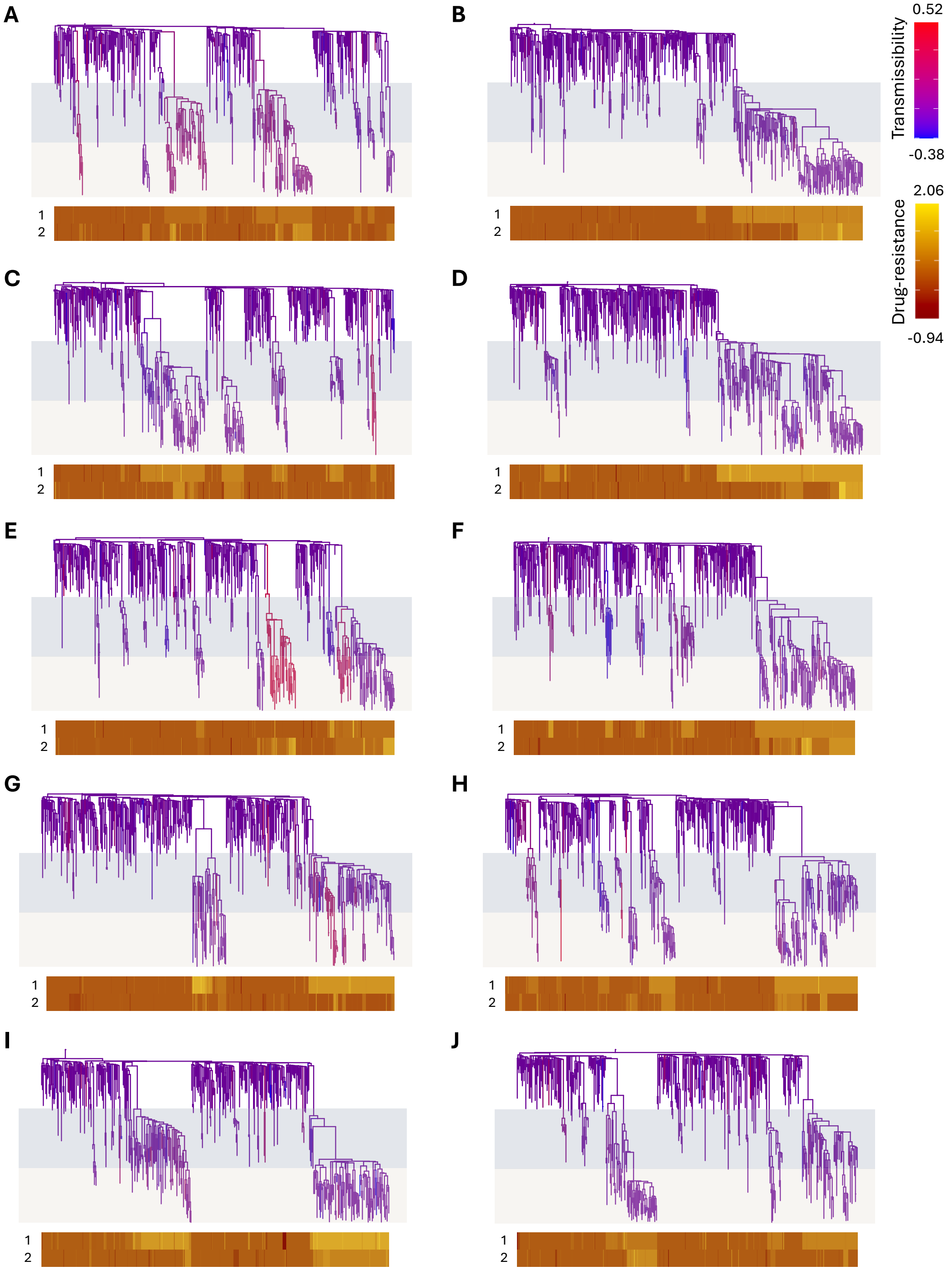


Figure S2. Genealogies of sampled SARS-CoV-2 pathogens illustrating the evolution of transmissibility and associated epi-eco-evo dynamics across ten simulation replicates. Ten randomly selected replicates from the 43 successful replicates in Figure 4A are shown. Time progresses from top (earlier) to bottom (later), and shaded backgrounds denote treatment epochs (drug 1: grey; drug 2: beige). Branches are colored by the transmissibility trait value (standardized link-scale genetic value; blue: lowest, red: highest), and two-row tip-aligned heatmaps indicate pathogen-specific drug-resistance trait values (brown: lowest, yellow: highest) for drug 1 (row 1) and drug 2 (row 2). Across replicates, lineages with higher transmissibility and drug-resistance trait values consistently achieved greater prevalence under the corresponding treatment regimes. Simulation details are provided in Sections 3.1.1 and S4.1.

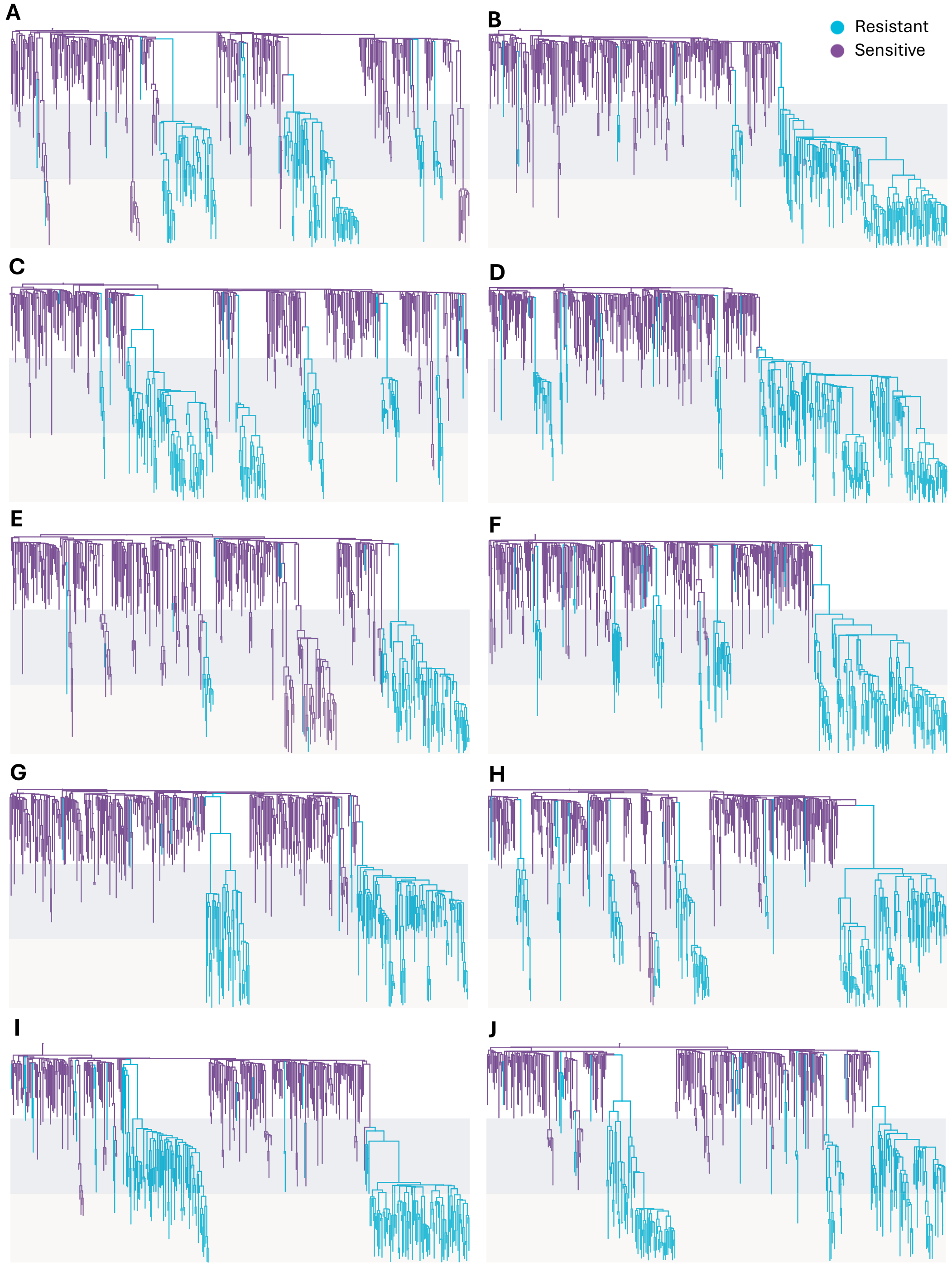


Figure S3. Genealogies of sampled SARS-CoV-2 pathogens illustrating the emergence of resistance to the first drug and associated epi-eco-evo dynamics across ten simulation replicates. These genealogies are identical to those in Figure S2, but branches are colored by resistance status to the first drug (turquoise: resistant, purple: sensitive). For visualization, resistance status is defined by the sign of the standardized link-scale drug-resistance genetic value, $G_{\mathrm{drug}} > 0$: a branch is labeled resistant if $G_{\mathrm{drug}} > 0$, implying its per-tick survival under treatment exceeds the baseline survival s (at $G_{\mathrm{drug}} = 0$); otherwise, it is labeled sensitive.

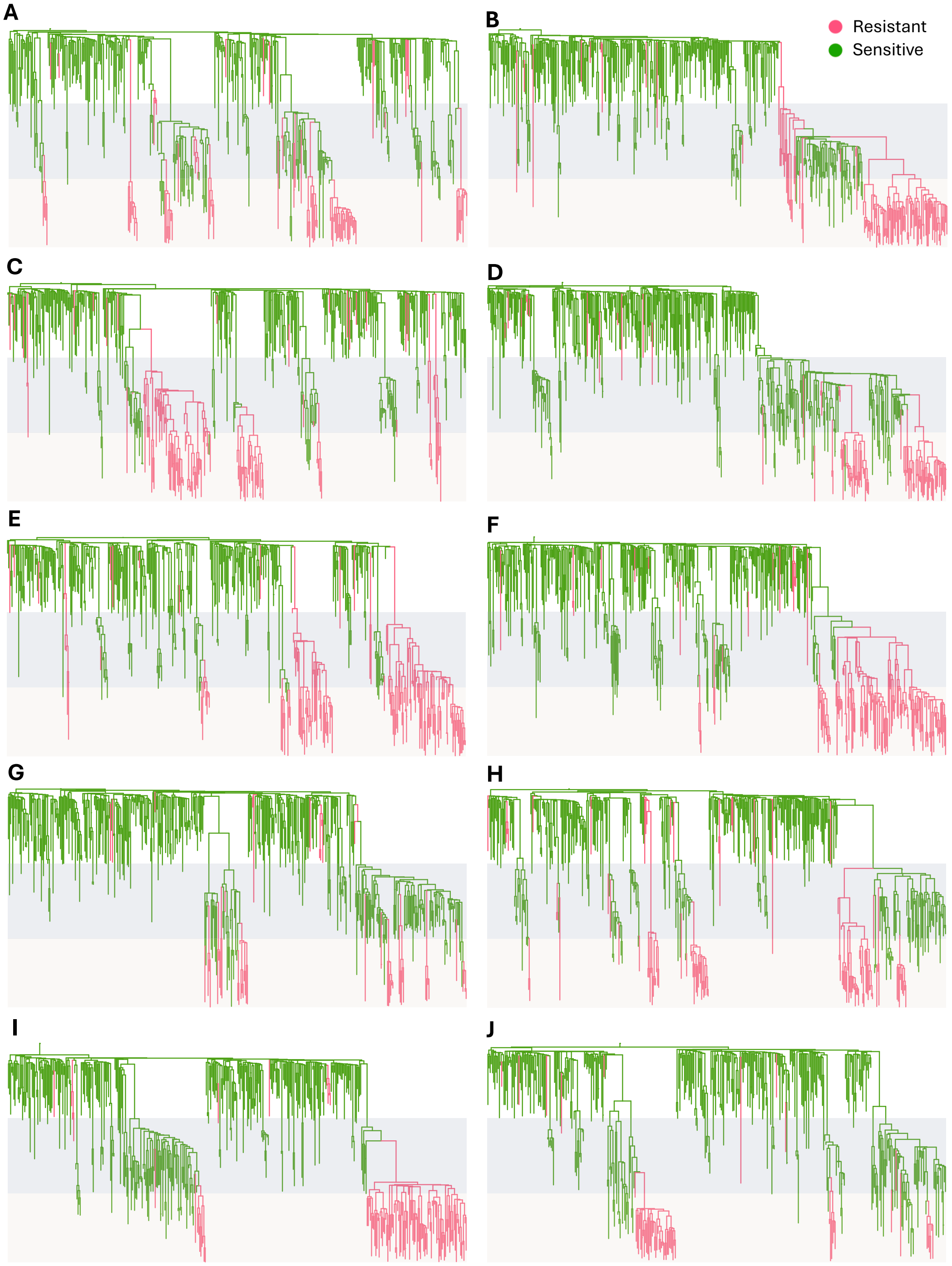


Figure S4. Genealogies of sampled SARS-CoV-2 pathogens illustrating the emergence of resistance to the second drug and associated epi-eco-evo dynamics across ten simulation replicates. These genealogies are identical to those in Figure S2, but branches are colored by resistance status to the second drug (salmon: resistant, green: sensitive). The figure design follows Figure S3.

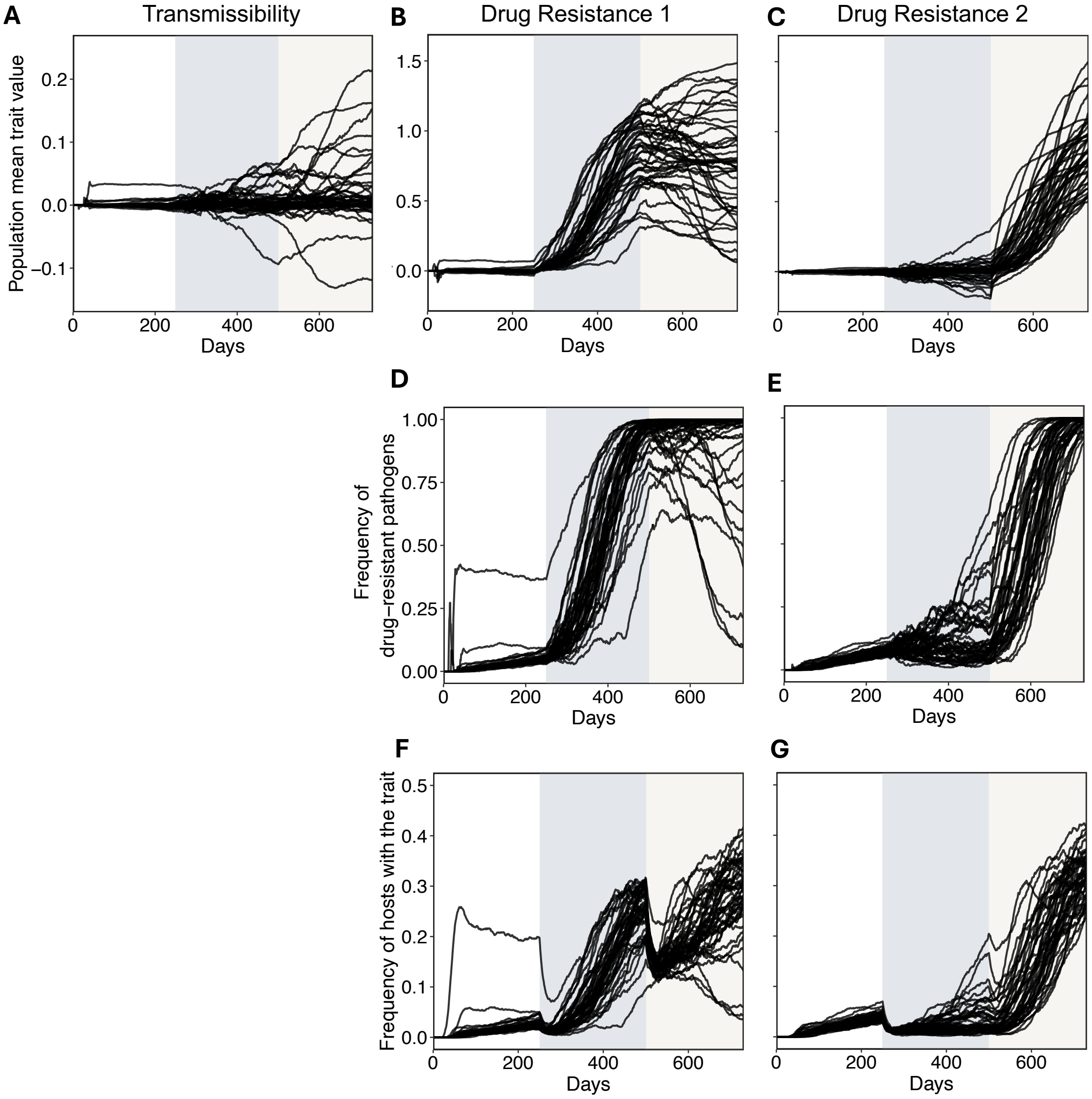


Figure S5. Time trajectories of population mean trait values and frequencies of drug-resistant pathogen lineages and hosts carrying drug-resistant strains. These trajectories are derived from the same 43 successful replicates in Figure 4A. Shaded regions indicate drug treatment periods: first treatment (grey) and second treatment (beige). Black lines represent individual simulation replicates. (A) Population mean transmissibility trait value. (B) Population mean resistance trait value to the first drug. (C) Population mean resistance trait value to the second drug. (D) Proportion of pathogens exhibiting resistance (positive trait value) to the first drug. (E) Proportion of pathogens exhibiting resistance (positive trait value) to the second drug. (F) Proportion of hosts carrying pathogens resistant to the first drug. (G) Proportion of hosts carrying pathogens resistant to the second drug.

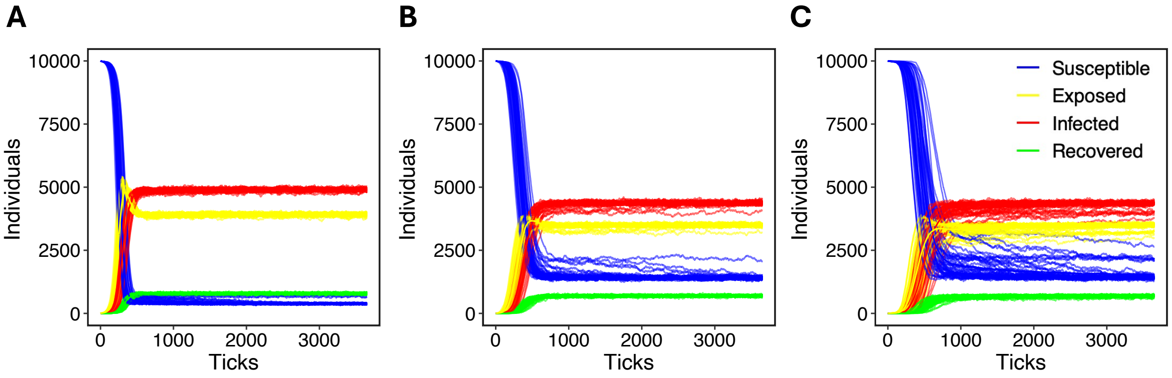


Figure S6. Effects of superspreaders and contact network structure on *Mtb* epidemic dynamics. Each curve represents compartment size trajectories (SEIR) from one simulation replicate, with 50 replicates per scenario. Simulation details are provided in Sections 3.1.2 and S4.2. (A) Erdős–Rényi network with the "Uniform Selection" host–seed matching scheme (Figure 5A). (B) Barabási–Albert network with highly transmissible seeds matched to highly connected hosts and less transmissible seeds matched to less connected hosts (Figure 5B). (C) Barabási–Albert network with highly transmissible seeds matched to less connected hosts, and less transmissible seeds matched to highly connected hosts (Figure 5C).

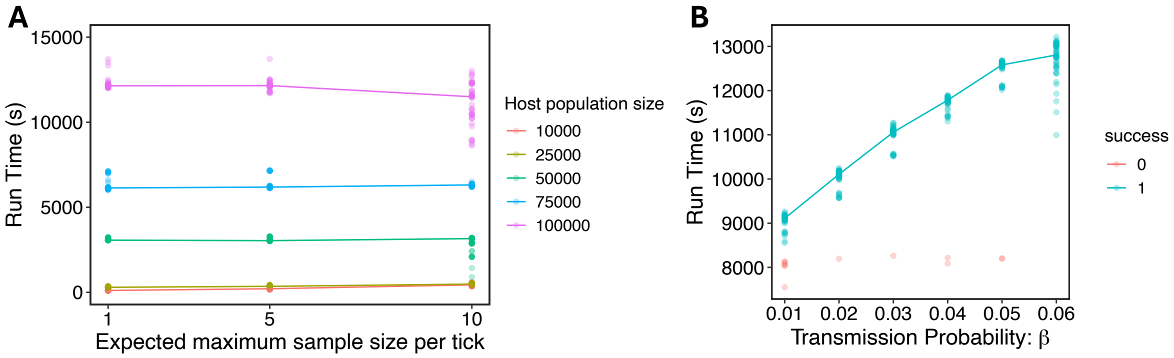


Figure S7. Computational runtime for varying outbreak sizes on a Linux system. Each parameter set was run for 50 replicates. Dots show runtimes for individual replicates; lines indicate the median runtimes across successful replicates (at least one sampled genome). (A) Runtime as host population size and sequential sampling intensity vary, using the same simulation parameters and pipeline as in Section 3.2 and Figure 6. Parameter settings are listed in Tables S5 and S6. Only successful replicates are shown. (B) Runtime as the base transmission probability β varies, with all other parameters fixed as listed in Table S5. Dots are colored by replicate success (teal, 1: at least one sampled genome) versus failure (red, 0: no sampled genomes). Memory usage was limited to 6 GB.

SI Example configuration files

Configuration File S1. Configuration for running `OutbreakSimulator` for the SARS-CoV-2 simulation with two treatment stages. Relevant files generated by the pre-simulation modules must be present in the base working directory `cwdir`. Descriptions of the parameters and simulation processes are provided in Sections 3.1.1 and S4.1.

```
{
  "BasicRunConfiguration": {
    "cwdir": "example1",
    "n_replicates": 50
  },
  "EvolutionModel": {
    "subst_model_parameterization": "mut_rate",
    "n_generation": 730,
    "mut_rate": 1.8e-06,
    "within_host_reproduction": false,
    "within_host_reproduction_rate": 0,
    "cap_withinhost": 1
  },
  "SeedsConfiguration": {
    "seed_size": 1,
    "use_reference": true
  },
  "GenomeElement": {
    "use_genetic_model": true,
    "ref_path": "EPI_ISL_402124.fasta",
    "traits_num": {
      "transmissibility": 1,
      "drug_resistance": 2
    },
    "trait_prob_link": {
      "link": "logit",
      "logit": {
        "alpha_trans": [0.5],
        "alpha_drug": [1, 1.5]
      }
    }
  },
  "NetworkModelParameters": {
    "host_size": 10000
  },
  "EpidemiologyModel": {
    "model": "SEIR",
    "epoch_changing": {
      "n_epoch": 3,
      "epoch_changing_generation": [250, 500]
    },
    "genetic_architecture": {
      "transmissibility": [1, 1, 1],
      "drug_resistance": [0, 1, 2]
    },
    "transition_prob": {
      "S_IE_prob": [0.04, 0.04, 0.04],
      "I_R_prob": [0.03, 0.03, 0.03],
      "R_S_prob": [0.05, 0.05, 0.05],
      "latency_prob": [1, 1, 1],
      "E_I_prob": [0.3, 0.3, 0.3],
      "I_E_prob": [0, 0, 0],
      "E_R_prob": [0, 0, 0],
      "sample_prob": [0.0002, 0.0003, 0.0003],
      "recovery_prob_after_sampling": [0, 0, 0],
      "surviv_prob": [0.01, 0.9, 0.9]
    },
    "massive_sampling": {
      "event_num": 0,
      "generation": [],
      "sampling_prob": [],
      "recovery_prob_after_sampling": []
    },
    "super_infection": false
  },
  "Postprocessing_options": {
    "do_postprocess": true,
    "tree_plotting": {
      "branch_color_trait": 1,
      "heatmap": "drug_resistance"
    },
    "sequence_output": {
      "vcf": true,
      "fasta": false
    }
  }
}
```

Configuration File S2. Configuration for running `OutbreakSimulator` for *Mtb* simulations with various host contact network structures. The network file specifying the contact structure and other relevant files generated by the pre-simulation modules must be present in the base working directory `cwdir`. Descriptions of the parameters and simulation processes are provided in Sections 3.1.2 and S4.2.

```
{
  "BasicRunConfiguration": {
    "cwdir": "example2",
    "n_replicates": 50
  },
  "EvolutionModel": {
    "subst_model_parameterization": "mut_rate",
    "n_generation": 3650,
    "mut_rate": 3.12e-10,
    "within_host_reproduction": false,
    "within_host_reproduction_rate": 0,
    "cap_withinhost": 1
  },
  "SeedsConfiguration": {
    "seed_size": 5,
    "use_reference": false
  },
  "GenomeElement": {
    "use_genetic_model": true,
    "ref_path": "GCF_000195955.2_ASM19595v2_genomic.fna",
    "traits_num": {
      "transmissibility": 1,
      "drug_resistance": 0
    },
    "trait_prob_link": {
      "link": "logit",
      "logit": {
        "alpha_trans": [3.5874],
        "alpha_drug": []
      }
    }
  },
  "NetworkModelParameters": {
    "host_size": 10000
  },
  "EpidemiologyModel": {
    "model": "SEIR",
    "epoch_changing": {
      "n_epoch": 1,
      "epoch_changing_generation": []
    },
    "genetic_architecture": {
      "transmissibility": [1],
      "drug_resistance": [0]
    },
    "transition_prob": {
      "S_IE_prob": [0.003],
      "I_R_prob": [0.008],
      "R_S_prob": [0.05],
      "latency_prob": [1],
      "E_I_prob": [0.01],
      "I_E_prob": [0],
      "E_R_prob": [0],
      "sample_prob": [1e-05],
      "recovery_prob_after_sampling": [0]
    },
    "massive_sampling": {
      "event_num": 0,
      "generation": [],
      "sampling_prob": [],
      "recovery_prob_after_sampling": []
    },
    "super_infection": false
  },
  "Postprocessing_options": {
    "do_postprocess": true,
    "tree_plotting": {
      "branch_color_trait": 1,
      "heatmap": "none"
    },
    "sequence_output": {
      "vcf": false,
      "fasta": false
    }
  }
}
```

References

1. McCullagh P, Nelder JA, 2019. *Generalized linear models*. CRC Press, 2nd edition.

2. Kwee LC, Liu D, Lin X, Ghosh D, Epstein MP, 2008. A powerful and flexible multilocus association test for quantitative traits. *The American Journal of Human Genetics*, 82(2):386–397.

3. Brandes N, Weissbrod O, Linial M, 2022. Open problems in human trait genetics. *Genome Biology*, 23(1):131.

4. Lynch M, Walsh B, et al., 1998. *Genetics and analysis of quantitative traits*, volume 1. Sinauer Sunderland, MA.

5. Piegorsch WW, 1992. Complementary log regression for generalized linear models. *The American Statistician*, 46(2):94–99.

6. Suresh K, Severn C, Ghosh D, 2022. Survival prediction models: an introduction to discrete-time modeling. *BMC Medical Research Methodology*, 22(1):207.

7. Skums P, Mohebbi F, Tsyvina V, Baykal PI, Nemira A, Ramachandran S, Khudyakov Y, 2022. SOPHIE: viral outbreak investigation and transmission history reconstruction in a joint phylogenetic and network theory framework. *Cell Systems*, 13(10):844–856.e4.

8. Hagberg A, Swart PJ, Schult DA, 2008. Exploring network structure, dynamics, and function using NetworkX. *Proceedings of the 7th Python in Science Conference*, SciPy 2008:11–16.

9. Gilbert EN, 1959. Random graphs. *The Annals of Mathematical Statistics*, 30(4):1141–1144.

10. Barabási AL, Albert R, 1999. Emergence of scaling in random networks. *Science*, 286(5439):509–512.

11. Fortunato S, 2010. Community detection in graphs. *Physics Reports*, 486(3-5):75–174.

12. Moshiri N, Ragonnet-Cronin M, Wertheim JO, Mirarab S, 2018. FAVITES: simultaneous simulation of transmission networks, phylogenetic trees and sequences. *Bioinformatics*, 35(11):1852–1861.

13. Zhao S, Magpantay FMG, 2025. Disease transmission on random graphs using edge-based percolation. *Mathematical Methods in the Applied Sciences*, 48(11):11265–11290.

14. Haller BC, Messer PW, 2023. SLiM 4: multispecies eco-evolutionary modeling. *The American Naturalist*, 201(5):E127–E139.

15. Fisher RA, 1923. XXI.—On the dominance ratio. *Proceedings of the Royal Society of Edinburgh*, 42:321–341.

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

17. Meyer HV, Birney E, 2018. Phenotypesimulator: A comprehensive framework for simulating multi-trait, multi-locus genotype to phenotype relationships. *Bioinformatics*, 34(17):2951–2956.

18. Porter HF, O'Reilly PF, 2017. Multivariate simulation framework reveals performance of multi-trait GWAS methods. *Scientific Reports*, 7(1):38837.

19. Tagami D, Bisschop G, Kelleher J, 2024. tstrait: a quantitative trait simulator for ancestral recombination graphs. *Bioinformatics*, 40(6):btae334.

20. Morrison J, 2025. GWASBrewer: An R Package for Simulating Realistic GWAS Summary Statistics. *Genetic Epidemiology*, 49(1):e22594.

21. Paré G, Mao S, Deng WQ, 2017. A machine-learning heuristic to improve gene score prediction of polygenic traits. *Scientific Reports*, 7(1):12665.

22. Mitchell TJ, Beauchamp JJ, 1988. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032.

23. Park T, Casella G, 2008. The Bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686.

24. Gelman A, Jakulin A, Pittau MG, Su YS, 2008. A weakly informative default prior distribution for logistic and other regression models. *Annals of Applied Statistics*, 2(4):1360–1383.

25. Allison PD, 1982. Discrete-time methods for the analysis of event histories. *Sociological Methodology*, 13:61–98.

26. Boyle P, Broadie M, Glasserman P, 1997. Monte Carlo methods for security pricing. *Journal of Economic Dynamics and Control*, 21(8-9):1267–1321.

27. Gaynor RC, Gorjanc G, Hickey JM, 2020. AlphaSimR: an R package for breeding program simulations. *G3 Genes|Genomes|Genetics*, 11(2):jkaa017.

28. Nelder JA, Wedderburn RWM, 1972. Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, 135(3):370–384.

29. Gelman A, 2008. Scaling regression inputs by dividing by two standard deviations. *Statistics in Medicine*, 27(15):2865–2873.

30. Hill WG, Goddard ME, Visscher PM, 2008. Data and theory point to mainly additive genetic variance for complex traits. *PLOS Genetics*, 4(2):e1000008.

31. Balding DJ, Nichols RA, 1995. A method for quantifying differentiation between populations at multi-allelic loci and its implications for investigating identity and paternity. *Genetica*, 96(1):3–12.

32. Jukes TH, Cantor CR, et al., 1969. Evolution of protein molecules. *Mammalian Protein Metabolism*, 3(24):21–132.

33. Mideo N, Alizon S, Day T, 2008. Linking within- and between-host dynamics in the evolutionary epidemiology of infectious diseases. *Trends in Ecology & Evolution*, 23(9):511–517.

34. Didelot X, Gardy J, Colijn C, 2014. Bayesian inference of infectious disease transmission from whole-genome sequence data. *Molecular Biology and Evolution*, 31(7):1869–1879.

35. Rasmussen DA, Volz EM, Koelle K, 2014. Phylodynamic inference for structured epidemiological models. *PLOS Computational Biology*, 10(4):e1003570.

36. Worby CJ, Read TD, 2015. 'SEEDY' (Simulation of Evolutionary and Epidemiological Dynamics): an R package to follow accumulation of within-host mutation in pathogens. *PLOS One*, 10(6):e0129745.

37. Cárdenas P, Corredor V, Santos-Vega M, 2022. Genomic epidemiological models describe pathogen evolution across fitness valleys. *Science Advances*, 8(28):eabo0173.

38. Biswas MHA, Paiva LT, de Pinho MDR, et al., 2014. A SEIR model for control of infectious diseases with constraints. *Mathematical Biosciences and Engineering*, 11(4):761–784.

39. Upadhyay RK, Pal AK, Kumari S, Roy P, 2019. Dynamics of an SEIR epidemic model with nonlinear incidence and treatment rates. *Nonlinear Dynamics*, 96(4):2351–2368.

40. MacPherson A, Louca S, McLaughlin A, Joy JB, Pennell MW, 2022. Unifying phylogenetic birth–death models in epidemiology and macroevolution. *Systematic Biology*, 71(1):172–189.

41. Stadler T, Kühnert D, Bonhoeffer S, Drummond AJ, 2013. Birth–death skyline plot reveals temporal changes of epidemic spread in HIV and hepatitis C virus (HCV). *Proceedings of the National Academy of Sciences*, 110(1):228–233.

42. Gavryushkina A, Welch D, Stadler T, Drummond AJ, 2014. Bayesian inference of sampled ancestor trees for epidemiology and fossil calibration. *PLOS Computational Biology*, 10(12):e1003919.

43. Stadler T, 2009. On incomplete sampling under birth–death models and connections to the sampling-based coalescent. *Journal of Theoretical Biology*, 261(1):58–66.

44. Zhukova A, Hecht F, Maday Y, Gascuel O, 2023. Fast and accurate maximum-likelihood estimation of multi-type birth–death epidemiological models from phylogenetic trees. *Systematic Biology*, 72(6):1387–1402.

45. Celentano M, DeWitt WS, Prillo S, Song YS, 2025. Exact and efficient phylodynamic simulation from arbitrarily large populations. *Proceedings of the National Academy of Sciences*, 122(20):e2412978122.

46. Zhou P, Yang XL, Wang XG, Hu B, Zhang L, Zhang W, Si HR, Zhu Y, Li B, Huang CL, *et al.*, 2020. A pneumonia outbreak associated with a new coronavirus of probable bat origin. *Nature*, 579(7798):270–273.

47. Khare S, Gurry C, Freitas L, B Schultz M, Bach G, Diallo A, Akite N, Ho J, TC Lee R, Yeo W, *et al.*, 2021. GISAID's role in pandemic response. *China CDC Weekly*, 3(49):1049–1051.

48. Snijder EJ, Bredenbeek PJ, Dobbe JC, Thiel V, Ziebuhr J, Poon LL, Guan Y, Rozanov M, Spaan WJ, Gorbalenya AE, *et al.*, 2003. Unique and conserved features of genome and proteome of SARS-coronavirus, an early split-off from the coronavirus group 2 lineage. *Journal of Molecular Biology*, 331(5):991–1004.

49. Du L, He Y, Zhou Y, Liu S, Zheng BJ, Jiang S, 2009. The spike protein of SARS-CoV—a target for vaccine and therapeutic development. *Nature Reviews Microbiology*, 7(3):226–236.

50. Wheeler DL, Barrett T, Benson DA, Bryant SH, Canese K, Chetvernin V, Church DM, DiCuccio M, Edgar R, Federhen S, *et al.*, 2007. Database resources of the national center for biotechnology information. *Nucleic Acids Research*, 36(suppl_1):D13–D21.