

STRUCTURED GENEALOGY PROCESSES

QIANYING LIN, AARON A. KING, AND EDWARD L. IONIDES

ABSTRACT. This is a followup paper on Markov Genealogy Process. We construct a continuous time Markov Process, called Structure Genealogy Process, to include structures in the focal population with a moderate adjustments in the setting of individuals, genealogies, and genealogy processes. We derive the exact expression of likelihood of a structured genealogy and develop simulation-based algorithms to conduct statistical inferences. Numerical simulation and real-world examples are included.

1. Introduction.

2. Mathematical settings.

Markov genealogy process. A population process $\mathcal{X}_t \in \mathbb{Z}^d$ as a non-explosive Markov jump process is defined, where $t \in \mathbb{R}_+$ indicates the time and the population is divided into $J \in \mathbb{N}$ demes, with initial-state distribution $p_0(x)$. We then define the jump rate functions $\alpha_u(t, x) \in \mathbb{R}_+$, where $u, x \in \mathbb{Z}^d$ are the jump event and states, respectively. Thus, the population process \mathcal{X}_t is a stochastic process by defining a sequence of jumps, $\omega = (t_k, u_k, n_k)_{k=1}^K$, where $K \in \mathbb{N} \cup \{\infty\}$ is the total number of jumps and for the k -th jump, $t_k \in \mathbb{R}_+$ is the time, $u_k \in \mathbb{Z}^d$ is the event type, and $n_k \in \mathbb{N}$ is the label of the individual who conducts u_k . The full construction can be referred to [King et al. \(2021\)](#). We define the following functions on $\omega \in \Omega$:

$$T_k(\omega) := t_k, \quad U_k(\omega) := u_k, \quad N_k(\omega) := n_k, \quad K(\omega) := K \quad (1)$$

and

$$T(\omega) := (T_k(\omega))_{k=0}^{K(\omega)}, \quad U(\omega) := (U_k(\omega))_{k=0}^{K(\omega)}, \quad N(\omega, j) := (N_k(\omega))_{k=0}^{K(\omega)}. \quad (2)$$

Births. Within a structured infective population, the parent and the child, at a given birth event, can be in different demes. That is, an individual in Deme i can give birth to a newborn individual in Deme j , where $i, j = 1, \dots, J$. We then define the birth function $B_{ij} : \mathbb{Z}^d \rightarrow \mathbb{N}$ for any $i, j = 1, \dots, J$. The set of birth events where an individual in Deme i gives birth to n individuals in Deme j is $\mathbf{B}_{ij} := B_{ij}^{-1}(\{1\})$. Trivially, the set of birth events in Deme j is $\mathbf{B}_j := \bigcup_i \mathbf{B}_{ij}$.

Transitions. Infective demes are interchangeable and we define the transition function from Deme i to j for any $i \neq j$ and $i, j = 1, \dots, J$ as $M_{ij} : \mathbb{Z}^d \rightarrow \mathbb{N}$ to represent the number of infectives transiting to Deme j from Deme i . Thus the set of transition is $\mathbf{M}_{ij} := M_{ij}^{-1}(\{1\})$ since we assume single transition at a single transition event. Similarly, we disallow coinciding transition events and other events.

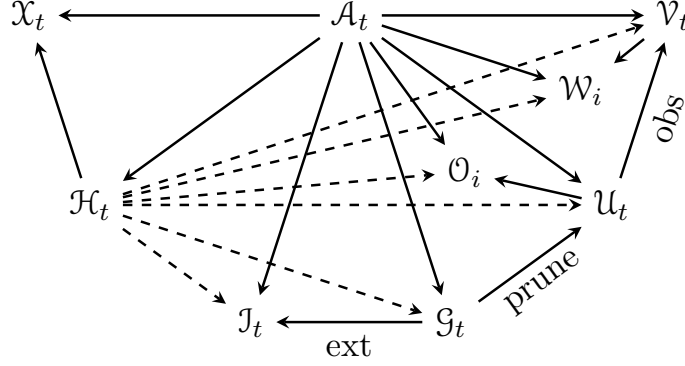


FIGURE 1. Relations among the various Markov processes discussed in the paper. Deterministic maps are indicated with solid arrows; random maps are shown as dashed arrows. All the maps shown commute. \mathcal{X}_t is the *population process*, a model of the dynamics of some system, which we take as a starting point. \mathcal{H}_t is the *history process*, which records the full history of \mathcal{X}_t . \mathcal{J}_t is the *inventory process*: at each time t , \mathcal{J}_t is an inventory of all extant individuals in the population, each of which has a globally unique name. \mathcal{G}_t is the *genealogy process*, which captures the precise genealogical relationships among all individuals in \mathcal{J}_t , as well as among any samples that have been taken from the population. \mathcal{V}_t is the *visible genealogy process*, which is \mathcal{G}_t pruned so that only relationships among samples remain. Finally \mathcal{W}_i is the *embedded chain of the visible genealogy process*, which is \mathcal{V}_{s_i} , s_i being the time of the i -th sample. All of these processes can be obtained via deterministic procedures applied to the *master process* \mathcal{A}_t , as described in the text.

Samples. We assume samples are taken serially and no births or deaths are coincided. We then define the sampling function on the probability space of the population process \mathcal{X}_t : $G : \mathbb{Z}^d \rightarrow \mathbb{N}_0$ is the number of samples taken at event u , then the set of sampling events is $\mathbf{G} := G^{-1}(\{1\})$. Of course, the non-coincidence assumption can be easily relaxed while it's not the focus in this paper.

Deaths and population size. The population, the set of infected individuals in the context of epidemiology, is structured, categorized into interchangeable demes. We can then define population size function and death functions in Deme j , respectively, as $I_j, D_j : \mathbb{Z}^d \rightarrow \mathbb{N}$ for any $j = 1, \dots, J$.

$$\alpha_u(t, x) > 0 \implies I_j(x + u) - I_j(x) = \sum_i B_{ij}(u) + \sum_i M_{ij}(u) - D_j(u) - \sum_i M_{ji}(u), \quad (3)$$

for all $x, u \in \mathbb{Z}^d$. We disallow simultaneous death at a single death event at this stage and define the set of death events in Deme j as $\mathbf{D}_j := D^{-1}(\{1\})$. We also insist, birth and death events don't coincide for any demes, i, j , that is, $\mathbf{B}_j \cap \mathbf{D}_i = \emptyset$. Eventually, we can have the function for total population size by letting $I(x) := \sum_{j=1}^J I_j(x)$ and it's trivial that

$$\alpha_u(t, x) > 0 \implies I(x + u) - I(x) = \sum_{i,j} B_{i,j}(u) - \sum_j D_j(u), \quad (4)$$



FIGURE 2.

3. Examples.

Some commonly used compartmental models in epidemiology are in fact structured, either explicitly or implicitly. Here, we demonstrate below how these representatives fit within the population processes we described previously.

SEIR model. The SEIR model is a simple extension of the most basic SIR model, by adding the latent infected compartment “E” into the system, where susceptible individual first become “exposed” when being infected, then turn into being infectious and eventually recover or get removed from the system. The infected population, which we are mostly interested in, is divided into two demes: (1) “E” represents those that are not able to spread the disease though infected and (2) “I” are those that are infectious. To fit within the aforementioned population process, we take $d = 4$ so that the state vector is $\mathcal{X} = (s, e, i, g)$, where s, e, i, g are the number of susceptibles, exposeds, infectives, and the cumulative number of genomic samples collected, respectively. We then summarize four types of jumps, with rate functions:

$$\alpha_u = \begin{cases} b(t) s i, & u = (-1, 1, 0, 0), s > 0, i > 0 \\ \sigma e, & u = (0, -1, 1, 0), e > 0, \\ \gamma i, & u = (0, 0, -1, 0), i > 0 \\ \psi(t, s, e, i, g), & u = (0, 0, 0, 1), e + i > 0 \\ 0, & \text{otherwise.} \end{cases}$$

The above shows the rates of being infected, progression from exposed to infectious, recovery, and sampling, respectively. As in [King et al. \(2021\)](#), the transmission $b(t)$ is time-dependent, and the sampling rate ψ is any function, as long as the constraints in Section 2 are satisfied. Therefore, $I_1(\mathcal{X}) = e$, $I_2(\mathcal{X}) = i$, $I(\mathcal{X}) = e + i$, $\mathbf{B}_{21} = \{(-1, 1, 0, 0)\}$, $\mathbf{M}_{12} = \{(0, -1, 1, 0)\}$, $\mathbf{D}_2 = \{(0, 0, -1, 0)\}$, $\mathbf{G} = \{(0, 0, 0, 1)\}$, and $\mathbf{B}_{11} = \mathbf{B}_{12} = \mathbf{B}_{22} = \mathbf{M}_{21} = \mathbf{D}_1 = \emptyset$.

SI²R model. We can customize the state and event vector to fit a complex system. In SI²R model, we have two different while interchangeable demes of infectious individuals, the *per capita* transmissibility therefore in the first deme is $b_1(t)$ whilst in the second deme is $b_2(t)$. Transition rates between Deme i and j are also defined: Deme i infection turns into Deme j at rate r_{ij} for $i, j = 1, 2$ and $i \neq j$. Furthermore, we also suppose that, once being infected, a susceptible individual becomes Deme 1 infectious with probability ρ while Deme 2 infectious with probability $1 - \rho$. To specify each event and the number of individuals involved, we define a relatively complex state vector by setting $d = 10$ and $\mathcal{X} = (s, i_1, i_2, b_{11}, b_{12}, b_{21}, b_{22}, m_{12}, m_{21}, g)$, for s, i_1 , and i_2 being the population size of the susceptible, the infectives in Deme 1, and those in Deme 2, respectively, and $b_{11}, b_{12}, b_{21}, b_{22}, m_{12}$, and m_{21} being the cumulative number of events of parents in Deme 1 giving to children in Deme 1 and 2, parents in Deme 2 giving birth to children in Deme 1 and 2, individuals transiting from Deme 1 to Deme 2, and vice versa, and sampling, respectively. In this case, we can summarize the event rates as:

$$\alpha_u = \begin{cases} b_1(t) s i_1 \rho, & u = (-1, 1, 0, 1, 0, 0, 0, 0, 0), s > 0, i_1 > 0 \\ b_1(t) s i_1 (1 - \rho), & u = (-1, 0, 1, 0, 1, 0, 0, 0, 0), s > 0, i_1 > 0 \\ b_2(t) s i_2 \rho, & u = (-1, 1, 0, 0, 0, 1, 0, 0, 0), s > 0, i_2 > 0 \\ b_2(t) s i_2 (1 - \rho), & u = (-1, 0, 1, 0, 0, 0, 1, 0, 0), s > 0, i_2 > 0 \\ r_{12} i_1, & u = (0, -1, 1, 0, 0, 0, 0, 1, 0), i_1 > 0 \\ r_{21} i_2, & u = (0, 1, -1, 0, 0, 0, 0, 0, 1), i_2 > 0 \\ \gamma i_1, & u = (0, -1, 0, 0, 0, 0, 0, 0, 0), i_1 > 0 \\ \gamma i_2, & u = (0, 0, -1, 0, 0, 0, 0, 0, 0), i_2 > 0 \\ \psi(t, s, i_1, i_2, g), & u = (0, 0, 0, 0, 0, 0, 0, 0, 1), i_1 + i_2 > 0 \\ 0, & \text{otherwise.} \end{cases}$$

The nine non-zero rates above represent those of infection into Deme 1 infected by individuals in Deme 1, infection into Deme 2 infected by individuals in Deme 1, infection in Deme 1 infected by individuals in Deme 2, infection into Deme 2 infected by individuals in Deme 1, transition from Deme 1 to Deme 2, transition from Deme 2 to Deme 1, recovery from Deme 1, recovery from Deme 2, and sampling, respectively. We then summarize the population functions and sets as follows:

$$\begin{aligned} I_1(\mathcal{X}) &= i_1, & I_2(\mathcal{X}) &= i_2, & I(\mathcal{X}) &= i_1 + i_2, \\ \mathbf{B}_{11} &= \{(-1, 1, 0, 1, 0, 0, 0, 0, 0)\}, & \mathbf{B}_{12} &= \{(-1, 0, 1, 0, 1, 0, 0, 0, 0)\}, \\ \mathbf{B}_{21} &= \{(-1, 1, 0, 0, 0, 1, 0, 0, 0)\}, & \mathbf{B}_{22} &= \{(-1, 0, 1, 0, 0, 0, 1, 0, 0)\}, \\ \mathbf{M}_{21} &= \{(0, -1, 1, 0, 0, 0, 0, 1, 0)\}, & \mathbf{M}_{12} &= \{(0, 1, -1, 0, 0, 0, 0, 0, 1)\}, \\ \mathbf{D}_1 &= \{(0, -1, 0, 0, 0, 0, 0, 0, 0)\}, & \mathbf{D}_2 &= \{(0, 0, -1, 0, 0, 0, 0, 0, 0)\}, \\ \mathbf{G} &= \{(0, 0, 0, 0, 0, 0, 0, 0, 1)\}. \end{aligned}$$

Migration. One special case of the SI²R model is that with one-time migration or introduction. In other words, an infectious individual from another population join our interested population at a certain point of time τ and then stay and spread the disease. In this case, if this specific individual is defined as Deme 1 while other infectious individuals as Deme 2, the transition rate r_{12} is a Dirac delta function $\delta(t)$, where $\delta(t) = \infty$ for $t = \tau$ and $\delta(t) = 0$ for $t \neq \tau$, and $r_{21} = 0$ all the time.

Superspreading. The definition of superspreading is ambiguous in epidemiology. Some researchers view it as a certain individual who has extremely strong transmissibility with high reproduction rate and call this individual *superspreader*; others avoid to individualize this phenomenon and consider it as an event where a cluster of infections appears in a single event within a small amount of time and define it as a *superspreading event*. The first definition can be represented by the SI²R model with two demes of infectious individuals: one with extremely strong transmissibility and the other with weak transmissibility. That is, $b_1(t) \gg b_2(t)$ if Deme 1 is the set of highly infective individuals and Deme 2 is the set of ordinary infective. In terms of the second definition, the involvement of polytomies and multifurcating trees adds an extra layer of complexity in the population model, along with the issues of measurement error and model robustness, such that we would like to postpone the development of a new framework to a future study.

4. Structure genealogy processes.

Inventory process. First, we define a global set $\mathcal{Q} := \mathbb{N}$, which is a collection of infinite unique names. For any Deme j of infective population, we define an inventory $\mathcal{I}_t^j(\omega) = \text{inven}(\omega|_t, j)$, a list to document all the extant infected individuals in Deme j at time t , by their names, where $\omega|_t$ is the sequence of jumps up to time t . On top of the global set \mathcal{Q} and the inven procedure, we can then define six operators:

- (1) $\text{next}(\mathcal{Q})$ returns the minimum integer in \mathcal{Q} , call it n , and update the global set $\mathcal{Q} = \mathcal{Q} \setminus \{n\}$;
- (2) $\text{initialize}(I) := \bigcup_{i=1}^I \{\text{next}(\mathcal{Q})\}$;
- (3) $\text{add}(\mathcal{I}) = \mathcal{I} \cup \{\text{next}(\mathcal{Q})\}$;
- (4) $\text{drop}(\mathcal{I}, n) = \mathcal{I} \setminus \{n\}$; and
- (5) $\text{movein}(\mathcal{I}, n) = \mathcal{I} \cup \{n\}$.

Now the deterministic and recursive procedure inven is described as follows, for ω is a finite jump sequence:

$$\text{inven}(\omega, j) := \begin{cases} \text{initialize}(I_j(\mathcal{X}_0(\omega))), & \text{if } K(\omega) = 0; \\ \text{add}(\text{inven}(\omega^-, j)), & \text{if } K(\omega) > 0 \text{ and } U_K(\omega) \in \mathbf{B}_j; \\ \text{drop}(\text{inven}(\omega^-, j), N_{K(\omega)}(\omega)), & \text{if } K(\omega) > 0 \text{ and } U_K(\omega) \in \mathbf{D}_j; \\ \text{movein}(\text{inven}(\omega^-, j), N_{K(\omega)}(\omega)), & \text{if } K(\omega) > 0 \text{ and } U_K(\omega) \in \bigcup_i \mathbf{M}_{ij}; \\ \text{drop}(\text{inven}(\omega^-, j), N_{K(\omega)}(\omega)), & \text{if } K(\omega) > 0 \text{ and } U_K(\omega) \in \bigcup_i \mathbf{M}_{ji}; \\ \text{inven}(\omega^-, j), & \text{otherwise.} \end{cases} \quad (5)$$

Eventually, we can define $\text{inven}(\omega) := \{\text{inven}(\omega, j)\}_{j=1}^J$ as a list of inventories, such that $\mathcal{I}_t(\omega) := \{\mathcal{I}_t^j(\omega)\}_{j=1}^J$.

Structured genealogies. Types of nodes are defined in a structured genealogy: *Leaves* are the extant members at current time and *Internal Nodes* represent ancestors to a certain subset of extant members, which we further define *Sampling Nodes* and *Transition Nodes*, whose differences will be clear in a moment.

For the sake of distinction and visualization, we define a set of *colored balls*, $(f, n) \in \mathbf{F} \times \mathbb{N}$, where n is the label and f is the color of the ball, and $\mathbf{F} := \{\text{green, black, blue, red, purple}\}$. Therefore, we can define a genealogy *node* as a quadruple (n, t, w, z) , where $n \in \mathbb{N}$ is the node's *name*, $t \in \mathbb{R}_+$ is its time, w is the node's *pocket* (i.e., unordered pair of colored balls), and $z \in \mathbb{N}$ is the node's *deme*. For a node p , we will use $n(p)$, $t(p)$, $w(p)$, and $z(p)$ to denote the name, time, pocket, and deme of p , respectively. We then assign black, green, blue, and purple balls to serve as pointers to the extant population at time t , to the internal nodes, to the samples, and to the transitions between demes, respectively.

Eventually, the current time t and the sequence of nodes, $\mathcal{G} = \left(t, (p_k)_{k=0}^{K-1}\right)$, explicitly define a structured genealogy \mathcal{G} , where $t \in \mathbb{R}_+$, $K \in \mathbb{N}$, and node p_k are subjected to conditions described in [King et al. \(2021\)](#). We then use $t(\mathcal{G})$, $K(\mathcal{G})$, and $p_k(\mathcal{G})$ to represent the time, the length (i.e., the number of nodes), and the k -th node of the genealogy \mathcal{G} , respectively. We define $P(\mathcal{G})$ as the node sequence of genealogy \mathcal{G} , where $p \in \mathcal{G}$ when p is one node in $P(\mathcal{G})$.

Effect of births, death, sampling and transistions. Births, deaths, samples, and transitions change the topology of a structured genealogy. A death at time t indicates the removal of a leaf in the genealogy and the dismiss of a node. The detailed definition can be found in [King et al. \(2021\)](#).

A birth at time t in a structured genealogy is similar to that in a unstructured one (King et al., 2021): a black ball is selected as the parent and a new node introduced as the child, who is in the same deme as the parent. Let b be randomly selected black ball, named n , which is the parent, and there exists a node $p \in \mathcal{G}$ holding $b \in w(p) = \{b, b'\}$ where b' is the other ball held by p . Here, we assume the parent is from Deme i (i.e., $n \in \mathcal{I}_i$) and the child is born into Deme j . We then can produce a new node $p' = (c, t, \{g, b''\}, i)$, where $c = \text{next}(\mathcal{Q})$, $g = (\text{green}, c)$, $b'' = (\text{black}, c)$, and update $\mathcal{I}_j = \mathcal{I}_j \cup \{c\}$. By swapping balls between p and p' , it ends up that $w(p) = \{g, b'\}$ and $w(p') = \{b, b''\}$. We denote the resulting sequence of nodes as $\text{add}(\mathcal{P}(\mathcal{G}), n)$.

A death at time t is defined in the same way as that in King et al. (2021) and we continue to use the notation $\text{drop}(\mathcal{P}(\mathcal{G}), n)$ to denote the resulting sequence of nodes for a death, where an individual with name n is selected randomly to be removed from the population.

A sample at time t results in the participation of a new node with a new blue ball. The procedure is similar to the previous sampling in King et al. (2021) and the birth mentioned above: a black ball, named n , where $n \in \mathcal{I}_j$, and held by p (i.e., $w(p) = \{b, b'\}$), is selected at random. Then a new node $p' = (c, t, \{g, b''\}, j)$ is generated, where $c = \text{next}(\mathcal{Q})$, $g = (\text{green}, c)$, and $b'' = (\text{blue}, q)$. At the end, p exchanges b for g with p' . Here, q is the name of the new blue ball, which is the ordinal number of the transition, $q = |\{b \in w(\mathcal{G}) \mid b \text{ is blue}\}|$. We use the notation $\text{sample}(\mathcal{P}(\mathcal{G}), n)$ as the resulting sequence of nodes.

A transition at time t leads to the inclusion of a new node with a new purple ball. Let b denote the black ball with name n and selected to indicate an extant in the population transits from his/her current deme to a new one. Assuming this selected individual is from Deme i (i.e., $i \in \mathcal{I}_i$) and transiting to Deme j ($j \neq i$), we then initialize a new node by taking $p' = (c, t, \{g, b''\}, i)$, where $c = \text{next}(\mathcal{Q})$, $g = (\text{green}, c)$, and $b'' = (\text{purple}, q)$. Note that, we update $\mathcal{I}_i = \mathcal{I}_i \setminus \{n\}$ and $\mathcal{I}_j = \mathcal{I}_j \cup \{n\}$. Similar to other events, we therefore swap b for g between nodes p and p' and insert p' at the last position of the node-sequence. Here, q is the name of the new purple ball, which is the ordinal number of the transition, $q = |\{b \in w(\mathcal{G}) \mid b \text{ is purple}\}|$. We call this new node p' the *Transition Nodes* in the genealogy and denote the resulting sequence of nodes by $\text{transition}(\mathcal{P}(\mathcal{G}), n)$.

Event times. Given a genealogy \mathcal{G} , the set of *genealogical event times*, $E(\mathcal{G})$, is the set of all node times. Its subsets, including those for different event types and those for different demes, are of interest. In particular, we define

$$\begin{aligned} E^j(\mathcal{G}) &:= \{t(p) \mid p \in \mathcal{G}, z(p) \text{ is } j\}, \\ A^j(\mathcal{G}) &:= \{t(p) \mid p \in \mathcal{G}, w(p) \text{ contains a green ball}, z(p) \text{ is } j\}, \\ C^j(\mathcal{G}) &:= \{t(p) \mid p \in \mathcal{G}, w(p) \text{ contains two green balls}, z(p) \text{ is } j\}, \\ L^j(\mathcal{G}) &:= \{t(p) \mid p \in \mathcal{G}, w(p) \text{ contains no green balls}, z(p) \text{ is } j\}, \\ S^j(\mathcal{G}) &:= \{t(p) \mid p \in \mathcal{G}, w(p) \text{ contains a blue ball}, z(p) \text{ is } j\}, \\ Z^j(\mathcal{G}) &:= \{t(p) \mid p \in \mathcal{G}, w(p) \text{ contains a purple ball}, z(p) \text{ is } j\}, \\ D^j(\mathcal{G}) &:= A^j(\mathcal{G}) \cap S^j(\mathcal{G}). \end{aligned}$$

With these definitions, $A^j(\mathcal{G})$ comprises the internal node times of nodes that are derived from nodes in Deme j , $C^j(\mathcal{G})$ is the set of branch times of nodes that are produced by nodes in Deme j , $L^j(\mathcal{G})$ contains the leaf times where the tips are in Deme j , $S^j(\mathcal{G})$ holds the sample times for samples in Deme j , $Z^j(\mathcal{G})$ holds the transition times derived from nodes in Deme j , and $D^j(\mathcal{G})$ is the set of *direct-descent times* derived from samples in Deme j . Trivially, $A(\mathcal{G}) = \cup_j A^j(\mathcal{G})$, $C(\mathcal{G}) = \cup_j C^j(\mathcal{G})$, $L(\mathcal{G}) = \cup_j L^j(\mathcal{G})$,

$S(\mathcal{G}) = \cup_j S^j(\mathcal{G})$, $Z(\mathcal{G}) = \cup_j Z^j(\mathcal{G})$, and $D(\mathcal{G}) = \cup_j D^j(\mathcal{G})$, representing all internal node times, branch times, leaf times, sample times, transition times, and direct-descent times for \mathcal{G} , respectively.

Structured genealogy process. The structured genealogy evolves over time, so we can proceed to define the *structured genealogy process* \mathcal{G}_t now. We define $\mathcal{G}_t(\omega) = (t, \text{geneal}(\omega|_t))$, for the end time $t \in \mathbb{R}_+$ and the sequence of jumps $\omega \in \Omega$. We first initialize the global set of names by letting $\mathcal{Q} = \mathbb{N} \setminus \{1, 2, \dots, I(\mathcal{X}_0(\omega))\}$, then the operation *geneal* is defined recursively as follows for $\omega \in \mathring{\Omega}$:

$$\text{geneal}(\omega) := \begin{cases} (k, 0, \{(\text{green}, k), (\text{black}, k)\}, 0)_{k=0}^{I(\mathcal{X}_0(\omega))}, & \text{if } K(\omega) = 0; \\ \text{add}(\text{geneal}(\omega-), N_{K(\omega)}(\omega)), & \text{if } K(\omega) > 0 \text{ and } U_{K(\omega)} \in \mathbf{B}; \\ \text{drop}(\text{geneal}(\omega-), N_{K(\omega)}(\omega)), & \text{if } K(\omega) > 0 \text{ and } U_{K(\omega)} \in \mathbf{D}; \\ \text{sample}(\text{geneal}(\omega-), N_{K(\omega)}(\omega)), & \text{if } K(\omega) > 0 \text{ and } U_{K(\omega)} \in \mathbf{G}; \\ \text{transition}(\text{geneal}(\omega-), N_{K(\omega)}(\omega)), & \text{if } K(\omega) > 0 \text{ and } U_{K(\omega)} \in \mathbf{M}; \\ \text{geneal}(\omega-), & \text{otherwise;} \end{cases} \quad (6)$$

where $\mathbf{B} := \cup_{i,j} \mathbf{B}_{ij}$, $\mathbf{D} := \cup_j \mathbf{D}_j$, and $\mathbf{M} := \cup_{i,j} \mathbf{M}_{ij}$. A deterministic map, *ext*, can be defined, such that $\mathcal{I}_t(\omega) = \text{ext}(\mathcal{G}_t(\omega))$ for all $\omega \in \Omega$.

Branch color.

Pruned genealogy process. The sampled genealogy contains only relations among genomic samples. In other words, information in a full genealogy about unsampled individuals and deme transitions are unobserved in a sampled genealogy, which we call it a *visible genealogy*. We divide the procedure of deriving a visible genealogy from a full genealogy into two separate steps: *pruning* and *obscuring*. The former drops all unsampled individuals (*i.e.*, black balls) in a genealogy and remains the relationship between samples and deme-transition records of samples, and the latter further erases all deme-transition records of samples and obtains an *obscured genealogy*.

Given a genealogy \mathcal{G} , we define $\mathcal{J} = \text{ext}(\mathcal{G})$. Let $\text{prune}(\mathcal{G})$ be the pruned genealogy derived by applying iterative drop operation to \mathcal{G} for each name $c \in \mathcal{J}$. Specifically, suppose $\mathcal{J}^j = \{n_0^j, \dots, n_{m_j-1}^j\}$ is the inventory for Deme j and $\mathcal{J} = \{\mathcal{J}^j\}_{j=1}^J$, where J is the total number of demes. Then we can define the set of all names in \mathcal{J} as $\mathcal{N} := \bigcup_{j=1}^J \mathcal{J}^j = \{c_0, \dots, c_{m-1}\}$, where $m = \sum_j m_j$. Let $P_0 = P(\mathcal{G})$ and $P_k = \text{drop}(P_{k-1}, c_{k-1})$, for $k = 1, \dots, m$. Then $\text{prune}(\mathcal{G}) := (t(\mathcal{G}), P_m)$. ?? illustrates the pruning procedure.

For $\omega \in \Omega$, we define the *pruned genealogy process*,

$$\mathcal{U}_t(\omega) := \text{prune}(\mathcal{G}_t(\omega)). \quad (7)$$

Note that \mathcal{U}_t so defined is itself a genealogy. ??B depicts one example of \mathcal{U}_t both graphically (as a tree) and diagrammatically (as a node-sequence).

Obscured genealogy process. We first define an operation *reverse* on the sequence of nodes of a genealogy as follows: (1) a random purple ball b with name q is selected such that there exists a node p , where $b \in w(p) = \{b, b'\}$ and b' is the other ball held by p ; (2) there also exists a node p' such that $g = (\text{green}, n(p)) \in w(p') = \{g, b''\}$ where b'' is the other ball held by p' ; (3) swapping balls between p and p' results in $w(p') = \{b', b''\}$ and $w(p) = \{g, b\}$; (4) we delete p from the node sequence. The resulting sequence of nodes is denoted as $\text{reverse}(P(\mathcal{G}), q)$.

Suppose for a genealogy \mathcal{G} , the set of the serial number of the purple balls is $\{0, 1, \dots, q\}$ and the list of inventories is $\mathcal{J} = \{\mathcal{J}^j\}_{j=1}^J$. $\text{obs}(\mathcal{G})$ applies the following steps on \mathcal{G} : (a) let $P_0 = P(\mathcal{G})$ and $P_k = \text{reverse}(P_{k-1}, k)$, for $k = 1, \dots, q$; and (b) for P_q , we update $\mathcal{J} = \bigcup_j \mathcal{J}^j$ and for each node $p \in P(\mathcal{G})$, $z(p) = 0$.

For $\omega \in \Omega$, we define the *obscured genealogy process*,

$$\mathcal{R}_t(\omega) := \text{obs}(\mathcal{G}_t(\omega)). \quad (8)$$

Note that \mathcal{R}_t so defined is a itself a genealogy. ??B depicts one example of \mathcal{R}_t both graphically (as a tree) and diagrammatically (as a node-sequence).

Visible genealogy process.

Node color. There are at most three kinds of nodes in a visible genealogy, distinguished by the contents of their pockets: (a) *green nodes*, which have two green balls in their pockets; (b) *blue nodes*, which have one blue and one green ball; (c) *red nodes*, which have one red and one blue ball. This distinction allows a compact graphical representation of the visible genealogy (??C) and proves useful in deriving our main results as well. Green nodes correspond to *coalescence points* (branch points) in the visible genealogy; red nodes, to leaves; blue nodes correspond to direct-descent events, i.e., samples which are directly ancestral to other samples. Thus, if \mathcal{V} is a visible genealogy, $E(\mathcal{V}) = C(\mathcal{V}) \cup D(\mathcal{V}) \cup L(\mathcal{V})$, where $C(\mathcal{V})$ is the set of times of the green nodes, $D(\mathcal{V})$ contains the times of the blue nodes, and $L(\mathcal{V})$ holds the times of the red nodes. Moreover, $S(\mathcal{V}) = D(\mathcal{V}) \cup L(\mathcal{V})$ is the set of sample times and $A(\mathcal{V}) = D(\mathcal{V}) \cup C(\mathcal{V})$ comprises the *attachment times*, i.e., those points at which a sampled lineage attaches to the visible genealogy subtended by earlier samples.

Deme proportion and lineage count. We can further define the deme proportion function $\mathcal{D}(t, \omega) := \{\mathcal{D}^j(t, \omega)\}_{j=1}^J$, where $\mathcal{D}^j(t, \omega) := \frac{|\mathcal{J}_t^j(\omega)|}{|\mathcal{J}_t(\omega)|}$.

Lineage count function may not be a càdlàg

Given a genealogy, \mathcal{V} , at each time $t \in \mathbb{R}_+$, there are a finite number, $\ell^j(t, \mathcal{V})$, of lineages present for Deme j in the genealogy at that time. Evidently, one has

$$\ell^j(t, \mathcal{V}) := \sum_{e \in C^j(\mathcal{V})} \mathbb{1}_{[e, \infty)}(t) - \sum_{e \in L^j(\mathcal{V})} \mathbb{1}_{[e, \infty)}(t) = \sum_{e \in A^j(\mathcal{V})} \mathbb{1}_{[e, \infty)}(t) - \sum_{e \in S^j(\mathcal{V})} \mathbb{1}_{[e, \infty)}(t), \quad (9)$$

and the lineage count for all demes $\ell(t, \mathcal{V}) = \sum_{j=1}^J \ell^j(t, \mathcal{V})$. With this definition, note that both ℓ^j and ℓ are càdlàg functions of t .

5. Likelihood, filtering equation, and algorithms.

6. Results.

7. Discussion.

References.

King, A. A., Lin, Q., & Ionides, E. L. (2021) Markov genealogy processes. *arXiv* **2105.12730**.