Population Genetics Models for Viral Data with Recombination

Anastasia Ignatieva

Thesis submitted for the degree of
Doctor of Philosophy

Department of Statistics
University of Warwick

October 2021

Contents

List of Figures

List of Tables

List of Abbreviations

ARG	Ancestral recombination graph
BDP	Birth-death process
BGW	Bienaymé–Galton–Watson
CPP	Coalescent point process
MCMC	Markov chain Monte Carlo
MERS-CoV	Middle East respiratory syndrome coronavirus
MRCA	Most recent common ancestor
RP	Reconstructed process
RRP	Reversed reconstructed process
SARS-CoV-2	Severe acute respiratory syndrome coronavirus 2
SFS	Site frequency spectrum
SMC	Sequentially Markov coalescent

Acknowledgements

I am deeply grateful to my supervisors, Paul Jenkins and Jotun Hein, for their insightful guidance and direction.

I would also like to express my appreciation to the Oxford-Warwick Statistics Programme, its students, organisers and funders (the Engineering and Physical Sciences Research Council and the Medical Research Council). It has been a fantastic experience, academically and otherwise. Special thanks go to my cohort pals, James, Natalia, Hector, Maud, Valerie, Bobby, William, Lorenzo, Alan, Emmanuelle, Lucy and Deborah.

I am thankful to everyone who has helped me understand the things mentioned in this thesis through useful conversations. In particular, thank you Jaro, Suzie, and the other indefatigable (and fatigable) attendees of the Warwick population genetics reading group. I also thank Jere Koskela and Mike Steel, for taking the time to read and examine this thesis.

Finally, huge thanks to my family, my friends, and to Ruadhaí, for their unwavering support and the many good times.

Declaration

This thesis is submitted to the University of Warwick in support of my application for the degree of Doctor of Philosophy. It has been composed by myself and has not been submitted in any previous application for any degree.

The work presented (including data generated and data analysis) was carried out by the author, apart from the following:

- C code for KwARG (the program described in Chapter 3) was extended from an earlier version written by Rune Lyngsø, which implemented the algorithm under the infinite sites assumption.

Parts of this thesis have been published by the author:

- Ignatieva et al. (2020);
- Ignatieva et al. (2021b);
- Ignatieva et al. (2021a).

Anastasia Ignatieva

Abstract

Utilising genetic sequencing data to infer the biological parameters that govern the evolution of a population is an important goal of population genetics. Common features of viral evolution mean that widely used modelling assumptions do not hold, such as that the population size is deterministic, that each site of the genome undergoes at most one mutation, or that recombination (individuals inheriting genetic material from two different parent genomes) is absent. In this thesis, models and methods are developed that relax these assumptions, and are thus particularly suited for the analysis of viral sequencing data.

Birth-death process models naturally capture the stochastic variation and exponential growth in population size that is commonly seen, for instance, with intra-host viral populations. I investigate the properties of sample genealogies when the population evolves according to a birth-death process, and focus in particular on the setting of the population size growing to infinity. Through utilising a time rescaling formalism, distributions characterising the process are derived explicitly, and the results show that the genealogy has an interesting structure in this setting.

The reconstruction of possible histories given a sample of genetic data in the presence of recombination is a challenging problem, and existing methods commonly assume the absence of recurrent mutation. I present KwARG, which implements a heuristic-based algorithm for finding plausible genealogical histories that are minimal or near-minimal in the number of posited recombination and recurrent mutation events. Through applying KwARG to reconstruct possible histories for samples of SARS-CoV-2 data, and combining the results with a principled statistical framework for recombination detection, I present evidence of ongoing recombination of SARS-CoV-2 within human hosts.

Chapter 1

Introduction

The field of population genetics was pioneered in the 1920s and 30s by Wright, Fisher and Haldane, whose work set out to quantify how evolutionary forces drive the accumulation of genetic variation in a population. Through the subsequent decades, advances were driven by using stochastic processes and diffusion theory to capture and understand the importance of random factors in molecular evolution. Following the improvement of sequencing technology in the latter half of the 20th century, the focus shifted from the classical aim of understanding the future evolutionary trajectory of an entire population, to the fundamental modern goal of utilising genetic sequencing data to infer the evolutionary history of a sample.

The central object of interest in mathematical population genetics is the genealogy of a sample: a graph that describes the relationships between the sequences by connecting them through shared ancestors in the past. This encodes all of the evolutionary events that have led to the observed present-day genetic diversity, which is created by the complex processes of genetic drift, mutation, recombination and other factors. In practice, the genealogy is unobserved, and methods must leverage the traces of shared history contained within sequencing data to understand the magnitude and interplay of these forces.

The main contributions of this thesis are in introducing methods for inferring genealogies and evolutionary parameters when analysing sequencing data sampled from viral populations. Many of the common modelling assumptions that might hold for other organisms (such as humans or bacteria) break down for viral genomes, such as constant or deterministically changing population size, or the absence of recurrent mutation. In addition, many methods that exist for the analysis of viral data adopt the simplifying assumption that recombination, an important evolutionary process that shapes the genetic diversity of many viruses, is absent. The methods developed in this thesis seek to address these shortcomings.

1.1 Sample genealogies

The notion of a genealogy will be explained through considering the Wright–Fisher model, a classical model of population dynamics. Starting with an initial population of N individuals, they reproduce in discrete non-overlapping generations, keeping the total population size constant, with each new individual selecting a parent uniformly at random from the previous generation and copying its genome. The genome consists of a number of linked sites (nucleotides or genetic loci); with some fixed probability u, each site is not copied exactly from the parent, but undergoes a mutation. An illustration of this population process is given in Figure 1.1.

The population is allowed to evolve for a large number of generations, before a sample of $n < N$ individuals is selected. If the history of the population is known, the genealogy of this sample is easily obtained through tracing the lineages ancestral to the sample backwards in time, to recover the timing of common ancestors and mutation events. An example of such a genealogy is illustrated in Figure 1.1: the sample of three individuals at the present time (outlined in red) is connected by the genealogy shown in red.

Unfortunately, the history of the population is unobserved in most usual settings, so the timing or sequence of evolutionary events in the sample's history must be inferred from the data at hand. There are two main categories of methods for doing this. Model-based inference requires the user to select a generative model, and relies on the estimation of mutation and recombination rates as model parameters. This generally involves integrating over the space of possible histories, which is usually intractable; methods rely on Markov chain Monte Carlo (MCMC) (e.g. Rasmussen et al., 2014) or importance sampling (e.g. Jenkins and Griffiths, 2011), but the problem remains computationally difficult. Moreover, model misspecification can play an important role, for instance when modelling viral evolution over a transmission network, where the relative importance of factors such as geographical structure, social clustering, and the impact of interventions may be difficult to ascertain.

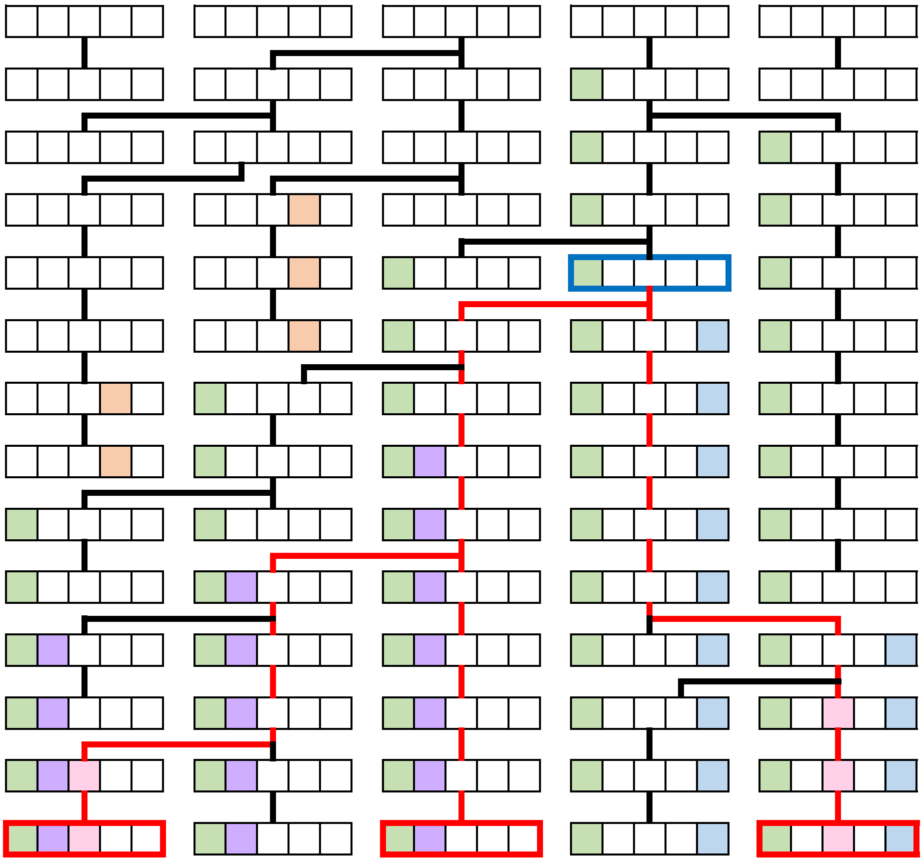

Figure 1.1: Illustration of the Wright–Fisher model, for a population of $N = 5$ individuals with 5 sites. Mutations shown as colours. Sample of $n = 3$ individuals sampled at the present time outlined in red, their genealogy is shown as red edges; MRCA outlined in blue.

The alternative is to constrain the problem by imposing restrictions on the solution space, for instance through searching for the most *parsimonious* solutions that minimise the complexity of the evolutionary history, or implementing heuristic methods to reconstruct the genealogy. In general, this means that event *rates* cannot be inferred, and interpretability of results and quantification of uncertainty are difficult to achieve. However, such methods can be very useful when it is not possible to select an appropriate model, and heuristic methods can be very efficient.

1.2 Models for sample genealogies

1.2.1 The coalescent

It may seem that there are myriad reasonable models that can describe the individual-level dynamics governing the reproductive behaviour of a population, and that the genealogy of a sample will thus depend on the particular choice of model. The groundbreaking work of Kingman (1982b,a) showed that, in fact, often the small-scale details are not important when the population size is large compared to the sample size. For many population models (including Wright–Fisher), under a suitable time rescaling, sample genealogies are described by a particular stochastic process called *the coalescent*. The important modelling assumptions for convergence to the basic coalescent are that the population is selectively neutral, and that there is no population structure (however, these assumptions can be relaxed, leading to modified versions of the coalescent).

1.2.1.1 Standard coalescent

The coalescent arises as a limiting process for the Wright–Fisher model by taking the limit $N \to \infty$ and rescaling time. It is straightforward to show (see e.g. Hein et al., 2004, Section 1.7.2) that the probability that $k \leq n$ sequences have k different parent genomes is

$$1 - \binom{k}{2}\frac{1}{N} + \mathcal{O}(N^{-2}).$$

Then, the probability that a coalescence occurs in a given generation is approximately $\binom{k}{2}/N$. The distribution of the time T_k (measured in generations) at which two of the k sequences had a common ancestor is then

$$\mathbb{P}(T_k \leq j) \approx 1 - \left(1 - \binom{k}{2}\frac{1}{N}\right)^j,$$

for $j = 1, 2, \ldots$ and large N. Measuring time in units of N generations (so $t = j/N$) and taking $N \to \infty$ gives

$$\mathbb{P}(T_k^c \leq t) \approx 1 - \left(1 - \binom{k}{2}\frac{1}{N}\right)^{Nt} \to 1 - \exp\left(-\binom{k}{2}t\right).$$

Looking backwards in time, the number of lineages ancestral to the sample thus decreases by one after an exponentially distributed waiting time with rate $\binom{k}{2}$, when there are $k = n, \ldots, 2$ remaining ancestral lineages. A realisation of the coalescent can be associated with a genealogy in the form of a binary tree on n leaves, by starting with n lineages and merging a randomly selected pair at each event time. Then a point at which two edges merge represents the time when sequences in the sample had a common ancestor, and the root of the tree corresponds to the most recent common ancestor (MRCA) of the entire sample (which is found with probability 1).

If each offspring individual undergoes a mutation with probability u in the Wright–Fisher model, then it is straightforward to show that on the coalescent time scale, mutations will occur on the branches of the genealogy according to a Poisson process with rate $\theta/2$, where $\theta := 2Nu$.

1.2.1.2 Changing population size

The assumption that the population size is constant can be relaxed by modelling the population size, backwards in time, by a positive function $N(t)$, with $N(0) = N$. While for the standard coalescent, time is rescaled in units of N generations, the effects of a changing population size will mean that genealogies will be stretched or compressed according to $N(t)$, as when the population size is low (resp. high), the probability of coalescence increases (resp. decreases). Griffiths and Tavaré (1994) defined the coalescent with variable population size through defining a population size intensity

$$\Lambda(t) = \int_0^t \lambda(u)du,$$

where $\lambda(t) = N/N(t)$. Letting $T^p_n, \ldots, T^p_2$ be the waiting times while there are $n, \ldots, 2$ lineages, the joint density of $(T^p_n, \ldots, T^p_2)$ is given by

$$f(t^p_n, \ldots, t^p_2) = \prod_{j=2}^{n} \binom{j}{2} \lambda(v_j) \exp\left(- \binom{j}{2} (\Lambda(v_j) - \Lambda(v_{j+1})) \right),$$

where $v_{n+1} = 0$ and $v_i = \sum_{j=i}^{n} t^p_i$.

To simulate a realisation of the coalescent with population size $N(t)$, one can:

1. draw waiting times $T^c_n = t^c_n, \ldots, T^c_2 = t^c_2$, where T^c_k is exponentially distributed with rate $\binom{k}{2}$;

2. set $v_k = \sum_{j=k}^{n} t^c_j$, $v_{n+1} = 0$;

3. solve $\Lambda(t^p_k + v_{k+1}) - \Lambda(v_{k+1}) = t^c_k$ for each t^p_k.

The $t^p_n, \ldots, t^p_2$ are then the inter-event waiting times for the coalescent with the given variable population size (Hein et al., 2004, Section 4.2.2).

There are also stochastic formulations of the coalescent, with the population size function allowed to be stochastic (Kaj and Krone, 2003; Parsons et al., 2010).

1.2.2 Birth-death models

In some settings, the dynamics of a population where individuals replicate and die independently of each other may be better modelled as a birth-death process, which, unlike the coalescent, naturally captures the stochasticity and exponential growth of the population size (Boskova et al., 2014; Stadler et al., 2015). The simple linear birth-death process (BDP) studied by Kendall (1948) is a popular neutral population model, in which individuals independently divide at rate λ and die at rate μ. A realisation of this process can be represented as a tree relating the individuals, with bifurcations corresponding to birth events, and terminating branches corresponding to death events. The process models the entire population, creating a birth-death tree such as that shown in the left panel of Figure 1.2, where lineages can go extinct before the present. The genealogy of surviving individuals can then be obtained by pruning these extinct lineages, shown in the middle panel. The process tracing out the genealogy is termed the *reconstructed process (RP)* (Nee et al., 1994).

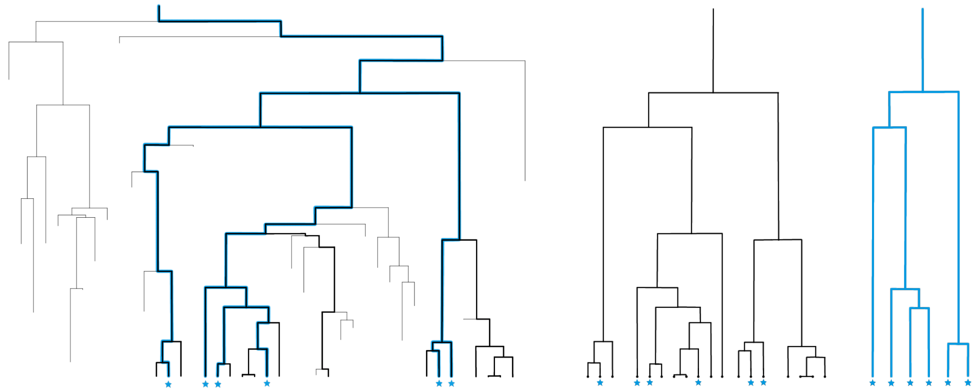

Figure 1.2: Left: birth-death tree with $\lambda = 0.1, \mu = 0.05$ and 18 individuals surviving to the present time. Middle: corresponding genealogy with complete sampling. Right: genealogy with incomplete sampling: each surviving individual is sampled independently with fixed probability $\psi = \frac{1}{3}$; blue stars indicate sampled individuals

1.2.2.1 Reconstructed process

Gernhard (2008a) considered the RP conditioned on having n extant individuals at the present and a given time of origin T. Gernhard noted a correspondence between this conditioned reconstructed process and a point process termed the *coalescent point process (CPP)*, as introduced by Aldous and Popovic (2005) for critical branching processes. The main idea is that, viewing the genealogy backwards in time, the death times of each sampled lineage are i.i.d with a given distribution. Instead of considering each coalescence event sequentially backwards in time as is natural for the coalescent, the genealogy can thus be constructed by drawing n realisations of a particular random variable with support $(0, T]$, giving the event times.

With this CPP formulation, and using the results of Thompson (1975), Gernhard (2008a) then derived the density of the k-th bifurcation time in the RP: first conditioned on T, then integrating it out with an improper uniform $(0, \infty)$ prior.

1.2.2.2 Sampling

Birth-death models additionally differ from the coalescent in that they must explicitly incorporate the method of selecting the sample. Two main sampling regimes have been considered in the literature. Bernoulli-type sampling assumes that each extant individual is sampled independently at the present time with some fixed probability ψ (Stadler, 2009; Wiuf, 2018; Stadler and Steel, 2019; Lambert, 2018). The other case is that of n-sampling, where n individuals are sampled from the full population of size N, which is conditioned to be greater than n (Stadler, 2009; Lambert, 2018).

1.2.2.3 Branching processes

There is also a substantial related body of work concerning Bienaymé–Galton–Watson (BGW) processes, considered in either discrete or continuous time, in which individuals reproduce independently according to a specified offspring distribution (in continuous time, when the number of offspring is either zero or two, this reduces to the special case of a birth-death process). Work on reduced trees (tracing the genealogy of a sample) goes back several decades; Fleischmann and Siegmund-Schultze (1977) showed that the reduced tree associated with a BGW process is itself a time-inhomogeneous BGW process.

Several papers have considered the question of coalescence times for a finite sample (e.g. O'Connell, 1995; Harris et al., 2020; Grosjean and Huillet, 2018; Burden and Soewongsono, 2019). O'Connell (1995) derives an expression for the coalescence time of a sample of size 2, as a fraction of the time since origin of the process. Harris et al. (2020) generalise these results to any sample size, and consider continuous-time BGW processes sampled after time T, with n-sampling, assuming that sampling happens a fixed time after the origin of the process; the exposition of their results explicitly applied to birth-death processes is restricted to the limit $T \to \infty$ for $n = 2$. Burden and Soewongsono (2019) consider the infinite-population limit of near-critical Galton–Watson process, arriving at the Feller diffusion and using this to derive properties of genealogies in the large population limit.

1.3 Recombination

For many species, the evolution of genetic variation within a population is driven by the processes of mutation and *recombination*, in addition to genetic drift. As described above, a typical mutation affects the genome at a single position, and may or may not spread through subsequent generations by inheritance. Recombination, on the other hand, occurs when a new haplotype is created as a mixture of genetic material from two different sources, which can drive evolution at a much faster rate through creating hybrids that carry phenotypic features of both parent genomes. In the context of viruses, recombination is a particularly important factor to consider in the development of treatments and vaccines, as it has the potential to have a drastic impact on the evolution of virulence, transmissibility, and evasion of the host's immunity (Simon-Loriere and Holmes, 2011).

A common restriction on how recombination operates is that of considering *crossover* recombination. Looking back at the Wright–Fisher model, recombination can be incorporated by allowing each offspring to inherit material from not one, but two parent genomes, with some fixed probability r. A recombination breakpoint is then chosen at random between loci, and genetic material to the left of the breakpoint is inherited from one parent, and that to the right from the other parent. This is illustrated in Figure 1.3.

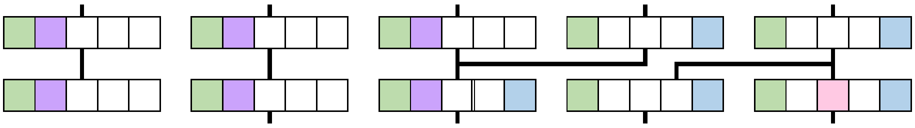

Figure 1.3: Illustration of crossover recombination. Third offsping individual has two parent genomes; material to the left (resp. right) of the recombination breakpoint just after site 3 is inherited from the parent on the left (resp. right).

This offers the simplest model of recombination; a common extension is to consider *gene conversion*, whereby a stretch of the genome (not necessarily including the endpoints) is inherited from one parent, and the rest from another parent genome.

1.3.1 Ancestral recombination graphs (ARGs)

An extension of the coalescent in the presence of recombination introduced by Griffiths and Marjoram (1997) is the *ancestral recombination graph (ARG)*. The topology of a genealogy in the presence of recombination can be a network with loops, rather than necessarily a binary tree. On the coalescent time scale, recombination events occur at the population scaled rate $\rho/2$, where $\rho := 2Nr$.

The exact definition of an ARG varies somewhat in the literature: here, an ARG is defined as a rooted, directed acyclic graph with recombination nodes (of in-degree two and out-degree one) as well as coalescent nodes (of in-degree one and out-degree two), with leaves corresponding to the sampled sequences. The graph is ultrametric, in the sense that the distances between each sampled sequence and the root are equal. The ARG *topology* refers to the graph when the branch lengths are ignored. An example of an ARG topology can be seen in the left panel of Figure 1.4. Mutations are represented as points on the edges, labelled by the sites they affect. Considering the graph backwards in time (from the bottom up), the point at which two edges merge represents the time at which some sequences in the data coalesced, or have found a common ancestor. A point at which an edge splits into two corresponds to a recombination: the parts of the genome to the left and to the right of the breakpoint (whose site number is labelled inside the blue recombination node) are inherited from two different parent particles. The network thus fully encodes the evolutionary events in the history of a sample.

	1	*2*	*3*	*4*
A	0	0	0	1
B	0	1	0	0
C	1	0	0	1
D	1	1	1	0
E	1	1	1	1

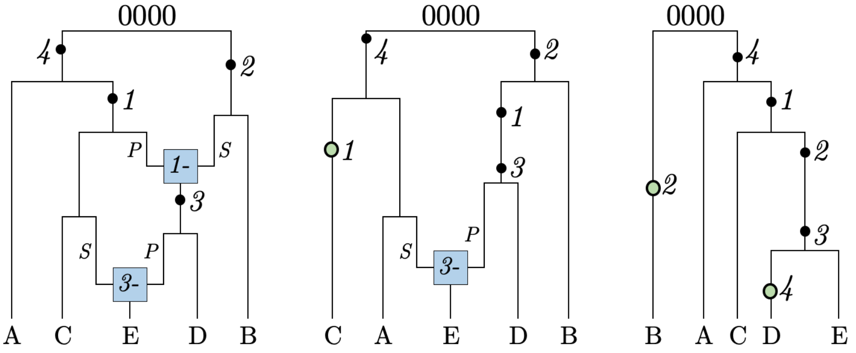

Figure 1.4: Three examples of ARGs. The dataset is shown on the left in binary format, with 0's and 1's corresponding to the ancestral and mutant state at each site, respectively. Mutation events are shown as black dots and labelled by the site they affect; green filled circles correspond to recurrent mutations. Recombination nodes (in blue) are labelled with the recombination breakpoint; material to the right (left) of the breakpoint is inherited from the parent connected by the edge labelled S (P) for "suffix" ("prefix").

Note that if we consider a region of the genome that falls between two recombination breakpoints, the restriction of the ARG to the genealogy of this region is a binary tree. Thus, the ARG can be broken up into a sequence of *local* or *marginal* trees. Wiuf and Hein (1999) introduced the idea of reframing the coalescent with recombination in terms of moving spatially along the sequence, rather than backwards in time. Unlike the backwards-in-time formulation, this spatial process does not have the Markov property, due to dependencies between local trees that are far apart on the genome. The sequentially Markov coalescent (SMC) is an approximation to the coalescent with recombination that imposes a Markov structure between adjacent local trees, introduced by McVean and Cardin (2005). Their simulation results suggest that genealogical events which contradict this Markov structure have relatively low probability, and so the SMC model has very similar properties to the coalescent with recombination.

1.3.2 Recombination detection

The detection of recombination from sequencing data is an important but notoriously difficult problem. Past recombination events can only be detected through considering the observed patterns of mutation, and the effects of recombination are not always obvious.

Crossover recombination can occur anywhere along a sequence, and the breakpoint position is also generally unobserved. Recombination can be undetectable unless mutations appear on specific branches of the genealogy (Hein et al., 2004, Section 5.11). Moreover, recombination events can produce patterns in the data that are indistinguishable from the effects of *recurrent* (or *homoplasic*) mutation (McVean et al., 2002): that is, two or more mutation events in a genealogical history that affect the same locus. In general, many different ARG topologies can generate the same input dataset.

1.3.2.1 Four gamete test

A commonly used assumption on the mutation process is that of *infinite sites*: that each site of the genome has only undergone at most one mutation; then the allele at each site can be denoted 0 (if it is ancestral) or 1 (if it is derived). The *four gamete* test (Hudson and Kaplan, 1985) can then detect the presence of recombination: if all four of the configurations 00, 01, 10, 11 are found at any pair of sites, the data could not have been generated through replication and mutation alone, and at least one recombination event must have occurred between the two corresponding sites. The sites are then termed *incompatible*. The dataset in Figure 1.4 consists of five sequences (rows labelled A-E) with four variable sites (columns labelled 1-4). Sites 3 and 4 contain the configurations 00 (in sequence B), 01 (in sequence C), 10 (in sequence D) and 11 (in sequence E); this means that these two sites are incompatible, and there must have been at least one recombination with a breakpoint between these two sites.

If the infinite sites assumption is violated, the four gamete test no longer necessarily indicates the presence of recombination, as the incompatibilities could instead have been generated through recurrent mutation (McVean et al., 2002). This is illustrated in Figure 1.4, where the same set of sequences is compatible with three different ARG topologies, containing different combinations of recombination and/or recurrent mutation events.

The infinite sites assumption may be reasonable for human DNA, for instance, as the mutation rate per nucleotide is relatively low, so mutations are unlikely to occur multiple times at the same position (in the absence of complicating biological factors). However, the genomes of RNA viruses are much shorter (Flint et al., 2009), and the mutation rate per site much higher, making recurrent mutation a common occurrence. For instance, multiple recurrent mutations were known to have arisen independently on the SARS-CoV-2 genome within months of the beginning of the COVID-19 pandemic (van Dorp et al., 2020a). As recombination is also common in RNA viruses (Simon-Loriere and Holmes, 2011), this complicates the detection of recombination from viral sequencing data.

1.3.2.2 Lower bound on number of recombinations

If recombination is detectable from a sample of sequencing data, a natural question to ask is how many recombination events must have occurred. The minimal number of (crossover) recombination events required to reconstruct a given dataset, denoted R_{min}, cannot be computed exactly in most cases, but several methods exist for computing a lower bound on R_{min}.

The Hudson-Kaplan bound (Hudson and Kaplan, 1985) can be viewed as an extension of the four gamete test. First, all incompatible pairs of sites in the dataset are identified: in Figure 1.5, incompatibilities for the dataset in Figure 1.4 are illustrated by drawing horizontal segments connecting pairs of incompatible sites. The lower bound can then be computed by adding the minimum number of recombination breakpoints so that all pairs of incompatible sites are separated by at least one recombination; this is a version of a classical optimisation problem that can be solved in linear time (Gusfield, 2014, Section 5.2.2.2). Figure 1.5 demonstrates one possible solution, giving a lower bound of two. In this case, the lower bound is equal to R_{min}, as Figure 1.4 demonstrates that it is possible to construct an ARG with two recombinations for this dataset. In general, however, the quality of the bound can be poor, significantly underestimating the number of recombinations that might be required to generate a valid ARG for the data (Gusfield et al., 2007).

$$\begin{array}{ccccc}
 & 1 & 2 & 3 & 4 \\
\text{A} & 0 & 0 & 0 & 1 \\
\text{B} & 0 & 1 & 0 & 0 \\
\text{C} & 1 & 0 & 0 & 1 \\
\text{D} & 1 & 1 & 1 & 0 \\
\text{E} & 1 & 1 & 1 & 1
\end{array}$$

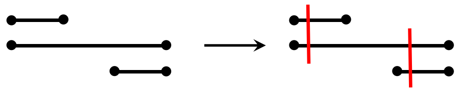

Figure 1.5: Computation of the Hudson-Kaplan lower bound. Horizontal segments link pairs of incompatible sites. Red vertical lines show possible locations of recombination breakpoints.

The haplotype bound (Myers, 2003; Myers and Griffiths, 2003) utilises the observation that each mutation and recombination event can generate at most one new sequence. Let $r(\mathcal{D})$ and $c(\mathcal{D})$ be the number of *distinct* rows and columns of a data matrix $\mathcal{D}$, respectively. Then the haplotype bound is defined to be $r(\mathcal{D}) - c(\mathcal{D}) - 1$. Myers and Griffiths introduced a number of ways to improve the bound; the most potent is through computing the haplotype bound locally on subsets of the data matrix, and using a composition method to then calculate a global bound. There is a trade-off between accuracy and computational time in choosing the number of subsets considered in calculating the local bounds.

1.4 ARG inference

Several methods have been developed that seek to reconstruct ARGs or sequences of local trees from the data. Some, in particular ARGweaver (Rasmussen et al., 2014), aim to infer a distribution over ARGs using computational methods, given a prior model. Other tools, including RENT+ (Mirzaei and Wu, 2017), tsinfer (Kelleher et al., 2019), and Relate (Speidel et al., 2019), instead output one set of local trees for each input dataset, using heuristic approaches. Finally, some methods impose a well-defined optimisation criterion on the ARG reconstruction problem through aiming to identify the most *parsimonious* histories, i.e. minimising the number of recombination events;

these include Beagle (Lyngsø et al., 2005), SHRUB (Song et al., 2005) and SHRUB-GC (Song et al., 2006). There also exist numerous other methods for the inference of recombination (e.g. Martin and Rybicki, 2000; Li and Stephens, 2003; Kosakovsky Pond et al., 2006; Boni et al., 2007) which do not explicitly reconstruct ARGs.

1.4.1 Model-based methods

ARGweaver infers a posterior distribution over ARGs compatible with a given dataset using MCMC techniques, assuming the SMC model and prior values for the population size and evolutionary rates. The main idea is to move spatially along each sequence, and form local trees assuming a Markov dependency structure, sequentially constructing ARGs for k sequences conditional on ARGs for $k-1$ sequences. By removing and re-adding individual sequences (or, more efficiently, entire subtrees) with an operation called *threading*, a Gibbs (resp. Metropolis-Hastings) sampler can be constructed.

ARGweaver infers the timing of each coalescence event as well as the ARG topology. By discretising time and enumerating tree topologies, the SMC is approximated by a model with a finite state space, enabling the use of hidden Markov model methods for threading; the level of time discretisation affects computational efficiency. The program generally runs in reasonable time for samples of under 100 sequences (Hubisz and Siepel, 2020), scaling poorly as the sequence length increases (Speidel et al., 2019), meaning that generally genome-scale data requires splitting into shorter segments and running ARGweaver on each segment separately (Hubisz and Siepel, 2020). Although this approach is more computationally intensive than heuristic methods, the key advantage of inferring a distribution over ARGs is that this allows for a broader range of questions to be addressed: for instance, the timing of genealogical events can be estimated, local trees can be examined explicitly, and uncertainty over the ARG topology and branch lengths can be quantified, allowing for more meaningful interpretation of the results of inference.

1.4.2 Heuristic methods

A number of methods have been developed recently that can utilise the efficiency of heuristics to generate plausible genealogies for very large datasets. The program tsinfer reconstructs a sequence of local trees compatible with an input dataset, utilising the tree sequence data format (Kelleher et al., 2016) to achieve impressive efficiency. First, a set of possible ancestors for the sampled sequences is generated, then a sequence of local trees is constructed connecting these ancestors, and finally the sampled sequences are matched against this tree sequence. The output differs from an ARG in that the location of recombination nodes is not specified (so it is not possible to tell how exactly the recombination event has transformed one local tree to the next), and (with the default settings) inferred trees can contain *polytomies*: nodes with in-degree one and out-degree greater than two.

RENT+ implements efficiency improvements for an earlier version of the algorithm called RENT (Wu, 2009). This also infers a sequence of local trees (rather than a full ARG) compatible with the data, through first constructing local trees for each site of the genome, and then refining this through a series of steps. The key idea of the method is that the local trees are improved *jointly*, with each local tree being affected by updates to neighbouring trees, made according to a set of pre-defined rules.

While tsinfer and RENT+ focus on inferring the local tree topologies (that is, without estimating branch lengths), Relate also incorporates a method for estimating the event times. First, a hidden Markov model approach similar to that of Li and Stephens (2003) is used to calculate a distance matrix that estimates the ordering of coalescence events. A custom algorithm then uses this to reconstruct local trees at each site. Branch lengths are estimated as a separate step, using MCMC with a coalescent prior.

1.4.3 Parsimony

As noted above, a sample of genetic sequences may have many possible histories, with many different corresponding ARGs. The *parsimony* approach to reconstructing ARG topologies given a sample of genetic data focusses on minimising the number of recombination and/or recurrent mutation events. This does not necessarily produce the most biologically plausible histories, but it does provide a useful lower bound on the number of events that must have occurred in the evolutionary pathway generating the sample. Thus, recombination can be detected in the history of a sample by considering whether the most plausible parsimonious solutions contain at least one recombination node.

Crucially, the parsimony approach does not require the assumption of a particular generative model for the data (such as the coalescent with recombination) beyond specifying the types of events that can occur. While this means that mutation and recombination *rates* cannot be inferred, it circumvents the need to specify a detailed model of population dynamics, which can be particularly challenging, for instance when working with viral datasets at the level of between-host transmission. A parsimony-based approach is particularly appropriate when the focus is on interrogating the hypothesis that recombination is present at all. It also allows for the explicit reconstruction of possible events in the history of a sample, and thus allows for the identification of recombinant sequences and discovery of patterns consistent with the effects of sequencing errors.

Previous work on reconstructing histories using parsimony has tackled recombination and recurrent mutation separately. Algorithms for reconstructing minimal ARGs generally make the infinite sites assumption, thus precluding recurrent mutation events, and the goal is to calculate the minimum number of crossover recombinations required to explain a dataset. Even with this constraint, the problem is NP-hard (Wang et al., 2001); exact algorithms are practical only for small datasets (Hein, 1990; Lyngsø et al., 2005), and general methods rely on heuristic approximations (Hein, 1993; Song et al., 2005; Minichiello and Durbin, 2006; Parida et al., 2008; Thao and Vinh, 2019).

1.4.3.1 Exact methods

In the absence of recombination, the goal of the maximum parsimony problem is to calculate the minimum number of recurrent mutations required to reconstruct a tree consistent with the data (denoted P_{min}). The problem of reconstructing maximally parsimonious trees is also NP-hard (Foulds and Graham, 1982); likewise, methods can only handle small datasets or are based on heuristics (Semple and Steel, 2003, Section 5.4). PAUP* (Swofford, 2003) is a program that implements exact (and a number of heuristic) methods for reconstructing parsimonious trees.

In the presence of recombination, Beagle is a method that reconstructs ARGs that are guaranteed to contain R_{min} recombination nodes, using a branch-and-bound approach. Given an input dataset, Beagle constructs sequences of coalescence, mutation and recombination events that could have generated the dataset, proceeding backwards in time until the MRCA is reached. By utilising lower bounds, the search space among possible histories is reduced, by abandoning any partially generated histories that are clearly not the most parsimonious.

1.4.3.2 Heuristic methods

Heuristic algorithms for reconstructing parsimonious ARGs generally implement the same ideas as for exact algorithms, but with various shortcuts to improve efficiency. SHRUB is a program implementing a heuristic algorithm to compute an upper bound on R_{min} and output a single ARG compatible with an input dataset (although due to stochastic steps within the search algorithm, each run of the program may produce different output ARGs). Like Beagle, SHRUB reconstructs histories backwards in time, but does not exhaustively search through all possible sequences of events. Studies of accuracy using simulated and real data (Song et al., 2005) have demonstrated that the computed upper bounds are reasonably close to R_{min} for moderate sample sizes (under 100) and relatively low values of θ and ρ.

SHRUB-GC is an extension of SHRUB incorporating gene conversion events: rather than computing an upper bound on R_{min}, it seeks to minimise T_{min}, defined as the total number of crossover recombination or gene conversion events required to reconstruct a dataset. The maximum gene conversion tract length can be specified as an input parameter; note that setting this to 1 is effectively equivalent to introducing recurrent mutation events. The program outputs a single most parsimonious ARG identified, although as with SHRUB, due to stochastic steps in the search algorithm, each run might produce different results.

1.5 Recombination of SARS-CoV-2 genomes

Viral recombination occurs when a single host cell is co-infected with different strains of the same virus, and during replication the genomes are reshuffled and combined before being packaged and released as new offspring virions, now potentially possessing very different pathogenic properties. The COVID-19 pandemic began following the emergence of the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2 virus) in late 2019. While the role of recombination between different coronaviruses in the *origins* of SARS-CoV-2 has been widely studied, understanding its potential for ongoing recombination within human hosts has proved difficult.

As noted in Section 1.3.2, the detection of ongoing recombination from a sample of genetic data is, in general, a very challenging problem. In evolutionary terms, a relatively short time period has passed since the start of the pandemic, so typical SARS-CoV-2 sequences differ only by a small number of mutations, meaning that recombination events are likely to be undetectable or leave only faint traces. Coronaviruses are known to have relatively high recombination rates (Su et al., 2016), and cell culture studies indicate that this holds true for SARS-CoV-2 (Gribble et al., 2021). This suggests that ongoing intra-host recombination since the start of the pandemic should be commonplace, but detection efforts have been thwarted by the slow accumulation of genetic diversity.

Early evidence of ongoing recombination in SARS-CoV-2 was presented by Yi (2020), who identified the presence of loops in reconstructed phylogenetic networks, which can arise as a consequence of recombination. A number of more recent reports have utilised methods based on classifying sequences into clades, and searching for those that appear to carry a mix of mutations characteristic to more than one clade. VanInsberghe et al. (2021) identified 1 175 possible recombinants out of 537 000 analysed sequences; Varabyou et al. (2021) identified 225 possible recombinants out of 84 000; Jackson et al. (2021) have identified a small number of putative recombinants circulating in the UK. These methods are sensitive to the classification of sequences into clades, do not allow for the detection of intra-clade recombinants (thus underestimating the overall extent of recombination), and do not incorporate a framework for quantifying how likely it is that an observed pattern of incompatibilities has arisen through recombination rather than recurrent mutation. A number of studies have also failed to detect recombination signal, through the analysis of linkage disequilibrium and similar techniques (De Maio et al., 2020; van Dorp et al., 2020b; Nie et al., 2020; Tang et al., 2020; Wang et al., 2020; Richard et al., 2020). In general, a relatively small number of putative recombinant sequences have been identified to date, and there is a lack of compelling evidence for widespread recombination in SARS-CoV-2. Given the aforementioned causes for studies to be underpowered, the overall extent and importance of ongoing recombination for SARS-CoV-2 remains not fully resolved.

Phylogenetic analysis of SARS-CoV-2 data largely assumes the absence of recombination. Recombination can significantly influence the accuracy of phylogenetic inference (Posada and Crandall, 2002), distorting the branch lengths of inferred trees and making mutation rate heterogeneity appear stronger

(Schierup and Hein, 2000). Moreover, when analysing data at the level of consensus sequences, the genealogy of a sample is related to the transmission network of the disease, with splits in the genealogy relating to the transmission of the virus between hosts. Models used for constructing genealogies and inferring evolutionary rates for this type of data cannot fully incorporate potentially important factors, such as geographical structure, patterns of social mixing, travel restrictions, and other non-pharmaceutical interventions, without making inference intractable. Relying on standard tree-based models can easily lead to biased estimates, with the extent of the error due to model misspecification being very difficult to quantify.

1.6 Overview

In Chapter 2, birth-death models for sample genealogies are considered. A stochastic process is defined which is the time reversal of the RP, simplifying the derivation of distributions characterising the event times in the genealogy (some of which are known, but were computed by substantially more cumbersome means in the literature). Then, the large population limit of the process is considered, as the Bernoulli sampling probability tends to 0, and properties of the genealogy are analysed in this setting.

In Chapter 3, KwARG, a heuristic parsimony-based method for reconstructing ARG topologies, is presented. KwARG outputs ARGs that are minimal or near-minimal in the number of posited recombination and/or recurrent mutation events, dropping the infinite sites assumption, differentiating it from other existing methods. Given an input dataset of aligned sequences, KwARG outputs a list of possible candidate solutions, each comprising a list of mutation and recombination events that could have generated the dataset; the relative proportion of recombinations and recurrent mutations in a solution can be controlled via specifying a set of 'cost' parameters. Analysis using simulated data shows that the algorithm performs well when compared against the existing methods described above, both in terms of providing close upper bounds on R_{min} and P_{min}, and reconstructing local trees with good accuracy.

In Chapter 4, KwARG is used to reconstruct possible genealogical histories for samples of SARS-CoV-2 sequences, with the goal of detecting ongoing recombination. A statistical framework is introduced for disentangling the effects of recurrent mutation from recombination in the history of a sample, providing a way of estimating the probability that ongoing recombination is present. Applying this to samples of sequencing data collected in England and South Africa, evidence of ongoing recombination is identified.

Discussion follows in Chapter 5.

Chapter 2

Birth-death models for genealogies

2.1 Introduction

In this chapter, I consider the genealogy of a sample from a population evolving according to a birth-death process. The time to origin is assumed to be random, with a uniform prior, and the sample size is conditioned to be n at the present, under Bernoulli-type sampling with sampling probability ψ. The *reversed reconstructed process (RRP)* is defined as a time reversal of the RP described in Section 1.2.2. Properties of the RRP are easily derived using standard methods for stochastic processes; this is used to re-derive several results, such as densities of event times, which have been given elsewhere in the literature (but the resulting proofs are significantly simpler and more intuitive).

A simulation algorithm is proposed for (incompletely) sampled RRPs using time rescaling. This is an alternative to existing algorithms (Hartmann et al., 2010; Stadler, 2011), which instead utilise a CPP formulation. The relationship between these two approaches is discussed.

Further, the correspondence between completely and incompletely sampled RRPs through time rescaling is derived. In related work, e.g. Stadler and Steel (2012), the approach taken of transforming birth and death rates meant that results could be derived only for a restricted set of parameter values, in particular for $1 - \psi \leq \mu/\lambda \leq 1$; this is especially restrictive when ψ is small. Here, it is shown instead that the completely and incompletely sampled RRPs are time-rescaled versions of each other, so distributions for the incompletely sampled case can be derived using a change of variables. This is used to derive the distribution of the length of a randomly chosen pendant edge, presented for a restricted range of λ and μ by Stadler and Steel (2012), for all parameter values.

Next, the scenario is considered in which the underlying population size in a birth-death process grows to infinity, but a finite sample of size n is obtained. This can be thought of as taking the limit $\psi \to 0$ for the Bernoulli sampling probability; the connection is discussed with the limit as the total population size tends to infinity for n-sampling, using results of Lambert (2018). The time transformation between the RRP in this setting and a pure-death process with rate 1 is discussed in detail; in this scenario, there are two distinct timescales, separating the time of the first event from the events nearer the root of the tree. The RRP tree becomes almost star-shaped: the terminal branch lengths tend to infinity, while the inter-event times at the top of the tree are approximately exponentially distributed, with rate depending on the remaining number of lineages. The time rescaling formalism is then used to derive, analytically, the density of the inter-event times, both for any $\psi \in (0, 1]$ and in the limit $\psi \to 0$; both results are new. It is then demonstrated that in the limit $\psi \to 0$, the event times are distributed as the order statistics of n logistic random variables, with mode $\log(1/\psi)$ (after a simple, linear, time rescaling). Further, it is shown that the inter-event times (thus distributed as the spacings between consecutive order statistics of n logistic random variables) are approximately exponentially distributed, with error bounded by $1/n$ in terms of Kolmogorov-Smirnov distance. The expectation of inter-event times is also shown to agree exactly with the expectation under this approximation.

In Section 2.2, the birth-death processes being considered are formally defined, and the notation used throughout is introduced. In Section 2.3, several known results are stated for inhomogeneous birth-death processes, and the notion of time rescaling for these processes is reviewed. In Section 2.4, the RRP of birth-death processes with Bernoulli sampling is considered. In Section 2.5, the limit of the sampling probability approaching 0 is investigated. Comparison between the RRP in the $\psi \to 0$ limit and the coalescent with exponential growth is presented in Section 2.6. Finally, discussion is presented in Section 2.7.

Illustrations of trees throughout were made using the R package `ape` (Paradis and Schliep, 2018).

2.2 Birth-death processes and time reversal

Consider a birth-death process $\mathcal{B}$ with birth rate λ and death rate $\mu \neq \lambda$ (shortened as BDP(λ, μ)). The process starts with one individual at time 0 and is run for time T since origin, at which point all n extant individuals are sampled. I assume a uniform (improper) prior on T, reiterating that this choice of prior is not novel, and has been treated, for instance, by Aldous and Popovic (2005) for the critical case $\lambda = \mu$, and Gernhard (2008a) and Wiuf (2018) for the supercritical case. In this section, calculations are given to show that with this choice of improper prior, after conditioning on the number of sampled individuals n, the time since origin T_n^* of the conditioned process is random with a particular, proper distribution. It is then demonstrated that the BDP(λ, μ) sampled a time T_n^* since origin and conditioned to have n sampled individuals is dual to the BDP(μ, λ), initialised with n individuals and run until first hitting state 0. This is not a new result, but it is crucial to the idea of considering the reconstructed process backwards in time from sampling, so it is included for completeness.

2.2.1 Prior on the time of origin

Let $\mathcal{B}_n$ denote the process $\mathcal{B}$ conditioned to have n sampled individuals, and denote by $\mathcal{B}_{n,s}$ this process with the sampling step happening at time s since origin. Both the subcritical ($\mu > \lambda$) and supercritical ($\lambda > \mu$) cases are considered here. Let $N(s)$ denote the number of individuals alive in $\mathcal{B}_n$ at time s since origin. Then the generating function of $N(s)$ is given by (Athreya and Ney, 1972, Chapter III, Section 5):

$$G(z) = \mathbb{E}(z^{N(s)}) = \frac{\mu(z-1)e^{(\lambda-\mu)s} - \lambda z + \mu}{\lambda(z-1)e^{(\lambda-\mu)s} - \lambda z + \mu}.$$

Then

$$p_s := \mathbb{P}(N(s) = 0) = G(0) = \frac{\frac{\mu}{\lambda-\mu}(e^{(\lambda-\mu)s} - 1)}{1 + \frac{\lambda}{\lambda-\mu}(e^{(\lambda-\mu)s} - 1)},$$

and

$$\mathbb{P}(N(s) = j) = (1 - p_s)\left(1 - \frac{\lambda}{\mu}p_s\right)p_s^{j-1}.$$

Analogously to Aldous and Popovic (2005, Section 2), define the probability measure

$$\mathbb{P}^*(\mathcal{B}_n \in \cdot) := \frac{\int_0^\infty \mathbb{P}(\mathcal{B}_{n,s} \in \cdot)\mathbb{P}(N(s) = n)ds}{\int_0^\infty \mathbb{P}(N(s) = n)ds}. \tag{2.2.1}$$

Making the observation that $\mathbb{P}(N(s) = j) = \frac{1}{\mu}\left(\frac{d}{ds}p_s\right)(p_s)^{j-1} = \frac{1}{j\mu}\frac{d}{ds}(p_s^j)$,

$$\int_0^\infty \mathbb{P}(N(s) = n)\, ds = \frac{1}{n\mu}\left[p_s^n\right]_0^\infty = \begin{cases} \frac{1}{n\mu} & \text{if } \mu > \lambda \\ \frac{1}{n\mu}\left(\frac{\mu}{\lambda}\right)^n & \text{if } \lambda > \mu. \end{cases}$$

Then the function

$$f_{T_n^*}(s) = \frac{\mathbb{P}(N(s) = n)}{\int_0^\infty \mathbb{P}(N(x) = n)dx} = \begin{cases} \frac{n\mu e^{(\mu-\lambda)s}\left[\frac{\mu}{\mu-\lambda}(e^{(\mu-\lambda)s}-1)\right]^{n-1}}{\left[1+\frac{\mu}{\mu-\lambda}(e^{(\mu-\lambda)s}-1)\right]^{n+1}} & \text{if } \mu > \lambda \\ \\ \frac{n\lambda e^{(\lambda-\mu)s}\left[\frac{\lambda}{\lambda-\mu}(e^{(\lambda-\mu)s}-1)\right]^{n-1}}{\left[1+\frac{\lambda}{\lambda-\mu}(e^{(\lambda-\mu)s}-1)\right]^{n+1}} & \text{if } \lambda > \mu \end{cases} \tag{2.2.2}$$

is a probability density on $[0, \infty)$ for the time since origin T_n^*, and (2.2.1) can be rewritten as

$$\mathbb{P}^*(\mathcal{B}_n \in \cdot) = \int_0^\infty f_{T_n^*}(s)\, \mathbb{P}(\mathcal{B}_{n,s} \in \cdot)ds.$$

Thus, after conditioning on the sample size n, $f_{T_n^*}(s)$ is a proper density for T_n^*.

2.2.2 Time reversal

The population size of the BDP(λ, μ) is a continuous-time Markov chain with the transition rates

$$q_{i,i+1} = \lambda i, \;\; q_{i,i-1} = \mu i.$$

As above, denote by $\{N(s),\ 0 \leq s \leq T_n^*\}$ the corresponding process associated with the complete tree, counting the population size up to the time of sampling, making the jump from 0 to 1 at time 0. Consider also the continuous-time Markov chain $\{\widehat{N}_n(\tau),\ 0 \leq \tau \leq T_n\}$, which has the reversed transition rates

$$q_{i,i+1} = \mu i, \;\; q_{i,i-1} = \lambda i,$$

started in state $\widehat{N}_n(0) = n$ and run until the first hitting time T_n of state 0. Then, mirroring Aldous and Popovic (2005, Lemma 2) for the critical case, the following holds:

Lemma 2.2.1.

$$\{N(T_n^* - \tau),\ T_n^* \geq \tau \geq 0\} \overset{d}{=} \{\widehat{N}_n(\tau),\ 0 \leq \tau \leq T_n\},$$

and in particular $T_n^* \overset{d}{=} T_n$*, where* $\overset{d}{=}$ *denotes equality in distribution.*

Proof. Fix the event times $\tau_0, \ldots, \tau_M$, with $\tau_M > \tau_{M-1} > \ldots \tau_1 > \tau_0 = 0$. Define the corresponding sequence of positive integers $k_M = 1, k_{M-1}, \ldots, k_2, k_1 = n$, with $|k_m - k_{m-1}| = 1$ and $k_{M+1} = 0$, describing the population size trajectory of the realisation of the birth-death process; reading from left to right, this has $b+1$ increases of size 1, and $n-1+b$ decreases of size 1 for some integer $b \geq 0$ with $n + 2b = M$. Then the event

> $\{$as τ decreases, $N(T_n^* - \tau)$ jumps from k_{m+1} to k_m for $\tau \in [\tau_m, \tau_m + d\tau_m]$ (for all $M \geq m \geq 1$) and makes no other jumps$\}$

has measure

$$d\tau_M \cdot \prod_{m=M}^{2} \left(e^{-k_m(\lambda+\mu)(\tau_m - \tau_{m-1})}\ k_m\, d\tau_{m-1}\right) \cdot \lambda^{b+1}\ \mu^{n-1+b} \cdot e^{-k_1 \tau_1}, \tag{2.2.3}$$

where $d\tau_M$ comes from the uniform prior, and ignoring terms of $o(d\tau_m)$. For the reversed process $\widehat{N}_n(\tau)$, the event

> $\{$as τ increases, $\widehat{N}_n(\tau)$ jumps from k_m to k_{m+1} in the interval $\tau \in [\tau_m, \tau_m + d\tau_m]$ (for all $1 \leq m \leq M$) and makes no other jumps$\}$

has probability

$$\prod_{m=1}^{M} \left(e^{-k_m(\lambda+\mu)(\tau_m - \tau_{m-1})}\ k_m d\tau_m\right) \cdot \mu^{n-1+b}\ \lambda^{b+1}, \tag{2.2.4}$$

ignoring terms of $o(d\tau_m)$; this is because reading the sequence of k_m's from right to left, there are $n-1+b$ increases of size 1 and $b+1$ decreases of size 1. The measure (2.2.3) is $1/k_1 = 1/n$ times (2.2.4), so after conditioning the probability measures of the two events are equal. □

This demonstrates the duality between the BDP(λ, μ) started with 1 individual at time 0 and reaching n individuals after the random time T_n^* (running "forwards" to the time of sampling), and the BDP(μ, λ), started from n individuals at time 0 and run until it reaches state 0 (running "backwards in time" from the sample). Next, the reconstructed process is considered, which tracks the genealogy of only the sampled individuals, making use of the duality between the forwards-in-time and backwards-in-time formulations.

2.2.3 The reversed reconstructed process (RRP)

The RP (forwards in time) describes the number of lineages in the BDP(λ, μ), which will have at least one surviving descendant in the sample. Nee et al. (1994) identified that the RP forwards in time is generated by an underlying time-inhomogeneous pure birth process, with birth rate per lineage at time s given by

$$\begin{aligned} \lambda P_1(s,T) &:= \lambda \cdot \mathbb{P}(\text{a single lineage born at time } s \text{ is not extinct by time } T) \\ &= \frac{\lambda(\lambda-\mu)}{\lambda - \mu e^{-(\lambda-\mu)(T-s)}} \\ &= \frac{\lambda e^{(\lambda-\mu)(T-s)}}{1 + \frac{\lambda}{\lambda-\mu}(e^{(\lambda-\mu)(T-s)} - 1)}, \end{aligned} \tag{2.2.5}$$

where T is the time of sampling and $P_1(s,T)$ is given by Kendall (1948). The state of the process at time s is the number of individuals alive at s with at least one descendant at T, with events corresponding to transitions from state j to $j+1$, $j \geq 1$.

It is advantageous to consider the process running backwards in time from the present, conditioning on the sample size n, and not explicitly conditioning on the time of origin of the process (which is generally unknown, and for which a uniform improper prior is imposed). The focus will thus be on the properties of the reversed reconstructed process, which is defined as the process tracking the genealogy of the initial population of the BDP(μ, λ), initialised at n individuals and run until the first hitting time of state 0. It is straightforward to show, similarly to Lemma 2.2.1, that the RP with birth rate (2.2.5) run for time T_n^* and reaching state n at the time of sampling is dual to the RRP which is started in state n at time 0 and stopped at the first hitting time of state 0, with death rate obtained by replacing $T-s$ by τ in (2.2.5) to account for the time reversal. Note that the time index τ increases into the past, and $\tau = 0$ denotes the time of sampling. The RRP is thus an inhomogeneous pure-death process, with death rate per lineage given by

$$m_\beta(\tau) = \frac{\lambda e^{(\lambda-\mu)\tau}}{1+\frac{\lambda}{\lambda-\mu}(e^{(\lambda-\mu)\tau}-1)}. \tag{2.2.6}$$

The subscript β refers to the time scale on which the RRP is measured.

To obtain the death rate of the RRP with Bernoulli sampling (where each lineage is sampled with a fixed probability ψ at time 0), replace $P_1(s,T)$ with the relevant probability $P_\psi(s,T)$ as derived by Yang and Rannala (1997):

$$P_\psi(s,T) = \frac{\psi(\lambda-\mu)}{\psi\lambda-(\mu-(1-\psi)\lambda)e^{-(\lambda-\mu)(T-s)}} = \frac{\psi e^{(\lambda-\mu)(T-s)}}{1+\frac{\psi\lambda}{\lambda-\mu}\left(e^{(\lambda-\mu)(T-s)}-1\right)},$$

which, following the same reasoning as for the case of complete sampling, gives the RRP death rate:

$$m_\gamma(\tau) = \frac{\psi\lambda e^{(\lambda-\mu)\tau}}{1+\frac{\psi\lambda}{\lambda-\mu}(e^{(\lambda-\mu)\tau}-1)}. \tag{2.2.7}$$

The subscript γ now refers to the units in which time is measured for the RRP with Bernoulli sampling. It will later be described how the two processes are linked through transformation of time, i.e. that a realisation of the completely sampled RRP can be transformed to a realisation of the Bernoulli sampled RRP, through rescaling time by some function g such that $g(\gamma)=\beta$. Thus, the subscripts denote the time scale of the corresponding process: Table 2.1 summarises the time units corresponding to each type of RRP, and introduces the notation used throughout.

For instance, denote by BDP(λ,μ,ψ) the birth-death population process where each individual divides independently with rate λ, dies independently with rate $\mu<\lambda$, with the rates measured in time units γ; at time 0, each surviving individual is sampled with a fixed probability ψ. The corresponding RRP, i.e. the process tracing out the genealogy of the sample from this population backwards in time from 0, is denoted by $X_\psi^\gamma := (X_\psi^\gamma(\tau) : \tau \geq 0)$. Let X_ψ^ξ denote the same process, but with time rescaled to units of $\xi = g(\gamma)$ for some time transformation g, i.e. $X_\psi^\xi(g(\tau)) = X_\psi^\gamma(\tau)$. The death rates of X_ψ^γ and X_ψ^ξ are denoted m_γ and m_ξ, respectively, with the subscripts denoting the time scale on which the rates are measured. The relationships between the time scales for the RRPs in Table 2.1 are presented in Appendix A for reference.

For the case of a critical branching process, measured in time units of α,

Population process	**Time unit**	**Notation**	**RRP notation**
Yule process, birth rate 1	t	Yule(1)	Y
Critical branching process, birth = death rate λ, sampling probability ψ	α	CBP(λ, ψ)	Z^{α}_{ψ}
Birth-death process, birth rate λ, death rate μ, complete sampling	β	BDP$(\lambda, \mu, 1)$	X^{β}_{1}
Birth-death process, birth rate λ, death rate μ, sampling probability ψ	γ	BDP(λ, μ, ψ)	X^{γ}_{ψ}
Birth-death process, birth rate λ', death rate μ', with $\lambda' - \mu' = 1$ sampling probability ψ	δ	BDP(λ', μ', ψ)	X^{δ}_{ψ}

Table 2.1: Summary of RRP notation

the death rate is given by taking the limit $\lambda \to \mu$ in (2.2.7):

$$m_{\alpha}(\tau) = \frac{\psi\lambda}{1 + \psi\lambda\tau}.$$

Note that for the case of a subcritical process (with $\lambda < \mu$), the population process backwards in time is supercritical. To ensure that the population reaches a common ancestor, the process must be conditioned on ultimate extinction; it can be shown that this is equivalent to swapping the birth and death rate (Waugh, 1958), indeed this is clear from (2.2.2) for the time of origin. Thus, the RRP death rate in the subcritical case will be the same as (2.2.7) but with λ and μ interchanged.

2.3 Background

In this section, the relevant known technical results, which will later be relied on, are reviewed.

2.3.1 Inhomogeneous pure-death processes

Consider a time-inhomogeneous pure-death process, with time measured in units ξ, starting with n individuals alive at time 0. Each individual dies independently at rate $m_\xi(\tau)$; if there are j individuals at time τ, the intensity is $jm_\xi(\tau)$. The rate function of the process is given by

$$\rho_\xi(\tau) = \int_0^\tau m_\xi(x)dx.$$

The transition probabilities, i.e. the probability of going from n to j individuals in time τ, are given by a binomial distribution (Bailey, 1964, p.112):

$$P_{nj}(\tau) = \begin{cases} \binom{n}{j}\big(1 - e^{-\rho_\xi(\tau)}\big)^{n-j}\big(e^{-\rho_\xi(\tau)}\big)^{j} & \text{for } j \leq n, \\ 0 & \text{otherwise,} \end{cases} \tag{2.3.1}$$

with $e^{-\rho_\xi(\tau)}$ being the probability that a lineage has not died by time τ. The distribution of time to origin is (Bailey, 1964, p.112):

$$F_{T_n}(\tau) = P(T_n < \tau) = \big(1 - e^{-\rho_\xi(\tau)}\big)^n, \tag{2.3.2}$$

and, by differentiating, the pdf is

$$f_{T_n}(\tau) = nm_\xi(\tau)e^{-\rho_\xi(\tau)}\big(1 - e^{-\rho_\xi(\tau)}\big)^{n-1}. \tag{2.3.3}$$

The density of the time of the k-th event is given by

$$\begin{aligned} f_{T_k}(\tau) &= \binom{n}{k} \cdot \underbrace{km_\xi(\tau)e^{-\rho_\xi(\tau)}\big(1 - e^{-\rho_\xi(\tau)}\big)^{k-1}}_{k\text{-th lineage dies at } \tau} \cdot \underbrace{\big(e^{-\rho_\xi(\tau)}\big)^{n-k}}_{n-k \text{ survive for at least } \tau} \\ &= \binom{n}{k}\, k\, m_\xi(\tau)\big(1 - e^{-\rho_\xi(\tau)}\big)^{k-1}\big(e^{-\rho_\xi(\tau)}\big)^{n-k+1}. \end{aligned} \tag{2.3.4}$$

2.3.2 Time rescaling

Consider a pure-death inhomogeneous process with death rate $m_\xi(\tau)$, with time measured in units of ξ. Suppose that time is rescaled in units of $\zeta = g(\xi)$, where g is strictly monotonic and differentiable. The death rate of the process then becomes, using a change of variables:

$$m_\zeta(\tau) = m_\xi(g^{-1}(\tau))\left|\frac{d}{d\tau}\, g^{-1}(\tau)\right|.$$

The time rescaling theorem (Meyer, 1971; Papangelou, 1972) states that any inhomogeneous point process with an integrable intensity function can be rescaled to a Poisson process with unit rate. The RRP can be thought of as a point process, with intensity given by its inhomogeneous death rate times the number of lineages. If the RRP (of any population process) has death rate $m_\xi(\tau)$, then rescaling time via the transformation $g = \rho_\xi$ rescales the RRP to a homogeneous pure-death process with death rate per lineage equal 1 (a time-reversed Yule rate 1 process).

2.3.3 Time-reversed Yule rate 1 process

Define the time-reversed Yule rate 1 process as a pure death process where each lineage dies independently at rate 1, denoted Y. This is the RRP of a forwards-in-time Yule process with birth rate 1. The inter-event time during which there are exactly j lineages is exponentially distributed with rate j. Using (2.3.3), the time to origin has density:

$$f_{T_n}(\tau) = ne^{-\tau}(1 - e^{-\tau})^{n-1},$$

and using (2.3.4), the time to k-th event has density:

$$f_{T_k}(\tau) = \binom{n}{k} k \left(1 - e^{-\tau}\right)^{k-1} \left(e^{-\tau}\right)^{n-k+1}. \tag{2.3.5}$$

The expectation of time to origin is $\sum_{j=1}^{n} \frac{1}{j}$. These results are identical to those derived by Gernhard (2008b).

2.4 Birth-death process with Bernoulli sampling

The RRP X_ψ^γ of a supercritical birth-death process is now considered in detail. First, using the formulation introduced in Section 2.2.3, some known properties of the process are re-derived, which will be readily available from the results given in Section 2.3. Then, using the fact that the RRP X_ψ^γ is a time rescaling of the RRP associated with a Yule rate 1 process, a simulation algorithm is proposed. Finally, using time rescaling, the relationship between completely and incompletely sampled RRPs is considered.

2.4.1 Properties of the process

Set $T_0 = 0$ and for $k \in \{1, \dots, n\}$ denote by T_k the time of the k-th event, backwards from the present time 0. At T_k, the number of lineages decreases from $n-k+1$ to $n-k$. For $k \in \{0, \dots n-1\}$, let $W_k := T_{k+1} - T_k$ denote the inter-event time.

2.4.1.1 Transition probabilities and densities of event times

The pure-death process formulation of X_ψ^γ is used to derive distributions characterising this process. The transition probabilities are, using (2.3.1):

$$P_{ij}(\tau) = \begin{cases} \binom{i}{j}\left(1 - e^{-\rho_\gamma(\tau)}\right)^{i-j}\left(e^{-\rho_\gamma(\tau)}\right)^j & \text{for } j \le i \\ 0 & \text{otherwise,} \end{cases}$$

where, by integrating the death rate in (2.2.7),

$$\rho_\gamma(\tau) = \int_0^\tau m_\gamma(x)dx = \log\left(1 + \frac{\psi\lambda}{\lambda-\mu}\left(e^{(\lambda-\mu)\tau} - 1\right)\right), \tag{2.4.1}$$

and

$$e^{-\rho_\gamma(\tau)} = \frac{1}{1 + \frac{\psi\lambda}{\lambda-\mu}(e^{(\lambda-\mu)\tau} - 1)}.$$

For $\tau \to \infty$ and fixed $\psi \in (0, 1]$, $\rho_\gamma(\tau) \to \infty$ and $e^{-\rho_\gamma(\tau)} \to 0$, so $P_{ij}(\tau) \to 0$ for all $j \neq 0$, and $P_{i0}(\tau) \to 1$. This implies that two individuals sampled at the present will eventually find a common ancestor in the past with probability 1.

The distribution of time to origin, using (2.3.2), is given by:

$$F_{T_n}^\psi(\tau) = P(T_n < \tau) = \left(1 - e^{-\rho_\gamma(\tau)}\right)^n = \left(\frac{\frac{\psi\lambda}{\lambda-\mu}(e^{(\lambda-\mu)\tau} - 1)}{1 + \frac{\psi\lambda}{\lambda-\mu}(e^{(\lambda-\mu)\tau} - 1)}\right)^n,$$

and its density, using (2.3.3), is

$$\begin{aligned} f_{T_n}^\psi(\tau) &= \frac{n \cdot \psi\lambda e^{(\lambda-\mu)\tau}}{1 + \frac{\psi\lambda}{\lambda-\mu}(e^{(\lambda-\mu)\tau} - 1)} \cdot \frac{1}{1 + \frac{\psi\lambda}{\lambda-\mu}(e^{(\lambda-\mu)\tau} - 1)}\left(\frac{\frac{\psi\lambda}{\lambda-\mu}(e^{(\lambda-\mu)\tau} - 1)}{1 + \frac{\psi\lambda}{\lambda-\mu}(e^{(\lambda-\mu)\tau} - 1)}\right)^{n-1} \\ &= n\psi\lambda e^{(\lambda-\mu)\tau}\frac{\left[\frac{\psi\lambda}{\lambda-\mu}\left(e^{(\lambda-\mu)\tau} - 1\right)\right]^{n-1}}{\left[1 + \frac{\psi\lambda}{\lambda-\mu}\left(e^{(\lambda-\mu)\tau} - 1\right)\right]^{n+1}}. \end{aligned} \tag{2.4.2}$$

Note that this agrees with (2.2.2) for the case $\psi = 1$. This result is also obtained in Stadler (2009, Lemma 3.1). Although the outcome is identical, the derivation given above is significantly simpler, and follows directly from the properties of the RRP as a stochastic process. In particular, the distribution function is immediately obtained from knowing the death rate; moreover, to obtain the pdf there is no need to integrate over the prior for the time of origin, as this is implicit in the time reversal.

Using (2.3.4), the waiting time to the k-th event is given by:

$$\begin{aligned}
f^{\psi}_{T_k}(\tau) &= \binom{n}{k}\, k\, m_\gamma(\tau)\big(1-e^{-\rho_\gamma(\tau)}\big)^{k-1}\big(e^{-\rho_\gamma(\tau)}\big)^{n-k+1} \\
&= \binom{n}{k}\, k\, \frac{\psi\lambda e^{(\lambda-\mu)\tau}}{1+\frac{\psi\lambda}{\lambda-\mu}(e^{(\lambda-\mu)\tau}-1)} \left(\frac{\frac{\psi\lambda}{\lambda-\mu}(e^{(\lambda-\mu)\tau}-1)}{1+\frac{\psi\lambda}{\lambda-\mu}(e^{(\lambda-\mu)\tau}-1)}\right)^{k-1} \cdot \\
&\quad \cdot \left(\frac{1}{1+\frac{\psi\lambda}{\lambda-\mu}(e^{(\lambda-\mu)\tau}-1)}\right)^{n-k+1} \\
&= \binom{n}{k}\, k\, \psi\lambda e^{(\lambda-\mu)\tau} \frac{\left[\frac{\psi\lambda}{\lambda-\mu}\big(e^{(\lambda-\mu)\tau}-1\big)\right]^{k-1}}{\left[1+\frac{\psi\lambda}{\lambda-\mu}\big(e^{(\lambda-\mu)\tau}-1\big)\right]^{n+1}}. \qquad (2.4.3)
\end{aligned}$$

This agrees with the result derived in Gernhard (2008a, Theorem 4.1) for the case of complete sampling; again, note that the result follows almost immediately from the properties of the RRP, which removes the need for deriving the related distributions by hand.

2.4.1.2 Simulating from the RRP

As described in Section 2.3.2, applying the time transformation $g_1 = \rho_\gamma$ rescales the RRP X^γ_ψ to the time-reversed Yule rate 1 process Y. From (2.4.1), this transformation is given by:

$$\begin{aligned}
t &= g_1(\gamma) = \log\left(1+\frac{\psi\lambda}{\lambda-\mu}\Big(e^{(\lambda-\mu)\gamma}-1\Big)\right), \\
\gamma &= g_1^{-1}(t) = \rho_\gamma^{-1}(t) = \frac{1}{\lambda-\mu}\log\left(1+\frac{\lambda-\mu}{\psi\lambda}\Big(e^{t}-1\Big)\right), \qquad (2.4.4)
\end{aligned}$$

and

$$X^\gamma_\psi(g_1^{-1}(\tau)) = Y(\tau) \text{ and } X^\gamma_\psi(\tau) = Y(g_1(\tau)),$$

meaning that X^γ_ψ rescaled in time units $g_1(\gamma)$ has the same death rate as Y. To see why this works, the death rate of X^γ_ψ when measured in units $t = g_1(\gamma)$

becomes:

$$\begin{aligned} m_t(\tau) &= m_\gamma(g_1^{-1}(\tau)) \left| \frac{d}{d\tau}\, g_1^{-1}(\tau) \right| \\ &= m_\gamma(\rho_\gamma^{-1}(\tau)) \left| \frac{d}{d\tau}\, \rho_\gamma^{-1}(\tau) \right| \\ &= m_\gamma(\rho_\gamma^{-1}(\tau)) \Big/ m_\gamma(\rho_\gamma^{-1}(\tau)) \\ &= 1. \end{aligned}$$

Note also that in the complete process, birth, death, and sampling events affect all individuals with equal probability, so the topologies of RRP trees are equal in law to those of Yule and coalescent trees (Aldous, 1996), and can thus be generated backwards in time by merging pairs of lineages selected uniformly at random. This suggests that to simulate from X_ψ^γ, it is possible to first simulate from Y, and then rescale the event times using the transformation given by (2.4.4). The method is summarised as Algorithm 1. This provides an alternative to the algorithms of Hartmann et al. (2010) and Stadler (2011), where first the time of origin is drawn from its distribution, and then the coalescent point process formulation is used to obtain the event times.

Algorithm 1: Simulating from the RRP X_ψ^γ

```
Input: $n$ individuals at time 0
Output: Realisation of a genealogy from the RRP $X_\psi^\gamma$
Draw $\widetilde{W}_j \sim \mathrm{Exp}(n-j)$ for $j = 0, \dots, n-1$, being the waiting times of $Y$;
Compute the event times $\widetilde{T}_{j+1} = \sum_{i=0}^{j} \widetilde{W}_i$;
Rescale the event times as $T_k = \frac{1}{\lambda-\mu} \log\Big(1 + \frac{\lambda-\mu}{\psi\lambda} \Big(\exp\Big(\widetilde{T}_k\Big) - 1\Big)\Big)$ for $k = 1, \dots, n$;
Construct a tree from $T_1, \dots, T_n$ by choosing a pair of lineages uniformly at random to coalesce at each event time.
```

Note that one can first derive distributions of interest for Y, and then use the change of variables given by (2.4.4) to obtain the equivalent results for X_ψ^γ. This will be used to derive the distribution of inter-event times W_k, analytically, in Section 2.5.3.

2.4.1.3 Relationship with coalescent point processes

Gernhard (2008a) gives the following CPP formulation for a supercritical process. To simulate an RRP for a sample of size n, first condition on the sample size and a time of origin T_n (possibly drawn from the distribution (2.3.2)), and then draw the times of the $n-1$ bifurcations in the tree i.i.d. from some specific density depending on T_n. Lambert and Stadler (2013) further give this density for the case of Bernoulli sampling. In a sense, conditioning on the time of origin, the event times can thus be simulated "horizontally", one-by-one for each sampled lineage, rather than "vertically", i.e. forwards or backwards in time.

The formulation of the RRP as a pure-death process also allows for simulation of the RRP lineage-by-lineage, conditioning on the sample size but not on the time of origin (producing a tree including the root edge). Because each lineage dies independently from the others, in order to simulate from X_ψ^γ for a sample of size n, the death times of each of the n lineages can be simulated independently, and then the lineages merged uniformly at random at each event time to create the tree. The death time of one lineage has density:

$$f_{T_{(1)}}^{\psi}(\tau) = \frac{\psi\lambda e^{(\lambda-\mu)\tau}}{\left[1+\frac{\psi\lambda}{\lambda-\mu}\left(e^{(\lambda-\mu)\tau}-1\right)\right]^2}, \tag{2.4.5}$$

which is obtained from (2.4.2) by substituting $n=1$; this can be simulated by drawing from an exponential rate 1 density, and rescaling time using (2.4.4). Therefore the relationship between CPP and the pure-death formulation is very direct. With the pure-death formulation, each of the n lineages dies independently with the same death rate. Conditioning on a time of origin T_n, the lineages still die independently, with death rate amended so that each event happens before T_n. The latter is exactly the CPP formulation of Gernhard (2008a).

The CPP formulation described by Lambert and Stadler (2013) also gives a method for simulating a Bernoulli RRP without conditioning on the sample size, as follows. Given a time of origin T, draw realisations $H_1^\psi, \ldots, H_N^\psi$ of a random variable H^ψ, with the stopping criterion that H_N^ψ is the first realisation that is greater than T. Then the $H_1^\psi, \ldots, H_{N-1}^\psi$ are the event times up to the MRCA for a sample of N lineages in a Bernoulli sampled RRP, conditioned on time of origin T. Note that in this case, setting $p = P(H^\psi > T)$, the number of sampled lineages is geometric with mass function $(1-p)^{n-1}p$, and the density of H^{ψ} given in Lambert and Stadler (2013, p.122) is exactly that in (2.4.5).

The pure-death formulation of the RRP highlights two differences between the genealogy of a birth-death process and the coalescent. Firstly, viewing the basic coalescent as a backwards in time pure-death process with rate $\frac{j(j-1)}{2}$ when there are j lineages, at each point in time the death rate of each individual lineage depends on the total number of lineages remaining; this dependence cannot be removed by conditioning on the time of origin (for $n > 2$). This implies that the process cannot be simulated by drawing the death time of each lineage independently from some density, as for the RRP. It is conjectured (Lambert and Stadler, 2013) that the coalescent does not have a CPP representation.

Secondly, the coalescent with variable population size, as described by Griffiths and Tavaré (1994), can be described as an inhomogeneous pure-death process, where the death rate is quadratic in the number of lineages and depends on a population size function. Because the death rate of the RRP is linear in the number of lineages, there is no population size function which would equate the two models exactly. The differences between the RRP and the coalescent with exponential growth is investigated in further detail in Section 2.6.

2.4.2 Relationship between completely and incompletely sampled RRPs

Stadler (2009) noted that there is a relationship between the RRP of the incompletely sampled BDP(λ, μ, ψ), and the RRP of the completely sampled BDP$(\widehat{\lambda}, \widehat{\mu}, 1)$, through the following transformation of the birth and death parameters:

$$\widehat{\lambda} = \psi\lambda, \quad \widehat{\mu} = \mu - \lambda(1-\psi). \tag{2.4.6}$$

Substituting (2.4.6) as the birth and death rates into (2.2.6) gives (2.2.7). Thus, the resulting process looks like the RRP of an incompletely sampled BDP(λ, μ, ψ) population process. However, as noted by Stadler and Steel (2012), $\widehat{\mu}$ can be negative (in particular, for very small values of ψ); for instance, with the parameters used in Figure 1.2, $\widehat{\mu} = -1/60$. In this case, the interpretation as an RRP of some birth-death process is problematic. Stadler and Steel (2012, 2019) discuss that when distributions are derived for the completely sampled process, this reparameterisation trick can be used to obtain the equivalent distributions for a process with incomplete sampling, but only for $\frac{\mu}{\lambda} \geq 1 - \psi$. Thus, this method of transforming the birth and death rates does not always produce a valid mapping between completely and incompletely sampled RRPs.

To avoid this issue, instead of transforming the birth and death parameters directly, I use a transformation of time, and demonstrate the relationship between the RRPs X_ψ^γ and X_1^β. This avoids introducing restrictions on the values of the parameters (λ, μ, ψ), and so allows distributions derived for the completely sampled process to be transformed for the case of incomplete sampling.

2.4.2.1 Time transformation from X_ψ to X_1

Define the transformation of time units g_2 as:

$$\begin{aligned}\beta = g_2(\gamma) &= \frac{1}{\lambda - \mu} \log\big(1 + \psi(e^{(\lambda-\mu)\gamma} - 1)\big), \\ \gamma = g_2^{-1}(\beta) &= \frac{1}{\lambda - \mu} \log\left(1 + \frac{1}{\psi}(e^{(\lambda-\mu)\beta} - 1)\right).\end{aligned} \tag{2.4.7}$$

This is a valid time transformation with $\gamma = 0 \iff \beta = 0$, and $\gamma = \beta$ when $\psi = 1$. Using a change of variable in (2.2.7), the death rate is:

$$\begin{aligned}m_\beta(\tau) &= m_\gamma(g_2^{-1}(\tau)) \cdot \left|\frac{dg_2^{-1}(\tau)}{d\tau}\right| \\ &= \frac{\psi\lambda(1 + \frac{1}{\psi}(e^{(\lambda-\mu)\tau} - 1))}{1 + \frac{\lambda}{\lambda-\mu}(e^{(\lambda-\mu)\tau} - 1)} \cdot \frac{\frac{1}{\psi}e^{(\lambda-\mu)\tau}}{1 + \frac{1}{\psi}(e^{(\lambda-\mu)\tau} - 1)} \\ &= \frac{\lambda e^{(\lambda-\mu)\tau}}{1 + \frac{\lambda}{\lambda-\mu}(e^{(\lambda-\mu)\tau} - 1)}.\end{aligned}$$

This is the death rate of the completely sampled RRP X_1^β as given in (2.2.6). Thus,

$$\begin{aligned}X_1^\beta(\tau) &= X_\psi^\gamma(g_2^{-1}(\tau)), \\ X_1^\beta(g_2(\tau)) &= X_\psi^\gamma(\tau).\end{aligned}$$

The RRP of a BDP(λ, μ, ψ) process is a time rescaled version of the RRP of a completely sampled BDP$(\lambda, \mu, 1)$ process. In effect, introducing incomplete sampling is equivalent to non-linearly rescaling the RRP of the BDP$(\lambda, \mu, 1)$ process using the time transformation (2.4.7).

2.4.2.2 Deriving results for X_ψ from X_1

Using the time transformation approach, distributions can be derived for X_1^β with complete sampling, and then the equivalent distribution results for X_ψ^γ can be obtained through a simple change of variables. As an example, Stadler and Steel (2012) derive the density of the length of a randomly chosen pendant edge (an edge adjacent to a sampled individual) for an incompletely sampled tree with the restriction $1 - \psi \leq \frac{\mu}{\lambda} \leq 1$; I complete the proof for the case $0 \leq \frac{\mu}{\lambda} \leq 1 - \psi$.

Proposition 2.4.1. *The density of the length of a randomly chosen pendant edge, E, of the RRP X_ψ^γ for any $0 \leq \mu < \lambda$ and $\psi \in (0, 1]$ is*

$$f_E^\psi(\tau) = \frac{2\psi\lambda(\lambda-\mu)^3 e^{(\lambda-\mu)\tau}}{\big(\lambda\psi e^{(\lambda-\mu)\tau} - [\mu - \lambda(1-\psi)]\big)^3}.$$

Proof. Mooers et al. (2012) give the density of the length of a pendant edge of a completely sampled RRP X_1^β as:

$$f_E^1(\tau) = \frac{2\lambda(\lambda-\mu)^3 e^{(\lambda-\mu)\tau}}{(\lambda e^{(\lambda-\mu)\tau} - \mu)^3}. \tag{2.4.8}$$

Using the time rescaling (2.4.7) and a change of variable, for X_ψ^γ this becomes:

$$\begin{aligned} f_E^\psi(\tau) &= f_E^1(g_2(\tau)) \left| \frac{d\, g_2(\tau)}{d\tau} \right| \\ &= \frac{2\lambda(\lambda-\mu)^3[1 + \psi(e^{(\lambda-\mu)\tau} - 1)]}{\big(\lambda\big[1 + \psi(e^{(\lambda-\mu)\tau} - 1)\big] - \mu\big)^3} \cdot \frac{\psi e^{(\lambda-\mu)\tau}}{1 + \psi(e^{(\lambda-\mu)\tau} - 1)} \\ &= \frac{2\psi\lambda(\lambda-\mu)^3 e^{(\lambda-\mu)\tau}}{\big(\lambda\psi e^{(\lambda-\mu)\tau} - [\mu - \lambda(1-\psi)]\big)^3}. \end{aligned}$$

□

Equivalence with the result of Stadler and Steel (2012, Section 4) for $\frac{\mu}{\lambda} \geq 1 - \psi$ is easily checked by substituting the birth rate $\widehat{\lambda}$ and death rate $\widehat{\mu}$ into (2.4.8).

2.5 Sampling from large populations

In this section, the setting is considered where the total population size is very large compared to the sample size n. This is a scenario often encountered in practice when collecting genetic data, particularly from viral populations, when the population size is unknown but can be presumed very large. An example will be mentioned within the discussion in Section 2.7.

This situation is to be distinguished from the limit as the sample size grows to infinity, which has been considered by Wiuf (2018). The scenario of interest here is when the total population tends to infinity, but a finite sample of size n is obtained. This can be interpreted as either the Bernoulli sampling probability ψ going to 0, or the total population size growing to infinity in the case of n-sampling. The similarity between these two regimes will be discussed in the following section.

In this section, for the sake of readability of the expressions, time is rescaled linearly in units of $\delta = (\lambda - \mu)\gamma$, writing $\lambda' = \frac{\lambda}{\lambda-\mu}, \mu' = \frac{\mu}{\lambda-\mu}$ with $\lambda' - \mu' = 1$. This simplifies the formulae, and is easy to reverse within any derived expressions. The RRP on this timescale is denoted X_ψ^δ, with death rate

$$m_\delta(\tau) = \frac{\psi\lambda' e^\tau}{1 + \psi\lambda'(e^\tau - 1)}.$$

The time transformation between X_ψ^δ and Y is given by $g_3 = \rho_\delta$, with

$$t = g_3(\delta) = \log\big(1 + \psi\lambda'(e^\delta - 1)\big), \tag{2.5.1}$$

$$\delta = g_3^{-1}(t) = \rho_\delta^{-1}(t) = \log\bigg(1 + \frac{1}{\psi\lambda'}(e^t - 1)\bigg). \tag{2.5.2}$$

2.5.1 Sampling method

Lambert (2018) showed the following relationship between the two sampling scenarios when considered from a CPP perspective. Bernoulli sampled trees can be generated using the CPP formulation; that is, conditioning on a time of origin T, the event times are i.i.d. according to a specific density (as described in Section 2.4.1.3). For n-sampling, if a CPP tree was generated with complete sampling (conditioned to have size at least n), and then n lineages chosen uniformly at random, then this would not have a CPP formulation (Lambert and Stadler, 2013). However, the genealogy of such an n-sample can be obtained by first drawing a sampling probability $\Psi = y$ from a specific distribution, and then generating a Bernoulli CPP of size n with sampling probability y. The distribution of Ψ has the form (Lambert, 2018, Theorem 3)

$$\frac{n(1-a)y^{n-1}}{(1-a(1-y))^{n+1}},$$

where $a = P(H < T)$ is the probability that the random variable corresponding to event times (in the complete tree) takes a value less than the specified time of origin.

The underlying population (of the complete tree) growing to infinity can be seen to correspond to the time of origin of the complete process growing to infinity, and thus the probability $a = P(H < T)$ approaching 1. In this case, the density of Ψ tends to a point mass at $y = 0$. This argument implies that the behaviour of the RRP for Bernoulli sampling with $\psi \to 0$, and for n-sampling when the underlying population grows to infinity, should be the same.

2.5.2 Relationship between X_ψ^δ and Y for small ψ

The effect of the time rescaling between X_ψ^δ and the time-reversed Yule rate 1 process Y is examined when $\psi \to 0$. In the following, it is assumed that λ' is fixed and very small compared to $1/\psi$.

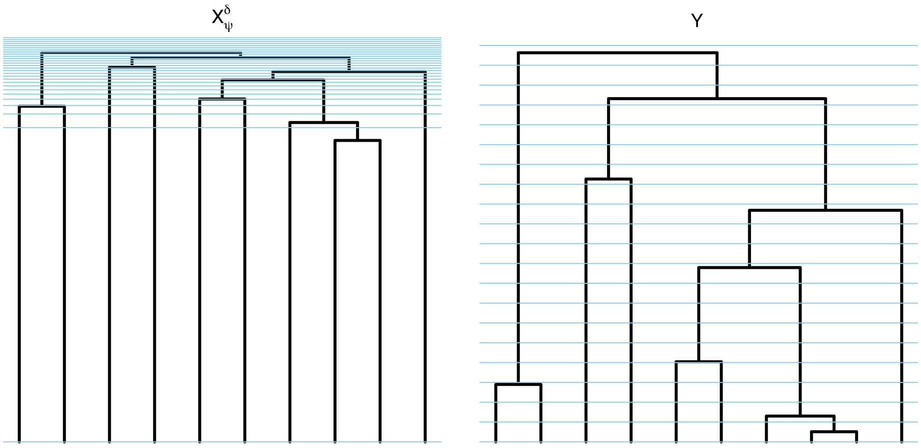

Figure 2.1: Left: realisation of X_ψ^δ with $\psi = e^{-20}, \lambda' = 2$. Right: same tree, rescaled in time units given by (2.5.1). Intervals delineated by blue lines in the left panel are rescaled to intervals of equal length in the right panel.

Consider the time rescaling given by (2.5.1): the process X_ψ^δ rescaled in units of $g_3(\delta)$ is a time-reversed Yule rate 1 process. This rescaling is illustrated in Figure 2.1 for a small value of ψ; the left panel shows a realisation of X_ψ^δ for $n = 10$. The right panel shows the same tree, but the intervals delineated by blue lines in the left panel are rescaled to intervals of equal length in the right panel.

Using the identity $\log(1+x) = \log(x) + \log(1+1/x)$ and a Taylor expansion in $\psi\lambda'$ around 0, (2.5.2) gives:

$$\delta - \log\left(\frac{1}{\psi\lambda'}\right) = \log\left(e^t - 1\right) + \mathcal{O}(\psi\lambda'). \tag{2.5.3}$$

For small t, $e^t - 1 \approx t$ and the transformation behaves as $\delta - \log(1/(\psi\lambda')) \approx \log(t)$. For large t, $\log(e^t - 1) \approx t$, so $\delta - \log(1/(\psi\lambda')) \approx t$. Thus, there are two time regimes, with a smooth transition between them.

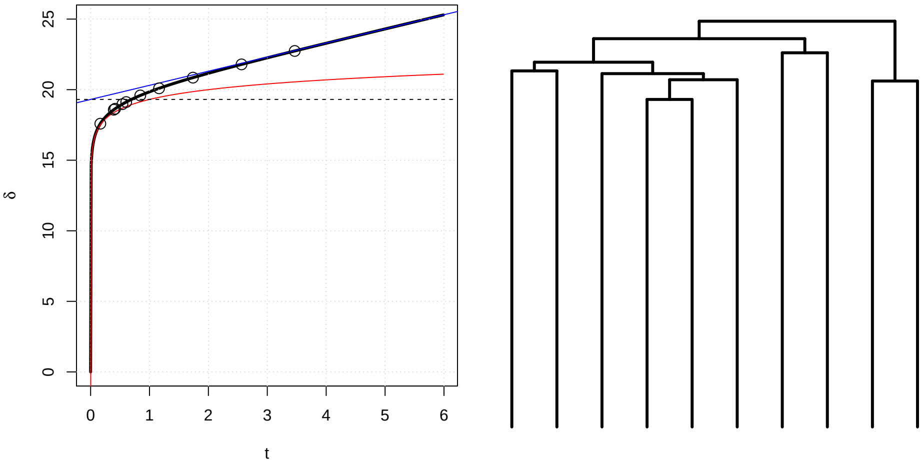

Figure 2.2: Left: solid black line shows time rescaling between δ and t (time units of Y). In blue: line $\delta = t - \log(\psi\lambda')$. In red: curve $\delta = \log(t) - \log(\psi\lambda')$. Dashed black line shows $\delta = -\log(\psi\lambda')$. Dots show simulated event times. Right: tree corresponding to the simulated event times. Parameters used: $\psi = e^{-20}, \lambda' = 2, \mu' = 1$.

This can be understood as follows. Under Bernoulli sampling, the sample size n is of order ψN, where N is the underlying population size in the complete tree. In the limit $\psi \to 0$, N is therefore $\mathcal{O}(\psi^{-1})$; this is very large compared to n, and no coalescences happen for a very long time: the probability of going from n to $n-1$ individuals in time τ is, from (2.3.1):

$$P_{n,n-1}(\tau) = \left(1 - e^{-\rho_\delta(\tau)}\right)\left(e^{-\rho_\delta(\tau)}\right)^{n-1},$$

where

$$e^{-\rho_\delta(\tau)} = \frac{1}{1 + \psi\lambda'(e^\tau - 1)}$$

is the probability of no event happening. This is very close to 1 until τ grows to the order of $\log(1/\psi)$.

With the time transformation above, a step of one unit of t approximately corresponds to taking a time step of $\log(1/\psi)$ in units of δ. At this point, $e^{-\rho_\delta(\tau)} \approx (1 + \lambda')^{-1}$ and the sample starts to coalesce. Then steps in t become roughly equal to steps in δ. In essence, we zoom back to a time when the underlying population was of order n, and then slow back down to linear time.

Figure 2.2 shows an example of the time rescaling (2.5.2) for $\psi = e^{-20}$, $\lambda' = 2$, $\mu' = 1$. The left panel shows δ against t; the horizontal axis is the time scale of the Y process, the vertical axis is the time scale of X^δ_ψ. The red line shows the curve $\delta = \log(t) + \log(1/(\psi\lambda'))$; the blue line shows $\delta = t + \log(1/(\psi\lambda'))$. The circles indicate a set of simulated event times for a sample size $n = 10$. For instance, the time to first event is $T_1 \sim \text{Exp}(n)$ on the horizontal axis; this is rescaled using (2.5.2) to get the corresponding time on the vertical axis. The right panel shows the corresponding RRP tree.

As $\psi \to 0$, $\log(1/(\psi\lambda')) \to \infty$, so the rescaled time of the first event in units of δ grows to infinity, and the reconstructed tree of X^δ_ψ becomes almost star-shaped. The terminal branches dominate the tree, but the inter-event times near the origin of the tree are still approximately exponentially distributed with rate depending on the remaining number of lineages, as the time rescaling for large t is approximately linear.

2.5.3 Density of inter-event times in the limit $\psi \to 0$

The density of inter-event times is now derived analytically, first for any $\psi \in (0, 1]$, then for the limit $\psi \to 0$.

Theorem 2.5.1. *The density of waiting times $W_k = T_{k+1} - T_k$ between events k and $k+1$, $k \in \{0, \ldots, n-1\}$, for the RRP X^δ_ψ with $\psi \in (0, 1]$, is:*

$$f^\psi_{W_k}(w) = \frac{(n-k)}{(n+1)} e^{-(n-k)w} \Big[(n+1) {}_2F_1(n-k+1, n-k+1; n+1; (1-\psi\lambda')(1-e^{-w})) - (1-\psi\lambda')(n-k+1) {}_2F_1(n-k+1, n-k+2; n+2; (1-\psi\lambda')(1-e^{-w})) \Big], \tag{2.5.4}$$

where ${}_2F_1$ is the ordinary hypergeometric function. In the case of a critical branching process with birth and death rate λ and RRP Z_ψ^α, this becomes:

$$\hat{f}^{\psi}_{W_k}(v) = \frac{(n-k+1)(n-k)}{n+1}\psi\lambda \cdot {}_2F_1(n-k+1, n-k+2; n+2; -\psi\lambda v).$$

Proof. In the time-reversed Yule rate 1 process, the density of waiting times between the k-th and $(k+1)$-th event, $k = 0, \ldots, n-1$, conditional on $T_k = s$ is

$$f_{W_k}(t|s) = (n-k)e^{-(n-k)((s+t)-s)} = (n-k)e^{-(n-k)t}. \qquad (2.5.5)$$

Using the time transformation (2.5.1), in units of δ the waiting time is

$$w = \rho_\delta^{-1}(s+t) - \rho_\delta^{-1}(s) = \log\left(\frac{\psi\lambda' + e^{s+t} - 1}{\psi\lambda' + e^s - 1}\right).$$

Rearranging, this gives

$$t = \log\left(e^w\left[1 - (1-\psi\lambda')e^{-s}\right] + (1-\psi\lambda')e^{-s}\right),$$

and

$$\frac{dt}{dw} = \frac{e^w(1-(1-\psi\lambda')e^{-s})}{e^w(1-(1-\psi\lambda')e^{-s}) + (1-\psi\lambda')e^{-s}}.$$

Thus, by using a change of variables in (2.5.5) and writing $\phi = 1 - \psi\lambda'$,

$$\begin{aligned} f^{\psi}_{W_k}(w|s) &= (n-k)e^w\left[1-(1-\psi\lambda')e^{-s}\right]\cdot \\ &\qquad \cdot \left[e^w(1-(1-\psi\lambda')e^{-s}) + (1-\psi\lambda')e^{-s}\right]^{-(n-k+1)} \\ &= (n-k)e^w\left[1-\phi e^{-s}\right]\left[e^w(1-\phi e^{-s}) + \phi e^{-s}\right]^{-(n-k+1)}. \end{aligned}$$

Since s is the time of the k-th event in the time-reversed Yule rate 1 process, it has density given by (2.3.5):

$$f_{T_k}(s) = \binom{n}{k-1}(n-k+1)\left(1-e^{-s}\right)^{k-1}\left(e^{-s}\right)^{n-k+1}.$$

The marginal distribution of W_k is thus

$$\begin{aligned}
f^{\psi}_{W_k}(w) &= \int_0^\infty f^{\psi}_{W_k}(w|s)\, f_{T_k}(s)ds \\
&= \binom{n}{k-1}(n-k+1)(n-k) \underbrace{\int_0^\infty e^w \frac{(1-\phi e^{-s})(1-e^{-s})^{k-1}(e^{-s})^{n-k+1}}{\big(e^w(1-\phi e^{-s})+\phi e^{-s}\big)^{n-k+1}} ds}_{A}\,.
\end{aligned}$$

Integrating using the change of variables $u = e^{-s}$:

$$\begin{aligned}
A &= e^w \int_0^1 \frac{(1-\phi u)(1-u)^{k-1}u^{n-k}}{\big(e^w(1-\phi u)+\phi u\big)^{n-k+1}} du \\
&= e^{-(n-k)w} \int_0^1 \frac{(1-\phi u)(1-u)^{k-1}u^{n-k}}{\big(1-\phi u(1-e^{-w})\big)^{n-k+1}} du \\
&= e^{-(n-k)w} \int_0^1 \left[\frac{(1-u)^{k-1}u^{n-k}}{\big(1-\phi u(1-e^{-w})\big)^{n-k+1}} - \phi \frac{(1-u)^{k-1}u^{n-k+1}}{\big(1-\phi u(1-e^{-w})\big)^{n-k+1}} \right] du.
\end{aligned}$$

Using the following identity for the ordinary hypergeometric function (Abramowitz and Stegun, 1964, p.558)

$$ {}_2F_1(a,b,c,x) = \frac{\Gamma(c)}{\Gamma(c-a)\Gamma(a)} \int_0^1 \frac{(1-t)^{c-a-1}t^{a-1}}{(1-xt)^b} dt $$

gives

$$\begin{aligned}
A = e^{-(n-k)w} \frac{(k-1)!(n-k)!}{(n+1)!} \Big[&(n+1){}_2F_1(n-k+1, n-k+1; n+1; \phi(1-e^{-w})) \\
&- \phi(n-k+1){}_2F_1(n-k+1, n-k+2; n+2; \phi(1-e^{-w})) \Big].
\end{aligned}$$

Thus,

$$\begin{aligned}
f^{\psi}_{W_k}(w) = \frac{(n-k)}{(n+1)} e^{-(n-k)w} \Big[&(n+1){}_2F_1(n-k+1, n-k+1; n+1; (1-\psi\lambda')(1-e^{-w})) \\
- (1-\psi\lambda')(n-k+1){}_2F_1(&n-k+1, n-k+2; n+2; (1-\psi\lambda')(1-e^{-w})) \Big].
\end{aligned}$$

For the RRP of a critical branching process, Z^{α}_{ψ}, the derivation is very similar. Using instead the time transformation

$$ v = \rho_\alpha^{-1}(s+t) - \rho_\alpha^{-1}(s) = \frac{1}{\psi\lambda}\Big[e^{s+t} - 1 - e^s + 1\Big] = \frac{1}{\psi\lambda} e^s(e^t - 1) $$

and following the same steps, the equivalent result is

$$\hat{f}^{\psi}_{W_k}(v) = \frac{(n-k+1)(n-k)}{n+1}\psi\lambda \cdot {}_2F_1(n-k+1, n-k+2; n+2; -\psi\lambda v).$$

□

Note that for $k = 0$, $f^{\psi}_{W_0}(w)$ reduces to the density of the first event, obtained by substituting $k = 1$ in (2.4.3). For $\psi \to 0$, the following holds:

Corollary 2.5.1. *The density of waiting times W_k between events k and $k+1$, $k \in \{1, \ldots, n-1\}$, in the limit $\psi \to 0$, is:*

$$f^{0}_{W_k}(w) = \frac{k(n-k)}{n+1}e^{-(n-k)w}{}_2F_1(n-k+1, n-k+1; n+2; 1-e^{-w}). \quad (2.5.6)$$

This is not a density for $k = 0$, i.e. for the waiting time to the first event: recall that for $\psi \to 0$ the first event time goes to infinity.

Proof. Substituting $\psi = 0$ into (2.5.4):

$$\begin{aligned} f^{0}_{W_k}(w) = \frac{(n-k)}{(n+1)}e^{-(n-k)w}\Big[&(n+1){}_2F_1(n-k+1, n-k+1; n+1; 1-e^{-w}) \\ &- (n-k+1){}_2F_1(n-k+1, n-k+2; n+2; (1-e^{-w}))\Big]. \end{aligned}$$

Identity (15.2.16) of Abramowitz and Stegun (1964, p. 558) gives:

$$\begin{aligned} ac(1-z)\, {}_2F_1(a+1, b; c; z) =& c[a-(c-b)z]\, {}_2F_1(a, b; c; z) + \\ & (c-a)(c-b)z\, {}_2F_1(a, b; c+1; z) \end{aligned} \quad (2.5.7)$$

Substituting $a+1$ instead of a in identity (15.2.20) of Abramowitz and Stegun (1964, p. 558) gives:

$$c(1-z)\, {}_2F_1(a+1, b; c; z) = c\, {}_2F_1(a, b; c; z) - (c-b)z\, {}_2F_1(a+1, b; c+1; z). \quad (2.5.8)$$

Multiplying (2.5.8) by a, equating with (2.5.7) and simplifying gives:

$$c\, {}_2F_1(a, b; c; z) - a\, {}_2F_1(b, a+1; c+1; z) = (c-a)\, {}_2F_1(a, b; c+1; z).$$

Thus,

$$f^{0}_{W_k}(w) = \frac{k(n-k)}{(n+1)}e^{-(n-k)w}{}_2F_1(n-k+1, n-k+1; n+2; 1-e^{-w}).$$

□

Note that using the transformation (Erdélyi et al., 1953, p.64)

$$ {}_2F_1(a,b;c;z) = (1-z)^{c-a-b}{}_2F_1(c-a,c-b;c;z), $$

the densities of the k-th and $(n-k)$-th waiting times are equal:

$$ \begin{aligned} f^0_{W_k}(w) &= \frac{k(n-k)}{n+1} e^{-(n-k)w}{}_2F_1(n-k+1,n-k+1;n+2;1-e^{-w}) \\ &= \frac{k(n-k)}{n+1} e^{-(n-k)w}e^{-(2k-n)w}{}_2F_1(k+1,k+1;n+2;1-e^{-w}) \\ &= \frac{k(n-k)}{n+1} e^{-kw}{}_2F_1(k+1,k+1;n+2;1-e^{-w}) \qquad (2.5.9) \\ &= f^0_{W_{n-k}}(w). \end{aligned} $$

This is an interesting property of the RRP tree in the limit. The inter-event times are symmetric, for instance the time it takes to go from $n-1$ to $n-2$ lineages, and the time it takes for the last lineage to die, have the same distribution.

To gain some insights into why this is true, consider the event times of the time-reversed Yule rate 1 process, which are distributed as the order statistics of n exponential rate 1 random variables, say $X_1 \leq X_2 \leq \ldots \leq X_n$. The form of equation (2.5.3) implies that in the limit $\psi \to 0$, the k-th event time T_k can be obtained via the transformation $T_k = \log(1/(\psi\lambda')) + \log\big(e^{X_k}-1\big)$. If $X \sim \text{Exp}(1)$, then $\log\big(e^X - 1\big)$ has the standard logistic distribution (George and Mudholkar, 1981). It thus follows that, in the limit, the shifted event time defined as $T'_k := T_k - \log(1/(\psi\lambda'))$ is distributed as the k-th order statistic of n draws from the standard logistic distribution, which has pdf

$$ f^0_{T'_{(1)}}(\tau') = \frac{e^{\tau'}}{(1+e^{\tau'})^2}. \qquad (2.5.10) $$

Note that this is equivalent to saying that T_k is distributed as the k-th order statistic of n draws from the logistic distribution with location parameter (mode) $\log(1/(\psi\lambda'))$ and scale 1. The same conclusion can also be reached by considering the CPP density (2.4.5), writing $\tau = \tau' + \log(1/(\psi\lambda'))$ and taking the limit $\psi \to 0$, which gives the density (2.5.10).

The limiting density of T'_k can also be obtained by applying the rescaling

$\delta = (\lambda - \mu)\gamma$ and writing $\lambda' = \frac{\lambda}{\lambda - \mu}$ in the density (2.4.3),

$$f_{T_k}^{\psi}(\tau) = \binom{n}{k} k \, \frac{\psi\lambda' e^{\tau}[\psi\lambda'(e^{\tau} - 1)]^{k-1}}{[1 + \psi\lambda'(e^{\tau} - 1)]^{n+1}},$$

writing $T_k' = T_k - \log(1/(\psi\lambda'))$ and taking the limit $\psi \to 0$ gives

$$f_{T_k'}^{0}(\tau') = \lim_{\psi \to 0} \binom{n}{k} k \, \frac{e^{\tau'}[e^{\tau'} - \psi\lambda']^{k-1}}{[1 + e^{\tau'} - \psi\lambda']^{n+1}} = \binom{n}{k} k \, \frac{[e^{\tau'}]^{k}}{[1 + e^{\tau'}]^{n+1}}, \qquad (2.5.11)$$

which, again, is the density of the k-th order statistic for the standard logistic distribution.

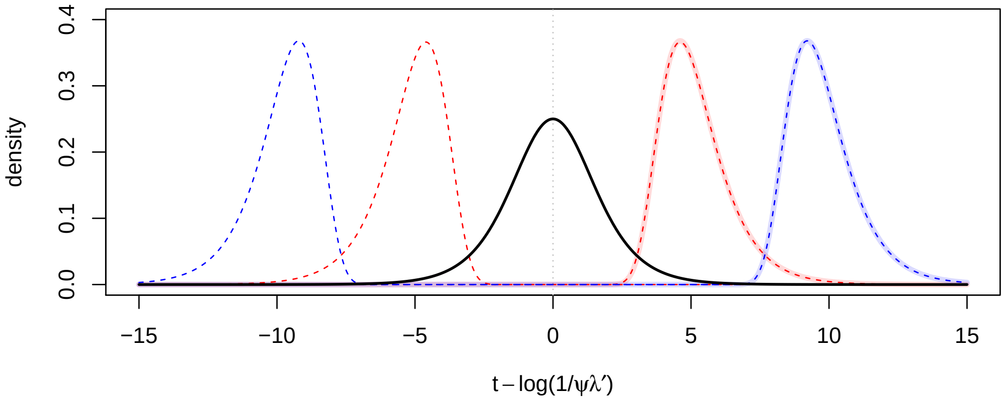

Figure 2.3: x-axis shows time shifted by $\log(1/(\psi\lambda'))$. Black solid line: standard logistic density (2.5.10). Dashed lines: density (2.5.11) of shifted time to first and last event for $n = 100$ (red) and $n = 10\,000$ (blue). Faint solid lines: Gumbel density with parameters $(\log n, 1)$ for $n = 100$ (red) and $n = 10\,000$ (blue).

As the logistic density (2.5.10) is symmetric around 0, the order statistics are also symmetric, with $T_k' \overset{d}{=} -T_{n-k+1}'$ (Arnold et al., 1992, pp. 26). This is illustrated in Figure 2.3: the black solid line shows the logistic density (2.5.10), and the red (blue) dashed lines show the densities of the first and last event times for $n = 100$ ($n = 10\,000$). Thus, the densities of the event times T_k and T_{n-k+1} are symmetric around $\log(1/(\psi\lambda'))$.

Moreover, as $T_{k+1}' \overset{d}{=} -T_{n-k}'$, this demonstrates that the inter-event times $W_k = T_{k+1} - T_k = T_{k+1}' - T_k'$ and $W_{n-k} = T_{n-k+1} - T_{n-k} = T_{n-k+1}' - T_{n-k}'$ are equal in distribution. The density derived in Corollary 2.5.1 is hence that of the gap between the k-th and $(k+1)$-th order statistic of n standard logistic random variables. See, for instance, Mahmuod and Ragab (1973, pp. 84): their equation (4.1) gives the density of the gap between the k-th and $(k+1)$-th order statistics for the logistic distribution, which appears in very different form, but becomes the density in Corollary 2.5.1 after some algebra. I am not aware of a simpler expression for this particular density.

Corollary 2.5.2. *The distribution function of the waiting time W_k between events k and $k+1$, $k \in \{1, \dots, n-1\}$, with $\psi \to 0$, is given by:*

$$F^0_{W_k}(w) = 1 - e^{-kw}{}_2F_1(k, k+1; n+1; 1-e^{-w}). \tag{2.5.12}$$

Proof. By integrating the density in (2.5.9):

$$F^0_{W_k}(w) = \frac{k(n-k)}{n+1} \int_0^w e^{-ku}{}_2F_1(k+1, k+1; n+2; 1-e^{-u})du$$

$$\begin{aligned}
&= \frac{k(n-k)}{n+1} \int_0^w e^{-u}e^{-(k-1)u}{}_2F_1(k+1, k+1; n+2; 1-e^{-u})du \\
&= \frac{k(n-k)}{n+1} \int_0^{1-e^{-w}} (1-z)^{k-1}{}_2F_1(k+1, k+1; n+2; z)dz \\
&= \frac{k(n-k)}{n+1} \Big[- \frac{(1-z)^k(n+1)}{k(n-k)}\; {}_2F_1(k, k+1; n+1; z) \Big]_0^{1-e^{-w}} \\
&= 1 - e^{-kw}{}_2F_1(k, k+1; n+1; 1-e^{-w}),
\end{aligned}$$

having used the substitution $z = 1 - e^{-u}$, and the identity (Erdélyi et al., 1953, p.102, eq. (25) with $n = 1$)

$$\int^z (1-x)^{a-2}{}_2F_1(a, b, c, x)\, dx = \frac{c-1}{(a-1)(b-c+1)}(1-z)^{a-1}{}_2F_1(a-1, b, c-1, z).$$

□

Another interesting property of this distribution is that it does not depend on the scaled birth rate $\lambda' = \frac{\lambda}{\lambda - \mu}$, as this parameter only appears as a factor in $\psi\lambda'$. In particular, take $\lambda' = 1 \implies \mu' = 0$. Thus, the inter-event times for the RRP X^δ_ψ have the same distributions as those of an incompletely sampled time-reversed Yule rate 1 process, in the limit $\psi \to 0$.

2.5.4 Time to origin

The distribution of shifted time to origin $T'_n = T_n - \log(1/(\psi\lambda'))$ is now considered in the limit $\psi \to 0$. Integrating the density in (2.5.11) for $k = n$, the distribution function of T'_n is given by

$$F^0_{T'_n}(\tau') = (1 + e^{-\tau'})^{-n}.$$

As n increases, the density of time to origin shifts to the right, away from $\log(1/(\psi\lambda'))$, so with high probability T'_n is much larger than 0. Figure 2.3 demonstrates this visually with examples of the density of T'_n for $n = 100$ and $n = 10\ 000$. Thus, for n large enough, this justifies introducing the approximation $1 + e^{-\tau'} \approx \exp\bigl(e^{-\tau'}\bigr)$, so the distribution of shifted time to origin can be approximated by

$$\widetilde{F}^0_{T'_n}(\tau') = \left[\exp\left(e^{-\tau'}\right)\right]^{-n} = \exp\left(-e^{-(\tau' - \log n)}\right).$$

This is a Gumbel distribution with location parameter (mode) $\log n$ and scale parameter 1. Figure 2.3 shows that this approximation provides a good fit, for $n = 100$ and $n = 10\ 000$.

This links to the results of Burden and Soewongsono (2019), who consider the diffusion limit (as the population size grows to infinity) of a near-critical Bienaymé-Galton-Watson process. Burden and Soewongsono (2019, Section 6) calculate numerically and plot the distribution of time to the MRCA, similarly shifted by the log of the population size at the time of sampling, and comment that as $n \to \infty$ this appears to converge to what looks like a Gumbel distribution. I have shown, analytically, that in the case of a supercritical birth-death process in the limit as $\psi \to 0$, the time to origin shifted by $\log(1/(\psi\lambda'))$ also converges to a Gumbel distribution, and in this case the location parameter depends on n.

2.5.5 Exponential approximation of inter-event times

Although Corollary 2.5.2 completely solves the question of what is the distribution of W_k as $\psi \to 0$, the appearance of ${}_2F_1$ in (2.5.12) somewhat obscures our insight into W_k. Here, it is shown that these waiting times are well approximated by exponential distributions with simple, time-homogeneous event rates.

Consider an exponential approximation to $f^0_{W_k}(w)$ with rate $k(n-k)/n$:

$$\widetilde{f}^0_{W_k}(w) = \frac{k(n-k)}{n}\exp\biggl(-\frac{k(n-k)}{n}w\biggr), \quad \widetilde{F}^0_{W_k}(w) = 1-\exp\biggl(-\frac{k(n-k)}{n}w\biggr), \tag{2.5.13}$$

for $k \in \{1,\dots,n-1\}$. The following result quantifies the accuracy of this approximation:

Proposition 2.5.1. *Suppose the waiting time distribution W_k, with distribution function* (2.5.12) *for $\psi \to 0$, is approximated by an exponential distribution* (2.5.13)*. Then the approximation error is bounded, uniformly in k, in terms of Kolmogorov-Smirnov distance:*

$$\sup_w \Bigl|F^0_{W_k}(w) - \widetilde{F}^0_{W_k}(w)\Bigr| < \frac{1}{n}.$$

Proof. Noting that

$$e^{-kw} = \exp\biggl(-\frac{k(n-k)}{n}w\biggr)\cdot\exp\biggl(-\frac{k^2}{n}w\biggr),$$

it follows that

$$\begin{aligned}
&\bigl|\widetilde{F}^0_{W_k}(w) - F^0_{W_k}(w)\bigr| = \\
&\quad = \left|1 - e^{-kw}{}_2F_1(k,k+1;n+1;1-e^{-w}) - 1 + \exp\biggl(-\frac{k(n-k)}{n}w\biggr)\right| \\
&\quad = \exp\biggl(-\frac{k(n-k)}{n}w\biggr)\cdot\left|\underbrace{\exp\biggl(-\frac{k^2}{n}w\biggr){}_2F_1(k,k+1;n+1;1-e^{-w})}_{=:\,h(w)} - 1\right|.
\end{aligned} \tag{2.5.14}$$

An upper bound on the maximum of this distance is required. The first exponential term decays rapidly to 0, while $h(0) = 1$ and h initially increases; the global maximum of h occurs near $w = 0$, where $h(w) - 1 \geq 0$. First, an upper bound is obtained for $h(w) - 1$, and then this is used to obtain an upper bound on (2.5.14). Using the mean value theorem (or, equivalently, Taylor's theorem to first order):

$$h(w) = h(0) + wh'(c) = 1 + wh'(c)$$

for some $c \in (0, w)$, with

$$\begin{aligned} h'(c) = &-\frac{k^2}{n}\exp\left(-\frac{k^2}{n}c\right){}_2F_1(k, k+1; n+1; 1-e^{-c}) \\ &+ \exp\left(-\frac{k^2+n}{n}c\right).\frac{k(k+1)}{n+1}{}_2F_1(k+1, k+2; n+2; 1-e^{-c}). \end{aligned}$$

Differentiating once more and considering the sign of the second derivative, $h''(0) < 0$, so h' has a maximum at $c = 0$; h' has no other extrema before it reaches 0. Thus,

$$h'(0) = -\frac{k^2}{n} + \frac{k(k+1)}{n+1} = \frac{k(n-k)}{n(n+1)},$$

so an upper bound on $h(w) - 1$ is given by

$$h(w) - 1 \leq \frac{k(n-k)}{n(n+1)}w.$$

Substituting this into (2.5.14),

$$\begin{aligned} \left|\widetilde{F}^0_{W_k}(w) - F^0_{W_k}(w)\right| &\leq \exp\left(-\frac{k(n-k)}{n}w\right) \cdot \big(h(w) - 1\big) \\ &\leq \exp\left(-\frac{k(n-k)}{n}w\right) \cdot \frac{k(n-k)}{n(n+1)}w. \end{aligned} \qquad (2.5.15)$$

This attains the maximum at $\hat{w} = \frac{n}{k(n-k)}$. Substituting this into (2.5.15),

$$\left|\widetilde{F}^0_{W_k}(w) - F^0_{W_k}(w)\right| \leq \frac{1}{e(n+1)} < \frac{1}{n}.$$

The approximation error is thus bounded by $\frac{1}{n}$. □

The density derived in Corollary 2.5.1 is nonintuitive, however this result shows that up to an error bounded by $1/n$, the distribution is actually approximately exponential. Note that the particular form of the exponential rate is such that $\widetilde{f}^0_{W_k}(w) = \widetilde{f}^0_{W_{n-k}}(w)$, so the symmetry between the k-th and $(n-k)$-th inter-event times is preserved in the approximation. Figure 2.4 shows an example of the (exact) density (2.5.4), for $\psi = 1$ on the left and very small ψ on the right; dotted lines in the latter case show the exponential approximations (2.5.13), demonstrating very close agreement for $n = 100$.

Wiuf (2018) gives results for the expectation of time to origin, and recur-

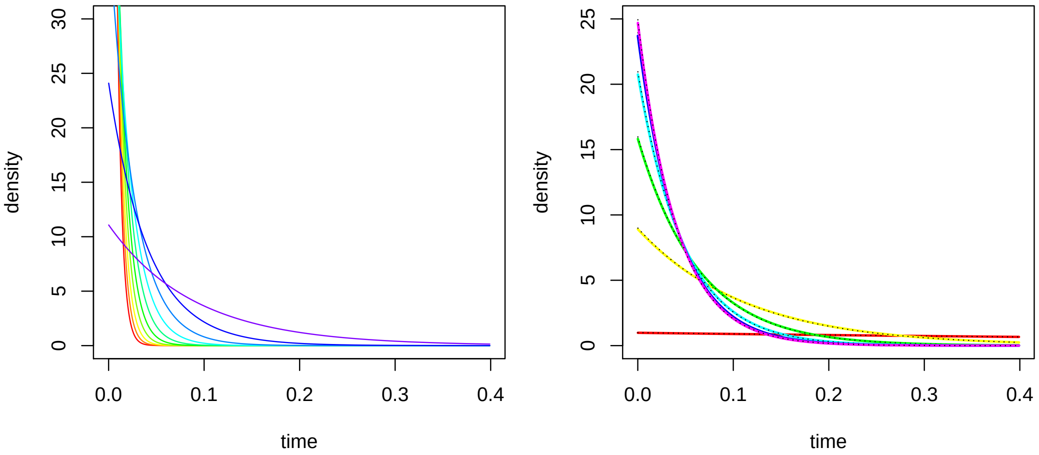

Figure 2.4: Inter-event time density, $n = 100$, $\lambda' = 2$, $\mu' = 1$. Left: with $\psi = 1$, colours (red to purple) correspond to event numbers $k = 0, 10, \ldots, 90$. Right: with $\psi = e^{-20}$, colours (red to purple) correspond to event numbers $k = 1, 10, 20, \ldots, 50$; dotted lines show exponential approximation (2.5.13). Note that the dotted lines overlay the coloured lines very closely.

sions for calculating the expectation of the other event times, for the RRP with Bernoulli sampling (not in the limit $\psi \to 0$). These results can be used to show that the expectation under the exponential approximation, being $n/(k(n-k))$, is *exact* in the limit $\psi \to 0$ (for any n).

Proposition 2.5.2. *The expectation of time to origin for $\psi \to 0$ is given by:*

$$\mathbb{E}(T_n) = \log\left(\frac{1}{\psi\lambda'}\right) + \sum_{j=1}^{n-1} \frac{1}{j} + \mathcal{O}(\psi). \tag{2.5.16}$$

Proof. Wiuf (2018, Appendix F) derives an expression for the expectation of time to origin, which in the present notation is

$$\mathbb{E}(T_n) = \log\left(\frac{1}{\psi\lambda'}\right) + \sum_{i=1}^{n} \frac{1}{i} - \sum_{i=1}^{n} \frac{1}{i} \frac{1}{\left(1 - \frac{1}{\psi\lambda'}\right)^{n-i}} - \frac{1}{(1 - \frac{1}{\psi\lambda'})^{n}} \log\left(\frac{1}{\psi\lambda'}\right). \tag{2.5.17}$$

The third term is

$$\sum_{i=1}^{n} \frac{1}{i} \frac{1}{\left(1 - \frac{1}{\psi\lambda'}\right)^{n-i}} = \frac{1}{n} + \frac{1}{n-1} \frac{1}{1 - \frac{1}{\psi\lambda'}} + \frac{1}{n-2} \left(\frac{1}{1 - \frac{1}{\psi\lambda'}}\right)^2 + \ldots = \frac{1}{n} + \mathcal{O}(\psi).$$

The fourth term in (2.5.17) is

$$\begin{aligned}\frac{1}{(1-\frac{1}{\psi\lambda'})^n}\log\left(\frac{1}{\psi\lambda'}\right) &= (-\psi\lambda')^n(1-\psi\lambda')^{-n}\log\left(\frac{1}{\psi\lambda'}\right)\\ &= -(-\psi\lambda')^n[1+\mathcal{O}(\psi\lambda')]\log(\psi\lambda')\\ &= \mathcal{O}((\psi\lambda')^n\log(\psi\lambda)),\end{aligned}$$

which is $\mathcal{O}(\psi)$ for $n > 1$. Thus, in the limit $\psi \to 0$,

$$\mathbb{E}(T_n) = \log\left(\frac{1}{\psi\lambda'}\right) + \sum_{i=1}^{n-1}\frac{1}{i} + \mathcal{O}(\psi).$$

□

This is an illuminating result, as the expectation is split into two parts. The first is $\log(1/(\psi\lambda'))$, corresponding to the first time rescaling regime, as described in Section 2.5.2. Near 0, a small step in t is equivalent to a step of order $\log(1/(\psi\lambda'))$ in units of δ. The second part is equivalent to the expectation of a sum of $n-1$ exponential waiting times with rate being the remaining number of lineages, corresponding to the second time rescaling regime, which is approximately linear.

This result agrees with the discussion in Section 2.5.3: recall that in the limit $\psi \to 0$, the shifted event time T'_k is distributed as the k-th order statistic of n standard logistic random variables, so $T_k = T'_k + \log(1/(\psi\lambda'))$ has expectation

$$\sum_{j=1}^{k-1}\frac{1}{j} - \sum_{j=1}^{n-k}\frac{1}{j} + \log\left(\frac{1}{\psi\lambda'}\right), \tag{2.5.18}$$

obtained by simplifying equation (4.8.6) in Arnold et al. (1992, p.82). Setting $k = n$, this becomes (2.5.16) up to the $\mathcal{O}(\psi)$ term. Notice also that using the Gumbel approximation for large n, as described in Section 2.5.4, gives the expectation of T'_n as $\log n + \widetilde{\gamma}$ (where $\widetilde{\gamma}$ is the Euler–Mascheroni constant). This is the limit of the harmonic sum in (2.5.16) as $n \to \infty$, so the expectations agree in this limit.

Wiuf (2018, Appendix D) derives a recursion for the expectations of event times, which in the present notation is:

$$\mathbb{E}_n(T_k) = \frac{n}{n-k}\mathbb{E}_{n-1}(T_k) - \frac{k}{n-k}\mathbb{E}_n(T_{k+1}), \tag{2.5.19}$$

where $\mathbb{E}_n(T_k)$ denotes the expectation of the k-th event time if the sample is of size n at time 0. Using this and the expression for time to origin given by Proposition 2.5.2, the following result is obtained:

Proposition 2.5.3. *The expectation of waiting times between events is given by*

$$\mathbb{E}(W_k) = \mathbb{E}(T_{k+1}) - \mathbb{E}(T_k) = \frac{n}{k(n-k)} + \mathcal{O}(\psi).$$

This agrees exactly with the expectation using the exponential approximation for $\psi \to 0$. This also agrees, up to the $\mathcal{O}(\psi)$ term, with the expectation of $T_{k+1} - T_k$ obtained using (2.5.18) in the limit $\psi \to 0$.

Proof. From Proposition 2.5.2, the expectation of time to origin for a sample of size n is:

$$\mathbb{E}_n(T_n) = \log\left(\frac{1}{\psi\lambda'}\right) + \sum_{j=1}^{n-1}\frac{1}{j} + \mathcal{O}(\psi),$$

which also implies that, for a sample of size $n-1$,

$$\mathbb{E}_{n-1}(T_{n-1}) = \log\left(\frac{1}{\psi\lambda'}\right) + \sum_{j=1}^{n-2}\frac{1}{j} + \mathcal{O}(\psi).$$

I proceed by induction on the event number k, to show that

$$\mathbb{E}_n(T_k) = \log\left(\frac{1}{\psi\lambda'}\right) + \sum_{j=1}^{n-1}\frac{1}{j} - \sum_{j=k}^{n-1}\frac{n}{j(n-j)} + \mathcal{O}(\psi). \tag{2.5.20}$$

This holds for event number $k = n-1$, as using (2.5.19):

$$\begin{aligned}
\mathbb{E}_n(T_{n-1}) &= n\mathbb{E}_{n-1}(T_{n-1}) - (n-1)\mathbb{E}_n(T_n) \\
&= n\left(\log\left(\frac{1}{\psi\lambda'}\right) + \sum_{j=1}^{n-1}\frac{1}{j} - \frac{1}{n-1}\right) - \\
&\qquad - (n-1)\left(\log\left(\frac{1}{\psi\lambda'}\right) + \sum_{j=1}^{n-1}\frac{1}{j}\right) + \mathcal{O}(\psi) \\
&= \log\left(\frac{1}{\psi\lambda'}\right) + \sum_{j=1}^{n-1}\frac{1}{j} - \frac{n}{n-1} + \mathcal{O}(\psi).
\end{aligned}$$

Suppose that (2.5.20) holds for some $k = n - i$, $i \in \{1, \dots, n-1\}$:

$$\mathbb{E}_n(T_{n-i}) = \log\left(\frac{1}{\psi\lambda'}\right) + \sum_{j=1}^{n-1}\frac{1}{j} - \sum_{j=1}^{i}\frac{n}{j(n-j)} + \mathcal{O}(\psi),$$

and so, equivalently, for $n-1$ lineages:

$$\mathbb{E}_{n-1}(T_{n-i-1}) = \log\left(\frac{1}{\psi\lambda'}\right) + \sum_{j=1}^{n-2}\frac{1}{j} - \sum_{j=1}^{i}\frac{n-1}{j(n-j-1)} + \mathcal{O}(\psi).$$

Then

$$\begin{aligned}
&\mathbb{E}_n(T_{n-i-1}) = \\
&= \frac{n}{i+1}\mathbb{E}_{n-1}(T_{n-i-1}) - \frac{n-i-1}{i+1}\mathbb{E}_n(T_{n-i}) \\
&= \underbrace{\log\left(\frac{1}{\psi\lambda'}\right) + \sum_{j=1}^{n-1}\frac{1}{j} + \mathcal{O}(\psi)}_{A} \\
&\qquad - \frac{n}{i+1}\left[\frac{1}{n-1} + \sum_{j=1}^{i}\frac{n-1}{j(n-j-1)} - (n-i-1)\sum_{j=1}^{i}\frac{1}{j(n-j)}\right] \\
&= A - \frac{n}{i+1}\left[\frac{1}{n-1} + \sum_{j=1}^{i}\left(\frac{1}{j} + \frac{1}{n-j-1}\right) - \frac{(n-i-1)}{n}\sum_{j=1}^{i}\left(\frac{1}{j} + \frac{1}{n-j}\right)\right] \\
&= A - \frac{1}{i+1}\left[\frac{n}{n-1} + (i+1)\sum_{j=1}^{i}\frac{1}{j} + n\sum_{j=1}^{i}\frac{1}{n-j-1} - (n-i-1)\sum_{j=1}^{i}\frac{1}{n-j}\right] \\
&= A - \frac{1}{i+1}\left[(i+1)\sum_{j=1}^{i}\frac{1}{j} + n\sum_{j=2}^{i}\frac{1}{n-j} + \frac{n}{n-i-1}\right. \\
&\qquad \left. - (n-i-1)\sum_{j=2}^{i}\frac{1}{n-j} + \frac{i+1}{n-1}\right] \\
&= A - \frac{1}{i+1}\left[(i+1)\sum_{j=1}^{i}\frac{1}{j} + (i+1)\sum_{j=1}^{i}\frac{1}{n-j} + \frac{(n-i-1)+(i+1)}{n-i-1}\right] \\
&= A - \frac{1}{i+1}\left[(i+1)\sum_{j=1}^{i+1}\frac{1}{j} + (i+1)\sum_{j=1}^{i+1}\frac{1}{n-j}\right] \\
&= \log\left(\frac{1}{\psi\lambda'}\right) + \sum_{j=1}^{n-1}\frac{1}{j} - \sum_{j=1}^{i+1}\frac{n}{j(n-j)} + \mathcal{O}(\psi).
\end{aligned}$$

Thus,

$$\begin{aligned}
\mathbb{E}_n(T_k) &= \log\left(\frac{1}{\psi\lambda'}\right) + \sum_{j=1}^{n-1}\frac{1}{j} - \sum_{j=1}^{n-k}\frac{n}{j(n-j)} + \mathcal{O}(\psi) \\
&= \log\left(\frac{1}{\psi\lambda'}\right) + \sum_{j=1}^{n-1}\frac{1}{j} - \sum_{j=k}^{n-1}\frac{n}{j(n-j)} + \mathcal{O}(\psi),
\end{aligned}$$

$$\implies \mathbb{E}(W_k) = \mathbb{E}(T_{k+1}) - \mathbb{E}(T_k) = \frac{n}{k(n-k)} + \mathcal{O}(\psi).$$

□

2.6 Connections to the coalescent with exponential growth in the large population limit

Looking backwards in time, the mean population size in the complete birth-death process decreases exponentially, with rate $\lambda - \mu$. Several studies have therefore sought to compare the properties of genealogies generated under the birth-death population model with those arising under the coalescent with exponential growth. Volz and Frost (2014) considered maximum likelihood estimates of the growth rate for simulated birth-death and coalescent trees, and found them to give very similar results. Boskova et al. (2014) used Bayesian estimation of birth-death parameters to compare the results of inference with birth-death and coalescent priors; the study found differences due to the coalescent having on average "longer" trees, although this effect appears to reduce as the sampling probability decreases.

Stadler et al. (2015) considered the distribution of the coalescence time for a sample of size two, comparing the birth-death model to the coalescent with exponential growth (finding them to be very different), and to some extensions of the coalescent incorporating stochastic population size trajectories. The method was to fix a time of origin for the birth-death process to be the time at which the *expected* population size for a birth-death tree is N, and use this to derive the event time density. This was then compared to the event time density for the coalescent with exponential growth with an initial population of size N. This approach may be problematic, however, as it induces an inherent difference in the models being considered: the total population size in the birth-death process at the time of sampling is stochastic, so is not directly comparable to the coalescent with a fixed initial population size. Conditional on n and ψ, the number of unsampled lineages $N-n$ is negative binomial (as this gives the number of failures before n successes with probability ψ). Thus, the total population size has density

$$f(N|n,\psi) = \binom{N-1}{n-1}\psi^n(1-\psi)^{N-n}. \tag{2.6.1}$$

A proof of this result using direct density calculations can be found in Stadler (2009, proof of Lemma 3.1).

In this section, the time to first event for the RRP in the $\psi \to 0$ limit is shown to be approximately Gompertz distributed, as is the time to first event for the coalescent with exponential growth, allowing for the two densities to be equated by matching up the parameters. The difference in event times between the two models is then investigated. Then, the approximate expected inter-event times derived in Section 2.5.5 are used to calculate the expected site frequency spectrum for the RRP, under the infinite sites assumption.

2.6.1 Time of first event

Define the time transformation

$$g_K(\tau) = \frac{1}{b}\log(1+abN\tau), \tag{2.6.2}$$

$$g_K^{-1}(\tau) = \frac{1}{abN}(e^{b\tau}-1). \tag{2.6.3}$$

This is the usual rescaling for the coalescent with exponential growth (Slatkin and Hudson, 1991), with growth rate b, generation time a and initial population size N. Event times for the coalescent with exponential growth can be simulated by drawing the sequence $t_1,\dots,t_{n-1}$ of inter-event times for the standard coalescent, setting $v_i = \sum_{j=1}^{i} t_i$, and then rescaling using (2.6.2) to get the event times $\tilde{v}_i = g_K(v_i)$ (e.g. Hein et al., 2004, Section 4.3).

The distribution of the time to first event is then

$$\begin{aligned} F_{T_1}^K(\tau) = \mathbb{P}(T_1 \le \tau) &= 1-\exp\left(-\binom{n}{2}\frac{1}{abN}(e^{b\tau}-1)\right) \\ &= 1-\exp\left(-\frac{n(n-1)}{2abN}(e^{b\tau}-1)\right). \end{aligned} \tag{2.6.4}$$

This is a Gompertz distribution with shape parameter $\frac{n(n-1)}{2abN}$ and scale b, which is a well known result for the coalescent with exponential growth (Slatkin and Hudson, 1991; Polanski et al., 2003).

For the RRP X^{γ}_{ψ}, the time to first event has distribution

$$F^{\psi}_{T_1}(\tau) = 1 - \left(1 + \frac{\psi\lambda}{\lambda - \mu}(e^{(\lambda-\mu)\tau} - 1)\right)^{-n}, \tag{2.6.5}$$

recalling the results of Section 2.4.1.1. Using the approximation $1 + x \approx e^x$ for small x, in the limit $\psi \to 0$

$$F^{\psi}_{T_1}(\tau) \approx 1 - \exp\left(-\frac{n\psi\lambda}{\lambda - \mu}(e^{(\lambda-\mu)\tau} - 1)\right), \tag{2.6.6}$$

this is also a Gompertz distribution. Setting the growth rate to be $b = \lambda - \mu$, the generation time to be $a = 1/(2\lambda)$ (deduced to be the correct scaling by Volz et al., 2009), and the initial population size to be $N = (n-1)/\psi$ equates the distribution of time to first event for the RRP (2.6.6) to that for the coalescent with exponential growth (2.6.4). Thus, the idea of equating the two models by connecting the sampling probability of the RRP to the population size for the coalescent is explored next.

2.6.2 Effective sampling probability for the RRP

Consider the RRP $X^{\gamma}_{\psi_0}$, with parameters (λ, μ, ψ_0). The RRP is related to the complete tree at time 0 by the sampling probability ψ_0, which gives a sense of how the sample size compares to the total population size (which has density (2.6.1)). After some time $\tau > 0$ has passed, the parameters λ and μ remain fixed, but the relationship between the number of extant lineages in the sample and the total population size will have changed. This can be captured by defining the *effective* sampling probability at time τ, denoted $\widetilde{\psi}(\tau)$, with $\widetilde{\psi}(0) = \psi_0$ and $\widetilde{\psi}(s) = 1$ as $s \to \infty$ (as the process becomes equivalent to a completely sampled process in the limit, when all non-sampled lineages die out). In essence, this captures the Markov property of the process: after the first coalescence event, the RRP can be restarted with $n - 1$ lineages and a different sampling probability $\psi_1 \neq \psi_0$.

This is similar to a result derived by Wiuf (2018, Section 6), with the following distinction: in Wiuf's parametrisation, the sampling probability approaches $\widetilde{\psi}(s) = \frac{\lambda}{\lambda-\mu}$ as $s \to \infty$, because the limiting process for the RRP is implicitly taken to be the time-reversed Yule rate 1 process (rather than the

RRP with complete sampling). The proof by Wiuf (2018) considers joint densities of event times to derive this property; here, it follows from considering the properties of the time rescaling between X_ψ^γ and the completely sampled RRP X_1^β.

Consider the time rescaling between the RRPs X_1^β and X_ψ^γ, given by (2.4.7):

$$\gamma = g_\psi(\beta) = \frac{1}{\lambda - \mu} \log\left(1 + \frac{1}{\psi}(e^{(\lambda-\mu)\beta} - 1)\right) \tag{2.6.7}$$

$$\beta = g_\psi^{-1}(\gamma) = \frac{1}{\lambda - \mu} \log\bigl(1 + \psi(e^{(\lambda-\mu)\gamma} - 1)\bigr) \tag{2.6.8}$$

If w_0 is the waiting time to the first event on the time scale of X_1^β, it must be that ψ_0 and the new effective sampling probability ψ_1 are related by

$$g_{\psi_0}(w_0 + s) - g_{\psi_0}(w_0) = g_{\psi_1}(s),$$

for all $s > 0$. Substituting into (2.6.7),

$$\log\left(\frac{1 + \frac{1}{\psi_0}(e^{(\lambda-\mu)(w_0+s)} - 1)}{1 + \frac{1}{\psi_0}(e^{(\lambda-\mu)(w_0)} - 1}\right) = \log\left(1 + \frac{1}{\psi_1}(e^{(\lambda-\mu)s} - 1)\right),$$

$$\frac{\psi_0 + e^{(\lambda-\mu)(w_0+s)} - 1}{\psi_0 + e^{(\lambda-\mu)w_0} - 1} = 1 + \frac{1}{\psi_1}(e^{(\lambda-\mu)s} - 1).$$

Solving this,

$$\psi_1 = 1 - \frac{1 - \psi_0}{e^{(\lambda-\mu)w_0}}, \tag{2.6.9}$$

which can be verified by substitution. Note that this is self-consistent, in the sense that if the waiting time to the second event is w_1, the effective sampling probability at the time of the second event is

$$\psi_2 = 1 - \frac{1 - \psi_1}{e^{(\lambda-\mu)w_1}} = 1 - \frac{1 - 1 + \frac{1-\psi_0}{e^{(\lambda-\mu)w_0}}}{e^{(\lambda-\mu)w_1}} = 1 - \frac{1 - \psi_0}{e^{(\lambda-\mu)(w_0+w_1)}}, \tag{2.6.10}$$

and in general

$$\psi_k = 1 - \frac{1 - \psi_0}{e^{(\lambda-\mu)t_k}}, \tag{2.6.11}$$

where $t_k = w_0 + \ldots + w_{k-1}$ is the time of the k-th event on the time scale of X_1^β. More generally, at any time t measured in time units β, the effective sampling probability is

$$1 - \frac{1 - \psi_0}{e^{(\lambda-\mu)t}}. \tag{2.6.12}$$

Finally, applying the rescaling (2.6.8) to (2.6.12), the effective sampling probability at time τ on the time scale of X_{ψ}^{γ} is

$$\widetilde{\psi}(\tau) = 1 - \frac{1-\psi_0}{1+\psi_0(e^{(\lambda-\mu)\tau}-1)} = \frac{\psi_0 e^{(\lambda-\mu)\tau}}{1+\psi_0(e^{(\lambda-\mu)\tau}-1)}. \qquad (2.6.13)$$

This is equal to ψ_0 at $\tau = 0$ and tends to 1 as $\tau \to \infty$, as expected. Figure 2.5 shows a plot of this for $\psi_0 = e^{-20}$. The effective sampling probability starts increasing from near 0 by the expected time of the first event (red dotted line), so the total population at this stage is very large. At the expected time of origin (blue dotted line), the effective sampling probability is close to 1, so the total population size is close to the remaining number of sampled lineages. The effective sampling probability is 0.5 at approximately $-\log(\psi_0)/(\lambda-\mu)$.

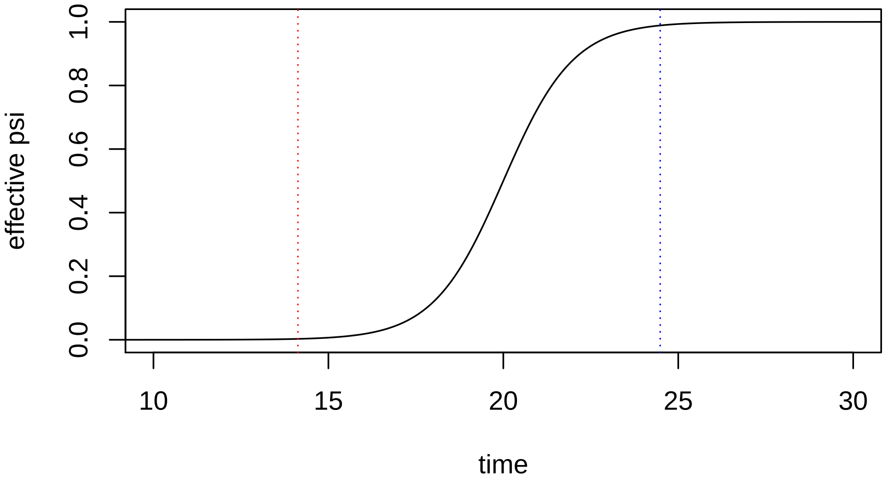

Figure 2.5: Effective sampling probability against time, with $\psi_0 = e^{-20}, \lambda = 2, \mu = 1$. Red dotted line: expected time of first event for $n = 100$. Blue dotted line: expected time of origin for $n = 100$.

Finally, (2.6.9) implies that to simulate ψ_1 given ψ_0, one can draw $Y \sim \text{Exp}(n)$ and set

$$\psi_1 = 1 - \frac{1-\psi_0}{1+\frac{\lambda-\mu}{\lambda}(e^Y-1)}.$$

In summary, if the k-th event of the RRP $X_{\psi_0}^{\gamma}$ occurs at time t_k, the waiting time to the next event will have the same distribution as that of $X_{\widetilde{\psi}(t_k)}^{\gamma}$ started with $n-k$ lineages, with $\widetilde{\psi}(t_k)$ given by (2.6.13).

2.6.3 Population size for the coalescent

A similar property holds for the coalescent with exponential growth: given an initial population size N_0, after the first coalescence the process can be restarted with $n-1$ lineages and a new initial population size N_1 (Ohtsuki and Innan, 2017). The distribution of N_1 can be found by considering

$$N_1 = N_0 e^{-(\lambda-\mu)T_1},$$

where T_1 has the Gompertz distribution given by (2.6.4). Then

$$T_1 = -\frac{1}{\lambda-\mu}\log\left(\frac{N_1}{N_0}\right) = \frac{1}{\lambda-\mu}\log\left(1+\left(\frac{N_0}{N_1}-1\right)\right).$$

By inverting the Gompertz cdf, it can be deduced that

$$\frac{N_0}{N_1} - 1 \sim \mathrm{Exp}\left(\frac{n(n-1)\lambda}{(\lambda-\mu)N_0}\right). \tag{2.6.14}$$

Thus, N_1 can be simulated by drawing $Y \sim \mathrm{Exp}\left(\frac{n(n-1)\lambda}{(\lambda-\mu)N_0}\right)$ and setting $N_1 = N_0/(1+Y)$.

2.6.4 Time of k-th event

As described above, in the limit $\psi \to 0$, the distributions of the time to first event for the two models can be equated by setting the initial population size N_0 for the coalescent with exponential growth to be equal to $(n-1)/\psi_0$, where ψ_0 is the sampling probability for the RRP. However, the distributions of subsequent event times can only be the same if the sequence of population sizes for the coalescent $\mathcal{N} = \{N_1, N_2, \ldots\}$ and the sequence $\mathcal{P} = \{(n-2)/\psi_1, (n-3)/\psi_2, \ldots\}$ for the RRP are equal in distribution. In other words, the two models will diverge if the entries of $\mathcal{P}$ and $\mathcal{N}$ have different distributions. Moreover, the quality of the approximation in (2.6.6) will degrade when the effective sampling probability becomes relatively large.

Figure 2.6 shows the percentage difference in median event times between the RRP and the coalescent with exponential growth (setting the initial size to $N_0 = (n-1)/\psi_0$), for three values of ψ_0. When ψ_0 is small, the event times agree very closely, up to the last few events. The left panels of Figure 2.7 demonstrates that this is because the population size sequences $\mathcal{P}$ and $\mathcal{N}$ agree closely for small ψ_0. For the last few events, the effective sampling

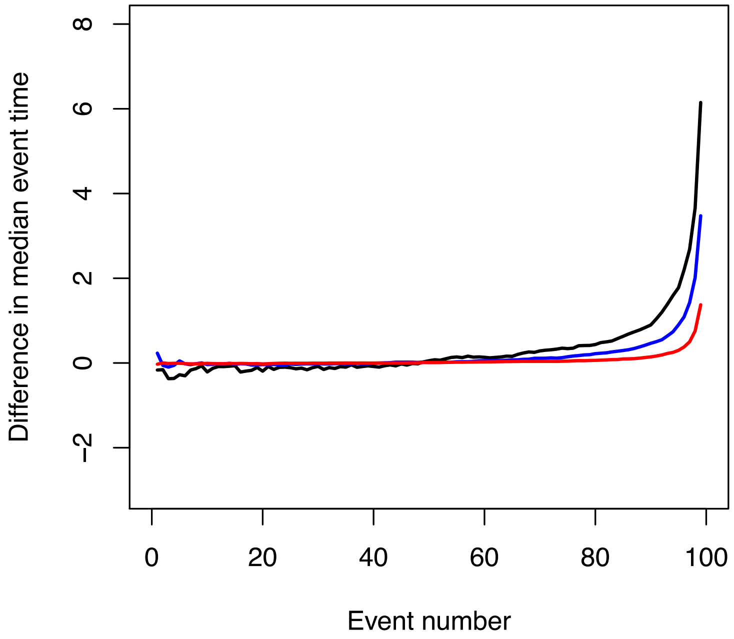

Figure 2.6: y-axis shows percentage difference in median event times for the coalescent with exponential growth and the RRP, against event number (x-axis). Parameters: $n = 100, \lambda = 2, \mu = 1$, based on 100 000 simulations for each event number. Black line: $\psi = 0.1$, blue line: $\psi = 0.001$, red line: $\psi = 10^{-10}$.

probability becomes large, and the remaining number of lineages small, causing the quality of the Gompertz approximation in (2.6.6) to deteriorate.

This investigation demonstrates that after matching up the time scaling parameters at time 0, the coalescent with exponential growth and the RRP are very similar, when the sample size is moderate to large. This suggests that the stark differences between the two models identified by Stadler et al. (2015) were likely to be due to restricting the study to $n = 2$, and not appropriately matching up the initial population sizes.

2.6.5 Site frequency spectrum

In this section, the expected site frequency spectrum (SFS) for the RRP is derived in the $\psi \to 0$ limit; this is a commonly used summary statistic capturing the extent to which mutations are shared by individuals in the sample. Suppose that mutations occur on the branches on the genealogy with fixed rate θ, under the infinite sites assumption. Let $S_n(j) =: \xi_j$ be the number of mutations shared by j individuals in a sample of size n (assuming that the ancestral type is known), and let the weight of an edge l at stage k, denoted

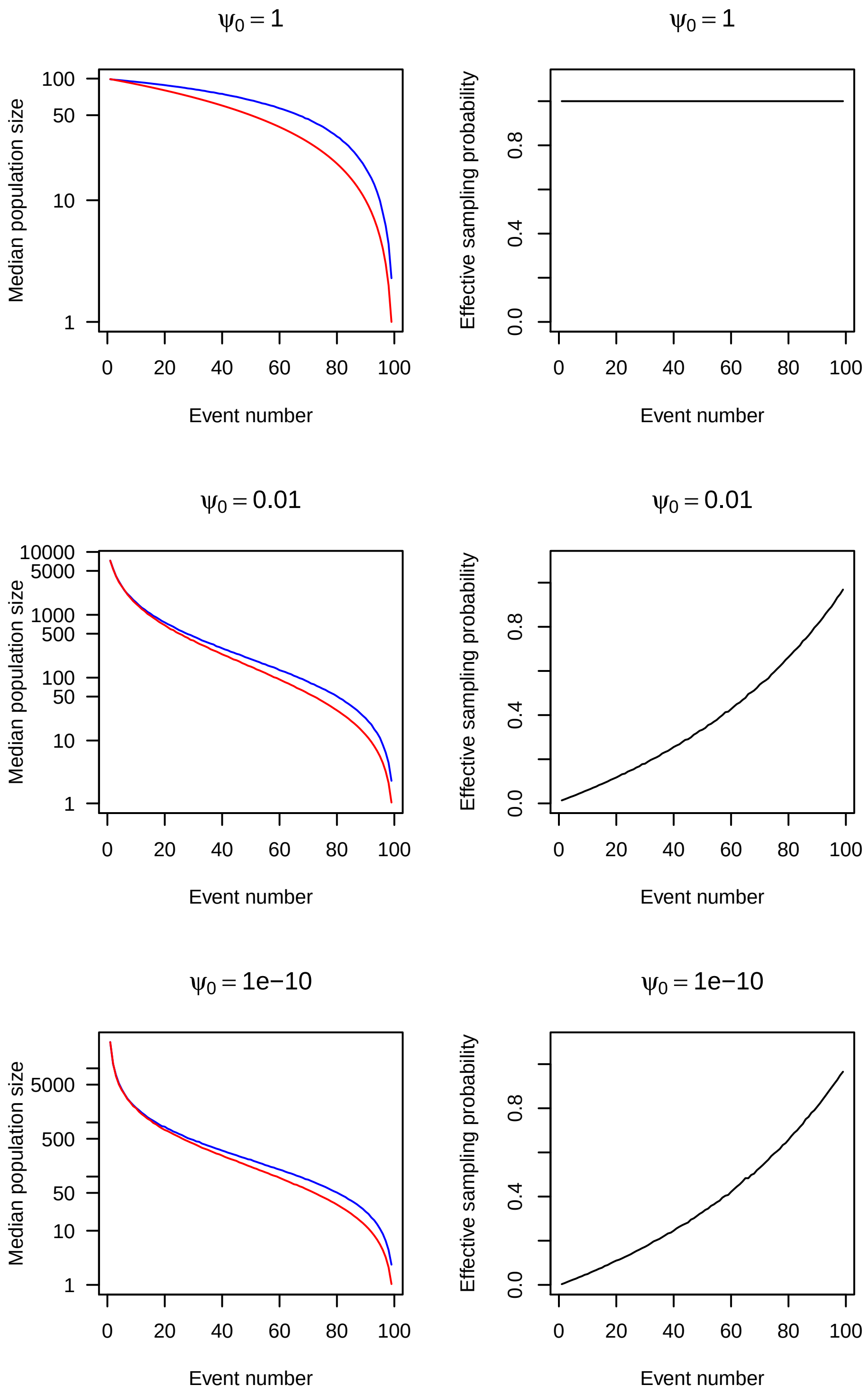

Figure 2.7: Left panels: median population size trajectory $\mathcal{N}$ for the coalescent with exponential growth (blue lines) and median $\mathcal{P}$ for the RRP (red lines), based on 1 000 simulations for each event number. Note log scale on the y-axis. Right panels: effective sampling probability for the RRP, versus event number. Parameters: $n = 100, \lambda = 2, \mu = 1$.

ω_{kl}, be the number of leaves it subtends, as illustrated in Figure 2.8.

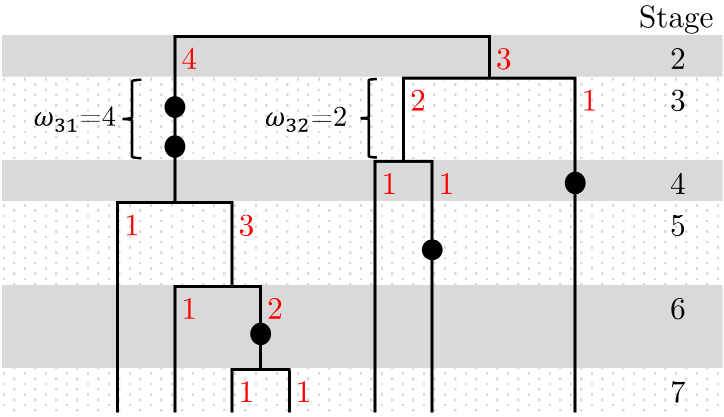

Figure 2.8: Shaded regions delineate the stages. Red labels beside edges show their weight. Black dots show mutations. Here $S_n = (\xi_1, \xi_2, \ldots, \xi_6) = (2, 1, 0, 2, 0, 0)$

Recall that in the limit $\psi \to 0$, the time to first event grows to infinity, hence the number of mutations with multiplicity one in the sample grows to infinity. Again fixing $\lambda - \mu = 1$ for simplicity, the following holds for mutations shared by between 2 and $n-1$ individuals in the sample:

Proposition 2.6.1. *The expectation of ξ_j for $j \in \{2, \ldots, n-1\}$ in the limit $\psi \to 0$ is given by*

$$\mathbb{E}(\xi_j) = \frac{n\theta}{j(j-1)}. \tag{2.6.15}$$

Proof. The proof follows the approach of Fu (1995), by considering the number of lineages in the tree when the mutations occur (see Hudson (2015) for a somewhat simpler argument). Call the time at which there are k lineages 'stage k'. At each stage, assign to each lineage l a weight ω_{kl}, being the number of individuals in the sample that are its descendants; note that $\sum_{l=1}^{k} \omega_{kl} = n$. Then $(\omega_{k1}, \ldots, \omega_{kk})$ is uniformly distributed over all vectors of length k with positive integer entries summing to n, and there are $\binom{n-1}{k-1}$ such vectors (Fu, 1995). This gives

$$\begin{aligned}
\mathbb{P}(\omega_{kl} = j) =: p(k,j) &= \binom{n-j-1}{k-2} \Big/ \binom{n-1}{k-1} \\
&= \frac{j!}{j!} \, \frac{(n-j-1)!}{(k-2)!(n-k-j+1)!} \, \frac{(k-1)!(n-k!)}{(n-1)!} \\
&= \frac{(n-k)}{j(j-1)} \, \frac{(k-1)(n-k-1)!}{((n-k-1)-(j-2))!(j-2)!} \, \frac{j!(n-j-1)!}{(n-1)!} \\
&= \frac{(n-k)}{j(j-1)}(k-1)\binom{n-k-1}{j-2} \Big/ \binom{n-1}{j}. \qquad (2.6.16)
\end{aligned}$$

Let L_m be the sum of the lengths of all branches with weight m. The mean duration of stage k (while there are k lineages) is $n/(k(n-k))$ by Proposition 2.5.3, so for $j \geq 2$

$$\mathbb{E}[\xi_j] = \theta \, \mathbb{E}[L_j]$$

$$\begin{aligned}
&= \theta \sum_{k=2}^{n-1} k \, \frac{n}{k(n-k)} \, p(k,j) \\
&= \frac{n\theta}{j(j-1)} \left[\sum_{k=2}^{n-1} (k-1) \binom{n-k-1}{j-2} \right] \Big/ \binom{n-1}{j}. \qquad (2.6.17)
\end{aligned}$$

The sum can be computed by using a version of the Chu–Vandermonde identity:

$$\sum_{a=0}^{b} \binom{a}{c} \binom{b-a}{d-c} = \binom{b+1}{d+1}.$$

Setting $c = 1,\ a = k-1,\ b = n-2,\ d = j-1$, and noting that the summand with $k = (n-1)$ is zero,

$$\sum_{k=2}^{n-1} (k-1) \binom{n-k-1}{j-2} = \binom{n-1}{j},$$

which completes the proof. □

The expected number of polymorphic sites with non-singleton mutations is

$$\sum_{j=2}^{n-1} \frac{n\theta}{j(j-1)} = \frac{n(n-2)\theta}{n-1},$$

this is approximately $n\theta$ for large n.

Proposition 2.6.2. *The expectation of ξ_1 is given by*

$$\mathbb{E}(\xi_1) = n\theta\left(\log\left(\frac{1}{\psi\lambda'}\right) - 1\right) + \mathcal{O}(\psi). \tag{2.6.18}$$

Proof. The expectation of the time to first event is

$$\begin{aligned}\mathbb{E}(W_0) &= \mathbb{E}(T_n) - \sum_{k=1}^{n-1}\mathbb{E}(W_k)\\ &= \log\left(\frac{1}{\psi\lambda'}\right) + \sum_{j=1}^{n-1}\frac{1}{j} - \sum_{k=1}^{n-1}\frac{n}{k(n-k)} + \mathcal{O}(\psi)\\ &= \log\left(\frac{1}{\psi\lambda'}\right) - \sum_{j=1}^{n-1}\frac{1}{j} + \mathcal{O}(\psi). \end{aligned} \tag{2.6.19}$$

Thus,

$$\begin{aligned}\mathbb{E}(L_1) &= \sum_{k=2}^{n-1} k\frac{n}{k(n-k)}p(k,1) + n\log\left(\frac{1}{\psi\lambda'}\right) - n\sum_{j=1}^{n-1}\frac{1}{j} + \mathcal{O}(\psi)\\ &= \sum_{k=2}^{n-1}\frac{n(k-1)}{(n-k)(n-1)} + n\log\left(\frac{1}{\psi\lambda'}\right) - n\sum_{j=1}^{n-1}\frac{1}{j} + \mathcal{O}(\psi)\\ &= \frac{n}{n-1}\sum_{k=2}^{n-1}\frac{k-1}{n-k} + n\log\left(\frac{1}{\psi\lambda'}\right) - n\sum_{j=1}^{n-1}\frac{1}{j} + \mathcal{O}(\psi)\\ &= n\sum_{j=2}^{n-1}\frac{1}{j} + n\log\left(\frac{1}{\psi\lambda'}\right) - n\sum_{j=1}^{n-1}\frac{1}{j} + \mathcal{O}(\psi)\\ &= n\log\left(\frac{1}{\psi\lambda'}\right) - n + \mathcal{O}(\psi). \end{aligned} \tag{2.6.20}$$

Multiplying by θ gives the result. □

Durrett (2013, Theorem 2) derives the expected SFS for the Moran model with exponential growth; the resulting formulas are the same as (2.6.15) and (2.6.18) asymptotically as $N \to \infty$:

$$\mathbb{E}(\xi_i)\begin{cases} \to \frac{n\theta}{bi(i-1)} & \text{for } i \geq 2,\\ \sim \frac{n\theta}{b}\log(Nb) & \text{for } i = 1, \end{cases} \tag{2.6.21}$$

where b is the population growth rate (which here was fixed to be 1 for simplicity), N the current population size, and $a_N \sim b_N$ means $a_N/b_N \to 1$ as $N \to \infty$. This implies that, similarly to the event times, in the large population limit the expected SFS under the RRP converges to that of the coalescent with exponential growth.

Lambert (2008) derives an explicit expression for the expected SFS for general coalescent point processes, and Dinh et al. (2020) integrates this formula for a birth-death process with Bernoulli sampling. Taking the limit $1 - \frac{\psi\lambda}{\lambda-\mu} \to 1$ in their equation (8) gives

$$\mathbb{E}[\xi_j] = \theta \frac{n+j-1}{j(j-1)}.$$

The difference with (2.6.15) is due to the treatment of time of origin. In Lambert (2008), when a CPP is constructed, one of the n individuals in the sample is conditioned to survive forever (in effect, conditioning on the time of origin going to ∞). This differs from the model considered here, as the time of origin is random, being the maximum of the n i.i.d. draws of individual lifetimes.

2.7 Discussion

Previous work on the applications of birth-death processes to modelling genealogies has largely concentrated on studying the evolution of species, or the transmission of pathogens. Intra-host viral infection is another setting where birth-death models may be very appropriate, but evolution happens on a different scale, both in terms of time (infections are relatively short-term) and population size, which can grow very large. For instance, the viral load of SARS-CoV-2 ranges between around $10^4 - 10^7$ copies per ml in throat swabs of infected patients (Pan et al., 2020). This motivates the development of birth-death models that consider sample genealogies in the large-population limit.

In this chapter, I have demonstrated that viewing the RRP as an inhomogeneous pure-death process allows for relatively simple and intuitive derivations of its properties. The time rescaling approach allows for results derived for completely sampled RRPs to be transformed to those for incomplete sampling, using a simple change of variables, with no restrictions on the parameter values. Moreover, the time rescaling between the time-reversed Yule rate 1 process and the RRP can be used to simulate the RRP in a straightforward way, by simulating each event time sequentially.

In the limit $\psi \to 0$, this rescaling can be decomposed into two timescales. The RRP tree becomes star-shaped, with terminal branch lengths tending to infinity, but inter-event times at the top of the tree are approximately exponential with a rate depending on n and the event number. This has interesting implications for data analysis, as it suggests that the number of singleton mutations in a small sample from a very large population tends to infinity, but the number of shared mutations does not. Indeed, Dinh et al. (2020) consider the expected frequency spectrum of mutations using a birth-death model with the infinite sites assumption. Although this is not explicitly discussed, the results of their simulations show that for small values of ψ, the expected number of singletons is orders of magnitude larger than that of mutations shared by multiple individuals. In applying their method to cancer data, Dinh et al. (2020) consider small values of ψ with the population size being very large compared to the sample size—the results presented in Section 2.5 provide an insight into the properties of the genealogy in this case.

As can be seen from the results presented in this chapter and related work, properties of the genealogy of a sample obtained from a population following a birth-death process are notably different from those arising under the coalescent, particularly when ψ is large and the sample size is close to being of the same order as the population size. The coalescent is widely used in statistical inference for intra-host viral populations (e.g. Dialdestoro et al., 2016). However, the choice of model should be appropriate to the relative scale of the biological application, and the individual-level population dynamics are arguably likely to be better modelled by a birth-death process. When considering the scenario of a small sample obtained from a very large population, the differences between the coalescent with exponential growth and the small-ψ limit of the birth-death model appear to fade.

An important question is thus whether, for samples of viral genetic sequencing data, birth-death models can provide better inference on the evolutionary dynamics of such populations. Answering this would require development of new methods for statistical inference that condition on the data and incorporate the natural processes governing such populations, such as high rates of mutation, recombination, and rapid demographic changes. This also presents interesting challenges in making full use of the increasingly rich and plentiful sequencing data available for viral organisms.

Chapter 3

Reconstruction of parsimonious ARGs with KwARG

3.1 Introduction

In this chapter, I introduce KwARG ("quick ARG"), a software tool (written in C) which implements a greedy heuristic-based parsimony algorithm for reconstructing histories that are minimal or near-minimal in the number of posited recombination and mutation events. The algorithm starts with the input dataset and generates plausible histories backwards in time, adding coalescence, mutation, recombination, and recurrent mutation events to reduce the dataset until the common ancestor is reached. By tuning a set of cost parameters for each event type, KwARG can find solutions consisting only of recombinations (giving an upper bound on R_{min}), only of recurrent mutations (giving an upper bound on P_{min}), or a combination of both event types. KwARG handles both the 'infinite sites' and 'maximum parsimony' scenarios, as well as interpolating between these two cases by allowing recombinations as well as recurrent mutations and sequencing errors, which is not offered by existing methods. Recalling Figure 1.4, KwARG finds all three types of solution for the given dataset.

KwARG shows excellent performance when benchmarked against exact methods on small datasets, and outperforms existing parsimony-based heuristic methods on large, more complex datasets while maintaining computational efficiency; KwARG also achieves very good accuracy in reconstructing local tree topologies. The source code and executables are available on GitHub at `https://github.com/a-ignatieva/kwarg`, along with documentation and usage examples.

Details of the algorithm underlying KwARG are given in Section 3.2, with an explanation of the required inputs and expected outputs. In Section 3.3, the performance of KwARG on simulated data is benchmarked against exact methods and existing programs. An application of KwARG to a widely studied *Drosophila melanogaster* dataset (Kreitman, 1983) is described in Section 3.4. Discussion follows in Section 3.5.

3.2 Technical details

Consider a sample of genetic data, where the allele at each site can be denoted 0 or 1. The infinite sites assumption is not required, so that each site can undergo multiple mutation events. However, it is assumed that mutations correspond to transitions between exactly two possible states, excluding, for instance, triallelic sites.

3.2.1 Input

KwARG accepts data in the form of a binary matrix, or a multiple alignment in nucleotide or amino acid format. The sequence and site labels can be provided if desired. It is possible to specify a root sequence, or leave this to be determined. The presence of missing data is permitted; regardless of the type of input, the data is converted to a binary matrix $\mathcal{D}$, with entries '$\star$' denoting missing entries or material that is not ancestral to the sample.

3.2.2 Methods

KwARG reconstructs the history of a sample backwards in time, by starting with the data matrix $\mathcal{D}$ and performing row and column operations corresponding to coalescence, mutation, and recombination events, until only one ancestral sequence remains. By reversing the order of the steps, a forward-in-time history is obtained, showing how the population evolved from the ancestor to the present sample. When a choice can be made between multiple possible events, a neighbourhood of candidate ancestral states is constructed, using the same general method as that employed in the program Beagle (Lyngsø et al., 2005). A backwards-in-time approach has also been implemented in the programs SHRUB (Song et al., 2005), Margarita (Minichiello and Durbin, 2006) and GAMARG (Thao and Vinh, 2019), all of which adopt the infinite sites assumption but use different criteria for choosing amongst possible recombination events.

3.2.2.1 Construction of a history

For convenience, assume that the all-zero sequence is specified as the root, and 0 (resp. 1) entries of $\mathcal{D}$ correspond to ancestral (resp. mutated) sites. Suppose $\mathcal{D}_t$ is the data matrix obtained after $t-1$ iterations of the algorithm.

Say that two rows (columns) *agree* if they are equal at all positions where both rows (columns) contain ancestral material, and the sites (sequences) carrying ancestral material in one are a subset of the sites (sequences) carrying ancestral material in the other. At the beginning of the t-th step, KwARG first reduces $\mathcal{D}_t$, by repeatedly applying the 'Clean' algorithm (Song and Hein, 2003) through:

- deleting uninformative columns (consisting of all 0's);
- deleting columns containing only one 1 (corresponding to "undoing" a mutation present in only one sequence);
- deleting a row if it agrees with another row (corresponding to a coalescence event);
- deleting a column if it agrees with an adjacent column.

A run of the 'Clean' algorithm repeatedly applies these steps to $\mathcal{D}_t$, terminating when no further reduction is possible. Suppose the resulting data matrix is $\overline{\mathcal{D}}_t$. KwARG then constructs a neighbourhood $\mathcal{N}_t$ of candidate next states, each one obtained through one of the following operations:

- Pick a row and split it into two at a possible recombination point (as illustrated in Figure 3.1). Only a subset of possible recombining sequences and breakpoints needs to be considered; see Lyngsø et al. (2005, Section 3.3) for a detailed explanation.
- Remove a recurrent mutation, by selecting a column and changing a 0 entry to 1, or a 1 entry to 0. This is the event type that is disallowed by algorithms applying the infinite sites assumption.

Suppose a neighbourhood $\mathcal{N}_t = \{\mathcal{N}_t^1, \dots, \mathcal{N}_t^N\}$ is formed, consisting of all possible states that can be reached from $\overline{\mathcal{D}}_t$ through applying one of these operations. Then the reduced neighbourhood $\overline{\mathcal{N}}_t = \{\overline{\mathcal{N}}_t^1, \dots, \overline{\mathcal{N}}_t^N\}$ is formed by applying 'Clean' to each state in turn. Each state $\overline{\mathcal{N}}_t^i$ is then assigned a score $S(\overline{\mathcal{N}}_t^i, \mathcal{N}_t^i, \overline{\mathcal{D}}_t)$, combining (i) the cost $C\left(\mathcal{N}_t^i, \overline{\mathcal{D}}_t\right)$, defined below, of reaching the configuration $\mathcal{N}_t^i$ from $\overline{\mathcal{D}}_t$, (ii) a measure AM $\left(\overline{\mathcal{N}}_t^i\right)$ of the complexity of the resulting data matrix $\overline{\mathcal{N}}_t^i$, and (iii) a lower bound $L(\overline{\mathcal{N}}_t^i)$ on the remaining number of recombination and recurrent mutation events still required to reach the ancestral sequence from $\overline{\mathcal{N}}_t^i$. Finally, a state is selected, say $\overline{\mathcal{N}}_t^j$, based on its score, setting $\mathcal{D}_{t+1} = \overline{\mathcal{N}}_t^j$. The process of reducing the dataset followed by constructing a neighbourhood and choosing the best move is repeated, until all incompatibilities are resolved and the root sequence is reached. Pseudocode for the 'Clean' algorithm and KwARG is given in Section 3.2.4.

The construction of a history for the dataset given in Figure 1.4 is illustrated in Figure 3.1. The first step corresponds to the construction of a neighbourhood, two of the states $\mathcal{N}_1^1, \mathcal{N}_1^2 \in \mathcal{N}_1$ are pictured. Then, the 'Clean' algorithm is applied to each state in the neighbourhood (illustrated as a series of steps following blue arrows). From the resulting reduced neighbourhood $\{\overline{\mathcal{N}}_1^1, \overline{\mathcal{N}}_1^2, \dots\}$, the state $\overline{\mathcal{N}}_1^2$ is selected; the other illustrated path is abandoned. This process is repeated until all incompatibilities are resolved and the empty state is reached. Following the path of selected moves in this figure left-to-right corresponds to the events encountered when traversing the leftmost ARG in Figure 1.4 from the bottom up. If instead the state $\overline{\mathcal{N}}_2^1$ were selected at the second step of the algorithm, the resulting path would correspond to the ARG in the centre of Figure 1.4.

3.2.2.2 Score

When considering which next step to take, better choices can be made by considering not just the cost of the step itself, but also the complexity of the configuration it leads to. This is the principle behind the well-known A* algorithm (Hart et al., 1968), where the choice of the next node to explore while traversing a graph is informed by using a heuristic estimate of the remaining distance. KwARG applies the same principle in a greedy fashion, attempting to find a minimal history by following a path of moves selected with probability proportional to a heuristic quality score (as described further on in Section 3.2.2.5).

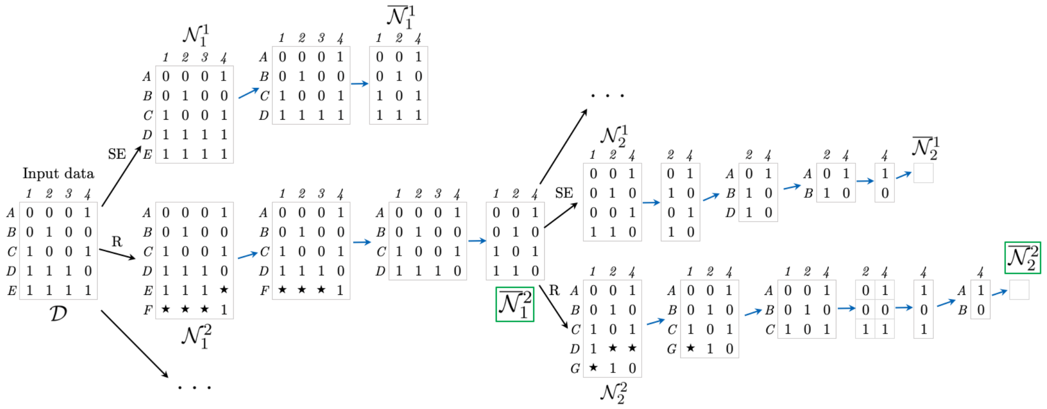
Input data
$\mathcal{D}$
SE
R
$\mathcal{N}_1^1$
$\overline{\mathcal{N}}_1^1$
$\mathcal{N}_1^2$
$\overline{\mathcal{N}}_1^2$
$\mathcal{N}_2^1$
$\overline{\mathcal{N}}_2^1$
$\mathcal{N}_2^2$
$\overline{\mathcal{N}}_2^2$
. . .

Figure 3.1: Example of a reconstructed history for the dataset in Figure 1.4. Stars '⋆' denote non-ancestral material. SE: recurrent mutation occurring on a terminal branch of the ARG. R: recombination event. A sequence of blue arrows corresponds to one application of the 'Clean' algorithm. Green boxes highlight the selected states.

The score implemented in KwARG is

$$S\left(\overline{\mathcal{N}}_t^i, \mathcal{N}_t^i, \overline{\mathcal{D}}_t\right) = \left(C\left(\mathcal{N}_t^i, \overline{\mathcal{D}}_t\right) + L\left(\overline{\mathcal{N}}_t^i\right)\right) \cdot \text{maxAM}\left(\overline{\mathcal{N}}_t\right) + \text{AM}\left(\overline{\mathcal{N}}_t^i\right), \quad (3.2.1)$$

where

$$L(\overline{\mathcal{N}}_t^i) = \begin{cases} R_{min}\left(\overline{\mathcal{N}}_t^i\right) & \text{if maxAM}(\overline{\mathcal{N}}_t) < 75, \\ HB\left(\overline{\mathcal{N}}_t^i\right) & \text{if } 75 \leq \text{maxAM}(\overline{\mathcal{N}}_t) < 200, \\ HK\left(\overline{\mathcal{N}}_t^i\right) & \text{otherwise.} \end{cases}$$

Here, $C\left(\mathcal{N}_t^i, \overline{\mathcal{D}}_t\right)$ denotes the cost of the corresponding event, defined in Section 3.2.2.3; maxAM$(\overline{\mathcal{N}}_t)$ denotes the maximum amount of ancestral material seen in any of the states in $\overline{\mathcal{N}}_t$, and AM$(\overline{\mathcal{N}}_t^i)$ gives the amount of ancestral material in state $\overline{\mathcal{N}}_t^i$. Incorporating a measure of the amount of ancestral material in a state helps to break ties by assigning a smaller score to simpler configurations.

The method of computing the lower bound L depends on the complexity of the dataset, with a trade-off between accuracy and computational cost. For relatively small datasets, it is feasible to compute R_{min} exactly using Beagle. HB refers to the haplotype bound, employing the improvements afforded by first calculating local bounds for incompatible intervals, and applying a composition method to obtain a global bound. HK refers to the Hudson-Kaplan bound; this is fast but less accurate, so is reserved for larger, more complex configurations. Note that these bounds are computed under the infinite sites assumption.

The particular form and components of the score were chosen through simulation testing; it was found that the given formula provides a good level of informativeness regarding the quality of a possible state.

3.2.2.3 Event cost

Each type of event is assigned a cost, which gives a relative measure of preference for each event type in the reconstructed history:

- C_R: the cost of a single recombination event, defaults to 1.
- C_{RR}: the cost of performing two successive recombinations, defaults to 2. It is sufficient to consider at most two consecutive recombination events before a coalescence (Lyngsø et al., 2005); this type of event also captures the effects of gene conversion.
- C_{RM}: the cost of a recurrent mutation. If $\mathcal{N}_t^i$ is formed from $\overline{\mathcal{D}}_t$ by a recurrent mutation in a column representing k agreeing sites, this corresponds to proposing k recurrent mutation events, so the cost is $C(\mathcal{N}_t^i, \overline{\mathcal{D}}_t) = k \cdot C_{RM}$.
- C_{SE}: this event is a recurrent mutation which affects only one sequence in the original dataset, i.e. it occurs on the terminal branches of the ARG. Thus, the event can be either a regular recurrent mutation, or an artefact due to sequencing errors. The cost can be set to equal C_{RM}, or lower if the presence of sequencing errors is considered likely.

KwARG allows the specification of a range of event costs as tuning parameters, as well as the number Q of independent runs of the algorithm to perform for each cost configuration. The proportions of recombinations to recurrent mutations in the solutions produced by KwARG can be controlled by varying the ratio of costs for the corresponding event types.

3.2.2.4 Default cost configuration

If the number of iterations $Q > 1$ is specified but no costs are input, KwARG runs each of the following 13 cost configurations Q times:

$$\begin{aligned}(C_{SE}, C_{RM}, C_R, C_{RR}) \in \{&(\infty, \infty, 1.0, 2.0), (1.0, 1.01, 1.0, 2.0), (0.9, 0.91, 1.0, 2.0),\\ &(0.8, 0.81, 1.0, 2.0), (0.7, 0.71, 1.0, 2.0), (0.6, 0.61, 1.0, 2.0),\\ &(0.5, 0.51, 1.0, 2.0), (0.4, 0.41, 1.0, 2.0), (0.3, 0.31, 1.0, 2.0),\\ &(0.2, 0.21, 1.0, 2.0), (0.1, 0.11, 1.0, 2.0), (0.01, 0.02, 1.0, 2.0),\\ &(1.0, 1.1, \infty, \infty)\}. \qquad (3.2.2)\end{aligned}$$

The effectiveness of this range of cost configurations will be illustrated on the Kreitman dataset in Section 3.4.1.

3.2.2.5 Selection probability

The method of selecting the next state from a neighbourhood of candidates will impact on the efficiency and performance of the algorithm. At one extreme, selecting at random amongst the states will mean that the solution space is explored more fully, but will be prohibitively inefficient in terms of the number of runs needed to find a near-optimal solution. On the other hand, always greedily selecting the move with the minimal score will quickly identify a small set of solutions for each cost configuration, at the expense of placing our faith in the ability of the score to assess the quality of the candidate states accurately.

Thus, a selection method is proposed that is intermediate between these two extremes, randomising the selection but focussing on moves with near-minimal scores. A pseudo-score for state $\overline{\mathcal{N}}_t^i$ is calculated:

$$\exp\Big(T \cdot \Big(1 - \widetilde{S}\left(\overline{\mathcal{N}}_t^i, \mathcal{N}_t^i, \overline{\mathcal{D}}_t\right)\Big)\Big), \qquad (3.2.3)$$

where

$$\widetilde{S}\left(\overline{\mathcal{N}}_t^i, \mathcal{N}_t^i, \overline{\mathcal{D}}_t\right) = \frac{S\left(\overline{\mathcal{N}}_t^i, \mathcal{N}_t^i, \overline{\mathcal{D}}_t\right) - \min_j S\left(\overline{\mathcal{N}}_t^j, \mathcal{N}_t^j, \overline{\mathcal{D}}_t\right)}{\max_j S\left(\overline{\mathcal{N}}_t^j, \mathcal{N}_t^j, \overline{\mathcal{D}}_t\right) - \min_j S\left(\overline{\mathcal{N}}_t^j, \mathcal{N}_t^j, \overline{\mathcal{D}}_t\right)},$$

and states in $\overline{\mathcal{N}}_t$ are selected with probability proportional to their pseudo-score. The annealing parameter T controls the extent of random exploration; $T = 0$ corresponds to choosing uniformly at random from the neighbourhood of candidates, and $T = \infty$ to always choosing a state with the minimal score.

The default value of $T = 30$ was chosen following simulation testing, which showed that this provides a good balance between efficiency and thorough exploration of the neighbourhood.

3.2.3 Output

The default output consists of the number of recombinations and recurrent mutations in each identified solution; an example for the Kreitman dataset is given in Table 3.1. Each iteration is assigned a unique random seed, which can be used to reconstruct each particular solution and produce more detailed outputs, such as a detailed list of events in the history, the ARG in several graph formats, or the corresponding sequence of marginal trees.

3.2.4 Pseudocode

Let $\mathcal{D}$ be an input data matrix with entries 0, 1 or $\star$. Denote by $\mathcal{D}_{i,j}$ the entry of $\mathcal{D}$ at position (i, j). Let $R_r(\mathcal{D}, i)$ and $R_c(\mathcal{D}, j)$ denote the resulting matrix when the i-th row or the j-th column of $\mathcal{D}$ is deleted, respectively. Let the history $\mathcal{H}$ be a set storing all of the intermediate states visited on the path from $\mathcal{D}$ to the root of the ARG. Algorithm 2 shows pseudocode for the 'Clean' algorithm with this notation.

Algorithm 2: Clean (adapted from Song and Hein, 2003)

```
Input: Dataset $\mathcal{D}$, history $\mathcal{H}$
Output: Reduced dataset $\overline{\mathcal{D}}$, updated history $\mathcal{H}'$
Initialise $C \leftarrow$ true, $\overline{\mathcal{D}} \leftarrow \mathcal{D}$, $\mathcal{H}' \leftarrow \mathcal{H}$;
while $C$ do
    if two distinct rows $i, j$ agree: $\overline{\mathcal{D}}_{i,k} \in \{\overline{\mathcal{D}}_{j,k}, \star\}\ \forall k$ then
        $\overline{\mathcal{D}} \leftarrow R_r(\overline{\mathcal{D}}, i)$, $\mathcal{H}' \leftarrow \mathcal{H}' \cup \overline{\mathcal{D}}$ ;
    else if there is a column $i$ such that $\overline{\mathcal{D}}_{k,i} = 1$ for exactly one $k$ then
        $\overline{\mathcal{D}} \leftarrow R_c(\overline{\mathcal{D}}, i)$, $\mathcal{H}' \leftarrow \mathcal{H}' \cup \overline{\mathcal{D}}$ ;
    else if two distinct neighbouring columns $i, j$ agree:
    $\overline{\mathcal{D}}_{k,i} \in \{\overline{\mathcal{D}}_{k,j}, \star\}\ \forall k$ then
        $\overline{\mathcal{D}} \leftarrow R_c(\overline{\mathcal{D}}, i)$, $\mathcal{H}' \leftarrow \mathcal{H}' \cup \overline{\mathcal{D}}$ ;
    else
        $C \leftarrow$ false;
end
return $(\overline{\mathcal{D}}, \mathcal{H}')$;
```

Define the following operations:

1. Recurrent mutation: $\widetilde{\mathcal{D}} = \mathrm{RM}(\mathcal{D}, i, j)$ is the result of a recurrent mutation in row i at column j; $\widetilde{\mathcal{D}}$ is obtained from $\mathcal{D}$ by changing the (i, j)-th entry from 0 to 1 or from 1 to 0.

2. Recombination: $\widetilde{\mathcal{D}} = \mathrm{Rec}(\mathcal{D}, i, j)$ is the result of a recombination in row i with breakpoint just after column j. Namely, $\widetilde{\mathcal{D}}$ is obtained from $\mathcal{D}$ by inserting a copy of the i-th row just below itself, and setting $\widetilde{\mathcal{D}}_{i,k} = \star\ \forall k \leq j$ and $\widetilde{\mathcal{D}}_{i+1,k} = \star\ \forall k > j$.

3. Two consecutive recombinations: $\widetilde{\mathcal{D}} = \mathrm{RRec}(\mathcal{D}, i, j, k, l)$ is the result of performing two recombinations, in rows i and k with breakpoints at j and l, respectively.

Note that for recombination events, not all row and column positions should to be considered, as some moves are guaranteed not to resolve any incompatibilities in the dataset. The ideas detailed by Lyngsø et al. (2005, Section 3.3) are applied to restrict the rows and breakpoints considered for recombination events. Suppose that as a result, $\mathcal{R}$ is the list of row and column indices (i, j) to consider for recombination events, and $\mathcal{RR}$ is the list of indices (i, j, k, l) to consider for two consecutive recombination events. Algorithm 3 shows pseudocode for constructing the neighbourhood of next candidate states; Algorithm 4 demonstrates the operation of KwARG.

Algorithm 3: Neighbourhood

```
Input: Dataset $\mathcal{D}$
Output: Neighbourhood $\mathcal{N}$
Initialise $\mathcal{N} \leftarrow \{\emptyset\}$;
for $(i, j) \in \mathcal{R}$ do
    $\mathcal{N} \leftarrow \mathcal{N} \cup \mathrm{Rec}(\mathcal{D}, i, j)$;
end
for $(i, j, k, l) \in \mathcal{RR}$ do
    $\mathcal{N} \leftarrow \mathcal{N} \cup \mathrm{RRec}(\mathcal{D}, i, j, k, l)$;
end
for all rows $i$ do
    for all columns $j$ such that $\mathcal{D}_{i,j} \neq \star$ do
        $\mathcal{N} \leftarrow \mathcal{N} \cup \mathrm{RM}(\mathcal{D}, i, j)$;
    end
end
return $\mathcal{N}$;
```

```
Algorithm 4: KwARG
  Input: Dataset $\mathcal{D}$
  Output: History $\mathcal{H}$
  Initialise $i \leftarrow 1,\ \mathcal{H} \leftarrow \{\mathcal{D}\},\ (\overline{\mathcal{D}}_1, \mathcal{H}) \leftarrow Clean(\mathcal{D}, \mathcal{H})$;
  while $\overline{\mathcal{D}}_i \neq \emptyset$ do
    $\overline{\mathcal{N}}_i \leftarrow \{\emptyset\},\ \mathcal{L}_i \leftarrow \{\emptyset\},\ S \leftarrow \{\emptyset\}$;
    $\mathcal{N}_i \leftarrow Neighbourhood(\overline{\mathcal{D}}_i) = \{\mathcal{N}_i^1, \mathcal{N}_i^2, \ldots\}$;
    for $j = 1$ to $|\mathcal{N}_i|$ do
      $(\overline{\mathcal{N}}_i^j, \mathcal{L}_i^j) \leftarrow Clean(\mathcal{N}_i^j, \mathcal{H} \cup \mathcal{N}_i^j)$;
      $\overline{\mathcal{N}}_i \leftarrow \overline{\mathcal{N}}_i \cup \overline{\mathcal{N}}_i^j,\ \mathcal{L}_i \leftarrow \mathcal{L}_i \cup \mathcal{L}_i^j$;
      $S \leftarrow S \cup \widetilde{S}\left(\overline{\mathcal{N}}_i^j, \mathcal{N}_i^j, \overline{\mathcal{D}}_i\right)$, where $\widetilde{S}\left(\overline{\mathcal{N}}_i^j, \mathcal{N}_i^j, \overline{\mathcal{D}}_i\right)$ is computed
      using (3.2.3);
    end
    Randomly draw an index $k$ from $\{1, \ldots, |\overline{\mathcal{N}}_i|\}$ with probabilities
    proportional to entries of $S$;
    Set $\overline{\mathcal{D}}_{i+1} \leftarrow \overline{\mathcal{N}}_i^k,\ \mathcal{H} \leftarrow \mathcal{L}_i^k$;
    $i \leftarrow i + 1$;
  end
  return $\mathcal{H}$;
```

3.3 Performance on simulated data

The performance of KwARG was tested based on two main criteria. Firstly, its performance was compared to that of exact methods, PAUP* and Beagle, to demonstrate that KwARG successfully reconstructs minimal histories in the mutation-only and recombination-only cases, respectively. Secondly, simulation studies were carried out to determine how accurately KwARG reconstructs local trees, compared against three other methods: tsinfer, RENT+, and ARGweaver. Finally, the performance of KwARG was compared to that of the parsimony-based heuristic methods SHRUB and SHRUB-GC. The dependence of the run time of KwARG on the number and length of sequences was also investigated through simulation studies.

3.3.1 Comparison to PAUP*

Disallowing recombination, the quality of computed upper bounds on P_{min} was tested by comparison with PAUP* (Swofford, 2003, version 4.0a168), which was used to compute the exact minimum parsimony score via branch-and-bound on 994 datasets simulated as follows.

Using msprime (Kelleher et al., 2016), 1 100 genealogies were simulated (parameters: 20 sequences, $N_e = 1$). For each tree, Seq-Gen (Rambaut and Grass, 1997) was used to add mutations (parameters: 1 000 sites, mutation rate per generation per site set by the scaling constant $s = 0.01$); only transitions were allowed, to fulfil the requirement that sites mutate between exactly two states. 1 063 datasets exhibited incompatibilities caused by recurrent mutations. KwARG was run for a total of $Q = 600$ iterations per dataset; 150 of these were used to estimate R_{min}, and 450 were run with a range of costs to estimate P_{min}. The runs were terminated after 10 minutes (if 600 iterations had not been completed by then, the results were discarded; this happened in 69 cases); a total of 994 successful runs were performed.

KwARG failed to find P_{min} in 11 (1.1%) cases out of 994. The results are illustrated in the left panel of Figure 3.2. Where KwARG failed to find an optimal solution, in all 11 cases it was off by just one recurrent mutation. Figure 3.2 also demonstrates that a substantial proportion of recurrent mutations do not create incompatibilities in the data, and the number of actual events often far exceeds P_{min}.

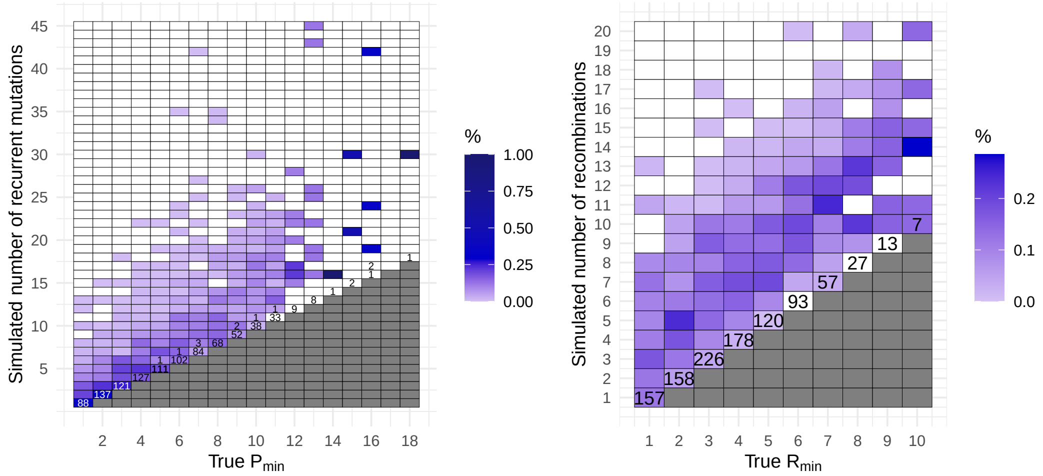

Figure 3.2: Left: number of simulated recurrent mutations against P_{min}. Right: number of simulated recombinations against R_{min}. Cell colouring intensity is proportional to the number of datasets generated for each pair of coordinates. Numbers in each cell correspond to the number of cases where for a dataset with the true minimum number of events given on the x-axis, KwARG inferred the number of events given on the y-axis (unlabelled cells correspond to 0 such cases).

3.3.2 Comparison to Beagle

Under the infinite sites assumption (disallowing recurrent mutation), the accuracy of KwARG's upper bound on R_{min} was tested by comparison with Beagle, on 1 037 datasets simulated as folllows.

Using msprime, 1 100 datasets were simulated under the infinite sites assumption (parameters: $N_e = 1$, mutation rate per generation per site 0.02, recombination rate per site 0.0003, 40 sequences of length 2 000bp). Of the generated datasets, 38 had no incompatible sites, and runs were terminated if Beagle took over 10 minutes to complete (which happened in 25 cases), leaving 1 037 datasets for testing. The parameters were chosen to produce datasets on which Beagle could be run within a reasonable amount of time; the value of R_{min} for the simulated datasets varied between 1 and 10.

Using the default annealing parameter $T = 30$, KwARG found R_{min} in all cases. In 97% of the runs, this took under 5 seconds of CPU time (on a 2.7GHz Intel Core i7 processor); all but one run took less than 40 seconds. In 93% of the runs, 1 iteration was sufficient to find an optimal solution; in 99% of the runs, 5 iterations were sufficient. Beagle found the exact solution in 5 seconds or less in 86% of cases; for datasets with a small R_{min} Beagle runs relatively quickly (median run time for $R_{min} = 5$ was 1 second, compared to KwARG's 0.3 seconds). For more complex datasets, KwARG finds an optimal solution much faster; for $R_{min} = 9$, the median run time of Beagle was 56 seconds, compared to KwARG's 3 seconds.

Setting $T = 10$ and $T = \infty$ resulted in 5 and 22 failures to find an optimal solution, respectively, when KwARG was run for $Q = 1\,000$ iterations per dataset (or terminated after 10 minutes have elapsed), demonstrating that setting the annealing parameters too low or too high results in deterioration of performance.

The right panel of Figure 3.2 illustrates the results, and shows the relationship between the true simulated number of recombinations and R_{min}. This demonstrates that in many cases, substantially more recombinations have occurred than can be confidently detected from the data.

3.3.3 Comparison to tsinfer, RENT+, and ARGweaver

The performance of KwARG in recovering the topology of simulated local trees was tested for a range of recombination and mutation rates (under the infinite sites assumption). For each combination of rates, 100 datasets were simulated, and from the output of each method, the Kendall–Colijn metric (Kendall and Colijn, 2016) was calculated between the inferred and true tree topologies at each variant site position, calculating the mean across all variant sites and averaging over the datasets. Note that ARGs contain more information than local trees, but there is no obvious way of comparing ARG topologies (and tsinfer only infers local trees, rather than full ARGs).

Datasets were simulated using msprime under the infinite sites assumption (parameters: $N_e = 10\,000$, 20 sequences of length 1 000bp), with a range of recombination rates ($\{1 \cdot 10^{-7}, 2 \cdot 10^{-7}, 4 \cdot 10^{-7}, 8 \cdot 10^{-7}, 1.6 \cdot 10^{-6}\}$ per site per generation) and mutation rates ($\{5 \cdot 10^{-8}, 1 \cdot 10^{-7}, 2 \cdot 10^{-7}, 4 \cdot 10^{-7}, 8 \cdot 10^{-7}, 1.6 \cdot 10^{-6}, 3.2 \cdot 10^{-6}, 6.4 \cdot 10^{-6}, 1.28 \cdot 10^{-5}\}$ per site per generation). These parameters were chosen to cover a broad range of the simulated number of recombinations and mutations. 100 datasets were simulated for each combination of rates.

RENT+, tsinfer, ARGweaver, and KwARG were run on each dataset. For tsinfer, the ancestral state must be specified at each variable site, and was set to the simulated truth. ARGweaver requires the specification of mutation and recombination rates; these were set to the simulation parameters used. ARGweaver was run for 1 200 iterations, discarding the first 1 000 as burn-in, and then sampling ARGs with intervals of 20 steps (obtaining 10 in total). KwARG was run for one iteration per dataset, with the parameters $T = 30$, $C_{SE} = C_{RM} = \infty$, and the known ancestral sequence set as the root.

For each dataset, the local trees output by each program were then compared to the simulated true trees, by calculating the Kendall–Colijn metric at each variable site position. As tsinfer can output trees with polytomies, these were resolved randomly before calculating the metric for the sake of fair comparison. The mean was then calculated across sites, and for each combination of recombination and mutation rate the metric was averaged across the datasets. The results are presented in Figure 3.3. A comparison of the run times of the programs used is illustrated in Figure 3.4.

All methods show very comparable performance across the range of considered scenarios, with KwARG slightly outperforming the other methods, based on the chosen metric, when the recombination rate is relatively low and the mutation rate relatively high. The same analysis was performed using the Robinson–Foulds metric (Robinson and Foulds, 1981), and this was found to give very similar results.

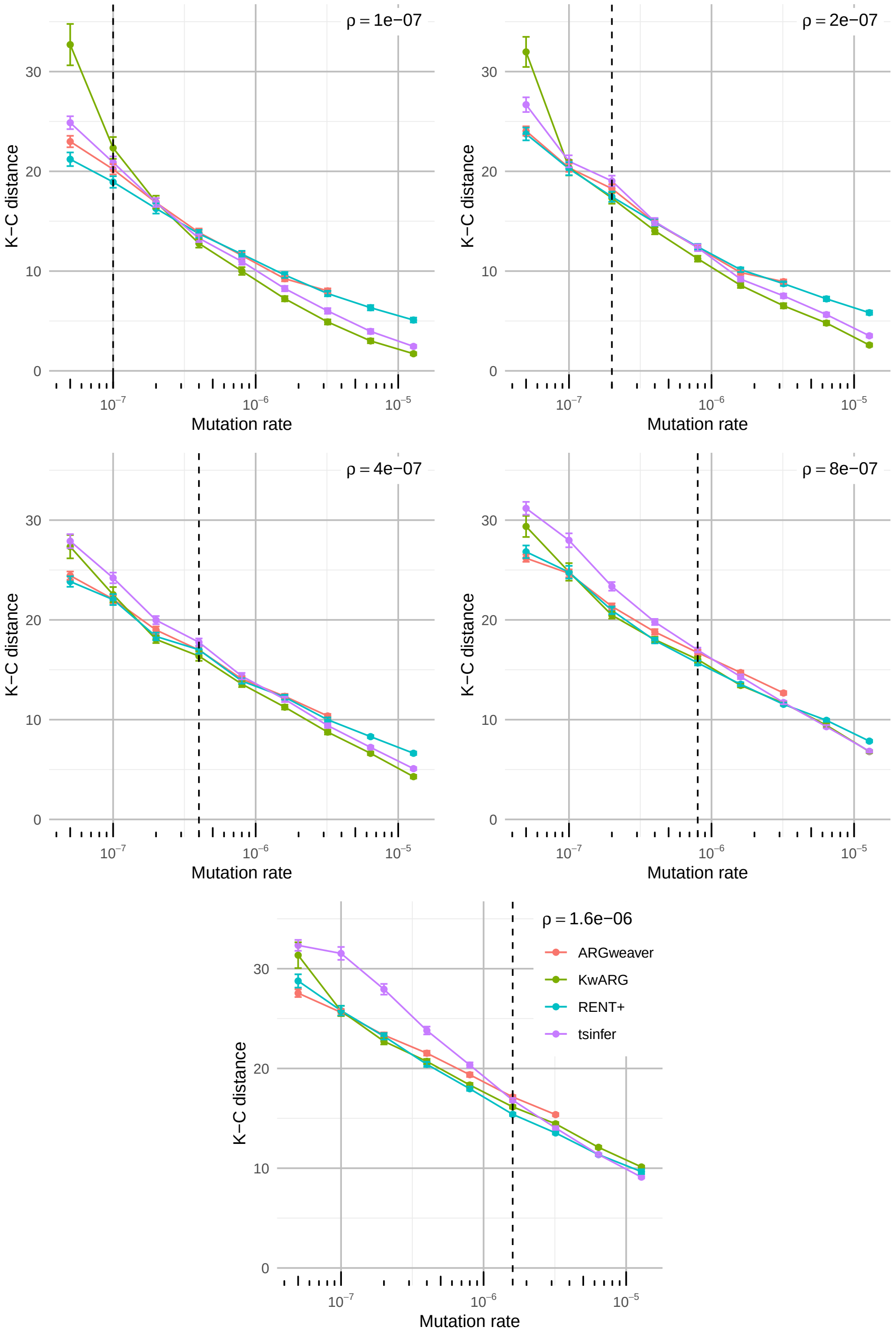

Figure 3.3: Comparison of performance in local tree recovery. Dashed vertical lines show the value of the recombination rate in each panel. Points correspond to mean values; error bars show mean ± standard error. ARGweaver results not shown past $\mu = 3.2 \cdot 10^{-6}$ due to prohibitively long run time. Lower K-C distance indicates better accuracy.

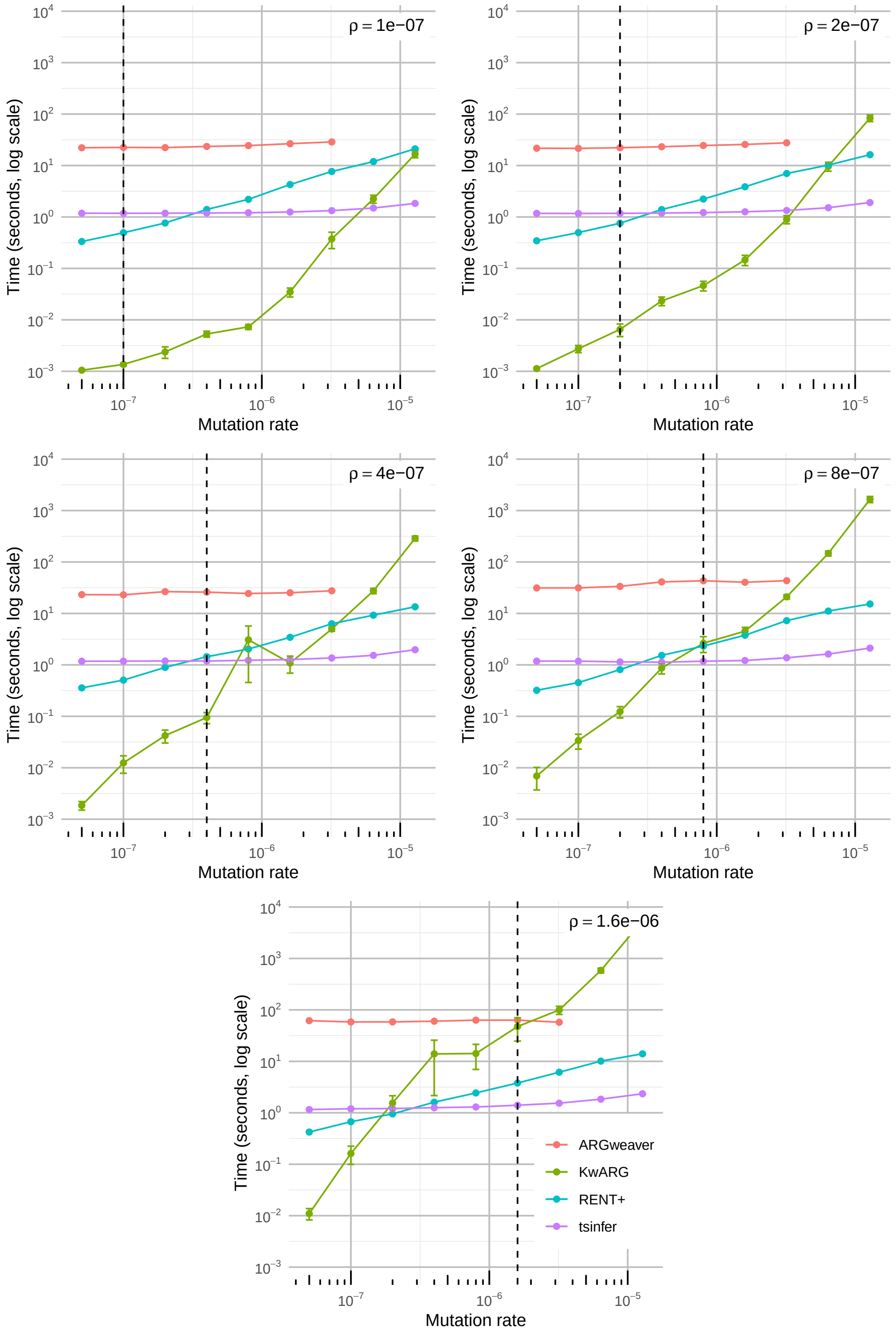

Figure 3.4: Comparison of time taken per dataset. Points show mean run time averaged over 100 datasets for each combination of rate parameters. Error bars show mean ± standard error.

3.3.4 Comparison to SHRUB and SHRUB-GC

The performance of KwARG on larger datasets was tested against the parsimony-based heuristic methods SHRUB and SHRUB-GC. Both methods implement a backwards-in-time construction of ARGs, using a dynamic programming approach to choose among possible recombination events. SHRUB produces an upper bound on R_{min} under the infinite sites assumption. SHRUB-GC also allows gene conversion events; setting the maximum gene conversion tract length to 1 makes this equivalent to recurrent mutation. The algorithm seeks to minimise the total number of events, essentially assigning equal costs to recombination and recurrent mutation. This differs from KwARG in that a single solution is produced for a given dataset, rather than a full range of solutions varying in the number of recombinations and recurrent mutations.

Using msprime and Seq-Gen, 300 datasets of 100 sequences were simulated, with a range of mutation and recombination rates and sequence lengths of 2 000, 5 000, 8 000 and 10 000 bp. For each dataset, KwARG was run for a total of $Q = 260$ iterations, with the default cost configurations and $T = 30$. The resulting upper bound on R_{min} was compared to that produced by SHRUB, and the minimum number of events over all identified solutions was compared to the solution produced by SHRUB-GC (configured to allow length-1 gene conversions).

KwARG obtained solutions at least as good as SHRUB's in 292 (97.3%) of 300 cases, outperforming it in 35 (11.7%) instances. KwARG obtained solutions at least as good as SHRUB-GC in 296 (98.7%) cases, outperforming it in 2 instances. The results and the run times are illustrated in Figures 3.5 and 3.6. On average, for relatively small and simple datasets, KwARG takes approximately the same time per one iteration as a run of SHRUB or SHRUB-GC, and outperforms both programs on more complex datasets.

3.3.5 Run time analysis

A comparison of the run times of KwARG against tsinfer, RENT+, and ARGweaver is presented in Figure 3.4. KwARG demonstrates good efficiency when the recombination and mutation rates are relatively low, and shows roughly linear growth in run time as the mutation rate increases.

The dependence of the run time of KwARG on the number and length of sequences was further investigated through simulations. First, the sequence length was fixed at 5 000bp, and datasets were simulated with varying numbers

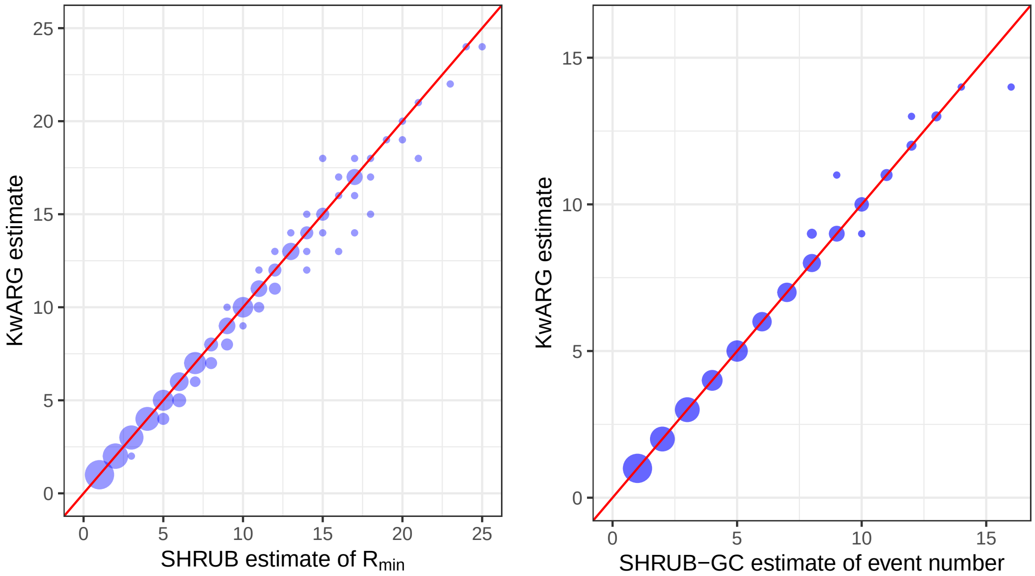

Figure 3.5: Comparison of KwARG to SHRUB and SHRUB-GC. x-axis: estimate produced by SHRUB (left) and SHRUB-GC (right). y-axis: estimate produced by KwARG. Instances where equally good solutions were found lie on the red diagonal line. Size of points is proportional to the number of corresponding datasets.

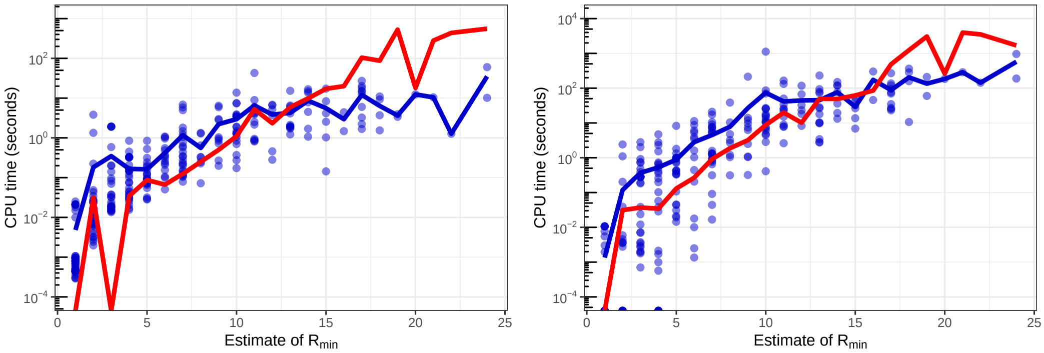

Figure 3.6: Blue points: time taken to run $Q = 20$ iterations of KwARG (left: disallowing recurrent mutations, right: allowing both recombination and recurrent mutation). Blue lines: mean values. Red line: mean run time of SHRUB (left) and SHRUB-GC (right). Time in seconds is given on a log scale.

of sequences (from 2 to 30) using msprime, with the infinite sites assumption (parameters: $N_e = 10\,000$, mutation rate $2 \cdot 10^{-7}$ per site per generation, recombination rate $2 \cdot 10^{-7}$ per site per generation). For each number of sequences, 500 simulations were carried out; for each dataset, KwARG was run once and the runtime recorded. The results are presented in the left panel of Figure 3.7. KwARG runs very quickly when the number of sequences is very low, and shows roughly exponential growth in run time when the number of sequences is 6 or more.

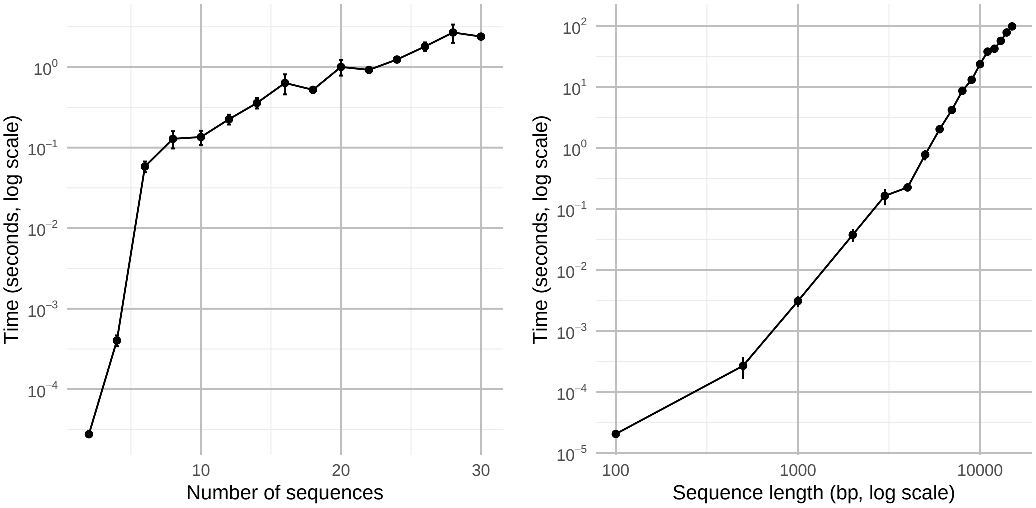

Figure 3.7: Run time versus number of sequences (left panel) and sequence length (right panel). Lines show mean run time over 500 (100) datasets; error bars show mean ± standard error.

Next, the number of sequences was fixed at 20, and datasets were simulated with varying sequence lengths (from 100 to 15 000bp) using msprime, with the infinite sites assumption (same parameters as above). For each sequence length, 100 simulations were carried out; for each dataset, KwARG was run once and the runtime recorded. The results are presented in the right panel of Figure 3.7. After an initial exponential increase (due to small datasets taking very little time per iteration), the run time scales roughly linearly in sequence length.

3.4 Application to Kreitman data

The performance of KwARG is now illustrated on the classic dataset of Kreitman (1983, Table 1); the size of the dataset is not close to the performance limit of KwARG, but this data has been widely used for benchmarking algorithms used for ARG reconstruction. The dataset consists of 11 sequences and 2 721 sites, of which 43 are polymorphic, of the alcohol dehydrogenase locus of *Drosophila melanogaster*. The data is shown in Figure 3.8, with columns containing singleton mutations removed for ease of viewing. Applying the 'Clean' algorithm, as described in Section 3.2.2.1, reduces this to a matrix of 9 rows and 16 columns.

Zeros correspond to:

	C	C	C	C	A	A	G	G	C	G	A	C	C	C	C	G	G	A	T	C	T	C	T	A	T	T	C	G	C	C
Wa-S	0	0	1	0	0	0	0	0	0	1	0	0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1	0	1	0
Fl-1S	0	0	0	0	0	1	0	0	0	1	0	0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1	0	1	0
Af-S	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0
Fr-S	0	0	1	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0
Fl-2S	0	0	1	1	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0
Ja-S	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Fl-F	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0
Fr-F	1	1	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1
Wa-F	1	1	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1
Af-F	1	1	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1
Ja-F	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0
	1	*2*	*3*	*4*	*5*	*9*	*11*	*12*	*13*	*16*	*17*	*18*	*19*	*20*	*22*	*23*	*24*	*26*	*27*	*28*	*29*	*30*	*31*	*32*	*33*	*34*	*35*	*36*	*37*	*38*

Figure 3.8: Illustration of the Kreitman dataset. The 11 sequences are labelled as presented by Kreitman (1983); polymorphic sites are labelled 1–43 and columns with singleton mutations are not shown.

3.4.1 Parameters

KwARG was run with the default parameters, $Q = 500$ times for each of 13 default cost configurations given in Section 3.2.2.4. An example of the output is shown in Table 3.1.

The effectiveness of using the default range of cost configurations is illustrated in Figure 3.9, which is based on the set of all possible minimal solutions identified for the dataset. Fixing $C_R = 1.0$ and $C_{RR} = 2.0$, each tile represents a pair (C_{SE}, C_{RM}). Each tile is coloured and labelled according to the corresponding cost-optimal solution, in the form $\{x, y, z\}$, giving the number of SE, RM and recombination events, respectively. For instance, if $C_{SE} = 0.5$ and $C_{RM} = 0.61$, the solutions $\{3, 0, 3\}$ (with cost $3 \cdot 0.5 + 3 \cdot 1.0 = 4.5$) and $\{5, 0, 2\}$ (with cost $5 \cdot 0.5 + 2 \cdot 1.0 = 4.5$) have the lowest costs over all feasible solutions.

The default cost configuration in (3.2.2) includes all pairs (C_{SE}, C_{RM}) on the diagonal in this plot, falling on the red line. This line crosses all optimal

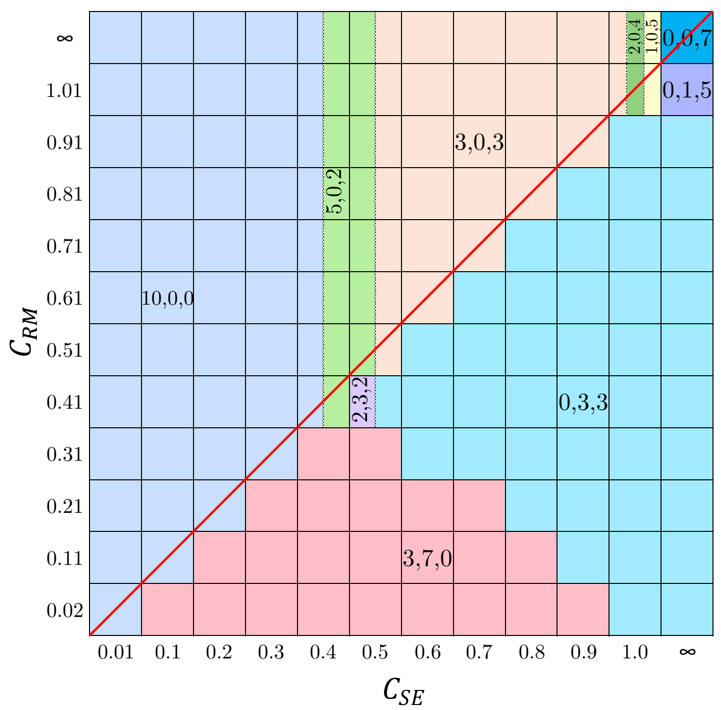

Figure 3.9: Solution tile plot for the Kreitman dataset.

solutions which maximise the number of SE events for each possible number of recombinations. Such events affect only a single sequence at a single site in the input dataset, so are, in a sense, more parsimonious than recurrent mutations occurring on internal branches.

3.4.2 Results

KwARG correctly identified the R_{min} of 7 and the P_{min} of 10 (confirmed by running Beagle and PAUP*, respectively). The 6 500 iterations of KwARG took just under 9 minutes to run. Of these, 1,829 (28%) resulted in optimal solutions; some are shown in Table 3.1. KwARG identified multiple combinations of recombinations and recurrent mutations that could have generated this dataset. By default, slightly cheaper costs are assigned to recurrent mutations if they happen on terminal branches, so the results show a bias towards solutions with more SE events for each given number of recombinations.

The ten recurrent mutations appearing in the solution in row 8 of Table 3.1 are highlighted on the dataset in Figure 3.8. It is striking that 7 of these 10 recurrent mutations affect the same sequence Fl-2S. In fact, these 7 recurrent mutations could be replaced by 3 recombination events affecting

	Seed	T	C_{SE}	C_{RM}	C_R	C_{RR}	SE	RM	R	$\sum_t \lvert\mathcal{N}_t\rvert$
1	2263536315	30.0	∞	∞	1.00	2.00	0	0	7	143
2	2347021759	30.0	0.90	0.91	1.00	2.00	1	0	6	853
3	1791455164	30.0	0.80	0.81	1.00	2.00	1	0	5	728
4	1684879495	30.0	0.60	0.61	1.00	2.00	2	0	4	783
5	1884182000	30.0	0.40	0.41	1.00	2.00	3	0	3	806
6	1900122424	30.0	0.20	0.21	1.00	2.00	5	0	2	702
7	2111915557	30.0	0.10	0.11	1.00	2.00	8	0	1	833
8	2888657821	30.0	0.01	0.02	1.00	2.00	10	0	0	715

Table 3.1: Example output of KwARG for the Kreitman dataset. SE: number of recurrent mutations occurring on terminal branches of the ARG (possible sequencing errors). RM: number of other recurrent mutations. R: number of recombinations. Last column gives the total number of neighbourhood states considered.

sequence Fl-2S, with breakpoints just after sites 3, 16, and 35; leaving the other identified recurrent mutations unchanged yields the solution in row 5 of Table 3.1. These findings suggest that the sequence may have been affected by cross-contamination or other errors during the sequencing process, or it could indeed be a recombinant mosaic of four other sequences in the sample. This recovers the results obtained by Stephens and Nei (1985), who posited the recombinant origins of sequence Fl-2S following manual examination of a reconstructed maximum parsimony tree, which also highlighted the five consecutive mutations identified by KwARG. The ARG corresponding to the solution in row 5 of Table 3.1, visualised using Graphviz (Ellson et al., 2004), is shown in Figure 3.10.

Examination of the identified solutions also shows that site 36 of sequence Ja-S "necessitates" two of the seven recombinations inferred in the minimal solution in the absence of recurrent mutation, while sites 3 and 9 in sequences Wa-S and Fl-1S, respectively, each create incompatibilities that could be resolved by one recombination.

3.5 Discussion

Methods for the reconstruction of parsimonious ARGs generally rely on the infinite sites assumption. When examining the output ARGs, it is often difficult to tell by how much the inferred recombination events actually affect the recombining sequences. As is the case with the Kreitman dataset, sometimes

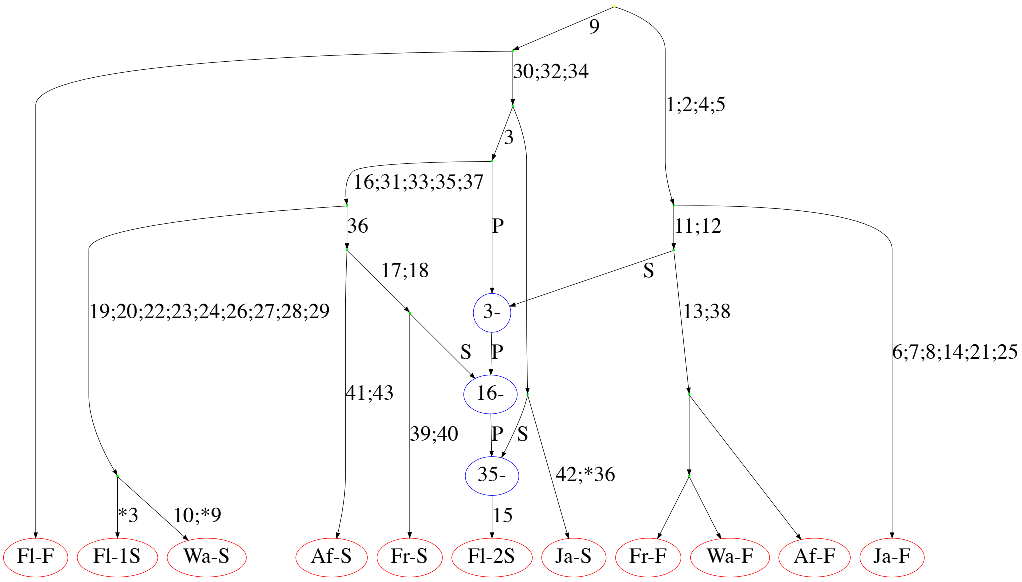

Figure 3.10: ARG constructed for the Kreitman data. Edges are labelled with sites undergoing mutations; recurrent mutations are prefixed with an asterisk. Recombination nodes, in blue, are labelled with the recombination breakpoint; material to the right (left) of the breakpoint is inherited from the parent connected by the edge labelled S (P) for “suffix” (“prefix”).

further examination reveals that two crossover recombination events have the same effect as one recurrent mutation, raising questions about which version of events is more likely. KwARG removes the need for such manual examination, and provides an automated way of highlighting such cases, which is particularly useful for larger datasets.

While KwARG performs well in inferring ARGs under the infinite sites assumption, it can be particularly useful in analysing genetic data from organisms whose genomes are reasonably likely to undergo recurrent mutation, such as viruses with relatively high mutation rates and short genomes. One such application is described further on, in Chapter 4, where the output of KwARG is combined with probabilistic arguments to investigate the presence of ongoing recombination in SARS-CoV-2.

The solutions identified by KwARG differ in the proportion of recurrent mutations to recombinations, ranging from an explanation that invokes only recombination events to one that invokes only mutation events. As is the case with other heuristic and parsimony-based methods, KwARG cannot offer uncertainty quantification for the inferred ARGs. Quantifying the likelihood of each scenario will be application-specific; for instance, one can choose a reasonable model of evolution for the population being studied, and identify the most likely solution under a range of reasonable mutation and recombination rates. When the presence or absence of recombination is not certain, then should the number of recurrent mutations needed to explain the dataset be infeasibly large, this provides evidence for the presence of recombination; this is the idea underlying the homoplasy test of Maynard Smith and Smith (1998). If the largest "reasonable" number of recurrent mutations is then estimated, KwARG can be used to say how many additional recombination events are required to explain the dataset.

KwARG performs well when compared against exact parsimony methods for the 'recombination-only' and 'mutation-only' scenarios. Because of the random exploration incorporated within KwARG, it should be run multiple times on the same dataset before selecting the best solutions; the optimal run length of KwARG will be constrained by timing and the available computational resources. To gauge whether KwARG has run enough iterations, one could proceed by calculating R_{min} and P_{min} either exactly (if the data is reasonably small) or using other heuristics-based methods (such as SHRUB or PAUP*), to confirm whether KwARG has found good solutions at these two extremes.

The range of solutions explored by KwARG is guided by the choice of cost parameters. As a rule of thumb, simulations have shown that if the mutation and recombination rates are similar, costs near one give good accuracy of solutions in terms of reconstructing local tree topologies; if the mutation rate is significantly higher (resp. lower) than the recombination rate, the cost should be set to less than (resp. greater than) one. As KwARG incorporates a degree of random exploration, a range of solutions will still be obtained; the best choice of parameters will depend strongly on the nature and aims of the analysis being performed.

For model-based inference, the modelling assumptions can clearly affect the quality of the results; however, a parsimony-based approach also makes the strong assumption that the minimal ARG can capture useful information about the history of a sample. The veracity of this assumption will depend on the true recombination rate. Based on comparisons with RENT+, tsinfer, and ARGweaver, KwARG achieves very good accuracy of inference of local tree topologies at least comparable to these other methods, particularly when the recombination rate is low to moderate and the mutation rate moderate to high. KwARG demonstrates relatively good accuracy even when the recombination rate is high and even though its express goal is to seek the most parsimonious, rather than necessarily the most likely, history. Moreover, for datasets with relatively few incompatibilities, the run time of KwARG is competitive with that of the other methods. It is also interesting to note that although all four programs incorporate very different approaches and heuristic algorithms, they demonstrate very similar performance in inferring local tree topologies over the range of considered scenarios.

The scalability of KwARG remains a challenge for large and more complex datasets. Performance gains could be readily achieved by running multiple iterations of KwARG in parallel, or incorporating more efficient ways of storing the intermediate states. Further improvements could also be obtained by amending the calculation of lower bounds within the cost function in order to account for the presence of recurrent mutation, which should make the scores more accurate, and hence the neighbourhood exploration more efficient. Other avenues for further work include explicitly incorporating gene conversion as a possible type of recombination event with a separate cost parameter, with a view to developing the underlying model of evolution to even more closely reflect biological reality.

Chapter 4

Recombination detection for SARS-CoV-2

4.1 Introduction

In this chapter, KwARG is used to detect and examine crossover recombination events in samples of SARS-CoV-2 viral consensus sequences. This approach provides a concrete way of describing their genealogical relationships, sidestepping the challenges presented by discrepancies in clade assignment, enabling the detection of intra-clade recombination, avoiding the need to specify a particular model of evolution, and allowing for the explicit identification of possible recombination events in the history of a sample. The method naturally handles both recombination and recurrent mutation, identifying a range of possible explicit genealogical histories for the dataset with varying proportions of both events types. Rather than using summary statistics calculated from the data, or focussing only on patterns of clade-defining mutations, the method utilises all of the information contained in the patterns of incompatibilities observed in a sample, allowing for powerful detection and identification of possible recombinants. Moreover, a nonparametric framework is presented for evaluating the probability of a given number of recurrent mutations, thus quantifying how many recombinations are likely to have occurred in the history of a dataset. This allows for a more thorough and statistically principled assessment of the extent to which ongoing recombination is occurring.

The presence of ongoing recombination in SARS-CoV-2 is investigated using publicly available data from GISAID (Elbe and Buckland-Merrett, 2017), collected between November 2020 and February 2021. Using data from South

Africa, the method detects recombination both when the sample contains sequences from multiple distinct lineages ('inter-clade'), as well as all from the same lineage ('intra-clade'). Further, the method can accurately detect consensus sequences carrying patterns of mutations that are consistent with recombination, flagging these sequences for further investigation—and, using data from England, it can identify both sequences arising as a result of sequencing errors due to sample contamination, aiding in identifying quality control issues, as well as sequences likely to be true recombinants. The method is validated using extensive simulation studies, and through application to Middle East respiratory syndrome coronavirus (MERS-CoV) data, for which evidence of recombination is identified, in agreement with previous studies.

Details of the data used are given in Section 4.2. An outline of the method is presented in Section 4.3, with details of genealogy reconstruction and evaluation of the resulting solutions given in Sections 4.4 and 4.5, respectively. Results are presented in Section 4.6, and discussion follows in Section 4.7.

4.2 Data

SARS-CoV-2 sequencing data is publicly available from GISAID at `gisaid.org` upon free registration. MERS-CoV data is publicly available from the NCBI Virus database at `ncbi.nlm.nih.gov/labs/virus`. Code used in processing the data and carrying out the analysis (with step-by-step instructions) is available at `github.com/a-ignatieva/sars-cov-2-recombination`.

4.2.1 SARS-CoV-2

Sequences were downloaded from GISAID, and aligned as described below. Masking was applied to sites at the endpoint regions of the genomes, any multi-allelic sites, regions with many missing nucleotides in multiple sequences, and sites identified by De Maio et al. (2020) as being highly homoplasic or prone to sequencing errors. Strict quality criteria were applied to remove any sequences with a large number of ambiguous nucleotides, excessive gaps, and groups of clustered mutations; additionally, sites identified by van Dorp et al. (2020a) as being prone to recurrent mutation were masked. These measures were aimed at reducing the possibility of including poor quality or contaminated sequences in the analysed samples, and also masking sites that are known to be highly homoplasic (either due to recurring sequencing errors, or due to the effects of selection).

The timing and location of samples was selected to coincide with periods of high transmission numbers, as this increases the probability of co-infection of the same host with multiple strains, which is a requirement for recombination to occur. Collection dates were also restricted to reasonably narrow windows, as KwARG assumes that the sequences are sampled contemporaneously. Four samples were analysed:

- from South Africa, collected in
  - November 2020: 50 sequences, with 25 from lineage B.1.351 (Beta variant), and 25 from other lineages;
  - February 2021: 38 sequences, all from lineage B.1.351;
- from England, collected in
  - November 2020: 80 sequences, with 40 sequences from lineage B.1.1.7 (Alpha variant) and 40 from other lineages within GISAID clade GR (which contains B.1.1.7);
  - December 2020 – January 2021 (40 sequences within GISAID clade GR).

A full table of acknowledgements for the data used is provided at `github.com/a-ignatieva/sars-cov-2-recombination/tree/main/GISAID_acknowledgements`.

4.2.1.1 Alignment and masking

SARS-CoV-2 sequences were downloaded from GISAID, filtering for those labelled as complete (>29 000bp, out of the total genome length of 29 903bp), collected from human hosts, and excluding any with more than 5% ambiguous nucleotides and incomplete collection dates. Although SARS-CoV-2 is an RNA virus, nucleotides will be referred to by their DNA type for consistency with the sequencing data (i.e. the base type T corresponds to U on the actual SARS-CoV-2 genome).

Alignment to the reference sequence collected in Wuhan in December 2019 (Wu et al., 2020) (GISAID accession: EPI_ISL_402125, GenBank: MN908947.3) was performed using MAFFT v7.475 (Katoh and Standley, 2013), with the options: `auto`, `keeplength`, `preservecase`, `addfragments`.

The following sites were masked from the data:

- the endpoint regions with a large number of missing nucleotides (1–55bp and 29 804–29 903bp);
- 322 further sites identified as problematic by De Maio et al. (2020) (prone to sequencing errors, known to be excessively homoplasic, or otherwise of questionable quality);
- any multi-allelic sites.

4.2.1.2 Quality criteria

Any sequences failing the following quality criteria were removed:

- at most 500 missing nucleotides (excluding start and end of alignment);
- at most 1 non-ACTG character;
- at most 25 gaps;
- no mutation clusters (more than 6 mutations in a window of 100 nucleotides, excluding known and verified clusters).

Nextclade (Hadfield et al., 2018, tool available at clades.nextstrain.org) was used to check sampled sequences against these criteria (and it was ensured that any sequences assigned a score of "bad" by the tool were removed). In addition, the 198 sites identified by van Dorp et al. (2020a, Supplementary Table S5) as potentially highly homoplasic were masked.

4.2.1.3 South Africa (November)

All sequences collected in South Africa in November 2020 were downloaded and aligned as described in Section 4.2.1.1. Removing 48 sequences flagged by the submitter as containing long stretches of ambiguous nucleotides, and applying the quality criteria in Section 4.2.1.2, left a total of 278 sequences.

The aligned sequences were split into the datasets SA_N (the 177 sequences labelled as belonging to variant 501Y.V2 (Beta) in GISAID) and SA_O (the other 101 sequences). A sample of 25 sequences from each of SA_O and SA_N was selected at random using SeqKit (Shen et al., 2016).

Masking was carried out as described in Section 4.2.1.1; in addition, sites 22 266–22 745 were masked, as many of the sequences contained a large number of ambiguous nucleotides at these positions. No further multi-allelic sites were identified. Of the total 1 125 masked positions, 28 corresponded to segregating sites in the dataset.

The resulting sample comprises 50 sequences with 207 variable sites. The corresponding GISAID accession numbers and collection dates are given in Appendix B, Table B.1.

4.2.1.4 South Africa (February)

All sequences collected in South Africa in February 2021 were downloaded, aligned, and masked as described in Section 4.2.1.1, also masking sites 22 266–22 745; no additional multi-allelic sites were identified. The quality filters detailed in Section 4.2.1.2 were applied. One sequence in the resulting sample was not from lineage B.1.351 and was removed. Of the total 1 125 masked positions, 17 corresponded to segregating sites.

The resulting sample consists of 38 sequences, all from lineage B.1.351, with 151 variable sites. The corresponding GISAID accession numbers and collection dates are given in Appendix B, Table B.2.

4.2.1.5 England (November)

All sequences labelled as clade GR, collected in England in November 2020, were downloaded and aligned as per Section 4.2.1.1. Exact duplicates of sequences in the dataset were removed, to avoid including identical sequences in the sample. The sequences were then split into datasets E_N (934 sequences labelled as belonging to lineage B.1.1.7) and E_O (the other 2 650 sequences).

A sample of 40 sequences from each of E_O and E_N was then selected at random using SeqKit. Sites were masked as detailed in Section 4.2.1.1. Three multi-allelic sites were identified and masked, at positions 12 067, 21 724, and 22 992. Of the total 477 masked positions, 10 corresponded to segregating sites in the dataset. The quality control criteria in Section 4.2.1.2 were *not* applied to this sample.

The resulting sample comprises 80 sequences with 363 variable sites. The corresponding GISAID accession numbers and collection dates are given in Appendix B, Table B.3.

4.2.1.6 England (January)

All sequences labelled as clade GR, collected in England in December 2020 to January 2021, were downloaded and aligned as per Section 4.2.1.1. Sites were masked as detailed in Section 4.2.1.1, and the quality filters detailed in Section 4.2.1.2 were applied. A sample of 38 sequences was selected at random using SeqKit, from among sequences uploaded by the COVID-19 Genomics UK Consortium; additionally, the sequence EPI_ISL_994038 (E39) identified as a potential recombinant by Jackson et al. (2021), and its potential parent sequence EPI_ISL_820233 (E40), were included. Five multi-allelic sites were identified and masked, at positions 21 255, 23 604, 24 914, 28 310, and 29 227. Of the total 660 masked positions, 35 corresponded to segregating sites in the dataset.

The resulting sample comprises 40 sequences with 276 variable sites. The corresponding GISAID accession numbers and collection dates are given in Appendix B, Table B.4.

4.2.2 MERS-CoV

MERS-CoV sequences were downloaded from the NCBI Virus database (Hatcher et al., 2017), filtering for those labelled as complete, human host, collected in Saudi Arabia in January to March 2015. Alignment to the reference sequence (HCoV-EMC/2012, accession number NC_019843.3) was performed using MAFFT, with the same options as in Section 4.2.1.1. Masking of the first and last 150 sites of the alignment was performed. Of the 300 masked sites, two were segregating in the dataset; no multi-allelic sites were identified. The resulting sample consists of 19 sequences with 197 variable sites. The corresponding accession numbers are given in Appendix B, Table B.5.

4.3 Methods

The method consists of two main steps. Firstly, using KwARG, plausible genealogical histories are reconstructed for each sample, with varying proportions of posited recombination and recurrent mutations events. Then, simulation is used to approximate the distribution of the number of recurrent mutations that might be observed in a dataset of the same size as each sample. This is used to establish which of the identified genealogical histories is more plausible for the data at hand, and thus whether the presence of recombination events in the history of the given sample is likely.

This can be framed in the language of statistical hypothesis testing. The 'null hypothesis' is the absence of recombination. The test statistic T is the number of recurrent mutations in the history of the dataset; the null distribution of T is approximated through simulation. The observed value T_{obs} is the minimal number of recurrent mutations required to explain the dataset in the absence of recombination, as estimated by KwARG. The 'p-value' is the probability of observing a number of recurrent mutations equal to or greater than T_{obs}. Small p-values allow the null hypothesis to be rejected, providing evidence that recombination has occurred. The reconstructed genealogies then allow for the detailed examination of possible recombination events in the history of the sampled sequences.

Note that very conservative assumptions are made throughout, both in processing the data and in estimating the distribution of the number of recurrent mutations. Moreover, the number of recurrent mutations required to explain a given dataset computed by KwARG is (or is close to) a lower bound on the actual number of such events, and is likely to be an underestimate, making the reported p-values larger (more stringent).

4.4 Reconstruction of genealogies

The first step in the approach is to use KwARG to reconstruct possible genealogical histories for the given datasets. For each dataset, KwARG was run $Q = 500$ times for each combination of the following values of the annealing parameter T and event costs $(C_{SE}, C_{RM}, C_R, C_{RR})$:

$$\begin{aligned} T &\in \{30, 50\} \\ (C_{SE}, C_{RM}, C_R, C_{RR}) &\in \{(\infty, \infty, 1, 2), (1.9, 1.91, 1, 2), (1.8, 1.81, 1, 2), \\ &\quad (1.7, 1.71, 1, 2), \ldots (0.1, 0.11, 1, 2), \\ &\quad (0.01, 0.02, 1, 2), (1.0, 1.1, \infty, \infty)\}. \end{aligned}$$

For MERS-CoV, the root was left unspecified. For SARS-CoV-2, the reference sequence used for alignment was set as the root. This reference sequence is a genome collected in Wuhan in December 2019 (Wu et al., 2020), giving the most likely rooting based on the available epidemiological evidence; our results do not change significantly if the root is left unspecified.

The results are presented in Table 4.1.

(a) South Africa (Nov)

R	RM	$\mathbb{P}(RM)$	p
10	0	0.28	1.00
8	1	0.35	0.72
6	2	0.23	0.37
4	3	0.10	0.14
3	4	0.03	0.04
2	5	0.01	0.01
1	7	0.00	$4 \cdot 10^{-4}$
0	9	0.00	$7 \cdot 10^{-6}$

(b) South Africa (Feb)

R	RM	$\mathbb{P}(RM)$	p
7	0	0.52	1.00
5	1	0.34	0.48
3	2	0.11	0.14
2	3	0.03	0.03
1	4	0.00	$5 \cdot 10^{-3}$
0	5	0.00	$7 \cdot 10^{-4}$

(c) England (Jan)

R	RM	$\mathbb{P}(RM)$	p
10	0	0.11	1.00
8	1	0.24	0.89
6	2	0.27	0.65
4	3	0.20	0.38
3	4	0.11	0.19
2	5	0.05	0.08
1	6	0.02	0.03
0	14	0.00	$1 \cdot 10^{-6}$

(d) MERS-CoV

R	RM	$\mathbb{P}(RM)$	p
9	0	0.42	1.00
7	1	0.36	0.58
6	2	0.16	0.22
5	3	0.05	0.06
4	4	0.01	0.01
3	5	0.00	$2 \cdot 10^{-3}$
2	10	0.00	$< 1 \cdot 10^{-6}$
1	12	0.00	$< 1 \cdot 10^{-6}$
0	16	0.00	$< 1 \cdot 10^{-6}$

Table 4.1: Summary of solutions identified by KwARG for each sample, and the probability of observing the corresponding number of recurrent mutations. First column: number of recombinations. Second column: number of recurrent mutations. Third column: probability of observing a number of recurrent mutations equal to that in the second column. Fourth column: corresponding p-values (probability of observing a number of recurrent mutations equal to or greater than that in the second column).

4.5 Evaluation of solutions

The next step is to determine which of the solutions identified by KwARG is more likely, by calculating the probability of observing the given number of recurrent mutations. To avoid making model-based assumptions on the genealogy of the sample, a nonparametric method is developed, inspired by the *homoplasy test* of Maynard Smith and Smith (1998).

The homoplasy test estimates the probability of observing the minimal number of recurrent mutations required to generate the sample in the absence of recombination, i.e. if the shape of the genealogy is constrained to be a tree. If this probability is very small, then it provides evidence for the presence of recombination. The test is particularly powerful when the level of divergence between sequences is very low, as is the case with SARS-CoV-2 data, although it appears prone to false positives in the presence of very strong mutation rate heterogeneity along the genome (Posada and Crandall, 2001). I calculate an empirical estimate $\widetilde{P}$ of mutation density along the genome using SARS-CoV-2 data, which does not suggest the presence of extreme heterogeneity, and then use this to simulate the distribution of the number of recurrent mutations that are observed in a sample.

The i-th entry of the vector $\widetilde{P}$, for $i \in \{1, \dots, 29\,903\}$, gives an estimated probability that when a mutation occurs, it affects the i-th site of the genome. Briefly, this estimate is calculated by examining the locations of sites that have undergone at least one mutation (segregating sites) using GISAID data collected in February 2021. If the mutation rate were constant along the genome, one would expect segregating sites to be spread uniformly throughout the genome; uneven clustering of the mutations gives an indication of mutation rate heterogeneity. A nonparametric method (wavelet decomposition) is used to estimate $\widetilde{P}$ from the observed positions of segregating sites, taking into account the dependence of the mutation rate on the base type of the nucleotide undergoing mutation, which is significant for SARS-CoV-2 (Simmonds, 2020; Koyama et al., 2020).

The estimate of $\widetilde{P}$ is then used to approximate the distribution of the number of recurrent mutations observed in a sample, using a simulation approach. The process of mutations falling along the genome is simulated until the simulated number of segregating sites matches that observed in the sample; the vector $\widetilde{P}$ controls where on the genome each mutation falls. The number of recurrent mutations (instances where mutations fall on the same site multiple times) is recorded, and after repeating this procedure a histogram of the results is constructed.

4.5.1 Distribution of the number of recurrent mutations

Let M be the length of the genome, and let m be the number of observed variable sites in the sample. The goal is to estimate the distribution of the number of recurrent mutations that have occurred; that is, the excess number of mutation events beyond the minimum m needed to explain the variability in the sample.

Regardless of any modelling assumptions on the evolution of a given sample or the genealogical relationships between the sequences, it is clear that at least m mutation or sequencing error events must have occurred in the history of the sample (here, a 'sequencing error' refers to the variant at a site being incorrectly called during the sequencing process). Suppose that each time such an event occurs (disregarding which particular sequence is affected), a position on the genome is selected at random with replacement, according to a probability vector P of length M. This corresponds to assuming that (i) such events occur independently from each other, (ii) all sequences have the same probabilities P of a mutation or sequencing error event occurring at each particular site. Moreover, assume that (iii) if a site undergoes at least one mutation in the history of the sample, the site is segregating in the data; and (iv) any sequencing errors fall on each site with probability proportional to P. The validity of these assumptions is discussed below in Section 4.5.3.

The number of recurrent mutations in a sample with m variable sites can then be simulated using Algorithm 5. This is a 'balls-into-bins' type simulation, in which balls are placed one-by-one into M bins, each time selecting a bin at random with probability proportional to P, until m bins contain at least one ball; the output is the total number of balls thrown minus m. Executing Algorithm 5 multiple times and calculating a histogram of the results gives an approximation to the distribution of the number of recurrent mutations given the number m of observed segregating sites.

4.5.2 Mutation rate heterogeneity along the genome

Parts of the genome with a relatively higher mutation rate are more likely to undergo recurrent mutation, so it is important to incorporate the effects

Algorithm 5: Simulating the number of recurrent mutations conditional on observing m variable sites

```
Input: $M$, $m$, $P$
Output: Number of recurrent mutations $\widetilde{m}$
Initialise $\widetilde{m} = 0$, $S = \{\emptyset\}$;
while $|S| < m$ do
    Draw $s$ from $\{0, \ldots, M\}$ with probabilities proportional to $P$;
    if $s \notin S$ then
        $S \leftarrow S \cup s$;
    end
    $\widetilde{m} \leftarrow \widetilde{m} + 1$ ;
end
$\widetilde{m} \leftarrow \widetilde{m} - m$ ;
return $\widetilde{m}$;
```

of mutation rate heterogeneity. An empirical estimate of mutation density is used to approximate the variation in mutation rate along the genome.

4.5.2.1 Data

All 17 908 sequences in GISAID collected around the world between 1 and 3 February 2021 were downloaded, filtering for sequences labelled as complete (>29 000bp), high coverage, and excluding any with more than 5% ambiguous nucleotides. Alignment was performed as described in Section 4.2.1.1. SNP-sites (Page et al., 2016) was used to extract the positions of the 13 747 identified segregating sites; a vector $\overline{P}$ of length 29 903 was then formed, with a 1 entry at position i if there had been at least one mutation at position i of the genome, and 0 otherwise.

Note that an alternative approach would be to fit a tree to the sequencing data (using maximum likelihood, for instance), count the minimum number of mutations required at each site of the genome, and use this to estimate P. However, this was found to result in very noisy estimates, and provide worse quantification of mutation rate heterogeneity (which was confirmed through simulation studies).

4.5.2.2 Smoothing

The mutation density along the genome was then estimated nonparametrically from $\overline{P}$ by smoothing using wavelet decomposition, as implemented in the R package `wavethresh` (Nason et al., 2010). This method was chosen as it does not require selecting a particular model, and it captures both fine-scale and broad variation in mutation density, allowing for the calculation of a smoothed estimate of $\overline{P}$ incorporating both local and large-scale rate heterogeneity.

Wavelet decomposition can be used to obtain an estimate of a signal from a set of discrete observations. The main idea is similar to that of the Fourier transform, whereby a function is decomposed into a sum of projections onto a particular basis. The distinction is that while the Fourier transform captures only global properties of the signal, wavelet decomposition can be used to analyse variation in the data at both local and increasingly coarser scales (Nason, 2008).

Given $M = 2^n$ observations of sites, corresponding to the entries of $\overline{P}$ (padding the vector $\overline{P}$ to the nearest power of 2 by reflecting the data at the endpoints), n iterations are performed, and at the i-th iteration, (1) coefficients are computed using (non-overlapping) subsets of 2^i neighbouring observations, and (2) these coefficients are used to refine a smoothed estimate of the data using the chosen wavelet basis. The computation of coefficients and the smoothed approximations is governed by the choice of wavelet shape; I used Daubechies' least-asymmetric wavelets (Daubechies, 1988) with six vanishing moments (other choices of wavelet basis produced similar results).

Wavelet *shrinkage* can be used to obtain a smoothed estimate of the observations and remove noise: coefficient selection is performed by only keeping coefficients with values above a certain threshold and setting the others to zero. There are myriad ways of calculating such a threshold (Nason, 2008); I applied the empirical Bayes method of Johnstone and Silverman (2005b) implemented in the R package `EbayesThresh` (Johnstone and Silverman, 2005a).

As the mutation rate is dependent on the base type of the nucleotide undergoing mutation (Simmonds, 2020; Koyama et al., 2020), $\overline{P}$ was split into four parts by the corresponding base type in the reference sequence, and the wavelet decomposition and thresholding performed separately for each part before joining them back together. The resulting smoothed estimate $\widetilde{P}$ is shown in Figure 4.1. The total estimated mutation probability for each base type closely matches the actual proportion of mutations that fall on sites of each base type in the data, as desired. The smoothing method has clearly identified both localised and long-range variation in mutation density along the genome.

To check consistency of the results across time periods, data from September–November 2020 was also used to produce smoothed estimates of $\overline{P}$ (consisting of 41 376 sequences with 14 263 variable sites). The resulting estimate was

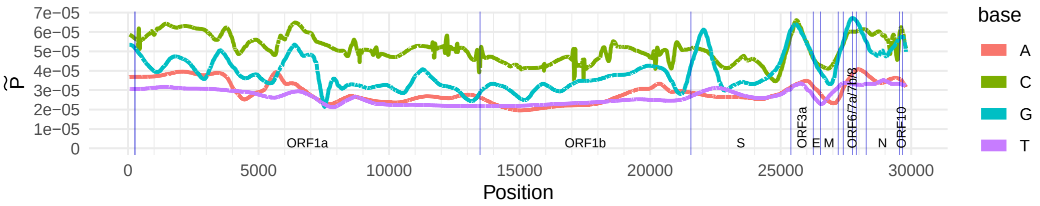

Figure 4.1: Estimate $\widetilde{P}$ of the probability of a mutation falling on each site of the SARS-CoV-2 genome. Blue vertical lines mark endpoints of the labelled open reading frames (ORFs) and genes as per Wu et al. (2020).

found to agree closely to that obtained using the February data, so the latter was used in further analysis.

4.5.3 Validity of assumptions

The validity of the assumptions stated in Section 4.5.1 is now considered in detail. Assumption (i) appears reasonable for the data at hand. Assumption (iii) can be violated if a mutation arising on a branch of the genealogy subsequently reverses through recurrent mutation: either on the same branch before it splits, or independently on every child branch subtending the mutation. Note that the probability of such events depends on the distribution of branch lengths in the genealogy; simulations using the standard coalescent model show that the probability of such events is small. Moreover, such events can never create incompatibilities in the data, so their possibility can be ignored, as the solutions identified by KwARG will never include such recurrent mutation events.

Regarding assumption (ii), as the mutation rates depend on the base type, it will not be true in reality that all sequences have exactly the same probabilities P of mutating at each particular site, as this will depend on the nucleotides carried by the sequence. However, the effect of this violation should be negligible, given the relatively low overall rate of mutation for SARS-CoV-2.

To make the approximation even more conservative, m is increased by adding back the number of masked segregating sites (which are as stated in Sections 4.2.1.3 to 4.2.1.6), and further the number of sites is multiplied by a penalty factor of $F = 1.1$, which is justified in Section 4.5.3.1 below. Thus, assumption (iv) is addressed by noting that sites that are excessively prone to sequencing errors have been masked, so correspondingly M is decreased by the number of masked sites and the corresponding entries of $\widetilde{P}$ are deleted. It is then reasonable to assume that sequencing errors occurring at the non-masked sites affect each site with the same probabilities as mutations. The effects of this assumption being violated are explored further in Section 4.5.3.2.

4.5.3.1 Choice of penalty factor

As noted above, the number of segregating sites in the sample is multiplied by a penalty factor F before performing the simulations. This results in a larger number of recurrent mutations being simulated, skewing the distribution to the right and thus ensuring that the p-values calculated from the simulated distribution are reasonably conservative. This is necessary because, as with any regression method that aims to (partially) de-noise the data, there is a risk that the fitted curve underestimates the true mutation rate heterogeneity, which would result in the expected number of recurrent mutations being underestimated, leading to false positives.

The choice of $F = 1.1$ was validated through simulation studies. First, a "true" mutation rate map P_{true} was simulated for 29,903 sites, as a realisation of an autoregressive process. Then, 20 000 mutations were simulated to fall on the genome (allowing sites to mutate multiple times), and the vector $\overline{P}$ was re-created by marking which sites had (or had not) undergone at least one mutation. The method described in Section 4.5.2 was then applied to fit an estimated mutation density P_{fit}. Finally, 10 000 simulations of Algorithm 5 were used to get an estimate of the null distribution: first, using P_{true} with $m \in \{100, 300, 500\}$ sample segregating sites, then using P_{sim} with $m \cdot F$ sample segregating sites, for $F \in \{1.0, 1.1, 1.2, 1.3, 1.4, 1.5\}$.

This procedure was repeated 500 times for each combination of m and F. The results are presented in Figure 4.2. This demonstrates that without the penalty term, the fitted mutation density may indeed fail to capture all of the mutation rate heterogeneity that is present; for instance, when considering a sample with 300 segregating sites, in 46% of cases the 95th percentile of the simulated distribution will be lower than that of the true distribution. The results demonstrate that a value of $F = 1.1$ appears sufficient to negate this effect, without excessively increasing the false negative rate.

4.5.3.2 Presence of highly homoplasic sites

Violations of assumption (iv) can occur if some (non-masked) sites along the genome are highly homoplasic, which can occur due to the effects of selection,

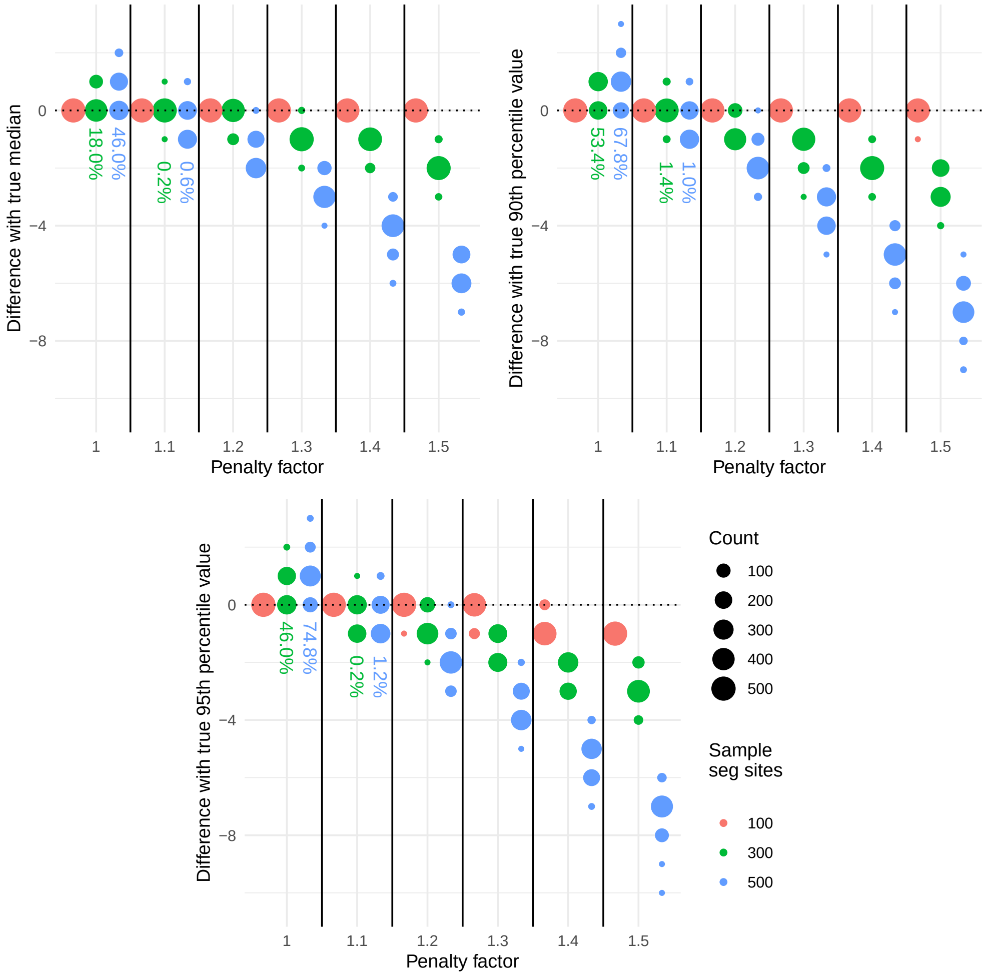

Figure 4.2: Comparison of simulated null distributions using P_{true} and P_{fit}. Points show the difference between the true and simulated median (left panel), 90th percentile (middle), 95th percentile (right), with size proportional to the number of observations, split by penalty factor F (x-axis) and the number of sample segregating sites m (colours). Ideally, the points should be concentrated around 0; values above (below) 0 may result in false positives (false negatives) when using the estimated null distribution. Percentages show the proportion of cases lying above 0.

or as an artifact of the sequencing process. If this assumption were violated, the estimate $\widetilde{P}$ would be missing ‘spikes’ of high probability at the corresponding positions, biasing the simulated null distribution to underestimate the number of recurrent mutations, and potentially leading to false positive results.

The extent to which a violation of assumption (iv) affects the resulting inference was assessed through simulation studies. For each $i \in \{0, 1, 5, 10, 20, 50, 100, 200\}$, i sites of the genome were chosen, and the corresponding probabilities in $\widetilde{P}$ were multiplied by a factor $H \in \{2, 5, 10, 20, 50\}$ to give the vectors $\widetilde{P}_{i,H}$. This recreates the effect of having i sites which are highly homoplasic (with the extent of this controlled by H); an example of $\widetilde{P}_{50,2}$ is shown in Figure 4.3.

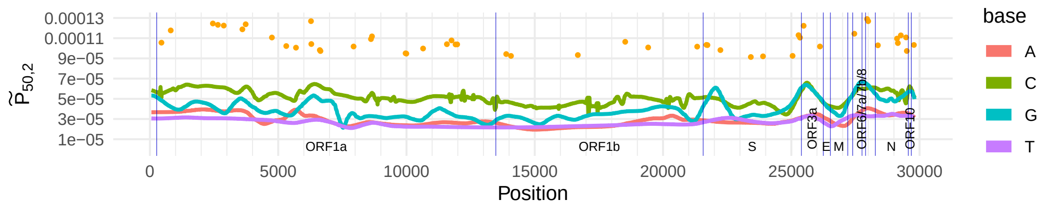

Figure 4.3: Mutation density estimate $\widetilde{P}$ adjusted by selecting $i = 50$ sites and multiplying the corresponding entry of $\widetilde{P}$ by $H = 2$ (resulting values shown in orange), to recreate the presence of 50 highly homoplasic sites.

For each combination of i and H, 200 datasets of 80 sequences were simulated using msprime, with parameters that appear reasonable for SARS-CoV-2:

- $N_e = 1 \cdot 10^6$, exponential growth rate of 1.5 (no appropriate published estimates of these parameters could be identified, but this choice was found to give reasonable values of MRCA time and number of segregating sites for the simulated datasets);
- binary mutation model (finite sites);
- mutation rate per site per generation given by the entries of $\widetilde{P}_{i,H} \times 2 \cdot 10^{-5} \times 29\,903$. This was calculated based on:
  - a mean mutation rate of $8 \cdot 10^{-4}$ per site per year (as used by Nextstrain (Hadfield et al., 2018), accessed through nextstrain.org/ncov/global);
  - a generation time of 7.5 days (Li et al., 2020);
  - giving a mean mutation rate of $2 \cdot 10^{-5}$ per site per generation.

Note that some considerations that may affect viral genealogies were not incorporated into the model (such as multiple mergers and effects of spatial structure), both for simplicity and due to the difficulty in identifying realistic assumptions and reasonable parameters values.

For each dataset, KwARG was run 200 times (parameters: $T = 30$, $Q = 100$, $(C_{SE}, C_{RM}, C_R, C_{RR}) \in \{(1, 1.1, \infty, \infty), (0.01, 0.02, 1.00, 2.00)\}$) to calculate the minimal number of recurrent mutations needed to explain the dataset in the absence of recombination. A p-value was then calculated, using the null distribution simulated using the un-adjusted vector $\widetilde{P}$ and 10 000 iterations of Algorithm 5, with m set to the number of segregating sites in the dataset multiplied by the penalty factor $F = 1.1$.

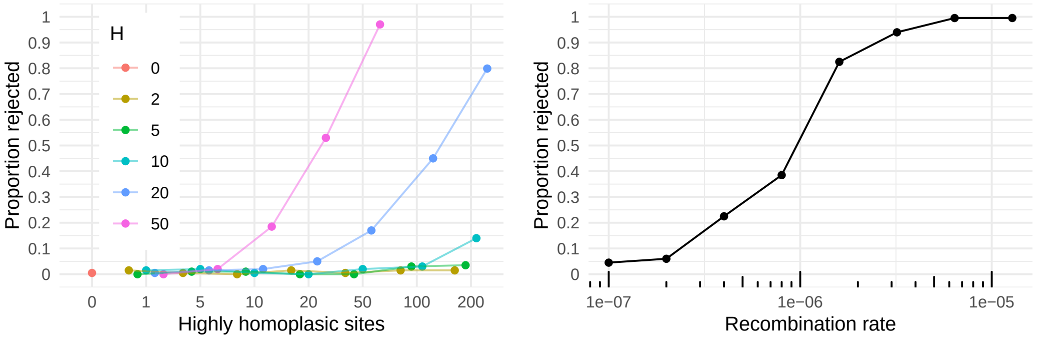

Figure 4.4: Left panel: x-axis shows number of added highly homoplasic sites, with the corresponding entries of $\widetilde{P}$ multiplied by the factor H (colours); y-axis shows the proportion of simulated datasets (out of 200 for each combination of parameters) for which the null hypothesis was (incorrectly) rejected with $p < 0.05$. Right panel: x-axis shows recombination rate (per site per generation) used to simulate 200 datasets, y-axis shows proportion of datasets for which the null hypothesis was rejected with $p < 0.05$.

The proportion of times the null hypothesis was (incorrectly) rejected, with $p < 0.05$, is shown in the left panel of Figure 4.4. False positives were seen in only 0.5% of cases when there are no highly homoplasic sites, demonstrating that the method conservatively overestimates the computed p-values. The proportion of false positives only increases significantly when a large number of extremely homoplasic sites is present, showing that the method is reasonably robust to violations of this assumption. Having applied several stringent quality filters and implemented a conservative strategy in masking sites known to be homoplasic, seeing a large number of extremely hypermutable sites appears improbable, so the method is unlikely to falsely indicate the presence of recombination.

4.5.4 Detection rate vs recombination rate

I now investigate how the proportion of cases in which the null hypothesis is rejected varies with recombination rate. For several values of the recombination rate $1 \cdot 10^{-7} \leq \rho \leq 1 \cdot 10^{-5}$ (per site per generation), 200 datasets were simulated using msprime with the parameters given in Section 4.5.3.2 (using the unadjusted vector $\widetilde{P}$), and the same method used to calculate a p-value for each dataset. It was recorded how often the null hypothesis of no recombination could be rejected (with $p < 0.05$).

The results are shown in the right panel of Figure 4.4, demonstrating that this occurred in 4.5% of cases for $\rho = 1 \cdot 10^{-7}$ per site per generation, rising to 99.5% of cases for $\rho = 1 \cdot 10^{-5}$ per site per generation. The simulations were performed using parameters that appear reasonable for SARS-CoV-2; the results suggest that the method is sufficiently powerful for detecting recombination if the recombination rate is higher than around $\rho = 1 \cdot 10^{-6}$ per site per generation $\approx 4 \cdot 10^{-5}$ per site per year (assuming a generation time of 7.5 days).

4.5.5 Null distribution for MERS-CoV

The same methodology as described above for SARS-CoV-2 was used to simulate the null distribution for MERS-CoV.

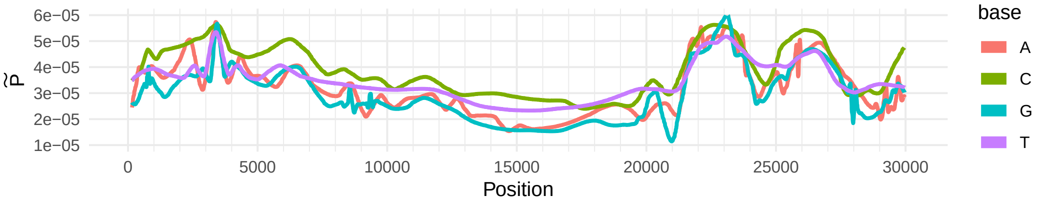

Figure 4.5: Estimate $\widetilde{P}$ of the probability of a mutation falling on each site of the MERS-CoV genome.

Sequences were downloaded from the NCBI Virus database (Hatcher et al., 2017), filtering for those of length at least 20 000bp, from human and camel hosts, across all time periods. Alignment to the reference sequence was performed as described in Section 4.2.2. The alignment comprised 700 sequences with 14 238 variable sites. The vector $\overline{P}$ was constructed, and wavelet decomposition was used to fit the estimate $\widetilde{P}$ in the same manner as described in Section 4.5.2; the result is shown in Figure 4.5.

4.5.6 Null distribution simulation

The null distribution was simulated using the estimate $\widetilde{P}$, after masking the appropriate sites for each dataset as stated in Section 4.2. For each dataset, 1 000 000 iterations of Algorithm 5 were run, with the parameters given in Table 4.2. The resulting probabilities and p-values are shown in the third and fourth columns of Table 4.1.

	SA (November)	SA (February)	England (November)
No. of segregating sites	206	150	363
Plus masked sites	29	18	10
Times penalty factor F	1.1	1.1	1.1
m	259	185	410
Length of genome	29 903	29 903	29 903
Less number of masked sites	1126	1126	477
M (= length of $\widetilde{P}$)	28 777	28 777	29 426

	England (January)	MERS-CoV
No. of segregating sites	276	197
Plus masked sites	35	2
Times penalty factor F	1.1	1.1
m	342	219
Length of genome	29 903	30 119
Less number of masked sites	660	300
M (= length of $\widetilde{P}$)	29 243	29 819

Table 4.2: Null distribution simulation parameters for each of the considered datasets.

4.6 Results

4.6.1 Identification of recombinant sequences

All sequences collected in England in December 2020 – January 2021, labelled as belonging to clade GR in GISAID, were downloaded and processed as described above in Section 4.2.1.6. The resulting sample comprises 40 sequences with 276 variable sites.

An illustration of the sample is provided in Figure 4.6. Choosing a solution with no recombinations, the sites of fourteen recurrent mutations identified by

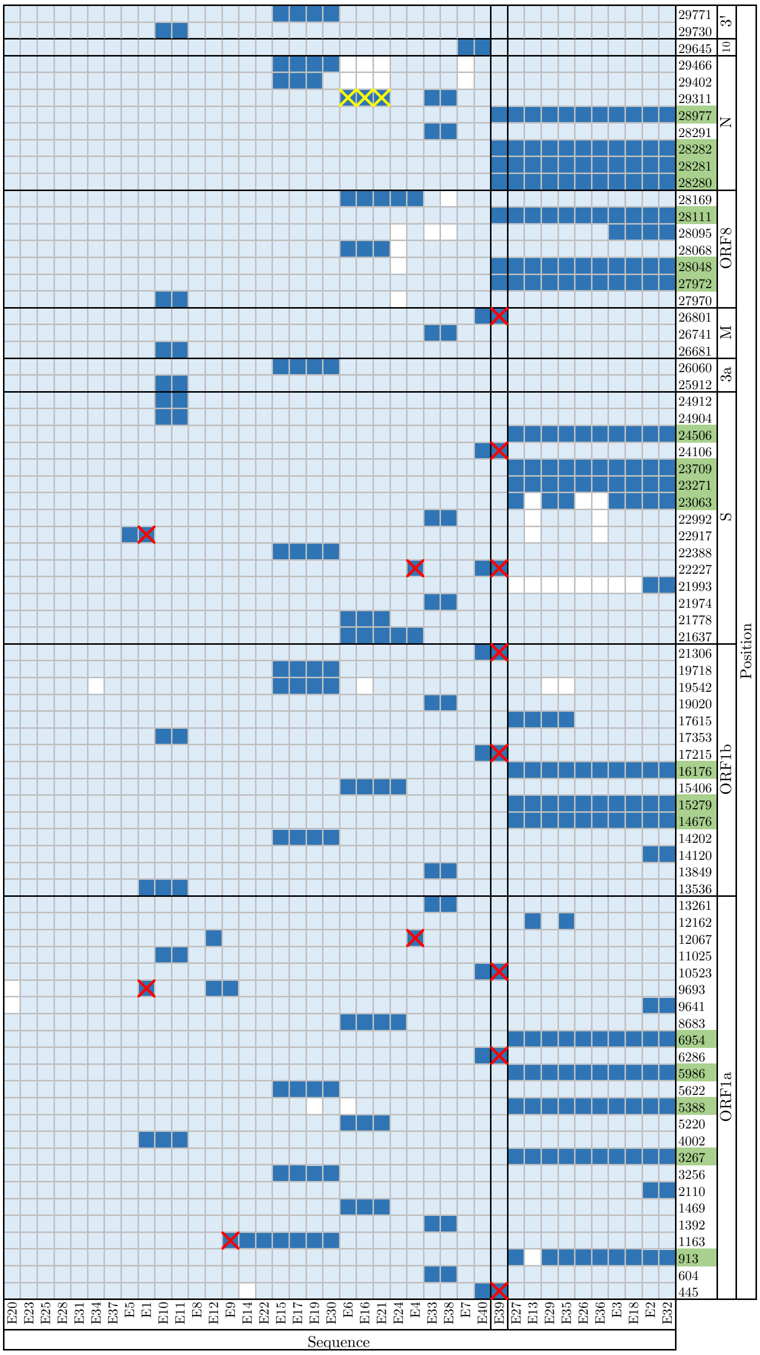

Figure 4.6: Summary of the England (January) dataset. Columns correspond to sequences, labelled on the bottom. Rows correspond to positions along the genome, labelled on the right; uninformative sites (with all 0's or 1's) and those with singleton mutations (with exactly one 1) are not shown. Light blue: ancestral state, dark blue: mutated state, white: missing data. Red crosses highlight sites of recurrent mutations identified by KwARG located on the terminal branches of the ARG (affecting only one sequence). Yellow crosses highlight recurrent mutations on internal branches (hence affecting multiple sequences). Sites bearing the characteristic mutations of lineage B.1.1.7 (Rambaut et al., 2020) are highlighted in green.

KwARG are highlighted with red (resp. yellow) crosses, where the recurrent mutations fall on the terminal (resp. internal) branches of the ARG. The sequencing protocol used by the COVID-19 Genomics UK Consortium, the submitters of the data, generates short amplicons of under 400bp in length, and none of the identified sites of recurrent mutations fall into the same amplicon region, making it less likely that the results are due to sample contamination or other sequencing artifacts. The probability of observing the required $T_{obs} = 14$ or more recurrent mutations is $p = 1 \cdot 10^{-6}$, which strongly indicates the presence of recombination.

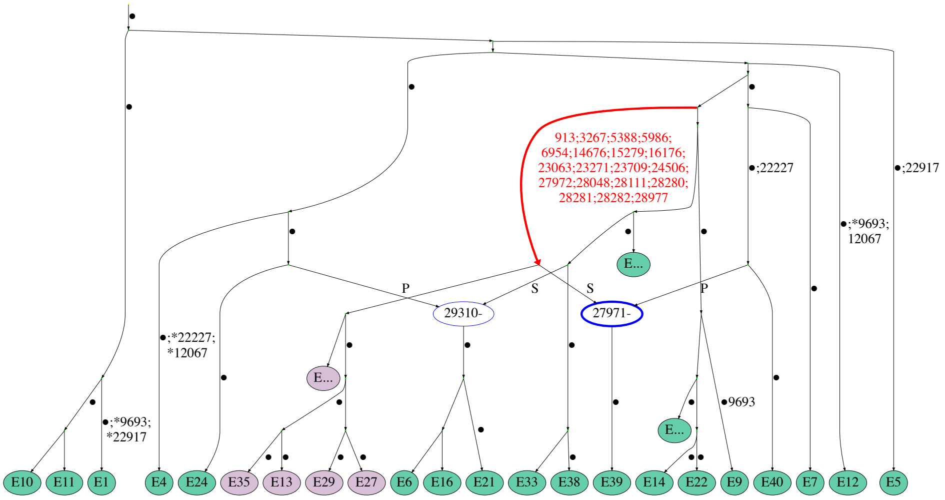

Figure 4.7: Example of an ARG for the England (January) dataset. Recombination nodes are shown in blue, labelled with the recombination breakpoint, with the offspring sequence inheriting part of the genome to the left (right) of the breakpoint from the parent labelled “P” (“S”). Recurrent mutations are prefixed with an asterisk. Edge carrying the characteristic mutations of lineage B.1.1.7 is highlighted in red; nodes corresponding to sequences from lineage B.1.1.7 are coloured purple. For ease of viewing, some parts of the ARG have been collapsed into nodes labelled “E...”. Edges are labelled by positions of mutations (some mutated sites are not explicitly labelled and are denoted by a dot instead).

Considering the results in Table 4.1c, three recurrent mutations can have the same effect as six of the identified recombination events (compare row $(R, RM) = (10, 0)$ with $(R, RM) = (4, 3)$), suggesting that recurrent mutation offers a more parsimonious explanation for at least part of the patterns seen in the data. One of these recurrent mutations consistently occurs at site 22 227; the other two can be placed either at the same site 9 693, or at sites 9 693 and 12 067. The probability of observing five or fewer recurrent mutations is

0.97, which suggests that, with high probability, at least two recombination have occurred in the history of the sample. An example of an ARG with two recombination events is shown in Figure 4.7.

It is striking that eight of the recurrent mutations seen in Figure 4.6 can be placed in the same sequence E39. Indeed, Figure 4.7 shows that the corresponding incompatibilities in the data can be resolved by just one recombination event between sequence E40 and a sequence from lineage B.1.1.7; the corresponding recombination node is shown in bold. The sequence E39 has previously been identified as a possible recombinant by Jackson et al. (2021), demonstrating that the method can clearly highlight mosaic sequences in addition to quantifying the probability that recombination has occurred in the history of the dataset.

4.6.2 Detection of intra-clade recombination

All sequences collected in South Africa in February 2021 were downloaded and processed as described above in Section 4.2.1.4. The resulting sample comprises 38 sequences with 151 variable sites, all from the same lineage B.1.351.

Initial examination of the solutions identified by KwARG show that at least eight recurrent mutations are required to construct a valid ARG for this sample in the absence of recombination. However, it was noted that three of these recurrent mutations occur at the same site 28 254. This may imply that the site is highly mutable, which could be due to repeated sequencing errors, or as a consequence of selection. Note that this demonstrates the usefulness of the presented approach in identifying potentially highly homoplasic sites.

This position was masked from the sample before re-running the analysis. The probability of observing the re-calculated value of $T_{obs} = 5$ or more recurrent mutations is $p = 7 \cdot 10^{-4}$, strongly suggesting the presence of recombination. The probability of observing two or fewer recurrent mutations is 0.97, which indicates that with high probability, at least three recombination events have occurred in the history of the dataset.

4.6.3 Detection of inter-clade recombination

All sequences collected in South Africa in November 2020 were downloaded and processed as described above in Section 4.2.1.3, to create a sample of 50 sequences with 207 variable sites, with 25 belonging to lineage B.1.351 (labelled

SAN1-SAN25), and 25 to other lineages (labelled SAO1-SAO25).

An initial run of KwARG demonstrated that, notably, one recurrent mutation occurs at site 28 254, further suggesting that this site is excessively prone to recurrent mutation. This site was therefore masked before re-running the analysis. An illustration of the sample is provided in Figure 4.8. The sites of nine recurrent mutations identified by KwARG are highlighted with red crosses (choosing a solution with no recombinations, and where the recurrent mutations fall on the terminal branches of the ARG). The probability of observing the required $T_{obs} = 9$ or more recurrent mutations is $p = 7 \cdot 10^{-6}$, strongly suggesting the presence of recombination.

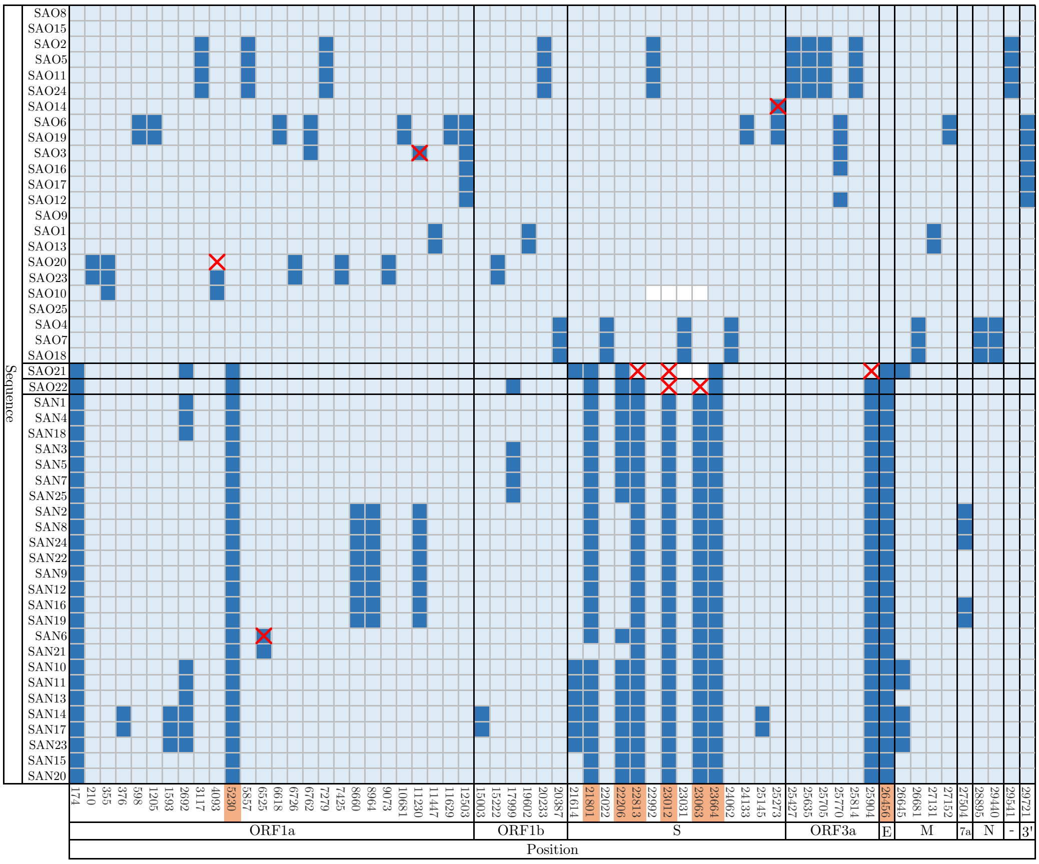

Figure 4.8: Summary of the South Africa (November) dataset. Rows correspond to sequences, labelled on the left. Columns correspond to positions along the genome; uninformative sites (with all 0's or 1's) and those with singleton mutations (with exactly one 1) are not shown. Light blue: ancestral state, dark blue: mutated state, white: missing data. Red crosses highlight sites of recurrent mutations identified by KwARG. Sites bearing the characteristic (non-synonymous) mutations of lineage B.1.351 (Tegally et al., 2020) are highlighted in orange.

The probability of observing three or fewer recurrent mutations is 0.96, which indicates that, with high probability, at least four recombination events have occurred in the history of the dataset. Indeed, Table 4.1 shows that three recurrent mutations can remove the necessity of six recombination events, suggesting that recurrent mutation offers a more parsimonious explanation than recombination for the remaining incompatibilities in the data. Examination of the KwARG solutions shows that these recurrent mutations consistently occur at sites 4 093, 11 230, and 25 273. An ARG with recurrent mutations at these three sites is shown in Figure 4.9; edges carrying the characteristic mutations of lineage B.1.351 are highlighted in red.

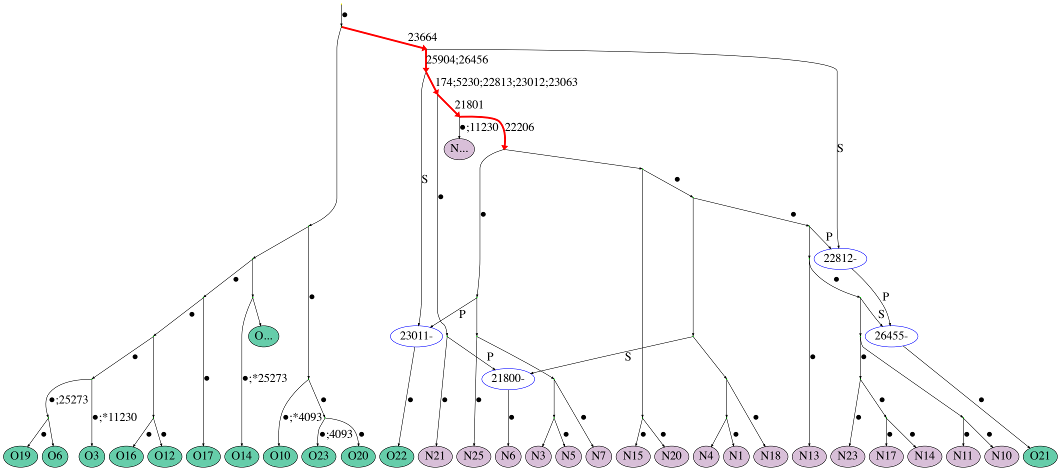

Figure 4.9: Example of an ARG for the South Africa (November) dataset (the "SA" prefix of each sequence reference number is dropped for ease of viewing). Recombination nodes are shown in blue, labelled with the recombination breakpoint, with the offspring sequence inheriting part of the genome to the left (right) of the breakpoint from the parent labelled "P" ("S"). Recurrent mutations are prefixed with an asterisk. For ease of viewing, some parts of the ARG have been collapsed into nodes labelled "O..." and "N..." (containing sequences labelled SAO and SAN, respectively). Edges are labelled by positions of mutations (some mutated sites are not explicitly labelled and are denoted by a dot instead).

The sequences SAO21 and SAO22 carry three and two of the identified nine recurrent mutations, respectively, when recombination is prohibited in reconstructing the genealogy. Both of these sequences carry some of the mutations characteristic of lineage B.1.351; this is demonstrated in Figure 4.10, where the two sequences are compared to two other typical sequences from lineage B.1.351. Examination of the KwARG solutions shows that a recombination in Sequence SAO21 just after site 22 812 has the same effect as the recurrent mutations at sites 22 813 and 23 012, and a recombination in Sequence SAO22 just after site 23 011 has the same effect as the recurrent mutations at sites 23 012 and 23 063. This suggests that the patterns of incompatibilities observed in these two sequences are consistent with recombination; a possible sequence of recombination events generating these sequences can be seen in the ARG in Figure 4.9.

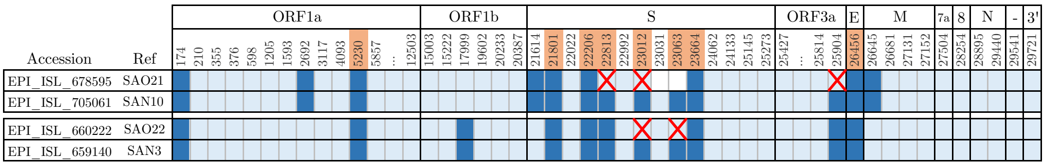

Figure 4.10: Comparison of sequences SAO21, SAO22 and the characteristic mutations for lineage B.1.351. Columns correspond to positions along the genome; uninformative sites (with all 0's or 1's) and those with singleton mutations (with exactly one 1) are not shown. Light blue: ancestral state, dark blue: mutated state, white: missing data. Red crosses highlight sites of recurrent mutations identified by KwARG. Sites bearing the characteristic (non-synonymous) mutations of lineage B.1.351 (Tegally et al., 2020) are highlighted in orange.

4.6.4 Identification of sequencing errors due to cross-contamination

All sequences labelled as GISAID clade GR, collected in England in November 2020, were aligned, masked, and processed as detailed above in Section 4.2.1.5. The quality criteria detailed in Section 4.2.1.2 were *not* applied in this case. The resulting sample comprises 80 sequences with 363 variable sites, 40 of which belong to lineage B.1.1.7 (labelled EN1-EN40) and 40 to other lineages (labelled EO1-EO40).

The results showed that in the absence of recombination, at least 15 recurrent mutations were required to explain the incompatibilities observed in this sample. However, it was identified that six of these recurrent mutations could be placed in the same sequence EO40, as illustrated in Figure 4.11. The sequence EO40 appeared to carry some of the mutations carried by sequence EO32, and some of the mutations characteristic of lineage B.1.1.7, strongly suggesting that this sequence was a recombinant.

These findings prompted further investigation by the submitters of this sequence, which revealed the signal to be the result of significant contamination

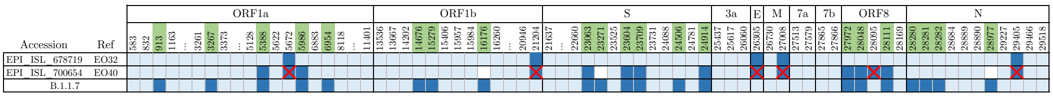

Figure 4.11: Comparison of sequences EO32, EO40 and the characteristic mutations of lineage B.1.1.7. Columns correspond to positions along the genome; uninformative sites (with all 0's or 1's) and those with singleton mutations (with exactly one 1) are not shown. Light blue: ancestral state, dark blue: mutated state, white: missing data. Red crosses highlight locations of the recurrent mutations identified by KwARG. Sites bearing the characteristic mutations of lineage B.1.1.7 (Rambaut et al., 2020) are highlighted in green.

of the genetic sample causing multiple errors in the consensus sequence, rather than a result of intra-host recombination. The sequence has subsequently been removed from GISAID.

4.6.5 Recombination detection for MERS-CoV data

MERS-CoV sequences collected in Saudi Arabia in January–March 2015 were downloaded from the NCBI virus database, and aligned, masked, and processed as described in Section 4.2.2. The resulting sample consists of 19 sequences with 197 variable sites.

The dataset is illustrated in Figure 4.12. The locations of recurrent mutations identified by KwARG are shows as red and yellow crosses, corresponding to recurrent mutations occurring on the terminal and internal branches of the ARG, respectively. In the absence of recombination, at least $T_{obs} = 16$ recurrent mutations are required, which has probability $p < 1 \cdot 10^{-6}$, strongly suggesting the presence of recombination. The probability of observing three or fewer recurrent mutations is 0.99, suggesting that at least five recombinations have occurred in the history of the sample. An ARG with five recombination nodes, showing a possible history of the dataset, is shown in Figure 4.13.

A group of four identical sequences (M16–M19, shown in purple in Figure 4.13) appear to carry a characteristic set of shared mutations that strongly differentiates them from the other sequences in the sample. Five of the identified recurrent mutations affect this group, occurring in a relatively short stretch of the genome, suggesting that these patterns are indicative of recombination with other sequences in the sample carrying these mutations.

Five of the other identified recurrent mutations can be placed in one sequence (M11), which appears to carry a mixture of mutations from the group

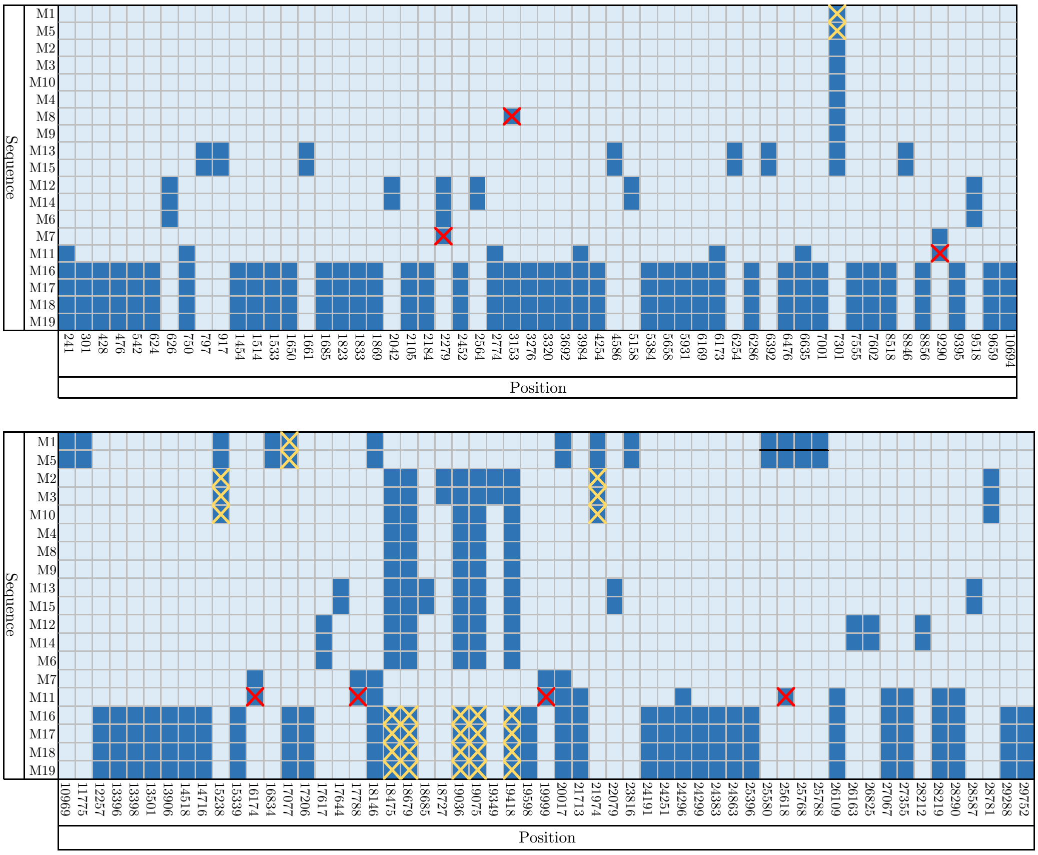

Figure 4.12: Summary of the MERS-CoV dataset. Rows correspond to sequences, labelled on the left. Columns correspond to positions along the genome; uninformative sites (with all 0's or 1's) and those with singleton mutations (with exactly one 0 or 1) are not shown. Light blue and dark blue denote differing allele types. Red crosses highlight sites of recurrent mutations identified by KwARG located on the terminal branches of the ARG (affecting only one sequence). Yellow crosses highlight recurrent mutations on internal branches (hence affecting multiple sequences).

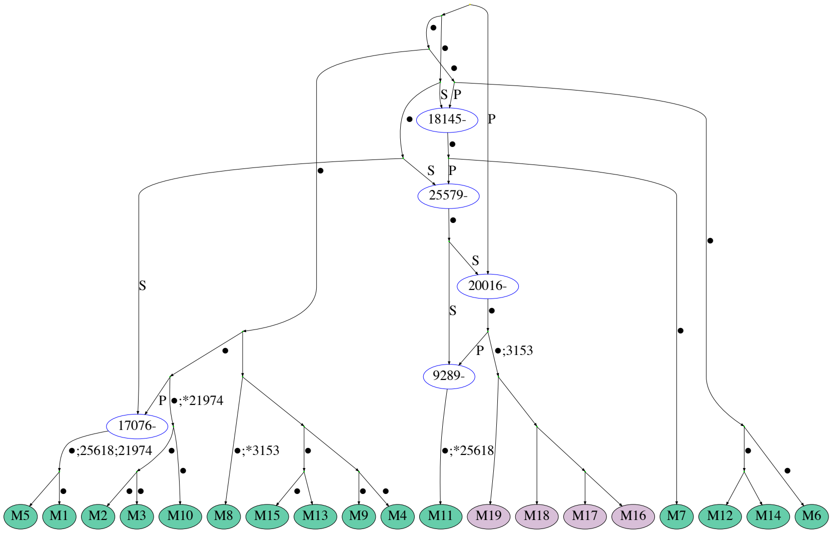

Figure 4.13: Example of an ARG for the MERS-CoV dataset. Recombination nodes are shown in blue, labelled with the recombination breakpoint, with the offspring sequence inheriting part of the genome to the left (right) of the breakpoint from the parent labelled "P" ("S"). Recurrent mutations are prefixed with an asterisk. Edges are labelled by positions of mutations (some mutated sites are not explicitly labelled and are denoted by a dot instead).

identified above and other sequences in the sample, which is consistent with recombination. This sequence does not match any others in the dataset, so it is possible that this is the result of sequencing errors or sample contamination. If this sequence is removed from the sample, at least $T_{obs} = 9$ recurrent mutations are still required to explain the observed incompatibilities, which has probability $p < 1 \cdot 10^{-6}$, still strongly suggesting that recombination is present. This agrees with previous reports of within-host recombination for MERS-CoV (Zhang et al., 2016; Dudas and Rambaut, 2016; Sabir et al., 2016).

4.7 Discussion

The method presented here offers a clear and principled framework for recombination detection, which can be interpreted as a hypothesis testing approach. Very conservative assumptions are made throughout, demonstrating on both real and simulated data that the method achieves a very low rate of false positive results, while offering powerful detection of recombination at even relatively low values of recombination rate. Nonparametric techniques are used at each stage, to avoid making assumptions on the process generating the data, and thus circumvent issues with model misspecification. The method allows us to gain clear insights into the evolutionary events that may have generated the given sequences, offering easily interpretable results. The method detects sequences carrying patterns consistent with recombination, demonstrating its effectiveness as a tool for flagging sequences with distinctive patterns of incompatibilities for further detailed investigation.

The results clearly indicate the presence of recombination in the history of the analysed SARS-CoV-2 sequencing data; based on the analysis of statistical power of the method, this suggests a likely recombination rate greater than around $4 \cdot 10^{-5}$ per site per year. One of the main limitations of the method is that KwARG does not scale well to large datasets. However, while studies relying on clade assignment and statistics such as linkage disequilibrium have identified that recombination occurs at very low levels (VanInsberghe et al., 2021; Varabyou et al., 2021) or is unlikely to be occurring at a detectable level (De Maio et al., 2020; van Dorp et al., 2020b; Nie et al., 2020; Tang et al., 2020; Wang et al., 2020; Richard et al., 2020) even when analysing vast quantities of sequencing data, the method is powerful enough to detect the presence of recombination using even relatively small samples. Moreover, the testing framework could potentially be used in combination with other methods for reconstructing ARGs, including ones not relying on the parsimony assumption, with appropriate modifications to control the false positive rate and ensure validity of the results.

Recombination can occur when the same host is co-infected by two different strains, which has been noted to occur in COVID-19 patients (Samoilov et al., 2020), and could become more likely with the emergence of more transmissible variants. Note that the potential mosaic sequences identified in the South Africa sample from November are represented only once in the data. This could be due to a lack of onward transmission, as recombinants are likely to reach a detectable level at a relatively late stage in the infection cycle. It could also indicate that the sequences arose due to either contamination of the sample during processing, or the misassembly of two distinct (non-recombinant) strains present in the same sample, as was identified to be the case for one sequence in the England sample from November.

Note that while any sites known to be highly homoplasic were masked, it cannot be ruled out that some of the identified recurrent mutations did arise multiple times as a consequence of selection or as a result of repeated sequencing errors. However, as demonstrated, the solutions presented by KwARG can be examined for the presence of highly mutable sites, and it was identified using both samples from South Africa that this appears to be the case for site 28 254 (located proximal to the stop codon of ORF8).

The findings suggest that care should be taken when performing and interpreting the results of analysis based on the construction of phylogenetic trees for SARS-CoV-2 data. The presence of recombination, as well as other factors complicating the structure of the transmission network of the virus, strongly suggests that tree-based models are not appropriate for modelling SARS-CoV-2 genealogies, and inference of evolutionary rates based on such methods may suffer from errors due to model misspecification that are difficult to quantify.

Due to the high level of homogeneity between sequences, the effects of recombination will be either undetectable or indistinguishable from recurrent mutation in the majority of cases. However, as genetic diversity builds up over longer timescales, the effects of recombination may become more pronounced. Particularly in light of the recent emergence of new variants, the rapid evolution of the virus through recombination between strains with different pathogenic properties is a crucial risk factor to consider.

Chapter 5

Discussion

Models based on the coalescent commonly assume that the population size is constant or deterministically changing through time, which is an unrealistic assumption for many viral populations. Birth-death models offer a useful alternative, naturally capturing the stochastic variation and exponential growth of the population size. Results presented in Chapter 2 have shed light on interesting theoretical properties of birth-death sample genealogies, in the limit of the underlying population size growing to infinity—a realistic setting in the context of viral sequencing data.

Numerous extensions to the birth-death population model have been considered previously, incorporating different sampling schemes, population structure, time-dependent branching rates, and other factors. However, these models ignore the presence of recombination, which in practice can significantly distort the results of inference. Explicitly incorporating the process of recombination into birth-death models remains a significant open problem. A key idea of Chapter 2 was to use time rescaling to make deriving the properties of the relevant stochastic process much more tractable. It may be possible to apply similar insights to uncover the properties of genealogies in a birth-death-recombination framework, which would present both interesting theoretical insights into the properties of resulting ARGs, as well as open up the way for using these models for improved inference of recombination from real data.

In general, the detection of recombination and identification of recombinants in samples of sequencing data is an extremely important but very difficult problem. The presence of recombination has significant consequences for understanding the future evolutionary trajectory of a virus, as it can quickly create new hybrid genomes with unique pathogenic properties. For viral data collected at the level of one sequence per infected host, the genealogy is related to the transmission network of the pathogen, with aspects such as geographical structure and human interventions making it difficult to choose an appropriate model for genealogies, motivating the development of model-free genealogical methods for detecting recombination.

The work presented in Chapter 3 introduced a fundamental computational tool (KwARG) for reconstructing parsimonious ARG topologies, i.e. those that are minimal or near-minimal in the number of posited recombination and mutation events. The method does not require assuming a particular model, and can give a useful lower bound on the number of recombinations that must have occurred. This work incorporated several theoretical advances into a readily usable program that can be used by virologists in practice for the analysis of sequencing data. The usefulness of genealogical reconstruction methods in making the most of sequencing data to gain scientific insights into the evolution of biological organisms is clear. For example, ARGs generated using SHRUB have been used in identifying a gene affecting the body size of dogs (Sutter et al., 2007), and Relate has been used to analyse selection within human populations (Speidel et al., 2019). KwARG incorporates several aspects which make it particularly appropriate for the analysis of viral data.

While several methods for inferring ARGs and ARG topologies from data have been developed in recent years, a central issue with validating and comparing performance is the absence of metrics to compare the inferred ARGs. Several metrics exist for comparing trees, so typically comparisons between ARGs are performed by first breaking them up into local trees and averaging, as was done for the analysis in Chapter 3. However, this loses valuable information regarding where and how the recombinations occur in the ARGs. Moreover, while tree metrics allow for the definition of ‘tree space’, there is no equivalent formalised notion for ARGs. Developing this would allow several crucial questions to be addressed: for instance, how different algorithms (both model-based and heuristic) explore ARG space, how different are the inferred ARGs from the true ARGs that have generated the data, and how ‘far’ parsimonious ARGs are from the true ARGs. Moreover, these insights should help in designing better search algorithms, and in understanding their theoretical properties.

The extent of ongoing recombination of SARS-CoV-2 within infected hosts has been very difficult to quantify, with the problem further complicated by its relatively slow accumulation of genetic diversity. Through combining KwARG with a principled statistical framework for recombination detection, the results presented in Chapter 4 have demonstrated that ongoing recombination in SARS-CoV-2 is present at higher levels than previously suggested, highlighting serious problems with the widely used tree-based phylogenetic analyses that ignore the presence of recombination. This has significant scientific implications, of interest to a broad community of researchers within statistics, epidemiology and microbiology. There is a clear need for continuous monitoring of the sequenced genomes for new variants, to enable the early detection of novel recombinant genotypes, and for further work on the quantification of recombination rates and identification of recombination hotspots along the genome.

Bibliography

Abramowitz, M. and Stegun, I. A. (1964). *Handbook of mathematical functions: with formulas, graphs, and mathematical tables.* National Bureau of Standards, Washington, D.C.

Aldous, D. (1996). Probability distributions on cladograms. In *Random Discrete Structures*, pp. 1–18. Springer.

Aldous, D. and Popovic, L. (2005). A critical branching process model for biodiversity. *Advances in Applied Probability*, **37**(4), 1094–1115.

Arnold, B. C., Balakrishnan, N. and Nagaraja, H. N. (1992). *A first course in order statistics*, volume 54. SIAM, Philadelphia.

Athreya, K. B. and Ney, P. E. (1972). *Branching Processes.* Springer-Verlag, Berlin.

Bailey, N. T. (1964). *The elements of stochastic processes with applications to the natural sciences.* Wiley, New York.

Boni, M. F., Posada, D. and Feldman, M. W. (2007). An exact nonparametric method for inferring mosaic structure in sequence triplets. *Genetics*, **176**(2), 1035–1047.

Boskova, V., Bonhoeffer, S. and Stadler, T. (2014). Inference of epidemiological dynamics based on simulated phylogenies using birth-death and coalescent models. *PLoS Computational Biology*, **10**(11), e1003913.

Burden, C. J. and Soewongsono, A. C. (2019). Coalescence in the diffusion limit of a Bienaymé–Galton–Watson branching process. *Theoretical Population Biology*, **130**, 50–59.

Daubechies, I. (1988). Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, **41**(7), 909–996.

De Maio, N., Walker, C., Borges, R., Weilguny, L., Slodkowicz, G. and Goldman, N. (2020). Issues with SARS-CoV-2 sequencing data. `https://virological.org/t/issues-with-sars-cov-2-sequencing-data/473`.

Dialdestoro, K., Sibbesen, J. A., Maretty, L., Raghwani, J., Gall, A., Kellam, P., Pybus, O. G., Hein, J. and Jenkins, P. A. (2016). Coalescent inference using serially sampled, high-throughput sequencing data from intrahost HIV infection. *Genetics*, **202**(4), 1449–1472.

Dinh, K. N., Jaksik, R., Kimmel, M., Lambert, A. and Tavaré, S. (2020). Statistical inference for the evolutionary history of cancer genomes. *Statistical Science*, **35**(1), 129–144.

Dudas, G. and Rambaut, A. (2016). MERS-CoV recombination: Implications about the reservoir and potential for adaptation. *Virus Evolution*, **2**(1).

Durrett, R. (2013). Population genetics of neutral mutations in exponentially growing cancer cell populations. *The Annals of Applied Probability*, **23**(1), 230.

Elbe, S. and Buckland-Merrett, G. (2017). Data, disease and diplomacy: GISAID's innovative contribution to global health. *Global Challenges*, **1**(1), 33–46.

Ellson, J., Gansner, E. R., Koutsofios, E., North, S. C. and Woodhull, G. (2004). Graphviz and Dynagraph: static and dynamic graph drawing tools. In *Graph drawing software*, pp. 127–148. Springer, Heidelberg.

Erdélyi, A., Magnus, W., Oberhettinger, F., Tricomi, F. G. and Bateman, H. (1953). *Higher transcendental functions*, volume 1. McGraw-Hill, New York.

Fleischmann, K. and Siegmund-Schultze, R. (1977). The structure of reduced critical Galton-Watson processes. *Mathematische Nachrichten*, **79**(1), 233–241.

Flint, S. J., Racaniello, V. R., Enquist, L. W. and Skalka, A. M. (2009). *Principles of Virology, Volume I.* ASM Press, Washington, D.C.

Foulds, L. R. and Graham, R. L. (1982). The steiner problem in phylogeny is NP-complete. *Advances in Applied mathematics*, **3**(1), 43–49.

Fu, Y.-X. (1995). Statistical properties of segregating sites. *Theoretical Population Biology*, **48**(2), 172–197.

George, E. O. and Mudholkar, G. S. (1981). Some relationships between the logistic and the exponential distributions. In C. Taillie, G. Patil and B. Baldessari, editors, *Statistical Distributions in Scientific Work*, pp. 401–409. Springer, Dordrecht.

Gernhard, T. (2008a). The conditioned reconstructed process. *Journal of Theoretical Biology*, **253**(4), 769–778.

Gernhard, T. (2008b). New analytic results for speciation times in neutral models. *Bulletin of Mathematical Biology*, **70**(4), 1082–1097.

Gribble, J., Stevens, L. J., Agostini, M. L., Anderson-Daniels, J., Chappell, J. D., Lu, X., Pruijssers, A. J., Routh, A. L. and Denison, M. R. (2021). The coronavirus proofreading exoribonuclease mediates extensive viral recombination. *PLOS Pathogens*, **17**(1), e1009226.

Griffiths, R. C. and Marjoram, P. (1997). An ancestral recombination graph. In P. Donnelly and S. Tavaré, editors, *Progress in population genetics and human evolution*, pp. 257–270. Springer, New York.

Griffiths, R. C. and Tavaré, S. (1994). Sampling theory for neutral alleles in a varying environment. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, **344**(1310), 403–410.

Grosjean, N. and Huillet, T. (2018). On the genealogy and coalescence times of Bienaymé–Galton–Watson branching processes. *Stochastic Models*, **34**(1), 1–24.

Gusfield, D. (2014). *ReCombinatorics: the algorithmics of ancestral recombination graphs and explicit phylogenetic networks.* MIT press, Cambridge, Massachusetts.

Gusfield, D., Hickerson, D. and Eddhu, S. (2007). An efficiently computed lower bound on the number of recombinations in phylogenetic networks: Theory and empirical study. *Discrete Applied Mathematics*, **155**(6-7), 806–830.

Hadfield, J., Megill, C., Bell, S. M., Huddleston, J., Potter, B., Callender, C., Sagulenko, P., Bedford, T. and Neher, R. A. (2018). Nextstrain: Real-time tracking of pathogen evolution. *Bioinformatics*, **34**(23), 4121–4123.

Harris, S. C., Johnston, S. G. and Roberts, M. I. (2020). The coalescent structure of continuous-time galton-watson trees. *Annals of Applied Probability*, **30**(3), 1368–1414.

Hart, P. E., Nilsson, N. J. and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, **4**(2), 100–107.

Hartmann, K., Wong, D. and Stadler, T. (2010). Sampling trees from evolutionary models. *Systematic Biology*, **59**(4), 465–476.

Hatcher, E. L., Zhdanov, S. A., Bao, Y., Blinkova, O., Nawrocki, E. P., Ostapchuck, Y., Schäffer, A. A. and Brister, J. R. (2017). Virus variation resource–improved response to emergent viral outbreaks. *Nucleic Acids Research*, **45**(D1), D482–D490.

Hein, J. (1990). Reconstructing evolution of sequences subject to recombination using parsimony. *Mathematical Biosciences*, **98**(2), 185–200.

Hein, J. (1993). A heuristic method to reconstruct the history of sequences subject to recombination. *Journal of Molecular Evolution*, **36**(4), 396–405.

Hein, J., Schierup, M. and Wiuf, C. (2004). *Gene genealogies, variation and evolution: a primer in coalescent theory.* Oxford University Press, New York.

Hubisz, M. and Siepel, A. (2020). Inference of ancestral recombination graphs using argweaver. In *Statistical Population Genomics*, pp. 231–266. Humana, New York.

Hudson, R. R. (2015). A new proof of the expected frequency spectrum under the standard neutral model. *PLOS ONE*, **10**(7), e0118087.

Hudson, R. R. and Kaplan, N. L. (1985). Statistical properties of the number of recombination events in the history of a sample of DNA sequences. *Genetics*, **111**(1), 147–164.

Ignatieva, A., Hein, J. and Jenkins, P. A. (2020). A characterisation of the reconstructed birth–death process through time rescaling. *Theoretical Population Biology*, **134**, 61–76.

Ignatieva, A., Hein, J. and Jenkins, P. A. (2021a). Investigation of ongoing recombination through genealogical reconstruction for SARS-CoV-2. *bioRxiv*. doi:10.1101/2021.01.21.427579.

Ignatieva, A., Lyngsø, R. B., Jenkins, P. A. and Hein, J. (2021b). KwARG: Parsimonious reconstruction of ancestral recombination graphs with recurrent mutation. *Bioinformatics*. doi:10.1093/bioinformatics/btab351.

Jackson, B., Boni, M. F., Bull, M. J., Colleran, A., Colquhoun, R. M., Darby, A. C., Haldenby, S., Hill, V., Lucaci, A., McCrone, J. T., Nicholls, S. M., ine OToole, Pacchiarini, N., Poplawski, R., Scher, E., Todd, F., Webster, H. J., Whitehead, M., Wierzbicki, C., Loman, N. J., Connor, T. R., Robertson, D. L., Pybus, O. G. and Rambaut, A. (2021). Generation and transmission of interlineage recombinants in the SARS-CoV-2 pandemic. *Cell*. doi:10.1016/j.cell.2021.08.014.

Jenkins, P. A. and Griffiths, R. C. (2011). Inference from samples of DNA sequences using a two-locus model. *Journal of Computational Biology*, **18**(1), 109–127.

Johnstone, I. M. and Silverman, B. W. (2005a). EbayesThresh: R and S-Plus programs for empirical Bayes thresholding. *Journal of Statistical Software*, **12**, 1–38.

Johnstone, I. M. and Silverman, B. W. (2005b). Empirical Bayes selection of wavelet thresholds. *Annals of Statistics*, pp. 1700–1752.

Kaj, I. and Krone, S. M. (2003). The coalescent process in a population with stochastically varying size. *Journal of Applied Probability*, **40**(1), 33–48.

Katoh, K. and Standley, D. M. (2013). MAFFT multiple sequence alignment software version 7: Improvements in performance and usability. *Molecular Biology and Evolution*, **30**(4), 772–780.

Kelleher, J., Etheridge, A. M. and McVean, G. (2016). Efficient coalescent simulation and genealogical analysis for large sample sizes. *PLoS Computational Biology*, **12**(5), e1004842.

Kelleher, J., Wong, Y., Wohns, A. W., Fadil, C., Albers, P. K. and McVean, G. (2019). Inferring whole-genome histories in large population datasets. *Nature Genetics*, **51**(9), 1330–1338.

Kendall, D. G. (1948). On some modes of population growth leading to RA Fisher's logarithmic series distribution. *Biometrika*, **35**(1/2), 6–15.

Kendall, M. and Colijn, C. (2016). Mapping phylogenetic trees to reveal distinct patterns of evolution. *Molecular Biology and Evolution*, **33**(10), 2735–2743.

Kingman, J. F. C. (1982a). The coalescent. *Stochastic Processes and Their Applications*, **13**(3), 235–248.

Kingman, J. F. C. (1982b). On the genealogy of large populations. *Journal of Applied Probability*, **19**(A), 27–43.

Kosakovsky Pond, S. L., Posada, D., Gravenor, M. B., Woelk, C. H. and Frost, S. D. (2006). Automated phylogenetic detection of recombination using a genetic algorithm. *Molecular Biology and Evolution*, **23**(10), 1891–1901.

Koyama, T., Platt, D. and Parida, L. (2020). Variant analysis of SARS-CoV-2 genomes. *Bulletin of the World Health Organization*, **98**(7), 495.

Kreitman, M. (1983). Nucleotide polymorphism at the alcohol dehydrogenase locus of Drosophila melanogaster. *Nature*, **304**(5925), 412–417.

Lambert, A. (2008). The allelic partition for coalescent point processes. *Markov Process and Related Fields*, **15**(3), 359–386.

Lambert, A. (2018). The coalescent of a sample from a binary branching process. *Theoretical Population Biology*, **122**, 30–35.

Lambert, A. and Stadler, T. (2013). Birth-death models and coalescent point processes: The shape and probability of reconstructed phylogenies. *Theoretical Population Biology*, **90**, 113–128.

Li, N. and Stephens, M. (2003). Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data. *Genetics*, **165**(4), 2213–2233.

Li, Q., Guan, X., Wu, P., Wang, X., Zhou, L., Tong, Y., Ren, R., Leung, K. S., Lau, E. H., Wong, J. Y. et al. (2020). Early transmission dynamics in Wuhan, China, of novel coronavirus-infected pneumonia. *New England Journal of Medicine*.

Lyngsø, R. B., Song, Y. S. and Hein, J. (2005). Minimum recombination histories by branch and bound. In *International Workshop on Algorithms in Bioinformatics*, pp. 239–250. Springer.

Mahmuod, M. and Ragab, A. (1973). On order statistics in samples drawn from the logistic distribution. *Statistics: A Journal of Theoretical and Applied Statistics*, **4**(1), 81–88.

Martin, D. and Rybicki, E. (2000). RDP: Detection of recombination amongst aligned sequences. *Bioinformatics*, **16**(6), 562–563.

Maynard Smith, J. and Smith, N. H. (1998). Detecting recombination from gene trees. *Molecular Biology and Evolution*, **15**(5), 590–599.

McVean, G., Awadalla, P. and Fearnhead, P. (2002). A coalescent-based method for detecting and estimating recombination from gene sequences. *Genetics*, **160**(3), 1231–1241.

McVean, G. A. and Cardin, N. J. (2005). Approximating the coalescent with recombination. *Philosophical Transactions of the Royal Society B: Biological Sciences*, **360**(1459), 1387–1393.

Meyer, P.-A. (1971). Démonstration simplifiée d'un théorème de Knight. *Seminaire de Probabilites V Universite de Strasbourg, Lecture Notes in Mathematics*, **5**, 191–195.

Minichiello, M. J. and Durbin, R. (2006). Mapping trait loci by use of inferred ancestral recombination graphs. *The American Journal of Human Genetics*, **79**(5), 910–922.

Mirzaei, S. and Wu, Y. (2017). RENT+: an improved method for inferring local genealogical trees from haplotypes with recombination. *Bioinformatics*, **33**(7), 1021–1030.

Mooers, A., Gascuel, O., Stadler, T., Li, H. and Steel, M. (2012). Branch lengths on birth–death trees and the expected loss of phylogenetic diversity. *Systematic Biology*, **61**(2), 195–203.

Myers, S. (2003). *The detection of recombination events using DNA sequence data*. Ph.D. thesis, University of Oxford, Department of Statistics.

Myers, S. R. and Griffiths, R. C. (2003). Bounds on the minimum number of recombination events in a sample history. *Genetics*, **163**(1), 375–394.

Nason, G. (2008). *Wavelet methods in statistics with R*. Springer Science & Business Media, New York.

Nason, G. et al. (2010). wavethresh: Wavelets statistics and transforms, v.4.6.8. `https://CRAN.R-project.org/package=wavethresh`.

Nee, S., May, R. M. and Harvey, P. H. (1994). The reconstructed evolutionary process. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, **344**(1309), 305–311.

Nie, Q., Li, X., Chen, W., Liu, D., Chen, Y., Li, H., Li, D., Tian, M., Tan, W. and Zai, J. (2020). Phylogenetic and phylodynamic analyses of SARS-CoV-2. *Virus Research*, **287**, 198098.

O'Connell, N. (1995). The genealogy of branching processes and the age of our most recent common ancestor. *Advances in Applied Probability*, **27**(2), 418–442.

Ohtsuki, H. and Innan, H. (2017). Forward and backward evolutionary processes and allele frequency spectrum in a cancer cell population. *Theoretical Population Biology*, **117**, 43–50.

Page, A. J., Taylor, B., Delaney, A. J., Soares, J., Seemann, T., Keane, J. A. and Harris, S. R. (2016). SNP-sites: Rapid efficient extraction of SNPs from multi-FASTA alignments. *Microbial Genomics*, **2**(4).

Pan, Y., Zhang, D., Yang, P., Poon, L. L. and Wang, Q. (2020). Viral load of SARS-CoV-2 in clinical samples. *The Lancet Infectious Diseases*, **20**(4), 411–412.

Papangelou, F. (1972). Integrability of expected increments of point processes and a related random change of scale. *Transactions of the American Mathematical Society*, **165**, 483–506.

Paradis, E. and Schliep, K. (2018). ape 5.0: an environment for modern phylogenetics and evolutionary analyses in R. *Bioinformatics*, **35**, 526–528.

Parida, L., Melé, M., Calafell, F., Bertranpetit, J. and Genographic Consortium (2008). Estimating the ancestral recombinations graph (ARG) as compatible networks of SNP patterns. *Journal of Computational Biology*, **15**(9), 1133–1153.

Parsons, T. L., Quince, C. and Plotkin, J. B. (2010). Some consequences of demographic stochasticity in population genetics. *Genetics*, **185**, 1345–1354.

Polanski, A., Bobrowski, A. and Kimmel, M. (2003). A note on distributions of times to coalescence, under time-dependent population size. *Theoretical Population Biology*, **63**(1), 33–40.

Posada, D. and Crandall, K. A. (2001). Evaluation of methods for detecting recombination from DNA sequences: Computer simulations. *PNAS*, **98**(24), 13757–13762.

Posada, D. and Crandall, K. A. (2002). The effect of recombination on the accuracy of phylogeny estimation. *Journal of Molecular Evolution*, **54**(3), 396–402.

Rambaut, A. and Grass, N. C. (1997). Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Bioinformatics*, **13**(3), 235–238.

Rambaut, A., Loman, N., Pybus, O., Barclay, W., Barrett, J., Carabelli, A., Connor, T., Peacock, T., Robertson, D. and Volz, E. (2020). Preliminary genomic characterisation of an emergent SARS-CoV-2 lineage in the UK defined by a novel set of spike mutations. `https://virological.org/t/preliminary-genomic-characterisation-of-an-emergent-sars-cov-2-lineage-in-the-uk-defined-by-a-novel-set-of-spike-mutations/563`.

Rasmussen, M. D., Hubisz, M. J., Gronau, I. and Siepel, A. (2014). Genome-wide inference of ancestral recombination graphs. *PLoS Genetics*, **10**(5), e1004342.

Richard, D., Owen, C. J., van Dorp, L. and Balloux, F. (2020). No detectable signal for ongoing genetic recombination in SARS-CoV-2. *bioRxiv*. doi: 10.1101/2020.12.15.422866.

Robinson, D. F. and Foulds, L. R. (1981). Comparison of phylogenetic trees. *Mathematical Biosciences*, **53**(1-2), 131–147.

Sabir, J. S., Lam, T. T.-Y., Ahmed, M. M., Li, L., Shen, Y., Abo-Aba, S. E., Qureshi, M. I., Abu-Zeid, M., Zhang, Y., Khiyami, M. A. et al. (2016). Co-circulation of three camel coronavirus species and recombination of MERS-CoVs in Saudi Arabia. *Science*, **351**(6268), 81–84.

Samoilov, A., Kaptelova, V., Bukharina, A., Shipulina, O., Korneenko, E., Lukyanov, A., Grishaeva, A., Ploskireva, A., Speranskaya, A. and Akimkin, V. (2020). Change of dominant strain during dual SARS-CoV-2 infection. *medRxiv*. doi:10.1101/2020.11.29.20238402.

Schierup, M. H. and Hein, J. (2000). Consequences of recombination on traditional phylogenetic analysis. *Genetics*, **156**(2), 879–891.

Semple, C. and Steel, M. (2003). *Phylogenetics*. Oxford University Press, Oxford.

Shen, W., Le, S., Li, Y. and Hu, F. (2016). SeqKit: A cross-platform and ultrafast toolkit for FASTA/Q file manipulation. *PLOS ONE*, **11**(10), e0163962.

Simmonds, P. (2020). Rampant C→U hypermutation in the genomes of SARS-CoV-2 and other coronaviruses: Causes and consequences for their short- and long-term evolutionary trajectories. *mSphere*, **5**(3).

Simon-Loriere, E. and Holmes, E. C. (2011). Why do RNA viruses recombine? *Nature Reviews Microbiology*, **9**(8), 617–626.

Slatkin, M. and Hudson, R. R. (1991). Pairwise comparisons of mitochondrial DNA sequences in stable and exponentially growing populations. *Genetics*, **129**(2), 555–562.

Song, Y. S., Ding, Z., Gusfield, D., Langley, C. H. and Wu, Y. (2006). Algorithms to distinguish the role of gene-conversion from single-crossover recombination in the derivation of SNP sequences in populations. In *Annual International Conference on Research in Computational Molecular Biology*, pp. 231–245. Springer.

Song, Y. S. and Hein, J. (2003). Parsimonious reconstruction of sequence evolution and haplotype blocks. In *International Workshop on Algorithms in Bioinformatics*, pp. 287–302. Springer.

Song, Y. S., Wu, Y. and Gusfield, D. (2005). Efficient computation of close lower and upper bounds on the minimum number of recombinations in biological sequence evolution. *Bioinformatics*, **21**(suppl_1), i413–i422.

Speidel, L., Forest, M., Shi, S. and Myers, S. R. (2019). A method for genome-wide genealogy estimation for thousands of samples. *Nature Genetics*, **51**(9), 1321–1329.

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.

Stadler, T. (2011). Simulating trees with a fixed number of extant species. *Systematic Biology*, **60**(5), 676–684.

Stadler, T. and Steel, M. (2012). Distribution of branch lengths and phylogenetic diversity under homogeneous speciation models. *Journal of Theoretical Biology*, **297**, 33–40.

Stadler, T. and Steel, M. (2019). Swapping birth and death: Symmetries and transformations in phylodynamic models. *Systematic Biology*, **68**(5), 852–858.

Stadler, T., Vaughan, T. G., Gavryushkin, A., Guindon, S., Kühnert, D., Leventhal, G. E. and Drummond, A. J. (2015). How well can the exponential-growth coalescent approximate constant-rate birth–death population dynamics? *Proceedings of the Royal Society B: Biological Sciences*, **282**(1806), 20150420.

Stephens, J. C. and Nei, M. (1985). Phylogenetic analysis of polymorphic DNA sequences at the ADH locus in Drosophila melanogaster and its sibling species. *Journal of Molecular Evolution*, **22**(4), 289–300.

Su, S., Wong, G., Shi, W., Liu, J., Lai, A. C., Zhou, J., Liu, W., Bi, Y. and Gao, G. F. (2016). Epidemiology, genetic recombination, and pathogenesis of coronaviruses. *Trends in Microbiology*, **24**(6), 490–502.

Sutter, N. B., Bustamante, C. D., Chase, K., Gray, M. M., Zhao, K., Zhu, L., Padhukasahasram, B., Karlins, E., Davis, S., Jones, P. G. et al. (2007). A single igf1 allele is a major determinant of small size in dogs. *Science*, **316**(5821), 112–115.

Swofford, D. L. (2003). PAUP*: Phylogenetic analysis using parsimony (and other methods). Version 4. Sinauer Associates, Sunderland, Massachusetts.

Tang, X., Wu, C., Li, X., Song, Y., Yao, X., Wu, X., Duan, Y., Zhang, H., Wang, Y., Qian, Z. et al. (2020). On the origin and continuing evolution of SARS-CoV-2. *National Science Review*, **7**(6), 1012–1023.

Tegally, H., Wilkinson, E., Giovanetti, M., Iranzadeh, A., Fonseca, V., Giandhari, J., Doolabh, D., Pillay, S., San, E. J., Msomi, N., Mlisana, K., von Gottberg, A., Walaza, S., Allam, M., Ismail, A., Mohale, T., Glass, A. J., Engelbrecht, S., Van Zyl, G., Preiser, W., Petruccione, F., Sigal, A., Hardie, D., Marais, G., Hsiao, M., Korsman, S., Davies, M.-A., Tyers, L., Mudau, I., York, D., Maslo, C., Goedhals, D., Abrahams, S., Laguda-Akingba, O., Alisoltani-Dehkordi, A., Godzik, A., Wibmer, C. K., Sewell, B. T., Lourenço, J., Alcantara, L. C. J., Pond, S. L. K., Weaver, S., Martin, D., Lessells, R. J., Bhiman, J. N., Williamson, C. and de Oliveira, T. (2020). Emergence and rapid spread of a new severe acute respiratory syndrome-related coronavirus 2 (SARS-CoV-2) lineage with multiple spike mutations in South Africa. *medRxiv*. doi:10.1101/2020.12.21.20248640.

Thao, N. T. P. and Vinh, L. S. (2019). A hybrid approach to optimize the number of recombinations in ancestral recombination graphs. In *Proceedings of the 2019 9th International Conference on Bioscience, Biochemistry and Bioinformatics*, pp. 36–42.

Thompson, E. A. (1975). *Human evolutionary trees*. Cambridge University Press, Cambridge.

van Dorp, L., Acman, M., Richard, D., Shaw, L. P., Ford, C. E., Ormond, L., Owen, C. J., Pang, J., Tan, C. C., Boshier, F. A. et al. (2020a). Emergence of genomic diversity and recurrent mutations in SARS-CoV-2. *Infection, Genetics and Evolution*, **83**, 104351.

van Dorp, L., Richard, D., Tan, C. C., Shaw, L. P., Acman, M. and Balloux, F. (2020b). No evidence for increased transmissibility from recurrent mutations in SARS-CoV-2. *Nature Communications*, **11**(5986).

VanInsberghe, D., Neish, A., Lowen, A. C. and Koelle, K. (2021). Recombinant SARS-CoV-2 genomes circulated at low levels over the first year of the pandemic. *Virus Evolution*. doi:10.1093/ve/veab059.

Varabyou, A., Pockrandt, C., Salzberg, S. L. and Pertea, M. (2021). Rapid detection of inter-clade recombination in SARS-CoV-2 with Bolotie. *Genetics*, **218**(3), iyab074.

Volz, E. M. and Frost, S. D. (2014). Sampling through time and phylodynamic inference with coalescent and birth–death models. *Journal of The Royal Society Interface*, **11**(101), 20140945.

Volz, E. M., Pond, S. L. K., Ward, M. J., Brown, A. J. L. and Frost, S. D. (2009). Phylodynamics of infectious disease epidemics. *Genetics*, **183**(4), 1421–1430.

Wang, H., Kosakovsky Pond, S. L., Nekrutenko, A. and Nielsen, R. (2020). Testing recombination in the pandemic SARS-CoV-2 strains. `https://virological.org/t/testing-recombination-in-the-pandemic-sars-cov-2-strains/492`.

Wang, L., Zhang, K. and Zhang, L. (2001). Perfect phylogenetic networks with recombination. *Journal of Computational Biology*, **8**(1), 69–78.

Waugh, W. A. O. (1958). Conditioned Markov processes. *Biometrika*, **45**(1-2), 241–249.

Wiuf, C. (2018). Some properties of the conditioned reconstructed process with Bernoulli sampling. *Theoretical Population Biology*, **122**, 36–45.

Wiuf, C. and Hein, J. (1999). Recombination as a point process along sequences. *Theoretical Population Biology*, **55**(3), 248–259.

Wu, F., Zhao, S., Yu, B., Chen, Y.-M., Wang, W., Song, Z.-G., Hu, Y., Tao, Z.-W., Tian, J.-H., Pei, Y.-Y. et al. (2020). A new coronavirus associated with human respiratory disease in china. *Nature*, **579**(7798), 265–269.

Wu, Y. (2009). New methods for inference of local tree topologies with recombinant SNP sequences in populations. *IEEE/ACM Transactions on ComputationalBbiology and Bioinformatics*, **8**(1), 182–193.

Yang, Z. and Rannala, B. (1997). Bayesian phylogenetic inference using DNA sequences: a Markov Chain Monte Carlo method. *Molecular Biology and Evolution*, **14**(7), 717–724.

Yi, H. (2020). 2019 novel coronavirus is undergoing active recombination. *Clinical Infectious Diseases*, **71**(15), 884–887.

Zhang, Z., Shen, L. and Gu, X. (2016). Evolutionary dynamics of MERS-CoV: potential recombination, positive selection and transmission. *Scientific Reports*, **6**(1), 1–10.

Appendix A

Summary of RRPs

RRP	Y	Z_ψ^α	X_1^β	X_ψ^γ	X_ψ^δ
Time variable	t	$\alpha = \frac{1}{\lambda\psi}(e^t - 1)$ $t = \log(1 + \psi\lambda\alpha)$	$\beta = \frac{1}{\lambda - \mu} \log\Big(1 + \frac{\lambda - \mu}{\lambda}(e^t - 1)\Big)$ $t = \log\Big(1 + \frac{\lambda}{\lambda - \mu}\Big(e^{(\lambda-\mu)\beta} - 1\Big)\Big)$	$\gamma = \frac{1}{\lambda - \mu} \log\Big(1 + \frac{1}{\psi}(e^{(\lambda-\mu)\beta} - 1)\Big)$ $\beta = \frac{1}{\lambda - \mu} \log\Big(1 + \psi(e^{(\lambda-\mu)\gamma} - 1)\Big)$	$\delta = (\lambda - \mu)\gamma$ $\gamma = \frac{1}{\lambda - \mu}\delta$
Corresponding complete process	Yule(1)	CBP(λ, ψ)	BDP$(\lambda, \mu, 1)$	BDP(λ, μ, ψ)	BDP(λ', μ', ψ)
m (death rate of the RRP, per lineage)	1	$\frac{\psi\lambda}{1 + \psi\lambda\alpha}$	$\frac{\lambda e^{(\lambda-\mu)\beta}}{1 + \frac{\lambda}{\lambda-\mu}(e^{(\lambda-\mu)\beta} - 1)}$	$\frac{\psi\lambda e^{(\lambda-\mu)\gamma}}{1 + \frac{\psi\lambda}{\lambda-\mu}(e^{(\lambda-\mu)\gamma} - 1)}$	$\frac{\psi\lambda' e^{\delta}}{1 + \psi\lambda'(e^{\delta} - 1)}$
$\rho = \int m$	t	$\log(1 + \psi\lambda\alpha)$	$\log\Big(1 + \frac{\lambda}{\lambda - \mu}\Big(e^{(\lambda-\mu)\beta} - 1\Big)\Big)$	$\log\Big(1 + \frac{\psi\lambda}{\lambda - \mu}\Big(e^{(\lambda-\mu)\gamma} - 1\Big)\Big)$	$\log\Big(1 + \psi\lambda'\Big(e^{\delta} - 1\Big)\Big)$
$e^{-\rho}$	e^{-t}	$\frac{1}{1 + \psi\lambda\alpha}$	$\frac{1}{1 + \frac{\lambda}{\lambda-\mu}\Big(e^{(\lambda-\mu)\beta} - 1\Big)}$	$\frac{1}{1 + \frac{\psi\lambda}{\lambda-\mu}\Big(e^{(\lambda-\mu)\gamma} - 1\Big)}$	$\frac{1}{1 + \psi\lambda'\Big(e^{\delta} - 1\Big)}$

Appendix B

SARS-CoV-2 and MERS-CoV data

Other lineages			Lineage B.1.351		
Accession	Date	Ref	Accession	Date	Ref
EPI_ISL_660225	02/11/2020	SAO1	EPI_ISL_736958	20/11/2020	SAN1
EPI_ISL_660257	18/11/2020	SAO2	EPI_ISL_696481	19/11/2020	SAN2
EPI_ISL_736993	25/11/2020	SAO3	EPI_ISL_660637	03/11/2020	SAN3
EPI_ISL_660643	01/11/2020	SAO4	EPI_ISL_678632	11/11/2020	SAN4
EPI_ISL_660229	16/11/2020	SAO5	EPI_ISL_736932	25/11/2020	SAN5
EPI_ISL_736985	25/11/2020	SAO6	EPI_ISL_678641	12/11/2020	SAN6
EPI_ISL_736926	26/11/2020	SAO7	EPI_ISL_700422	04/11/2020	SAN7
EPI_ISL_696462	19/11/2020	SAO8	EPI_ISL_696503	25/11/2020	SAN8
EPI_ISL_660655	03/11/2020	SAO9	EPI_ISL_700470	12/11/2020	SAN9
EPI_ISL_660625	05/11/2020	SAO10	EPI_ISL_736983	24/11/2020	SAN10
EPI_ISL_660231	16/11/2020	SAO11	EPI_ISL_736936	19/11/2020	SAN11
EPI_ISL_678608	15/11/2020	SAO12	EPI_ISL_700487	06/11/2020	SAN12
EPI_ISL_660163	05/11/2020	SAO13	EPI_ISL_736935	26/11/2020	SAN13
EPI_ISL_660232	17/11/2020	SAO14	EPI_ISL_700443	13/11/2020	SAN14
EPI_ISL_700488	05/11/2020	SAO15	EPI_ISL_736939	24/11/2020	SAN15
EPI_ISL_660652	01/11/2020	SAO16	EPI_ISL_700554	02/11/2020	SAN16
EPI_ISL_660622	07/11/2020	SAO17	EPI_ISL_696505	25/11/2020	SAN17
EPI_ISL_660651	02/11/2020	SAO18	EPI_ISL_696518	24/11/2020	SAN18
EPI_ISL_678612	15/11/2020	SAO19	EPI_ISL_700589	12/11/2020	SAN19
EPI_ISL_696509	24/11/2020	SAO20	EPI_ISL_736959	20/11/2020	SAN20
EPI_ISL_678595	18/11/2020	SAO21	EPI_ISL_696453	20/11/2020	SAN21
EPI_ISL_660222	09/11/2020	SAO22	EPI_ISL_696521	24/11/2020	SAN22
EPI_ISL_696468	18/11/2020	SAO23	EPI_ISL_736964	19/11/2020	SAN23
EPI_ISL_660230	16/11/2020	SAO24	EPI_ISL_736928	24/11/2020	SAN24
EPI_ISL_660626	07/11/2020	SAO25	EPI_ISL_678629	13/11/2020	SAN25

Table B.1: GISAID accession numbers, collection dates, and references of sequences in the South Africa (November) sample.

Accession	Date
EPI_ISL_1048548	01/02/2021
EPI_ISL_1048553	02/02/2021
EPI_ISL_1048554	02/02/2021
EPI_ISL_1048555	02/02/2021
EPI_ISL_1048562	01/02/2021
EPI_ISL_1366778	02/02/2021
EPI_ISL_1366779	02/02/2021
EPI_ISL_1366781	01/02/2021
EPI_ISL_1366782	18/02/2021
EPI_ISL_1366783	25/02/2021
EPI_ISL_1366793	04/02/2021
EPI_ISL_1366840	05/02/2021
EPI_ISL_1366864	05/02/2021
EPI_ISL_1366869	05/02/2021
EPI_ISL_1366877	05/02/2021
EPI_ISL_1366887	06/02/2021
EPI_ISL_1366888	05/02/2021
EPI_ISL_1371923	23/02/2021
EPI_ISL_1371924	21/02/2021
EPI_ISL_1371925	15/02/2021
EPI_ISL_1371926	09/02/2021
EPI_ISL_1371927	19/02/2021
EPI_ISL_1371928	21/02/2021
EPI_ISL_1371929	20/02/2021
EPI_ISL_1371930	09/02/2021
EPI_ISL_1371931	21/02/2021
EPI_ISL_1371932	17/02/2021
EPI_ISL_1371933	23/02/2021
EPI_ISL_1371995	05/02/2021
EPI_ISL_1371996	15/02/2021
EPI_ISL_1371999	09/02/2021
EPI_ISL_1372000	07/02/2021
EPI_ISL_1372001	08/02/2021
EPI_ISL_1372002	09/02/2021
EPI_ISL_1372003	24/02/2021
EPI_ISL_1372004	18/02/2021
EPI_ISL_1372005	17/02/2021
EPI_ISL_1372006	17/02/2021

Table B.2: GISAID accession numbers, collection dates, and references of sequences in the South Africa (February) sample

Other lineages			Lineage B.1.1.7		
Accession	Date	Ref	Accession	Date	Ref
EPI_ISL_662468	12/11/2020	EO1	EPI_ISL_708881	30/11/2020	EN1
EPI_ISL_664402	06/11/2020	EO2	EPI_ISL_705071	22/11/2020	EN2
EPI_ISL_702752	19/11/2020	EO3	EPI_ISL_657548	09/11/2020	EN3
EPI_ISL_650455	13/11/2020	EO4	EPI_ISL_702338	27/11/2020	EN4
EPI_ISL_667977	14/11/2020	EO5	EPI_ISL_656730	08/11/2020	EN5
EPI_ISL_642566	02/11/2020	EO6	EPI_ISL_709730	26/11/2020	EN6
EPI_ISL_661404	11/11/2020	EO7	EPI_ISL_702093	28/11/2020	EN7
EPI_ISL_679726	01/11/2020	EO8	EPI_ISL_675080	15/11/2020	EN8
EPI_ISL_654967	10/11/2020	EO9	EPI_ISL_673518	15/11/2020	EN9
EPI_ISL_659205	05/11/2020	EO10	EPI_ISL_704716	30/11/2020	EN10
EPI_ISL_659013	01/11/2020	EO11	EPI_ISL_676036	13/11/2020	EN11
EPI_ISL_662253	11/11/2020	EO12	EPI_ISL_704695	02/11/2020	EN12
EPI_ISL_660027	04/11/2020	EO13	EPI_ISL_704619	21/11/2020	EN13
EPI_ISL_646293	04/11/2020	EO14	EPI_ISL_658341	08/11/2020	EN14
EPI_ISL_664758	12/11/2020	EO15	EPI_ISL_661750	14/11/2020	EN15
EPI_ISL_659140	05/11/2020	EO16	EPI_ISL_665414	02/11/2020	EN16
EPI_ISL_661929	14/11/2020	EO17	EPI_ISL_703736	26/11/2020	EN17
EPI_ISL_641906	03/11/2020	EO18	EPI_ISL_658292	08/11/2020	EN18
EPI_ISL_661483	11/11/2020	EO19	EPI_ISL_709568	26/11/2020	EN19
EPI_ISL_656165	06/11/2020	EO20	EPI_ISL_704601	22/11/2020	EN20
EPI_ISL_658415	08/11/2020	EO21	EPI_ISL_656409	08/11/2020	EN21
EPI_ISL_655916	08/11/2020	EO22	EPI_ISL_668252	12/11/2020	EN22
EPI_ISL_637180	02/11/2020	EO23	EPI_ISL_661854	12/11/2020	EN23
EPI_ISL_673482	15/11/2020	EO24	EPI_ISL_703229	19/11/2020	EN24
EPI_ISL_703087	19/11/2020	EO25	EPI_ISL_657799	08/11/2020	EN25
EPI_ISL_675115	13/11/2020	EO26	EPI_ISL_708945	30/11/2020	EN26
EPI_ISL_664943	04/11/2020	EO27	EPI_ISL_679428	22/11/2020	EN27
EPI_ISL_706068	02/11/2020	EO28	EPI_ISL_676194	13/11/2020	EN28
EPI_ISL_657282	08/11/2020	EO29	EPI_ISL_683471	24/11/2020	EN29
EPI_ISL_679916	06/11/2020	EO30	EPI_ISL_676012	13/11/2020	EN30
EPI_ISL_673815	15/11/2020	EO31	EPI_ISL_705063	22/11/2020	EN31
EPI_ISL_678719	16/11/2020	EO32	EPI_ISL_659491	05/11/2020	EN32
EPI_ISL_705061	19/11/2020	EO33	EPI_ISL_668018	12/11/2020	EN33
EPI_ISL_646457	03/11/2020	EO34	EPI_ISL_702918	19/11/2020	EN34
EPI_ISL_656970	08/11/2020	EO35	EPI_ISL_657622	08/11/2020	EN35
EPI_ISL_647347	01/11/2020	EO36	EPI_ISL_704698	01/11/2020	EN36
EPI_ISL_650406	08/11/2020	EO37	EPI_ISL_679302	21/11/2020	EN37
EPI_ISL_661700	13/11/2020	EO38	EPI_ISL_704606	22/11/2020	EN38
EPI_ISL_658474	08/11/2020	EO39	EPI_ISL_703148	19/11/2020	EN39
EPI_ISL_700654	09/11/2020	EO40	EPI_ISL_645527	05/11/2020	EN40

Table B.3: GISAID accession numbers, collection dates, and references of sequences in the England (November) sample

Accession	Date	Ref	Accession	Date	Ref
EPI_ISL_878756	13/01/2021	E1	EPI_ISL_868555	18/01/2021	E21
EPI_ISL_778191	20/12/2020	E2	EPI_ISL_885546	18/01/2021	E22
EPI_ISL_836766	04/01/2021	E3	EPI_ISL_816845	31/12/2020	E23
EPI_ISL_720681	02/12/2020	E4	EPI_ISL_736552	11/12/2020	E24
EPI_ISL_735634	13/12/2020	E5	EPI_ISL_731132	10/12/2020	E25
EPI_ISL_816235	29/12/2020	E6	EPI_ISL_820022	25/12/2020	E26
EPI_ISL_799427	22/12/2020	E7	EPI_ISL_1054040	30/01/2021	E27
EPI_ISL_777127	17/12/2020	E8	EPI_ISL_881303	12/01/2021	E28
EPI_ISL_811454	01/01/2021	E9	EPI_ISL_838888	04/01/2021	E29
EPI_ISL_1242096	27/01/2021	E10	EPI_ISL_950899	27/12/2020	E30
EPI_ISL_735656	13/12/2020	E11	EPI_ISL_709038	04/12/2020	E31
EPI_ISL_863458	13/01/2021	E12	EPI_ISL_842015	01/01/2021	E32
EPI_ISL_1178212	25/01/2021	E13	EPI_ISL_835329	05/01/2021	E33
EPI_ISL_777970	18/12/2020	E14	EPI_ISL_741276	08/12/2020	E34
EPI_ISL_782374	26/12/2020	E15	EPI_ISL_813970	26/12/2020	E35
EPI_ISL_868478	07/01/2021	E16	EPI_ISL_1051452	22/01/2021	E36
EPI_ISL_762877	16/12/2020	E17	EPI_ISL_1046024	27/01/2021	E37
EPI_ISL_1050650	29/01/2021	E18	EPI_ISL_836823	04/01/2021	E38
EPI_ISL_708906	04/12/2020	E19	EPI_ISL_994038	12/01/2021	E39
EPI_ISL_740955	02/12/2020	E20	EPI_ISL_820233	14/12/2020	E40

Table B.4: GISAID accession numbers, collection dates, and references of sequences in the England (January) sample

Accession	Submitters	Date	Ref
KY688118.1	Paden, C. R., et al.	07/02/2015	M1
KT806044.1	Lu, X., et al.	09/02/2015	M2
KT806045.1	Lu, X., et al.	22/02/2015	M3
KT806047.1	Lu, X., et al.	27/03/2015	M4
KT806048.1	Lu, X., et al.	07/02/2015	M5
KT806049.1	Lu, X., et al.	15/02/2015	M6
KT806051.1	Lu, X., et al.	05/02/2015	M7
KT806052.1	Lu, X., et al.	02/02/2015	M8
KT806053.1	Lu, X., et al.	02/02/2015	M9
KT806054.1	Lu, X., et al.	13/02/2015	M10
KT806055.1	Lu, X., et al.	10/02/2015	M11
KT026453.1	Park, W. B., et al.	10/02/2015	M12
KT026454.1	Park, W. B., et al.	01/03/2015	M13
KT026455.1	Park, W. B., et al.	10/02/2015	M14
KT026456.1	Park, W. B., et al.	01/03/2015	M15
KR011263.1	Lu, X., et al.	21/01/2015	M16
KR011264.1	Lu, X., et al.	21/01/2015	M17
KR011265.1	Lu, X., et al.	26/01/2015	M18
KR011266.1	Lu, X., et al.	06/01/2015	M19

Table B.5: NCBI Virus database accession numbers, collection dates, and references of sequences in the MERS-CoV sample