Simultaneous Inference of Past Demography and Selection from the Ancestral Recombination Graph under the Beta Coalescent

Kevin Korfmann[1*], Thibaut Sellinger[1*,2*], Fabian Freund[3,5], Matteo Fumagalli[4], Aurélien Tellier[1+]

[1] Population Genetics, Department of Life Science Systems,

Technical University of Munich (Liesel-Beckmann-Strasse 2, 85354 Freising, Germany)

[2] Department of Environment and Biodiversity, Paris Lodron University of Salzburg

[3] Institute of Plant Breeding, Seed Science and Population Genetics, University of Hohenheim

(Fruwirthstrasse 21, 70599 Stuttgart, Germany)

[4] Department of Biological and Behavioural Sciences, Queen Mary University of London

(Mile End Road, London E1 4NS, UK)

[5] Department of Genetics and Genome Biology, University of Leicester

(University Road, Leicester LE1 7RH, UK)

[*] both author contributed equally and share first authorship

[+] Corresponding author, aurelien.tellier@tum.de

Abstract

The reproductive mechanism of a species is a key driver of genome evolution. The standard Wright-Fisher model for the reproduction of individuals in a population assumes that each individual produces a number of offspring negligible compared to the total population size. Yet many species of plants, invertebrates, prokaryotes or fish exhibit neutrally skewed offspring distribution or strong selection events yielding few individuals to produce a number of offspring of up to the same magnitude as the population size. As a result, the genealogy of a sample is characterized by multiple individuals (more than two) coalescing simultaneously to the same common ancestor. The current methods developed to detect such multiple merger events do not account for complex demographic scenarios or recombination, and require large sample sizes. We tackle these limitations by developing two novel and different approaches to infer multiple merger events from sequence data or the ancestral recombination graph (ARG): a sequentially Markovian coalescent (SMβC) and a graph neural network (GNN*coal*). We first give proof of the accuracy of our methods to estimate the multiple merger parameter and past demographic history using simulated data under the β-coalescent model. Secondly, we show that our approaches can also recover the effect of positive selective sweeps along the genome. Finally, we are able to distinguish skewed offspring distribution from selection while simultaneously inferring the past variation of population size. Our findings stress the aptitude of neural networks to leverage information from the ARG for inference but also the urgent need for more accurate ARG inference approaches.

Keywords— kingman coalescent, beta coalescent, selective sweep, deep learning, graph neural networks, population genetics, multiple merger coalescent, sequentially markovian coalescent, ancestral recombination graph

Introduction

With the availability of genomes of increasing quality for many species across the tree of life, population genetics models and statistical methods have been developed to recover the past history of a population/species from whole genome sequence data from several individuals [82, 53, 77, 83, 80, 5, 4, 85, 40, 41]. Indeed, the inference of the past demographic history of a species, *i.e.* population expansion, contraction, or bottlenecks, extinction/colonisation, is not only interesting in its own right, but also essential to calibrate genome-wide scans to detect genes under (*e.g.* positive or balancing) selection [85, 42]. A common feature of inference methods that make full use of whole genome sequences is the underlying assumption of a Kingman coalescent process [47] to describe the genealogy distribution of a sample. The Kingman coalescent process and its properties stem from using the traditional forward-in-time Wright-Fisher (WF) model to describe the reproduction mechanism of a population. Besides non-overlapping generations, a key assumption of the neutral WF model is that an individual offspring chooses randomly (*i.e.* uniformly) its parents from the previous generation. More precisely, each chromosome chooses a parental chromosome from the previous generation. Thus, a key parameter is the distribution of the number of offspring that parents can have. In the WF model, due to the binomial sampling, the distribution of offspring number per parent is well approximated by a Poisson distribution with both mean and variance equal to one. This implies that parents will most likely have zero, one, or two offspring individuals, but it is improbable that one parent would have many offspring individuals (*i.e.* on the order of the population size, under the Wright-Fisher haploid model the probability for a parent to have 10 or more offspring is $\approx 10^{-8}$). The assumption of small variance in offspring distribution between individual parents is realistic for species with low juvenile mortality (so-called type I and II survivorship in ecology), such as mammals.

As genome sequence data become available for a wide variety of species with different biological traits and/or life cycles, the applicability of the Kingman coalescent relying on the WF model can be questioned [84, 2, 3, 64, 43, 61, 87, 58, 29]. Indeed, for some species, such as fish, with high fecundity and high juveniles mortality (type III survivorship), it is expected that the variance in reproduction between parents can be much larger than under the Poisson distribution [87]. This effect is termed as sweepstake reproduction [34, 2]. Neutral processes such as strong seed banking [11], high fecundity with skewed offspring distribution [34, 24], extremely strong and recurrent bottlenecks [9, 20], and strong selective processes (*i.e.* positive selection) [23, 16, 17, 33, 3] are theoretically shown to deviate from the classic WF model in a way that the genealogies can no longer be described by a Kingman coalescent process. Under such conditions, a new class of processes arise to describe the genealogy distribution, a class where multiple individuals can coalesce and/or multiple distinguished coalescence events can occur simultaneously [73, 60, 22, 72, 66, 13]. Generally, this class of genealogical processes is called the Multiple Merger Coalescent (MMC). MMC models are more biologically appropriate than the Kingman coalescent to study many species of fish [25, 2, 3, 34], invertebrates (insects, crustaceans, etc.), viruses [56], bacteria [58, 62], plants and their pathogens [87]. While we would like to assess which population model best describes the species genealogy, field experiments to quantify the underlying reproduction mechanism of a species can be costly and time consuming at best, or intractable at worst. Therefore, an alternative solution is to use inference methods based on genome data to identify which model best describes

the genealogy of a given species/population.

In this study we use the so-called β-coalescent, a specific class of MMC models. Unlike under the WF model, under MMC models the ploidy level strongly affects the distribution of genealogies [8]. For simplicity, in this study we focus on haploid organisms. It is demonstrated that if the probability of a parent to have k or more offspring is proportional to $k^{-\alpha}$, where $1 < \alpha < 2$, then the genealogy can be described by a Λ-coalescent [79]. The latter is a general class of coalescent process describing how and how fast ancestral lineages merge [66, 72]. When using the Beta($2-\alpha$,α) distribution as a probability measure for the Λ-coalescent, the transition rates (*i.e.* coalescent rate) can be analytically obtained leading to the β-coalescent, a specific MMC model. If α tends to 2, then the coalescent process converges to a Kingman coalescent (up to a scaling constant) [8, 50, 51]. If α tends to one, the model tends to a Bolthausen-Sznitman coalescent process (*i.e.* dominated by strong multiple merger events) [13]. The β-coalescent has the property that the observed polarized Site Frequency Spectrum (SFS) of a sample of single nucleotide polymorphisms (SNPs) exhibits a characteristic U-shape with an excess of rare and high frequency variants (compared to the Kingman coalescent) [76]. Current methods to draw inference under MMC models leverage information from the summary statistics extracted from full genome data such as Site Frequency Spectrum (SFS, or derived summary statistics) [51, 33, 71], minor allele frequency [69] or copy number alteration [43]. It is shown that the SFS is robust to the effect of recombination [51, 69] and its shape allows to discriminate between simple demographic models (population expansion or contraction) under the Kingman coalescent and MMC models with constant population size [51, 50, 25]. However, methods relying on genome-wide SFS have two main disadvantages. First, in absence of strong prior knowledge, they can suffer from non-identifiability [40] as several complex neutral demographic and/or selective models under the Kingman or MMC models can generate similar SFS distributions. Second, as they summarize the collection of underlying genealogies, they require high sample sizes (>50) to produce trustworthy results [51, 50, 25], relying on experimental designs which are prohibitive for the study of non-model species. To tackle these limitations, we develop two methods that integrate recombination events along the genome in order to leverage more information from full genome data, thus requiring fewer samples.

In species undergoing sexual reproduction, recombination events break the genealogy of a sample at different position of the genome (*i.e.* the genealogy of a sample varies along the genome), leading to what is called the Ancestral Recombination Graph (ARG) [37, 8]. Because all the genealogical information is contained in the ARG, in this study we aim at the interpretation of the ARGs to recover model parameters in presence of multiple merger events. With the development of the sequentially Markovian coalescent theory [57, 55, 92], it becomes tractable to integrate linkage disequilibrium over chromosomes in inferences based on the Kingman coalescent [53]. Hence, we first develop an SMC approach based on the β-coalescent named the Sequentially Markovian β Coalescent (SMβC). The β-coalescent has the additional property that, under recombination, long range dependency can be generated between coalescent trees along the genome if multiple-merger events happen in a single generation [8]. In other words, coalescent trees which are located at different places in the genome, and expected to be unlinked from one another [63], would show non-zero correlation in their topology and coalescent times. This is because coalescent trees from different genomic regions may all be affected by the

same MMC event (merger event of multiple lineages in the past) which then leaves traces in the genome at several loci [9]. To overcome the theoretically predicted non-Markovian property of the distribution of genealogies along the genome under the β-coalescent with recombination [8], we develop a second method based on deep learning (DL) trained from efficient coalescent simulations [7]. In evolutionary genomics, DL approaches trained by simulations are shown to be powerful inference tools [82, 49]. Previous work demonstrated that DL approach can help overcome problems mathematically insolvable or computationally intractable in the field of population genetics [82, 6, 91, 95, 28, 21, 67, 18, 39]. The novelty of our neural network relies on its structure (Graph Neural Network, GNN) and its training algorithm based on the ARG of a sample, or its tree sequence representation [44]. GNNs are an emerging category of DL algorithm [15, 93, 19, 98] that benefit by using irregular domain data (*i.e.* graphs). GNNs are designed for the prediction of node features [48, 94], edge features (link prediction) [97, 78], or additional properties of entire graphs [96, 52]. Therefore, GNNs represent a new tool to address the large dimensionality of ARGs, while simultaneously leveraging information from the genealogy (namely topology and age of coalescent events) as a substantial improvement over convolutions of genotype matrices, as currently done in the field [74].

We first quantify the bias of previous SMC methods (MSMC and MSMC2 [77, 90]) when performing inference of past population size variation under the β-coalescent. We then describe our two methods, SMβC and GNN*coal*, and demonstrate their statistical power as well as their respective limitations. From simulated tree-sequence (*i.e.* ARG) and sequence (*i.e.* SNPs) data, we assess the accuracy of both approaches to recover the past variation of population size and the α parameter of the Beta-distribution. This parameter indicates how frequent and strong multiple merger events occur. We demonstrate that our approaches can infer the evolutionary mechanism responsible for multiple merger events and distinguish local selection events from genome-wide effects of multiple mergers. We highlight the limits of the Markovian property of SMC to describe data generated under the β-coalescent. Finally, we show that both our approaches can model and identify the presence of selection along the genome while simultaneously accounting for non-constant population size, recombination, and skewed offspring distribution. Thus our methods represents a major and necessary leap forward in the field of population genetic inferences.

Materials and Methods

In our study we first assume the true ARG to be known. Hence, the ARG of the sample is given as input to our methods to estimate recover model parameters of interest (*e.g.* the α parameter and/or the past variation of population size). We then show the applicability of our methods by using as input simulated sequence data (*i.e.* SNPs) and/or ARG inferred using ARGweaver [68] from simulated sequence data.

SMC-based method

In this study, we use different SMC-based algorithms: two previously published, MSMC and MSMC2 [77, 90], and the new SMβC. In the latter, the software backbone stems from our previous eSMC [80, 81] whilst the theoretical framework originates from the MSMC algorithm [77] (see Supplementary Text S1). All approaches can use the ARG

or sequence data as input. Giving ARG as input for MSMC an MSMC2 is enabled by a re-implementation included in the R package eSMC2 [81]. The MSMC2 algorithm focuses on the coalescence time between two haploid samples along the genome. In the event of recombination, there is a break in the current genealogy and the coalescence time consequently takes a new value. A detailed description of the algorithm can be found in [26, 90]. The MSMC algorithm simultaneously analyses multiple sequences (up to 10) and follows the distribution of the first coalescence event in a sample of size $n > 2$ along the sequence based on the Kingman coalescent [47]. A detailed description of MSMC can be found in [77].

Our new approach, SMβC, is a theoretical extension of the MSMC algorithm, simultaneously analyzing multiple haploid sequences and focusing on the first coalescence event of a sample size 3 or 4. The SMβC follows the distribution of the first coalescence event of a sample along sequences assuming a β-coalescent process. Therefore, our SMβC allows for more than two ancestral lineages to join the first coalescence event, or new lineages to join an already existing binary (or triple) coalescent event. Hence, the SMβC extends the MSMC theoretical framework by adding hidden states at which more than two lineages coalesce. Currently, the SMβC has been derived to analyze for up to 4 sequences simultaneously (due to computational load and mathematical complexity). The emission matrix is similar to the one of MSMC. As in the MSMC software, the population size is assumed piece-wise constant in time and we discretize time in 40 bins throughout this study. A detailed description of SMβC can be found in Supplementary Text S1. To test and validate the theoretical accuracy of our approach, we first study its best case convergence (introduced in [81]) which corresponds to the model's performance when the true genealogy is given as input, *i.e.* as if the hidden states are known. Additionally, we also validate the practical accuracy of the SMβC on simulated sequence data taking the same input as the MSMC software [77], or using the inferred ARGs by ARGweaver [68]. All SMC approaches used in this manuscript are found in the R package eSMC2 (`https://github.com/TPPSellinger/eSMC2`).

GNN*coal* method

Inspired by results obtained from inferences based on tree sequence data [31, 81], we develop a graph neural network (GNN) taking tree sequence data as input. Our GNN is designed to infer population size along with the α parameter of the Beta distribution describing the distribution of offspring production. In practice, the ARG is reshaped into a sequence of genealogies (more precisely a sequence of undirected graphs), and then given as input to the GNN (similar to what is described above for the SMβC). In our analyses, we fixed the batch size to 500. This value represents the number coalescence trees being processed before updating parameters of the neural network. As batch size is fixed to 500, only simulations displaying at least 500 recombination events are considered for the training data sets. If more than 500 recombination events occur along the sequence, the ARG is truncated and the GNN will only take as input the first 500 genealogies and remove the rest. Thanks to the GNN architecture, the algorithm can account for the topology of the genealogy. Hence, the GNN leverages information from coalescence time and branch lengths but also from the topology of the ARG. This operation is known as a graph convolution. By doing so, the GNN is capable of learning from local features of the ARG and extract information from its complex structure. To learn from global ge-

nealogy patterns (which SMC-based methods cannot do), an additional pooling strategy is implemented as part of the network [96]. To do so, the ARG is broken into smaller ARGs (*i.e.* subgraphs) during the forward-pass step. To illustrate the GNN strategy, we visualize the compression-like process, from the coalescent trees (1) being processed by GNN*coal* (2,3) to the inferred variable of interest (4, 5) in Figure 1.

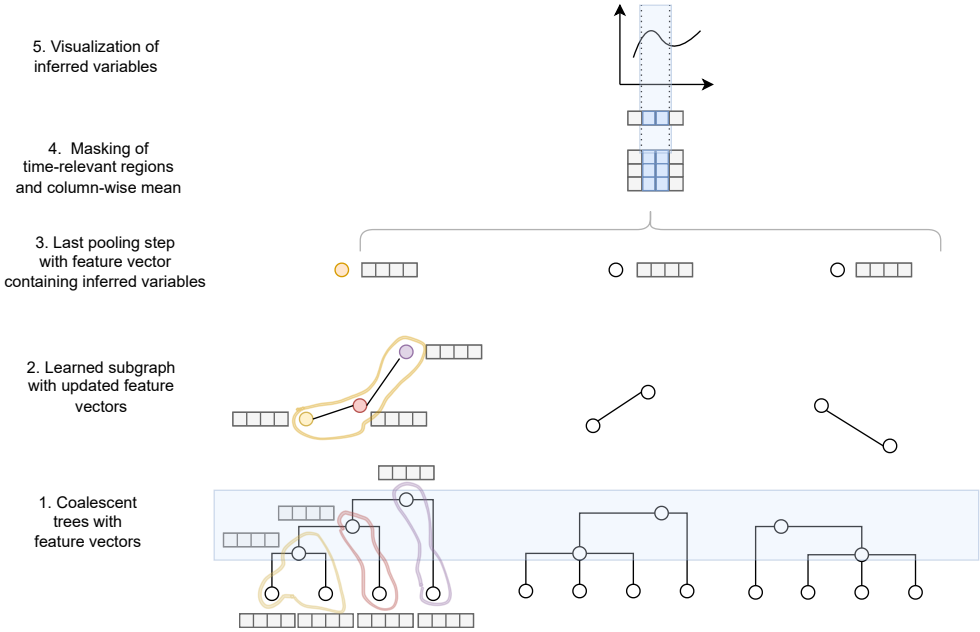

Fig. 1. Schematic representation of GNN*coal* processing an ARG Hierarchical pooling compression of a sequence of coalescent trees into the inferred variable of interest (i.e. demographic changes) using graph convolutions. Each coalescent ancestor or leaf node is initialized by a feature vector (light grey boxes) (1). Sub-graphs are generated by a pooling network with updated feature vectors and a final compression step is performed until ideally one node per graph remains (2-3). Lastly, the column-wise mean is taken after applying a time mask (blue - based on number of coalescent events), so that single feature vector remains (4-5). Detailed description of the graph convolution, feature vector initialization, pooling methodology, coalescent time mask construction, and dataset generation can be found in Supplementary Text S2.

To infer parameters from our neural network, we need to define an objective function to be optimized. We use a masked root-mean-squared error (RMSE) loss function as objective function which is computed for each inputted ARG (*i.e.* minimizing the average square difference between predicted and true parameter value). In practice, time is discretized (as for the SMβC) and time windows are defined. The true α value and true demography at 60 predefined time points are given as input to the GNN to compute the loss function. The GNN captures the stochastic complexity arising from the underlying demographic scenario and model parameters. Furthermore, our algorithm naturally defines an appropriate time window to have sufficient observation at each time point. A more detailed description of the GNN*coal* can be found in Supplementary Text S2. The code of the model architecture is implemented in *Pytorch* [65] using the extension *Pytorch Geometric* [27]. The model is available with the simulated training dataset at `https://github.com/kevinkorfmann/GNNcoal` and `https://github.com/kevinkorfmann/GNNcoal-analysis`.

ARGweaver

As the ARG is not known in practice, it needs to be inferred from sequence data. ARGweaver displays the best performance at recovering the ARG from whole genome polymorphism data at the sample sizes employed in this study (*i.e.* $\ll 50$) [68, 14]. Briefly, ARGweaver samples the ARG of n chromosomes/scaffolds conditional on the ARG of $n-1$ chromosomes/scaffolds. To this aim, ARGweaver relies on hidden Markov models while assuming a sequentially Markov coalescent process and a discretization of time, similarly to the SMC-based methods previously described. For a more detail description of the algorithm, we refer the reader to the supplementary material of [68].

Simulation of data

Validation dataset for both methods

The ARG is given as input to the DL approach and the SMβC (see [81]). We use msprime [7] to simulate the ARG of a sample (individuals are assumed to be haploid) under the β-coalescent based on [79, 8] or under the Kingman coalescent (under neutrality or selection). We simulate 10 sequences of 100 Mbp under five different demographic scenarios: 1) Constant population size; 2) Bottleneck with sudden decrease of the population size by a factor 10 followed by a sudden increase of population by a factor 10; 3) Expansion with sudden increase of the population size by a factor 10, 4) Contraction with sudden decrease of the population size by a factor 10; and 5) "Saw-tooth" with successive exponential decreases and increases of population size through time, resulting in continuous population size variation (as shown in [88, 77, 81]). We simulate data under different α values (*i.e.* parameters of the β-distribution) including values of 1.9 (almost no multiple merger events), 1.7, 1.5, and 1.3 (frequent and strong multiple merger events). Mutation and recombination rate (respectively μ and r) are set to 10^{-8} per generation per bp in order to obtain the best compromise between realistic values and number of SNPs. When specified, some specific scenarios assume recombination and mutation rate set to produce sufficient data or to avoid violation of the finite site hypothesis. All python scripts used to simulate data sets are available at `https://github.com/kevinkorfmann/GNNcoal-analysis`.

Additionally, to generate sequence data, we simulate 10 sequences of 10 Mbp under the five different demographic scenarios described above and for the same α values. For each scenario, 10 replicates are simulated. In order to obtain sufficient SNPs for inference, we simulate sequence data with mutation and recombination rate (respectively μ and r) of 10^{-8} per generation per bp when α is set to 1.9 and 1.7, 10^{-7} per generation per bp when α is set to 1.5, and 10^{-6} per generation per bp when α is set to 1.3.

Training dataset for the GNN*coal*

In our study we train two GNNs, one to infer past variation of population size through time along with α, and one for model selection. The training dataset used for both GNNs is described below.

Training dataset for the GNN inferring α and demography

We generate an extensive number of ARGs to train our GNN. The ARGs are simulated

under many demographic scenarios and α values. The model parameters are updated in supervised manner. The loss function is calculated for each batch with respect to how much the machine-learning estimates differ from to the true parameters used for simulation. The simulations strategy to recover past demographic history is based on the strategy described and used in [12, 74]. The idea of this approach is to generate a representative set of demographic scenarios over which the network generalizes to consequently infer similar demographic changes after training. More details on the training strategy can be found in Supplementary Text S2.

To improve the outputted demographic history, we introduce a smoothing of the demography allowing to infer continuous variation of population size through time. We do so by interpolating I time points cubically, and choosing w (set to 60) uniformly spaced new time points of the interpolation in log space. All time points more recent than ten generations in the past are discarded, since inference is too imprecise in the very recent present under our models. An example of this process can be seen in Supplementary Text S2.

Training dataset to disentangling coalescent and selection signatures

Beyond parameter inference, deep learning approaches can also be used for clustering. Hence, we train a GNN to disentangle between different scenarios and models. In total, we define eight classes, namely K (S0) (Kingman, no selection), K (WS) (Kingman, weak selection), K (MS) (Kingman, medium selection), K (SS) (Kingman, strong selection) and four different β-coalescent classes (]2.0-1.75],]1.75-1.5],]1.5-1-1.25],]1.25-1.01]) without selection. The three different selection regimes are defined, corresponding to $Ne \times s$ in: [0.1, 0.01[for SS, [0.01, 0.001[for MS, [0.001, 0.0001[for WS and [0] for absence of selection. Demography is kept constant and set to 10^5 individuals and sequence length is set to 10^5 bp. The simulation is discarded if it resulted in less than 2,000 obtained trees and is rerun with twice the sequence length until the tree number required is satisfied. This procedure avoids simulating large genome segments of which only a small fraction of trees is used for the given scenario during training and inference. The selection site is introduced in the centre of the respective sequence, so that 249 trees left and 250 right of the middle tree under selection form a training sample, using 500 trees for each sample. One hundred replicates are generated for each training sample. The complete training dataset consists of 1,000 parameter sets, 500 for the Kingman cases and 500 for the β-coalescent cases, with approximately 125 parameter sets per class. The model itself is trained on one epoch (number of time the data is analyzed), and the evaluation performed afterwards on 1,000 randomly generated parameter sets, with one replicate per parameter set. The same architecture used for demography estimation is employed with additional linear layers to reduce the number of output dimensions from 60 to 8. The loss function is set to a Cross-Entropy-Loss for the network to be trainable for categorical labels. Otherwise all architecture and training parameters is the same as described above and detailed in Supplementary Text S2.

Results

Inference bias under the wrongly assumed Kingman coalescent

We first study the effect of assuming a Kingman coalescent when the underlying true model is a β-coalescent (*i.e.* in presence of multiple merger events) by applying MSMC and MSMC2 to our simulated data. The inference results from MSMC and MSMC2 when the population undergoes a sawtooth demographic scenario are displayed in Figure 2. For $\alpha > 1.5$ the shape of the past demography is fairly well recovered. Decreasing the parameter α of the β-coalescent (*i.e.* higher probability of multiple merger events occurring) increases the variance of inferences and flattens the demography. Yet, both methods fail to correctly infer the correct population size, due to the scaling discrepancy between the Kingman and β-coalescent. Hence, we perform the same analysis and correct for the scaling effect of the MMC versus a Kingman coalescent in order to better capture the specific effects of assuming binary mergers only. The results are displayed in Figure S1. For $\alpha > 1.5$ the demography is accurately recovered providing we know the true value of α to adjust the y-axis (population size) scale. However, for smaller α values the observed variance is extremely high and a flattened past variation of population size is observed.

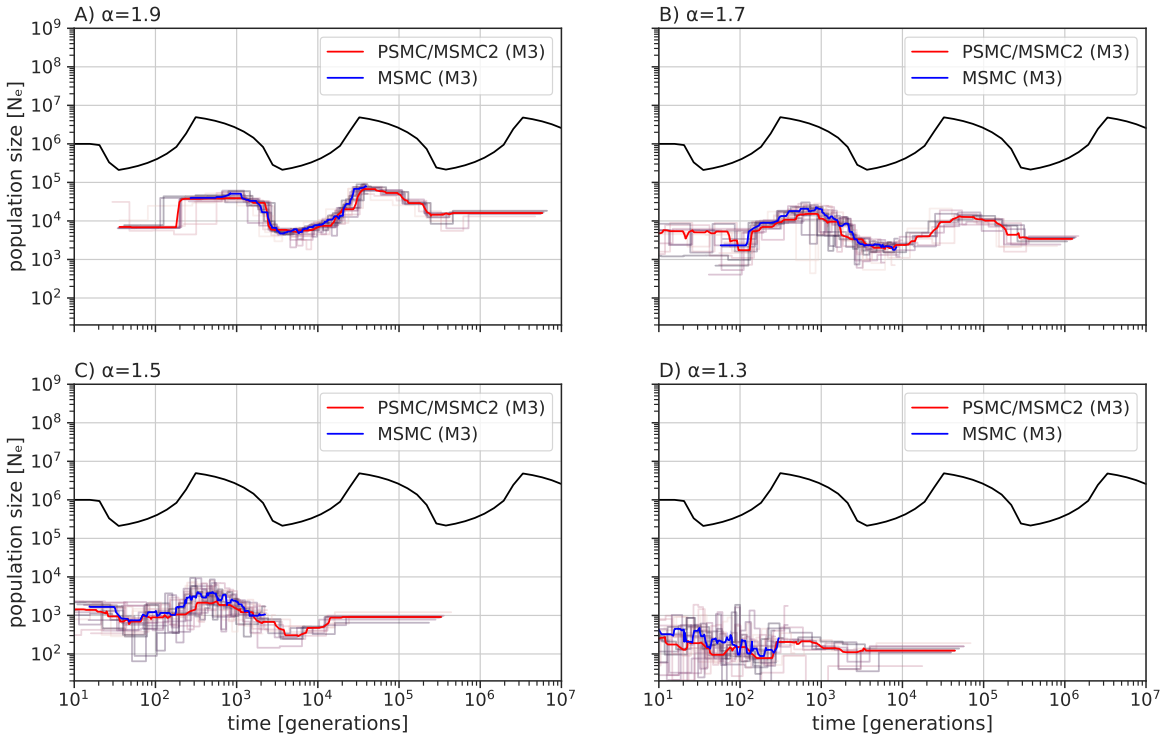

Fig. 2. Performance of MSMC and MSMC2 under a β-coalescent. Averaged estimated demographic history by MSMC (blue) and MSMC2 (red) based on 10 sequences of 100 Mb with $\mu = r = 10^{-8}$ per generation per bp over ten repetitions (while analyzing simultaneously 3 sequences, noted by M=3). Each repetition result is represented in light red (PSMC/MSMC2) or in light blue (MSMC). Population undergoes a sawtooth demographic scenario (black) for A) $\alpha = 1.9$, B) $\alpha = 1.7$, C) $\alpha = 1.5$, and D) $\alpha = 1.3$.

The limit of the Markovian hypothesis

As SMC approaches rely on the hypothesis of Markovian change in genealogy along the genome, we study the effect of α on the linkage disequilibrium of pairs of SNPs (r^2, [70, 59]) in data simulated under the Kingman Coalescent or the β-coalescent (with $\alpha = 1.5$ and $\alpha = 1.3$) and constant population size (Figure 3). Linkage monotonously decreases with distance under the Kingman coalescent. Under the β-coalescent a similar distribution is observed but with higher amount of LD. We find a higher variance in LD for smaller α values. More precisely, this increased variance results in the occurrence of high spikes of linkage disequilibrium along the genome (*e.g.* Figure 3 B). This stochastic increase of linkage along the genome demonstrates that the Markovian hypothesis used to model genealogies along the genome is violated under the β-coalescent due to the long range effect of strong multiple merger events [8].

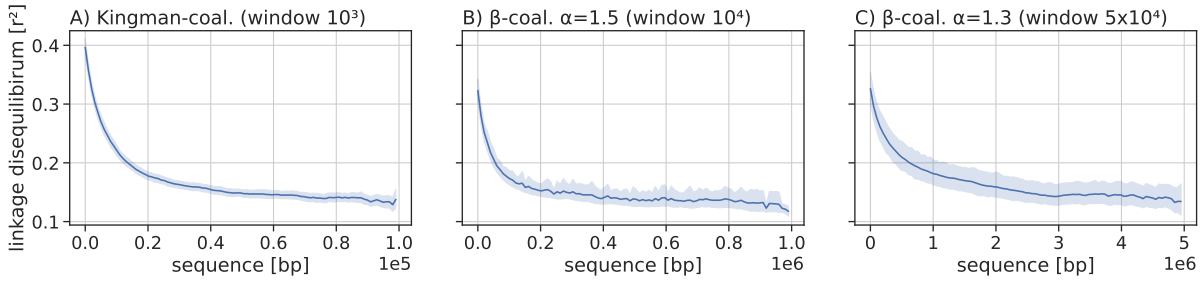

Fig. 3. Linkage disequilibrium under a Kingman and β-coalescent. Pairwise linkage disequilibrium between SNPs (r^2) under a Kingman and β-coalescent with $\alpha = 1.5$ and $\alpha = 1.3$ using 50 sequences of length A) 0.1 Mb, B) 1Mb, and C) 5Mb. The population size is constant at $N = 10^4$ for the Kingman model and $N = 10^6$ for the β-coalescent, with $\mu = 2 \times 10^{-7}$ and $r = 2 \times 10^{-8}$ per generation per bp. For each LD analysis, the linkage disequilibrium is calculated by averaging it over a window of size 10^3, 10^4 and 5×10^4 bp respectively in A), B) and C).

We further investigate the effect of multiple merger events on linkage disequilibrium. To do so, we first assume an SMC framework (*e.g.* MSMC2 or eSMC) to predict the transition matrix (*i.e.* matrix containing the probabilities for the coalescent time to change to another value along the genome) and looking at the absolute difference between the observed transition events. Under the Kingman coalescent, the distribution of coalescent times between two positions in a sample of size two ($n = 2$) is well approximated by the SMC as shown in Figure S2 (*i.e.* absence of structured difference between observed and predicted). However, under the β-coalescent (with $\alpha = 1.3$) we observe significant and structured differences between observed and predicted at times points where multiple merger events occur (Figure S3). In practice multiple merger events do not occur at each time point (as they remain rare events), unveiling a discrepancy between the expectation from the SMC (*i.e.* approximating the distribution of genealogies along the genome by a Markov chain) and the simulated data. This discrepancy does not stem from the simulator, because it correctly generates ARG under the β-coalescent model [8, 7], but from the limits of the SMC approximation to model events with long range effects on the ARG (Figure S3).

Inferring α and past demography on ARG

To test if our two approaches (GNN*coal* and SMβC) can recover the past variation of population size and the α parameter, we run both methods on simulated tree sequences under different α values and demographic scenarios. Figure 4 displays results for data simulated under a sawtooth past demography and for α ranging from 1.9, 1.7, 1.5 to 1.3. In all cases, the DL approach exhibits high accuracy and low variance to infer the variation of population size. For high α values (>1.5), the shape of population size variation is well recovered by SMβC. However, for smaller values, an extremely high variance is observed which demonstrates the limits of SMC inferences. On average, both approaches seem to recover fairly well the true α value (Figure 4 and Table 1 in S1), especially GNN*coal* which performs well by displaying high accuracy and lower standard deviation. We note that the variance in the estimation of α increases with diminishing α value. Moreover, increasing the number of simultaneously analyzed sequences by SMβC does not seem to improve the inferred α value (Table S1). These conclusions are also valid for the results in Figures S4-S7 and Table S1 based on inference under four additional demographic scenarios: constant population size, bottleneck, sudden increase and sudden decrease of population size.

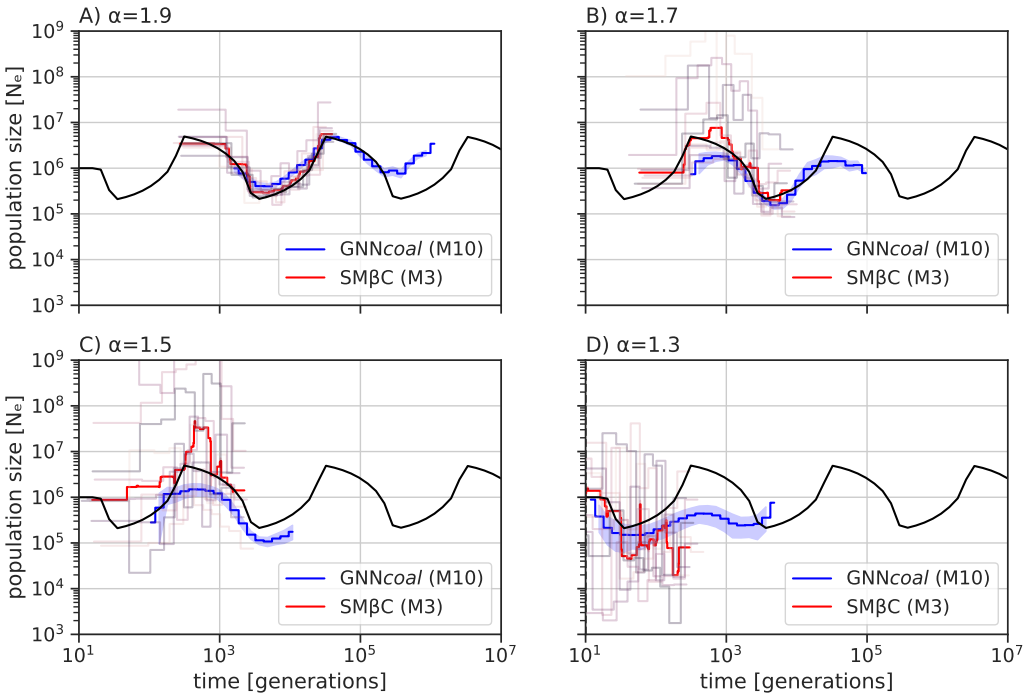

Fig. 4. Best-case convergence estimations of SMβC and GNN*coal* under a β-coalescent. Estimations of past demographic history by SMβC in red (median) and by GNN*coal* in blue (mean and 95% confidence interval, CI95; while analyzing simultaneously 3 or 10 sequences, noted by M=3 or M=10) when population undergoes a sawtooth demographic scenario (black) under A) $\alpha = 1.9$, B) $\alpha = 1.7$, C) $\alpha = 1.5$ and D) $\alpha = 1.3$. SMβC runs on 10 sequences and 100 Mb, GNN*coal* runs on 10 sequences and 500 trees, and $\mu = r = 10^{-8}$ per generation per bp.

Because when α diminishes, the effective population size decreases as well (*i.e.* the

population size calculated from SNPs assuming a Wright-Fisher model), the number of recombination events plummets for small values of $\alpha < 1.5$. To demonstrate the theoretical accuracy of the SMβC, *i.e.* the convergence to the correct values, we run SMβC on data simulated with mutation and recombination rate fifty times higher under similar scenarios as in Figure 4, namely to increase the amount of data (SNPs and number of independent coalescent trees by recombination). Results of SMβC for α values of 1.7, 1.5 and 1.3 are displayed on Table S2, where it appears that that SMβC can recover α with higher accuracy when given more data (*i.e.* recombination events).

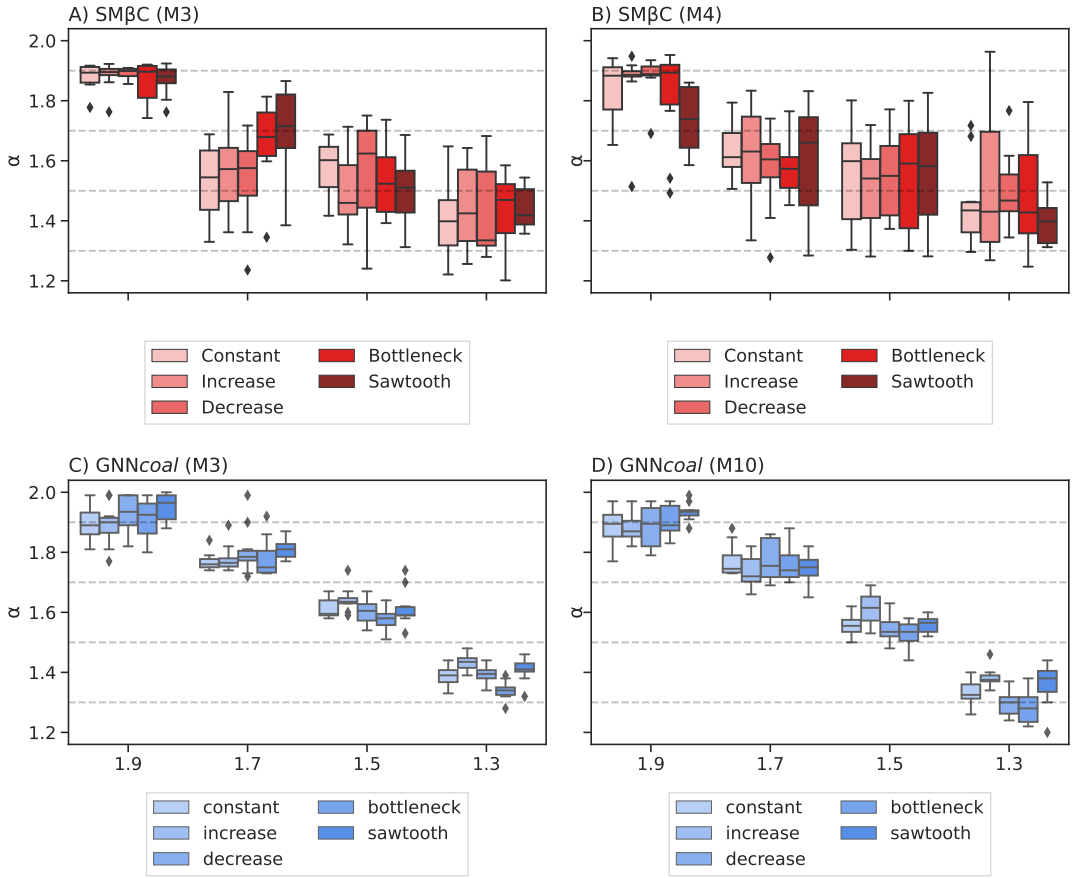

Fig. 5. Estimated values of α by SMβC and GNN*coal* over ten repetitions using 10 sequences of 100 Mb with $\mu = r = 10^{-8}$ per generation per bp under a β-coalescent process (with different α parameter). The analysis are run on five different demographic scenarios (Constant population size, Bottleneck, Sudden increase, Sudden decrease and a Sawtooth demography) using a sample size $n = 3$ for A) and C), $n = 4$ for B), and $n = 10$ for D). Grey dashed lines indicate the true α values.

Although 10 sequences are given to the SMβC in the previous analyses, the method can only analyze three or four simultaneously. However, the GNN*coal* approach can simultaneously analyze 10 sequences, that is the whole simulated ARG. As we observe that GNN*coal* has a higher performance than SMβC we wish to test whether the GNN*coal* better leverages information from the ARG or whether it benefits from simultaneously analyzing bigger sample size. Thus, we run GNN*coal* on the same dataset, but downsampling the coalescent trees to a sample size three. Results are displayed in Figure S4 to

Figure S7. Results with sample size three of the GNN*coal* approach are similar to results with sample size 10, demonstrating that the GNNs can better leverage information from the ARG in presence of multiple merger events (Figure S8).

Additionally, we test if both approaches can recover a Kingman coalescent from the ARG when data are simulated under the Kingman coalescent, namely both approach should recover $\alpha = 2$. To do so, we simulate the same five demographic scenarios as above under a Kingman coalescent and infer the α parameter along with the past variation of population size. Estimations of α values are provided in Table 1 and are systematically higher than 1.85, suggesting mostly binary mergers (*i.e* an underlying Kingman coalescent process). The associated inferred demographies are shown in Figures S9-S13. Both approaches correctly infer the past demographic shape up to the scaling discrepancy between the Beta and the Kingman coalescent mentioned above. Furthermore, we notice that the scaling effect only affects the y-axis for the SMβC but affect both axes for GNN*coal*.

As GNN*coal* was not trained on data simulated under the Kingman coalescent (especially with such high population size), some events fall beyond the scope of the GNN (due to the scaling discrepancy between the Beta and Kingman coalescence). Hence, we run our GNN*coal* on data simulated under the Kingman coalescent but with smaller population size (scaled down by a factor 100) to assure that all events fall within the scope of the GNN. Values of α inferred by the GNN*coal* and the SMβC under the five demographic scenarios are available in Table S3. The associated inference of population size are plotted in Figure S9-S12. Both approaches recover high α values (*i.e.*>1.85) suggesting a genealogy with almost exclusively binary mergers. In addition, both approaches accurately recover the shape of the past variation of population size up to a scaling constant (but only on the population size y-axis).

Inferring α and past demography from simulated sequence data

We first investigate results for both GNNcoal and SMβC when the ARG is reconstructed with ARGweaver [68], which is currently the best ARG inference approach for sample size smaller than 20 [14]. Demographic inference results by both approaches are displayed in Figure S14, and α inference results in Table S4. GNN*coal* does not manage to recover the shape of the demographic history from the inferred ARGs and fails to recover α by highly overestimating it. The α inferences of SMβC (when giving the inferred ARG as input) are better than GNN*coal* and hints toward the presence or absence of multiple merger events although variance is high. SMβC recovers the shape of the past variation of population size for $\alpha > 1.3$ but displays extremely high variance for $\alpha = 1.3$.
Second, we run SMβC on simulated sequence data and found that α is well recovered on average (Table 2) and results are similar to what is obtained when the true ARG is given. Concerning inference of past variation of population size, the shape of the past variation of population size is well recovered under the sawtooth demographic scenario for $\alpha > 1.3$ (Figure S15). In the other four scenarios, the shape of the demography is recovered in recent times, but underestimated in the past (Figure S16). Furthermore, as found above from inputted ARG, the variance generally increases with diminishing α.

Inferring MMC and accounting for selection

As specific reproductive mechanisms and selection can lead to the occurrence of multiple merger-like events, we train our neural network on data simulated under the β-coalescent, and under the Kingman coalescent in presence or absence of selection. We then use the trained GNN*coal* to cluster and determine if multiple merger events originate from skewed offspring distribution or positive selection, or if the data follows a neutral Kingman coalescent process. The classification results are displayed in Figure 6 in the form of a confusion matrix, that is the percentage of times the GNN*coal* correctly assigns the true model shown on the diagonal (evaluated on a test dataset of size 1,000 of known ARGs). Our approach can accurately select the model except in two cases. Our approach shows limited power to distinguish strong from average selection under the Kingman coalescent, as well as to distinguish the β-coalescent with a small amount of multiple merger events (*i.e.* $\alpha>1.75$) from the β-coalescent case with $1.75>\alpha>1.5$.

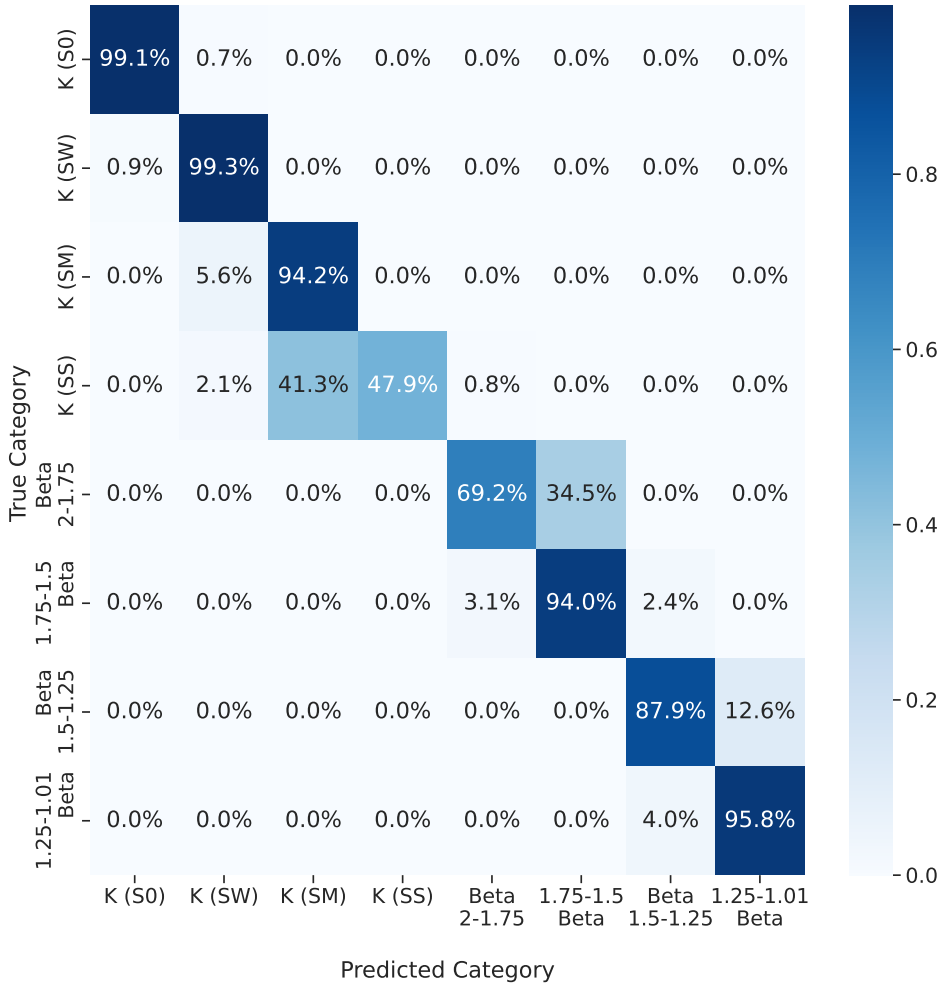

Fig. 6. Confusion matrix for Kingman and β-coalescent classification model under varying selection coefficients. Evaluation of classification accuracy for Kingman (K) and β-coalescent (B) for no selection (S0), weak selection (SW), medium selection (SM) and strong selection (SS) using a 1,000 repetition validation dataset. Population size was kept constant at $N = 10^5$ individuals, using a sample size $n = 10$ and $r = 10^{-8}$ per bp per generation.

To assess the effect of selection, we infer the α along the genome with both approaches from data simulated with strong positive selection or neutrality under a Kingman coalescent with population size being constant through time. SMβC infers α on windows of 10kbp along the genome, and GNN*coal* infers α every 20 trees along the genome. Results for the GNN*coal* approach and for the SMβC are displayed in Figure 7. Both approaches recover smaller α value around the locus under strong selection. However under neutrality or weak selection, inferred α values remain high ($>$1.6) along the genome.

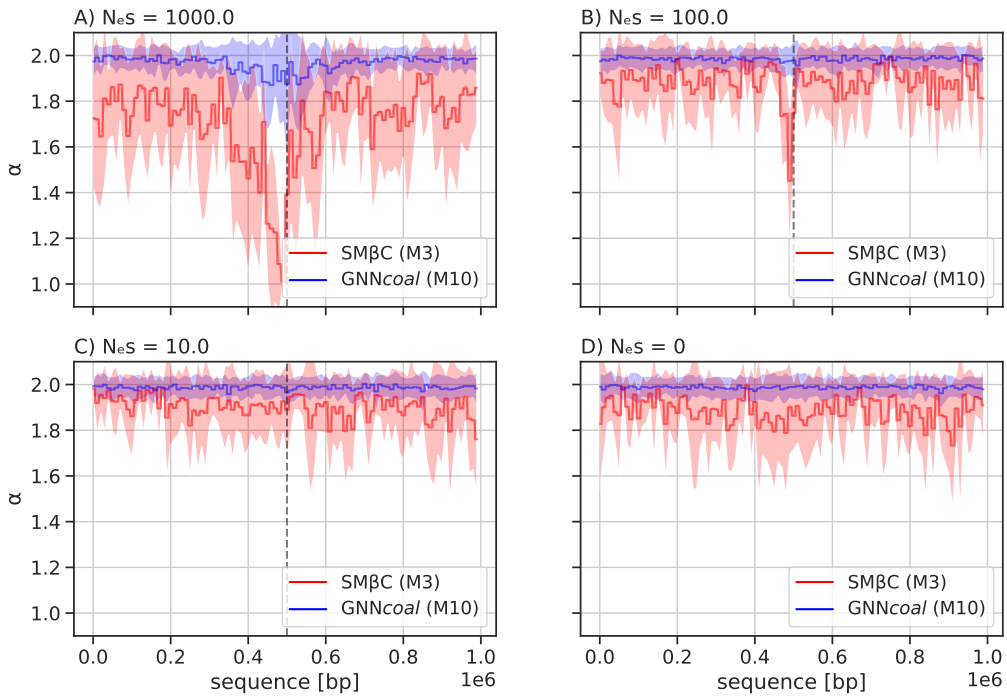

Fig. 7. Averaged estimations by GNN*coal* and SMβC under selection Estimations of α along the genome by the GNN*coal* approach and the SMβC when population undergoes as strong positive selective sweep event (at position 0.5 Mb) under different strengths of selection: A) $s = 0.01$, B)$s = 0.001$, C) $s = 0.0001$, and D) $s = 0$ meaning neutrality (mean and standard deviation for both methods). The population size is constant and set to $N = 10^5$ with $\mu = r = 10^{-8}$ per generation per bp. We hence have in A) $N_e \times s = 1000$,B) $N_e \times s = 100$, C) $N_e \times s = 10$ and D) $N_e \times s = 0$. SMβC uses 20 sequences of 1Mb (red) and GNN*coal* uses 10 sequences through down-sampling the sample nodes (blue)

Similarly, we run both approaches on data simulated under the β-coalescent (assuming neutrality) and we infer the α value along the genome. Inferred α values by both approaches are plotted in Figure 17 in S1. GNN*coal* is able to recover the α value along the genome (also with overestimation due to tree sparsity), but the SMβC systematically underestimate these α values. However, unlike in presence of positive selection at a given locus, the inferred α values are found in all cases to be fairly similar along the genome.

We finally simulate data under a strong selective sweeps or under neutrality and under a sawtooth demographic scenario. Under neutrality, our approach recovers high α value along the genome and can accurately recover the past variation of population size (only up to a scaling constant for GNN*coal*, since it was only trained on the β-coalescent) (Figure 8). Similarly, when the simulated data contains strong selection, a small α value is recovered at the locus under selection and the past variation of population size is accurately recovered, albeit with a small underestimation of population size in recent times for SMβC (Figure 8).

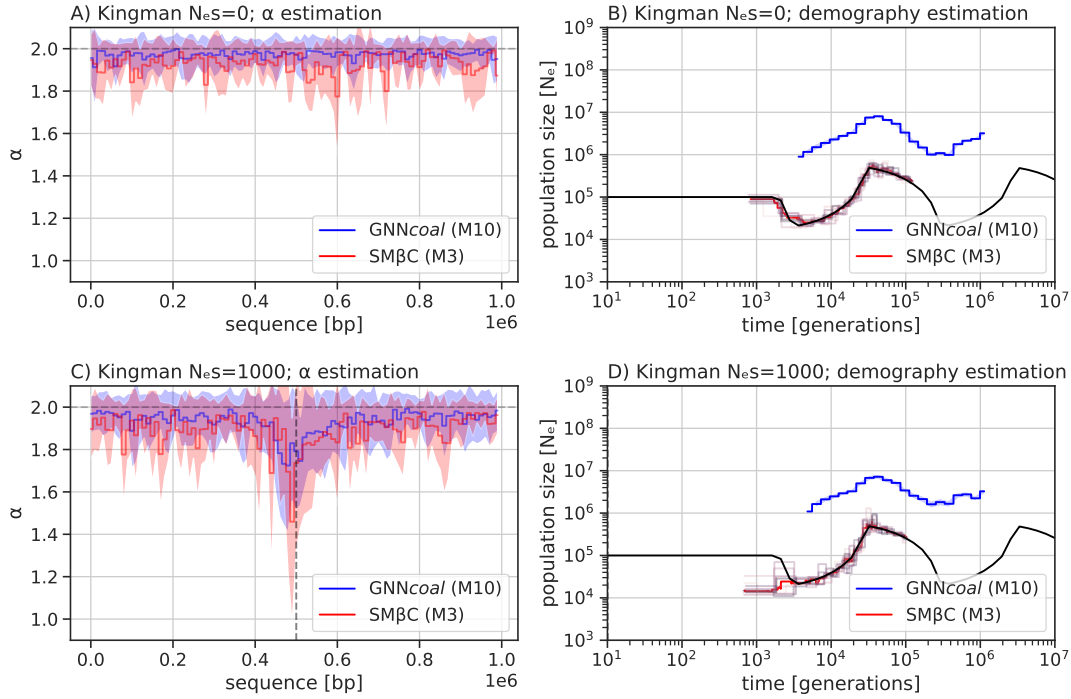

Fig. 8. Simultaneous estimations of α along the sequence under demographic change by GNN*coal* and SMβC. Simultaneous estimation of α along the genome under a partial sawtooth scenario: A) and B) in the absence of selection (mean and standard deviation for both methods), and C) and D) presence of selection with $N_eS = 1,000$ (mean and CI95 for GNN*coal* and median for SMβC). SMβC uses 20 sequences of 1Mb (red) and GNN*coal* uses 10 sequences through down-sampling the sample nodes (blue), and $\mu = r = 10^{-8}$ per generation per bp.

Discussion

With the rise of SMC approaches [53], most current methods leverage information from whole genome sequences by simultaneously reconstructing a portion of the ARG in order to infer past demographic history [53, 77, 88, 89], migration rates [46, 90], variation in recombination and mutation along the genome [5, 4], as well as ecological life history traits such as selfing or seed banking [80, 86]. However in previous studies authors discussed uncoupling both steps, namely reconstructing the ARG and then inferring parameters from its distribution [81, 31, 68]. Indeed, recent efforts have been made to improve approaches recovering the ARG [83, 45, 36, 68, 54, 14], as well as its interpretation [30, 81]. Our results on data simulated under the β-coalescent clearly show the strong effect of multiple merger events on the topology and branch length of the ARG. We find that the more multiple merger events occur, the more information concerning the past demography is lost. Both methods, whether given sequence data, the true or inferred ARG, can recover the α parameter and the variation of past population size for α values high enough (*i.e.* $\alpha > 1.3$), however for lower values of α a larger amount of data is necessary for any inference (which becomes nearly impossible when α tends to one). Both approaches can also recover the Kingman coalescent (*i.e.* α>1.8). We find the GNN*coal* approach outperforms the SMβC approach in almost all cases when given the true ARG, and one can use GNN*coal* to disentangle (classify) between models of β-coalescent and

Kingman models with selection.

Overall, our results provide a qualitative increase in the developments of inference methods for models with multiple merger events, a key step to understand the underlying reproduction mechanism of a species. We first develop a new SMC method which takes simultaneously three or four sequences to draw inference of the coalescent tree and transitions along the genome. While fairly accurate, SMβC is outperformed by GNN*coal* when given true ARGs as input. As we directly compare our theoretical SMC to the GNN based on the same input data (coalescent trees), we are ideally placed to dissect the mechanisms underlying the power of the GNN*coal* method. We identify four main reasons for the difference in accuracy between both methods in this specific case. First, the SMβC approach suffers from the limit of the sequential Markovian coalescent hypothesis along the genome when dealing with strong multiple merger events [8, 20]. Second, most current SMC approach rely on a discretization of the coalescent times into discrete hidden states, meaning that simultaneous mergers of three lineages may not be easily distinguished from two consecutive binary mergers occurring over a short period of time. Third, the SMβC relies on a complex hidden Markov model and due to computational and mathematical tractability, it cannot leverage information on a whole genealogy (as MSMC, SMβC only focuses on the first coalescent event) and cannot simultaneously analyze large sample size. Furthermore, the SMβC approach leverages information from the distribution of genealogies along the genome. It means that in the near absence of recombination events, both approaches cannot utilize any information from the genealogy itself. However, the GNN*coal* approach can overcome this by increasing sample size. Fourth, the SMβC is based on a coalescent model where α is constant in time. Yet multiple merger events do not appear regularly across the genealogical timescale, but occur at few random time points. Hence, the SMC approach suffers from a strong identifiability problem between the variation of population size and the α parameter (for low α values). As for example, if during one hidden state one strong multiple merger event occurs, multiple merger events are seldom observed and SMβC may rather assume a small population size at this time point (hidden state). This may explain the high variance of inferred population sizes under the β-coalescent.

By contrast, the GNN*coal* approach makes use of the whole ARG, and can easily scale to big sample size (>10), although it recovers α with high accuracy with sample size 3 only. Our interpretation is thus that the GNN*coal* is able of simultaneously leveraging information from topology and the age of coalescent events (nodes) across several genealogies (here 500). The GNN*coal* approach ultimately leverages information from observing recurrent occurrences of the same multiple merger events at different locations on the genome, while being aware of true multiple merger events from rapid successive binary mergers. We believe that our results pave the way towards the interpretability of GNN and DL methods applied to population genetics.

When applying both approaches to simulated sequence data (and not to true ARGs), both approaches behave differently. The GNN*coal* is not capable to accurately infer model parameter whether it is past variation of population size or α. In contrast, we find the SMβC to perform better than the GNN*coal* when dealing with sequence data (and not true ARG). SMβC is capable of recovering α and the shape of the demographic scenario in recent times irrespective of whether sequence data or ARG inferred by ARGweaver is

given as input. This is most likely due to the fact that the statistic used by the SMβC (*i.e.* first coalescent event in discrete time) is more coarse than the statistic used by the GNN*coal* (*i.e.* the exact ARG). We therefore speculate that the theoretical framework of the SMβC, although being in theory less accurate than GNN*coal*, is more robust and suited for application to sequence data. More specifically the issue being faced by the GNN*coal* is known as out-of-distribution inference [38], which requires the network to generalize over an untrained data distribution. More precisely, this happens because GNN*coal* is not trained using ARG inferred by ARGweaver. Building a training data set for GNN*coal* to overcome this issue is currently impractical due to the inference speed of ARGweaver (very slow). However, future work will aim on increasing robustness of GNN inferences (*e.g.* by adding uncertainty or multiple models during the training process). Improving the performance of GNN*coal* on sequence data requires a more suitable ARG inference methods (*i.e.* efficient and accurate), which can be used on a broad spectrum of data sets (*i.e.* under a more general class of model to avoid bias from potential hypothesis violations of the chosen ARG inference approach).

Among the major forces driving the genome evolution there are the past demographic history, the reproductive mechanisms and natural selection [40]. Hence, in the second part of this manuscript we focus on integrating selection in both approaches. Currently, no method (especially if relying only on SFS information) can account for the presence of selection, linkage, non-constant population size and multiple merger events [40] although recent theoretical framework might render this possible in the future [1]. As a first step to fill this gap, we demonstrate that the GNN*coal* approach can be used for model selection (classification), reducing the number of potential hypotheses. Determining which evolutionary forces are driving the genome evolution is key, as only under the appropriate neutral population model can results of past demography and selection scans be correctly interpreted [40, 42]. The high accuracy of the GNN*coal* approach in model selection is promising, especially as other methods based on the SFS alone [51, 43] have limits in presence of complex demographic scenarios which the GNN can possibly overcome, as it is easier to scale the GNN to estimate more parameters. We follow a thread of previous work [71, 35, 10], by integrating and recovering selection, multiple merger and population size variation by simply allowing each (fixed) region in the genome to have its own α parameter. In presence of strong selection, we find lower α value around the selected loci and high α value in neutral neighbouring regions. In presence of weak selection, no effect on the estimated α value is observed, demonstrating that weak selection can be modeled by binary merger (no multiple merger) and only has a local effect on the branch length (*i.e.* by shortening it). Hence, our results point out that strong selection can indeed be seen/modeled as a local multiple merger event (see [23, 10, 71]). In theory, both approaches could be able to infer the global α parameter linked to the reproductive mechanism, the local α parameter resulting from selection jointly with the variation of population size. However the absence of a simulator capable of simulating data with selection and non-constant population size under a β-coalescent model prevents us from delivering such proofs. We show strong evidence that under neutrality our approaches can recover a constant (and correct) α along the genome as well as the past variation of the population size. We further predict, that while selective processes favor coding regions, local variations in α as a consequence of sweepstake events should be indifferent to coding or non-coding regions. Hence, we suggest that current sequence simulators [7, 32] could be extended to include the aforementioned factors and *de facto* facilitate the

development of machine learning approaches.

Our study is unique in developing a new state of the art SMC approach to demonstrate that computational and mathematical problems can be overcome by DL (here GNN) approaches. The GNN*coal* approach is, in principle, not limited to the β-coalescent, and should work for other multiple merger models (*e.g.* Dirac coalescents [24]) with the appropriate training. Furthermore, our SMβC approach is the first step to build a full genome method with an underlying model accounting for positive selection. In the future, further implementations may be added for a more realistic approach. The α parameter should be varying along the genome (as a hidden state), as the recombination rate in the iSMC [5]. This would allow to account for the local effect of strong and weak selection [1]. The effect of the α parameter could be also changing through time to better model the non uniform occurrence of multiple merger events through time. Although it is mathematically correct to have α as a constant in time, it is erroneous in practice (Figure 2 in S1). We speculate that those additional features will allow to accurately model and infer multiple merger events, variation of population size, and selection at each position on the genome. We believe the DL approach could also be improved to recover more complex scenarios, providing in depth development on the structure of the graph neural networks, for example, by accounting for more features. At last, further investigation are required to make progress in the interpretability of the GNN methods, namely which statistics and convolution of statistics are used by GNN*coal* to infer which parameters.

Currently, the applications of our methods are constrained by the assumptions of haploidy. Throughout the study we have assumed haploid data, but the GNN*coal* and SMβC approach can easily be extended to diploidy [8]. Our results on inferred ARGs stress the need for efforts to be made on improving ARG inference [14]. Thanks to the SMC we are close to model the ARG allowing to infer demographic history, selection and specific reproductive mechanism. Moreover, the comparison of DL approaches with model driven *ad hoc* SMC methods may have the potential to help us solve issues (*e.g.* simultaneously inferring and accounting for recombination, variation of population size, different type of selection, population structure and the variation of the mutation and recombination rate along the genome) which have puzzled theoreticians and statisticians since the dawn of population genetics [40].

On a final note, as environmental considerations hit us all, we suggest that decreasing the computer and power resources needed to perform DL/ GNN analyses should be attempted [75]. Based on our study, we suggest that population genetics DL methods could be built as a two step process: 1) inferring ARGs, and 2) inferring demography and selection based on the ARGs. We speculate that general training sets based on ARGs could be build and be widely applicable for inference across many species with different life cycles and life history traits, while the inference of ARGs could be undertaken by complementary DL or Hidden Markov methods.

Tables

scenario	True α	α:SMβC,M=3	α:SMβC,M=4	α : GNN, M=3	α : GNN, M=10
Constant	2	1.97 (0.005)	1.97 (0.008)	1.99 (0.002)	1.99 (0.003)
Sawtooth	2	1.94 (0.017)	1.87 (0.019)	1.99 (0.002)	1.99 (0.003)
Bottleneck	2	1.97 (0.01)	1.97 (0.009)	1.99 (0.003)	1.99 (0.004)
Decrease	2	1.97 (0.007)	1.97 (0.008)	1.99 (0.003)	1.99 (0.004)
Increase	2	1.97 (0.007)	1.97 (0.008)	1.99 (0.004)	1.99 (0.002)

Table 1: Average estimated values of α by SMβC and GNN*coal* over ten repetitions under the Kingman coalescent using 10 haploid sequences of 10 Mb and $\mu = r = 10^{-8}$ per generation per bp. The standard deviation is indicated in brackets.

scenario	True α	α^*:SMβC,M=3
Constant	1.9	1.86 (0.16)
Bottleneck	1.9	1.89 (0.09)
Increase	1.9	1.93 (0.07)
Decrease	1.9	1.96 (0.04)
Sawtooth	1.9	1.76 (0.17)
Constant	1.7	1.82 (0.10)
Bottleneck	1.7	1.64 (0.23)
Increase	1.7	1.82 (0.10)
Decrease	1.7	1.89 (0.13)
Sawtooth	1.7	1.71 (0.27)
Constant	1.5	1.52 (0.30)
Bottleneck	1.5	1.64 (0.33)
Increase	1.5	1.57 (0.24)
Decrease	1.5	1.60 (0.18)
Sawtooth	1.5	1.66 (0.14)
Constant	1.3	1.31 (0.20)
Bottleneck	1.3	1.2 (0.17)
Increase	1.3	1.24 (0.13)
Decrease	1.3	1.57 (0.11)
Sawtooth	1.3	1.37 (0.16)

Table 2: Average estimated α values by SMβC on simulated sequence data over ten repetitions using 10 sequences of 10 Mb with recombination and mutation rate set to 1×10^{-8} for α 1.9 and 1.7, 1×10^{-7} for α 1.5 and 1×10^{-6} for α 1.3 per generation per bp under a Beta coalescent process. The analysis are run on five different demographic scenarios (Constant population size, Bottleneck, Sudden increase, Sudden decrease and a Sawtooth demography).

Acknowledgments

This work was supported by the BMBF-funded de.NBI Cloud within the German Network for Bioinformatics Infrastructure (de.NBI) (031A532B, 031A533A, 031A533B, 031A534A, 031A535A, 031A537A, 031A537B, 031A537C, 031A537D, 031A538A). KK is supported by a grant from the Deutsche Forschungsgemeinschaft (DFG) through the TUM International Graduate School of Science and Engineering (IGSSE), GSC 81, within the project GENOMIE QADOP. TS is supported by the Austrian Science Fund (project no. TAI 151-B). AT acknowledges funding from the DFG grant TE809/1-4 (project 254587930) and TE809/7-1 (project 317616126). FF and AT acknowledge funding from the DFG Priority Program SPP1590 on "Probabilistic Structures in Evolution". MF and AT acknowledge the support from the Imperial College - TUM Partnership award.

References

[1] Frederic Alberti, Carolin Herrmann, and Ellen Baake. Selection, recombination, and the ancestral initiation graph. *THEORETICAL POPULATION BIOLOGY*, 142:46–56, DEC 2021.

[2] Einar Arnason and Katrin Halldorsdottir. Nucleotide variation and balancing selection at the Ckma gene in Atlantic cod: analysis with multiple merger coalescent models. *PEERJ*, 3, FEB 24 2015.

[3] Einar Árnason, Jere Koskela, Katrín Halldórsdóttir, and Bjarki Eldon. Sweepstakes reproductive success via pervasive and recurrent selective sweeps. *Elife*, 12:e80781, 2023.

[4] Gustavo V. Barroso and Julien Y. Dutheil. Mutation rate variation shapes genome-wide diversity in *Drosophila melanogaster*. preprint, Evolutionary Biology, September 2021.

[5] Gustavo V. Barroso, Natasa Puzovic, and Julien Y. Dutheil. Inference of recombination maps from a single pair of genomes and its application to ancient samples. *PLOS Genetics*, 15(11), NOV 2019.

[6] CJ Battey, Peter L Ralph, and Andrew D Kern. Predicting geographic location from genetic variation with deep neural networks. *eLife*, 9:e54507, June 2020.

[7] Franz Baumdicker, Gertjan Bisschop, Daniel Goldstein, Graham Gower, Aaron P. Ragsdale, Georgia Tsambos, Sha Zhu, Bjarki Eldon, E. Castedo Ellerman, Jared G. Galloway, Ariella L. Gladstein, Gregor Gorjanc, Bing Guo, Ben Jeffery, Warren W. Kretzschumar, Konrad Lohse, Michael Matschiner, Dominic Nelson, Nathaniel S. Pope, Consuelo D. Quinto-Cortes, Murillo F. Rodrigues, Kumar Saunack, Thibaut Sellinger, Kevin Thornton, Hugo van Kemenade, Anthony W. Wohns, Yan Wong, Simon Gravel, Andrew D. Kern, Jere Koskela, Peter L. Ralph, and Jerome Kelleher. Efficient ancestry and mutation simulation with msprime 1.0. *GENETICS*, 220(3), MAR 3 2022.

[8] Matthias Birkner, Jochen Blath, and Bjarki Eldon. An Ancestral Recombination Graph for Diploid Populations with Skewed Offspring Distribution. *Genetics*, 193(1):255–290, JAN 2013.

[9] Matthias Birkner, Jochen Blath, Martin Moehle, Matthias Steinruecken, and Johanna Tams. A modified lookdown construction for the Xi-Fleming-Viot process with mutation and populations with recurrent bottlenecks. *arXiv:0808.0412*, 2008.

[10] Gertjan Bisschop, Konrad Lohse, and Derek Setter. Sweeps in time: leveraging the joint distribution of branch lengths. *GENETICS*, 219(2), OCT 2021.

[11] Jochen Blath, Adrian Gonzalez Casanova, Noemi Kurt, and Maite Wilke-Berenguer. The seed bank coalescent with simultaneous switching. *Electronic Journal of Probability*, 25, 2020.

[12] Simon Boitard, Willy Rodríguez, Flora Jay, Stefano Mona, and Frédéric Austerlitz. Inferring population size history from large samples of genome-wide molecular data - an approximate bayesian computation approach. 12(3):e1005877.

[13] Erwin Bolthausen and A-S Sznitman. On ruelle's probability cascades and an abstract cavity method. *Communications in mathematical physics*, 197(2):247–276, 1998.

[14] Debora Y. C. Brandt, Xinzhu Wei, Yun Deng, Andrew H. Vaughn, and Rasmus Nielsen. Evaluation of methods for estimating coalescence times using ancestral recombination graphs. *GENETICS*, 221(1), MAY 5 2022.

[15] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, jul 2017.

[16] E. Brunet, B. Derrida, A. H. Mueller, and S. Munier. Noisy traveling waves: Effect of selection on genealogies. *Europhysics Letters*, 76(1):1–7, OCT 2006.

[17] E. Brunet, B. Derrida, A. H. Mueller, and S. Munier. Effect of selection on ancestry: An exactly soluble case and its phenomenological generalization. *Physical Review E*, 76(4, 1), OCT 2007.

[18] Klara Elisabeth Burger, Peter Pfaffelhuber, and Franz Baumdicker. Neural networks for self-adjusting mutation rate estimation when the recombination rate is unknown. *PLOS Computational Biology*, 18(8):1–17, 08 2022.

[19] Wenming Cao, Zhiyue Yan, Zhiquan He, and Zhihai He. A comprehensive survey on geometric deep learning. *IEEE Access*, 8:35929–35949, 2020.

[20] Adrián González Casanova, Verónica Miró Pina, and Arno Siri-Jégousse. The Symmetric Coalescent and Wright-Fisher models with bottlenecks. *arXiv:1903.05642 [math]*, September 2020. arXiv: 1903.05642.

[21] Jianhai Chen, Pan Ni, Xinyun Li, Jianlin Han, Ivan Jakovlic, Chengjun Zhang, and Shuhong Zhao. Population size may shape the accumulation of functional mutations following domestication. *BMC Evolutionary Biology*, 18, JAN 19 2018.

[22] P Donnelly and TG Kurtz. Particle representations for measure-valued population models. *Annals of Probability*, 27(1):166–205, JAN 1999.

[23] R Durrett and J Schweinsberg. A coalescent model for the effect of advantageous mutations on the genealogy of a population. *Stochastic Processes and their Applications*, 115(10):1628–1657, OCT 2005.

[24] B Eldon and J Wakeley. Coalescent processes when the distribution of offspring number among individuals is highly skewed. *Genetics*, 172(4):2621–2633, APR 2006.

[25] Bjarki Eldon, Matthias Birkner, Jochen Blath, and Fabian Freund. Can the Site-Frequency Spectrum Distinguish Exponential Population Growth from Multiple-Merger Coalescents? *Genetics*, 199(3):841+, MAR 2015.

[26] Malaspinas et al. A genomic history of Aboriginal Australia. *Nature*, 538(7624):207+, OCT 13 2016.

[27] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with PyTorch geometric.

[28] Lex Flagel, Yaniv Brandvain, and Daniel R Schrider. The Unreasonable Effectiveness of Convolutional Neural Networks in Population Genetic Inference. *Molecular Biology and Evolution*, 36(2):220–238, 12 2018.

[29] Fabian Freund, Elise Kerdoncuff, Sebastian Matuszewski, Marguerite Lapierre, Marcel Hildebrandt, Jeffrey D. Jensen, Luca Ferretti, Amaury Lambert, Timothy B. Sackton, and Guillaume Achaz. Interpreting the pervasive observation of U-shaped Site Frequency Spectra. preprint, Evolutionary Biology, April 2022.

[30] L. M. Gattepaille, M. Jakobsson, and M. G. B. Blum. Inferring population size changes with sequence and SNP data: lessons from human bottlenecks. *Heredity*, 110(5):409–419, MAY 2013.

[31] Lucie Gattepaille, Torsten Guenther, and Mattias Jakobsson. Inferring Past Effective Population Size from Distributions of Coalescent Times. *Molecular Biology and Evolution*, 204(3):1191+, NOV 2016.

[32] Benjamin C. Haller, Jared Galloway, Jerome Kelleher, Philipp W. Messer, and Peter L. Ralph. Tree-sequence recording in slim opens new horizons for forward-time simulation of whole genomes. *MOLECULAR ECOLOGY RESOURCES*, 19(2):552–566, MAR 2019.

[33] Rebecca B. Harris and Jeffrey D. Jensen. Considering genomic scans for selection as coalescent model choice. *GENOME BIOLOGY AND EVOLUTION*, 12(6):871–877, JUN 2020.

[34] Dennis Hedgecock and Alexander I. Pudovkin. Sweepstakes reproductive success in highly fecund marine fish and shellfish: a review and Commentary. *Bulletin of Marine Science*, 87(4):971–1002, OCT 2011.

[35] Hussein A. Hejase, Ziyi Mo, Leonardo Campagna, and Adam Siepel. A deep-learning approach for inference of selective sweeps from the ancestral recombination graph. *MOLECULAR BIOLOGY AND EVOLUTION*, 39(1), JAN 7 2022.

[36] Melissa Hubisz and Adam Siepel. Inference of ancestral recombination graphs using argweaver. In JY Dutheil, editor, *STATISTICAL POPULATION GENOMICS*, volume 2090 of *Methods in Molecular Biology*, pages 231–266. 2020.

[37] RR HUDSON. Properties of a neutral allele model with intragenic recombination. *Theoretical Population Biology*, 23(2):183–201, 1983.

[38] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Machine Learning*, 110(3):457–506, March 2021.

[39] Ulas Isildak, Alessandro Stella, and Matteo Fumagalli. Distinguishing between recent balancing selection and incomplete sweep using deep neural networks. *Molecular Ecology Resources*, 21(8):2706–2718, November 2021.

[40] Parul Johri, Charles F. Aquadro, Mark Beaumont, Brian Charlesworth, Laurent Excoffier, Adam Eyre-Walker, Peter D. Keightley, Michael Lynch, Gil McVean, Bret A.

Payseur, Susanne P. Pfeifer, Wolfgang Stephan, and Jeffrey D. Jensen. Recommendations for improving statistical inference in population genomics. *PLOS Biology*, 20(5):e3001669, May 2022.

[41] Parul Johri, Brian Charlesworth, and Jeffrey D. Jensen. Toward an evolutionarily appropriate null model: Jointly inferring demography and purifying selection. *GENETICS*, 215(1):173–192, MAY 2020.

[42] Parul Johri, Kellen Riall, Hannes Becher, Laurent Excoffier, Brian Charlesworth, and Jeffrey D. Jensen. The impact of purifying and background selection on the inference of population history: Problems and prospects. *MOLECULAR BIOLOGY AND EVOLUTION*, 38(7):2986–3003, JUL 2021.

[43] Mamoru Kato, Daniel A. Vasco, Ryuichi Sugino, Daichi Narushima, and Alexander Krasnitz. Sweepstake evolution revealed by population-genetic analysis of copy-number alterations in single genomes of breast cancer. *Royal Society of Open Science*, 4(9), SEP 2017.

[44] Jerome Kelleher, Kevin R. Thornton, Jaime Ashander, and Peter L. Ralph. Efficient pedigree recording for fast population genetics simulation. 14(11):e1006581.

[45] Jerome Kelleher, Yan Wong, Anthony W. Wohns, Chaimaa Fadil, Patrick K. Albers, and Gil McVean. Inferring whole-genome histories in large population datasets (vol 51, pg 1330, 2019). *Nature Genetics*, 51(11):1660, NOV 2019.

[46] Younhun Kim, Frederic Koehler, Ankur Moitra, Elchanan Mossel, and Govind Ramnarayan. How Many Subpopulations Is Too Many? Exponential Lower Bounds for Inferring Population Histories. *Journal of Computational Biology*, 27(4):613–625, APR 1 2020.

[47] JFC Kingman. The Coalescent . *Stochastic Processes and their Applications*, 13, 1982.

[48] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. 2016.

[49] Kevin Korfmann, Oscar E Gaggiotti, and Matteo Fumagalli. Deep Learning in Population Genetics. *Genome Biology and Evolution*, 15(2):evad008, February 2023.

[50] Jere Koskela. Multi-locus data distinguishes between population growth and multiple merger coalescents. *STATISTICAL APPLICATIONS IN GENETICS AND MOLECULAR BIOLOGY*, 17(3), JUN 2018.

[51] Jere Koskela and Maite Wilke Berenguer. Robust model selection between population growth and multiple merger coalescents. *Mathematical Biosciences*, 311:1–12, MAY 2019.

[52] John Boaz Lee, Ryan Rossi, and Xiangnan Kong. Graph Classification using Structural Attention. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1666–1674, London United Kingdom, July 2018. ACM.

[53] Heng Li and Richard Durbin. Inference of human population history from individual whole-genome sequences. *Nature*, 475(7357):493–U84, JUL 28 2011.

[54] Ali Mahmoudi, Jere Koskela, Jerome Kelleher, Yao-ban Chan, and David Balding. Bayesian inference of ancestral recombination graphs. *PLOS COMPUTATIONAL BIOLOGY*, 18(3), MAR 2022.

[55] P Marjoram and JD Wall. Fast "coalescent" simulation. *BMC Genetics*, 7, MAR 15 2006.

[56] Sebastian Matuszewski, Marcel E. Hildebrandt, Guillaume Achaz, and Jeffrey D. Jensen. Coalescent processes with skewed offspring distributions and non-equilibrium demography. *Genetics*, 2017.

[57] GAT McVean and NJ Cardin. Approximating the coalescent with recombination. *Philosophical Transactions of the Royal Society B-Biological Sciences*, 360(1459):1387–1393, JUL 29 2005.

[58] F Menardo, S Gagneux, and F Freund. Multiple merger genealogies in outbreaks of Mycobacterium tuberculosis. *Molecular Biology and Evolution*, 07 2020. msaa179.

[59] Alistair Miles, pyup io bot, Murillo R, Peter Ralph, Nick Harding, Rahul Pisupati, Summer Rae, and Tim Millar. cggh/scikit-allel: v1.3.3.

[60] M Mohle and S Sagitov. A classification of coalescent processes for haploid exchangeable population models. *Annals of Probability*, 29(4):1547–1562, OCT 2001.

[61] Ana Y. Morales-Arce, Rebecca B. Harris, Anne C. Stone, and Jeffrey D. Jensen. Evaluating the contributions of purifying selection and progeny-skew in dictating within-host Mycobacterium tuberculosis evolution. *Evolution*, 74(5):992–1001, MAY 2020.

[62] Richard A. Neher and Oskar Hallatschek. Genealogies of rapidly adapting populations. *Proceedings of the National Academy of Sciences*, 110(2):437–442, January 2013.

[63] Dominic Nelson, Jerome Kelleher, Aaron P. Ragsdale, Claudia Moreau, Gil McVean, and Simon Gravel. Accounting for long-range correlations in genome-wide simulations of large cohorts. *PLOS Genetics*, 16(5), MAY 2020.

[64] Hiro-Sato Niwa, Kazuya Nashida, and Takashi Yanagimoto. Reproductive skew in japanese sardine inferred from dna sequences. *ICES Journal of Marine Science*, 73(9):2181–2189, 2016.

[65] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. October 2017.

[66] J Pitman. Coalescents with multiple collisions. *Annals of Probability*, 27(4):1870–1902, OCT 1999.

[67] Xinghu Qin, Charleston W. K. Chiang, and Oscar E. Gaggiotti. Deciphering signatures of natural selection via deep learning. *bioRxiv*, 2021.

[68] Matthew D. Rasmussen, Melissa J. Hubisz, Ilan Gronau, and Adam Siepel. Genome-wide inference of ancestral recombination graphs. *PLOS GENETICS*, 10(5), MAY 2014.

[69] Daniel P Rice, John Novembre, and Michael M Desai. Distinguishing multiple-merger from kingman coalescence using two-site frequency spectra. *bioRxiv*, 2018.

[70] Alan R. Rogers and Chad Huff. Linkage disequilibrium between loci with unknown phase. 182(3):839–844.

[71] Andrew M. Sackman, Rebecca B. Harris, and Jeffrey D. Jensen. Inferring demography and selection in organisms characterized by skewed offspring distributions. *GENETICS*, 211(3):1019–1028, MAR 2019.

[72] S Sagitov. The general coalescent with asynchronous mergers of ancestral lines. *Journal of Applied Probability*, 36(4):1116–1125, DEC 1999.

[73] S Sagitov. Convergence to the coalescent with simultaneous multiple mergers. *Journal of Applied Probability*, 40(4):839–854, DEC 2003.

[74] Théophile Sanchez, Jean Cury, Guillaume Charpiat, and Flora Jay. Deep learning for population size history inference: Design, comparison and combination with approximate bayesian computation. 21(8):2645–2660.

[75] Nicolae Sapoval, Amirali Aghazadeh, Michael G. Nute, Dinler A. Antunes, Advait Balaji, Richard Baraniuk, C. J. Barberan, Ruth Dannenfelser, Chen Dun, Mohammadamin Edrisi, R. A. Leo Elworth, Bryce Kille, Anastasios Kyrillidis, Luay Nakhleh, Cameron R. Wolfe, Zhi Yan, Vicky Yao, and Todd J. Treangen. Current progress and open challenges for applying deep learning across the biosciences. *Nature Communications*, 13(1):1728, December 2022.

[76] Ori Sargsyan and John Wakeley. A coalescent process with simultaneous multiple mergers for approximating the gene genealogies of many marine organisms. *Theoretical population biology*, 74(1):104–114, 2008.

[77] Stephan Schiffels and Richard Durbin. Inferring human population size and separation history from multiple genome sequences. *Nature Genetics*, 46(8):919–925, AUG 2014.

[78] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks, 2017.

[79] J Schweinsberg. Coalescent processes obtained from supercritical Galton-Watson processes. *Stochastic Processes and their Applications*, 106(1):107–139, JUL 2003.

[80] Thibaut Paul Patrick Sellinger, Diala Abu Awad, Markus Moest, and Aurelien Tellier. Inference of past demography, dormancy and self-fertilization rates from whole genome sequence data. *PLOS Genetics*, 16(4), APR 2020.

[81] Thibaut Paul Patrick Sellinger, Diala Abu-Awad, and Aurelien Tellier. Limits and convergence properties of the sequentially markovian coalescent. *MOLECULAR ECOLOGY RESOURCES*, 21(7):2231–2248, OCT 2021.

[82] Sara Sheehan and Yun S. Song. Deep Learning for Population Genetic Inference. *PLOS Computational Biology*, 12(3), MAR 2016.

[83] Leo Speidel, Marie Forest, Sinan Shi, and Simon R. Myers. A method for genome-wide genealogy estimation for thousands of samples. *Nature Genetics*, 51(9):1321+, SEP 2019.

[84] Matthias Steinruecken, Matthias Birkner, and Jochen Blath. Analysis of DNA sequence variation within marine species using Beta-coalescents. *Theoretical Population Biology*, 87:15–24, AUG 2013.

[85] Wolfgang Stephan. Selective Sweeps. *Genetics*, 211(1):5–13, January 2019.

[86] Stefan Struett, Thibaut Sellinger, Sylvain Glémin, Aurélien Tellier, and Stefan Laurent. Inference of evolutionary transitions to self-fertilization using whole-genome sequences. *bioRxiv*, 2022.

[87] Aurelien Tellier and Christophe Lemaire. Coalescence 2.0: a multiple branching of recent theoretical developments and their applications. *Molecular Ecology*, 23(11):2637–2652, JUN 2014.

[88] Jonathan Terhorst, John A. Kamm, and Yun S. Song. Robust and scalable inference of population history froth hundreds of unphased whole genomes. *Nature Genetics*, 49(2):303–309, FEB 2017.

[89] Gautam Upadhya and Matthias Steinrücken. Robust Inference of Population Size Histories from Genomic Sequencing Data. preprint, Genetics, May 2021.

[90] Ke Wang, Iain Mathieson, Jared O'Connell, and Stephan Schiffels. Tracking human population structure through time from whole genome sequences. *PLOS Genetics*, 16(3), MAR 2020.

[91] Zhanpeng Wang, Jiaping Wang, Michael Kourakos, Nhung Hoang, Hyong Hark Lee, Iain Mathieson, and Sara Mathieson. Automatic inference of demographic parameters using generative adversarial networks. *Molecular Ecology Resources*, 21(8):2689–2705, 2021.

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

[93] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

[94] Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. *CoRR*, abs/1603.08861, 2016.

[95] Burak Yelmen, Aurélien Decelle, Linda Ongaro, Davide Marnetto, Corentin Tallec, Francesco Montinaro, Cyril Furtlehner, Luca Pagani, and Flora Jay. Creating artificial human genomes using generative neural networks. *PLOS Genetics*, 17(2):1–22, 02 2021.

[96] Rex Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L. Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling.

[97] Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[98] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. 1:57–81.

Supplementary Figures and Tables: Simultaneous Inference of Past Demography and Selection from the Ancestral Recombination Graph under the Beta Coalescent

Kevin Korfmann, Thibaut Sellinger, Fabian Freund,
Matteo Fumagalli, Aurélien Tellier

1 Supplementary Figures

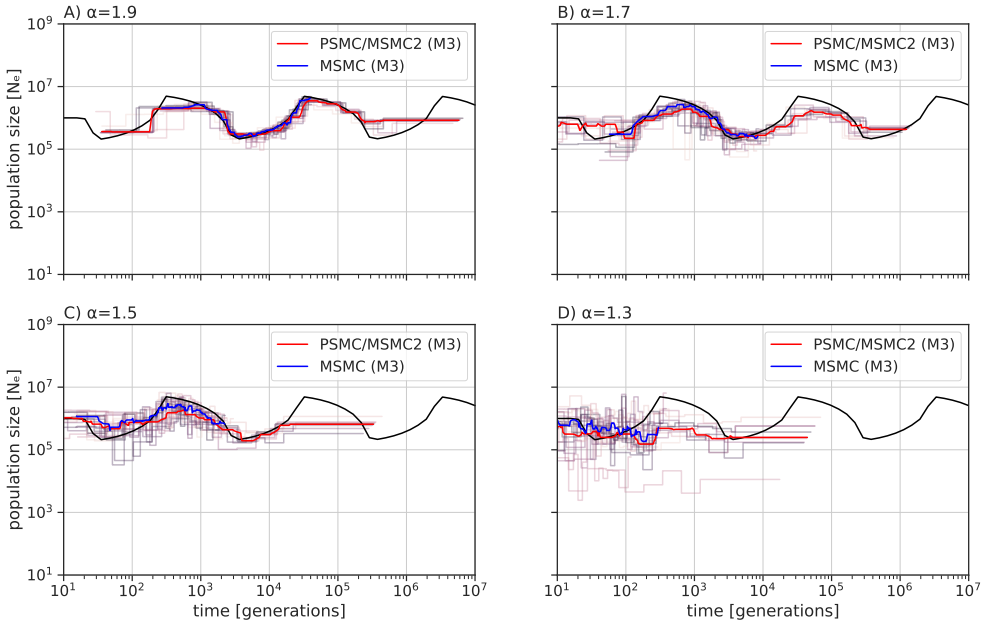

Fig. S1. Estimated demographic history by MSMC (blue) and MSMC2 (red) from the true ARG using 10 sequences of 100 Mb when population undergoes a sawtooth demographic scenario (black) for different α values: A) 1.9, B) 1.7, C) 1.5,and D) 1.3. The estimated population size is corrected in order to mask the scaling difference between the Kingman coalescent and the Beta coalescent (*i.e.* $Ne = (((mu_{estimated}/mu_{real})/scale)^{(}1/(alpha-1)))$, where $m = 1 + (1/((2^{(}alpha-1)) * (alpha-1)))$, $scale = (m^{alpha})/(alpha * beta(2-alpha, alpha))$ and $mu_{estimated} = \frac{\theta}{((2*sum(1/(1:(nb_{ind}-1))))*L)}$). The recombination and mutation rate are set to 1×10^{-8} per generation per bp.

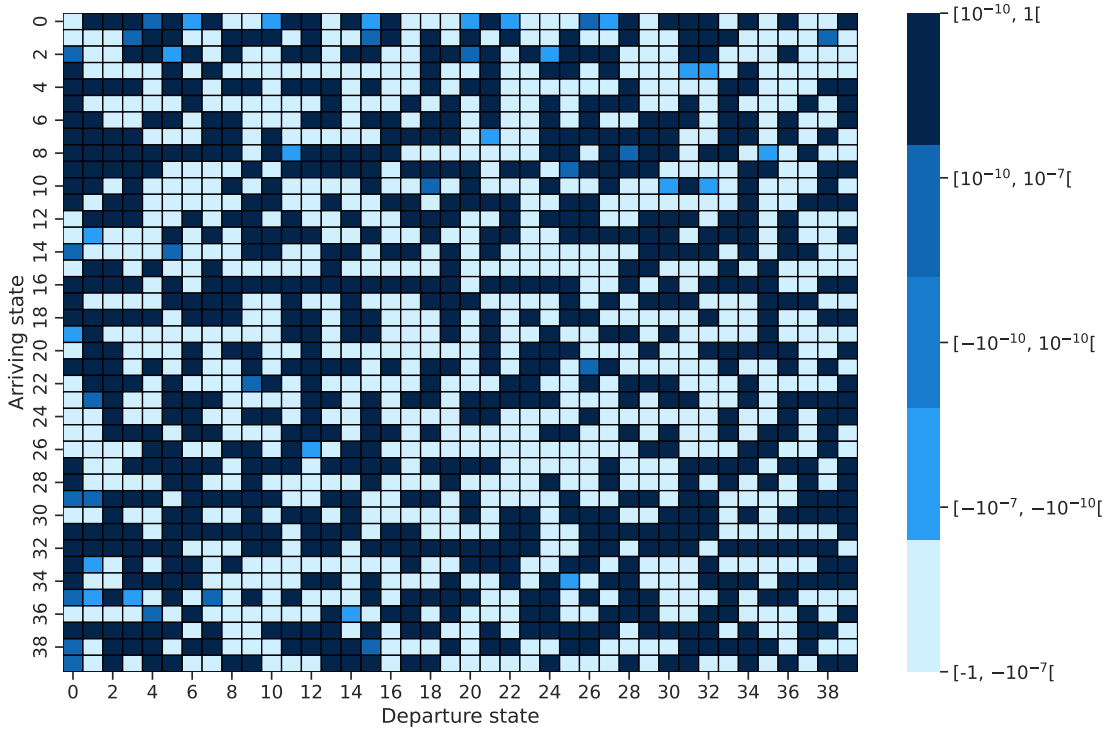

Fig. S2. Difference between the observed transition matrix and the theoretical prediction under the eSMC2 under a constant population size under the Kingman Coalescent

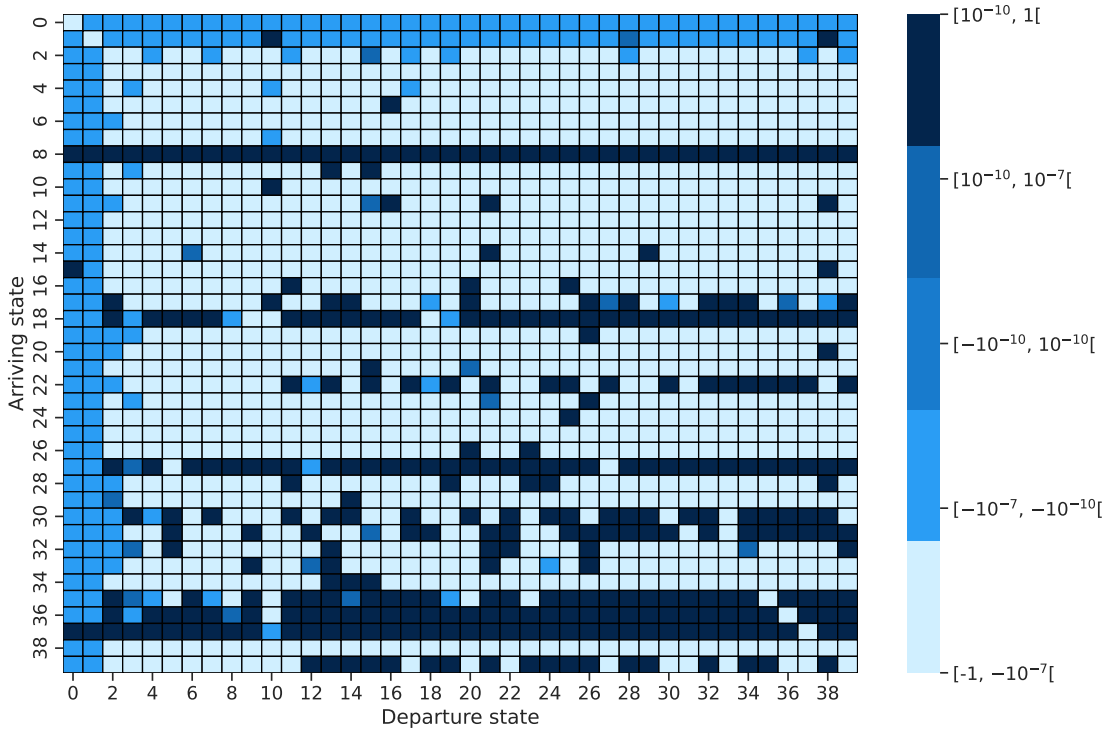

Fig. S3. Difference between the observed transition matrix and the theoretical prediction under the eSMC2 under a constant population size under the Beta Coalescent with $\alpha = 1.3$

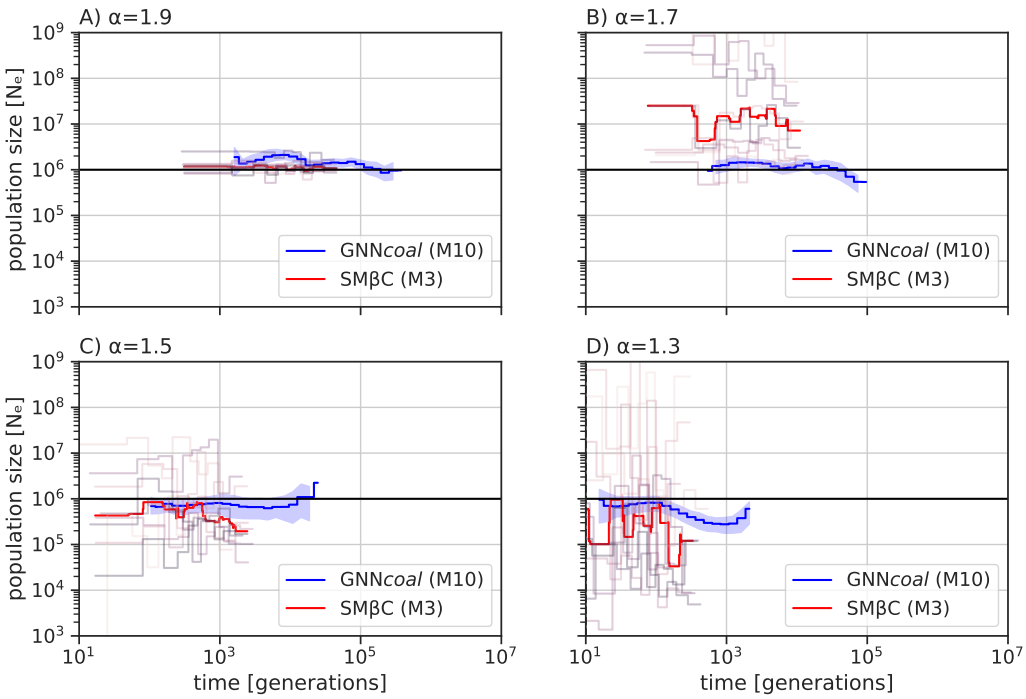

Fig. S4. Best-case convergence estimations of SMβC and GNN*coal* under a Beta coalescent. Estimations of past demographic history by SMβC using 10 sequences and 100 Mb in red (median) and by GNN using 10 sequences and 500 trees in blue (mean and CI95) when population undergoes a "constant" demographic scenario (black) under 4 different α values 1.9,1.7,1.5 and 1.3, respectively in A),B),C) and D). The recombination and mutation rate are set to 1×10^{-8} per generation per bp.

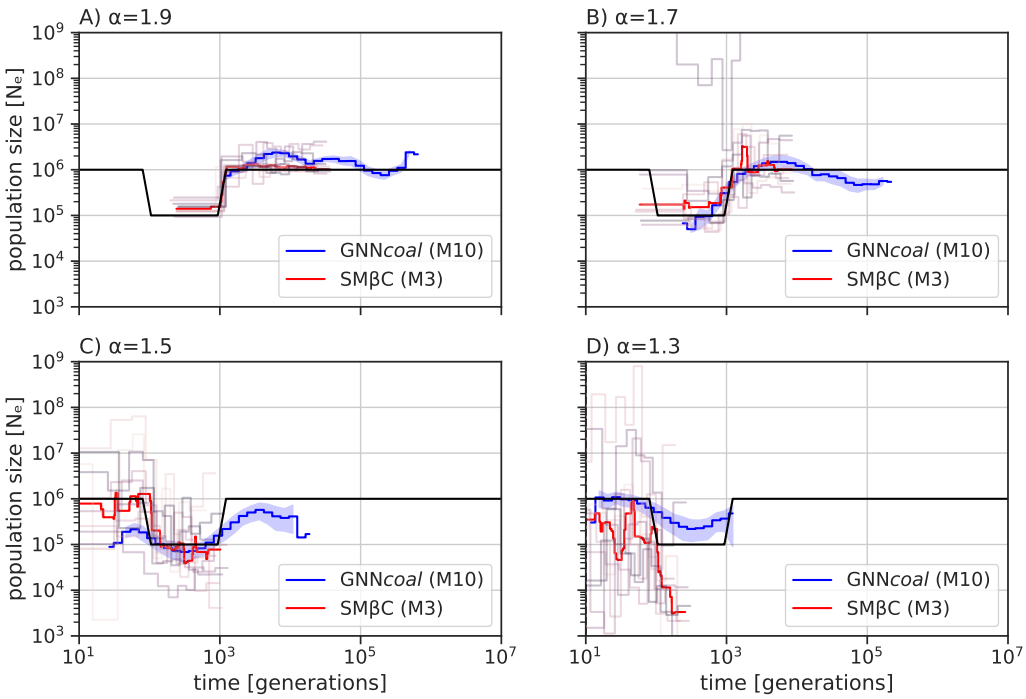

Fig. S5. Best-case convergence estimations of SMβC and GNN*coal* under a Beta coalescent. Estimations of past demographic history by SMβC using 10 sequences and 100 Mb in red (median) and by GNN using 10 sequences and 500 trees in blue (mean and CI95) when population undergoes a "bottleneck" demographic scenario (black) under 4 different α values A) 1.9, B) 1.7, c) 1.5 and D) 1.3. The recombination and mutation rate are set to 1×10^{-8} per generation per bp.

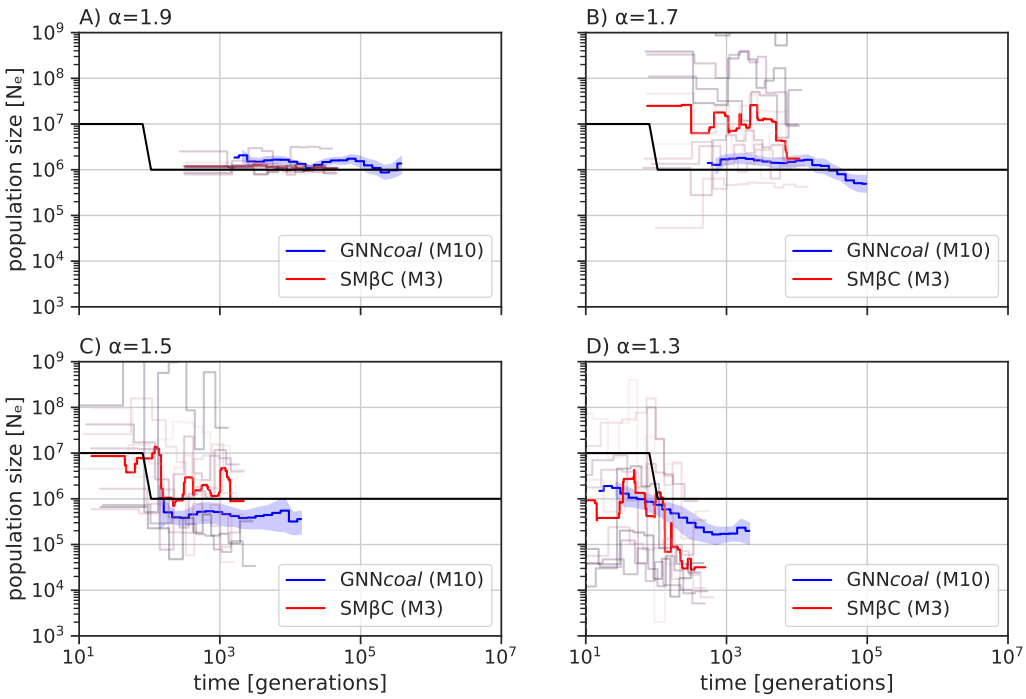

Fig. S6. Best-case convergence estimations of SMβC and GNN*coal* under a Beta coalescent. Estimations of past demographic history by SMβC using 10 sequences and 100 Mb in red (median) and by GNN using 10 sequences and 500 trees in blue (mean and CI95) when population undergoes a "increase" demographic scenario (black) under 4 different α values A) 1.9, B) 1.7, c) 1.5 and D) 1.3. The recombination and mutation rate are set to 1×10^{-8} per generation per bp.

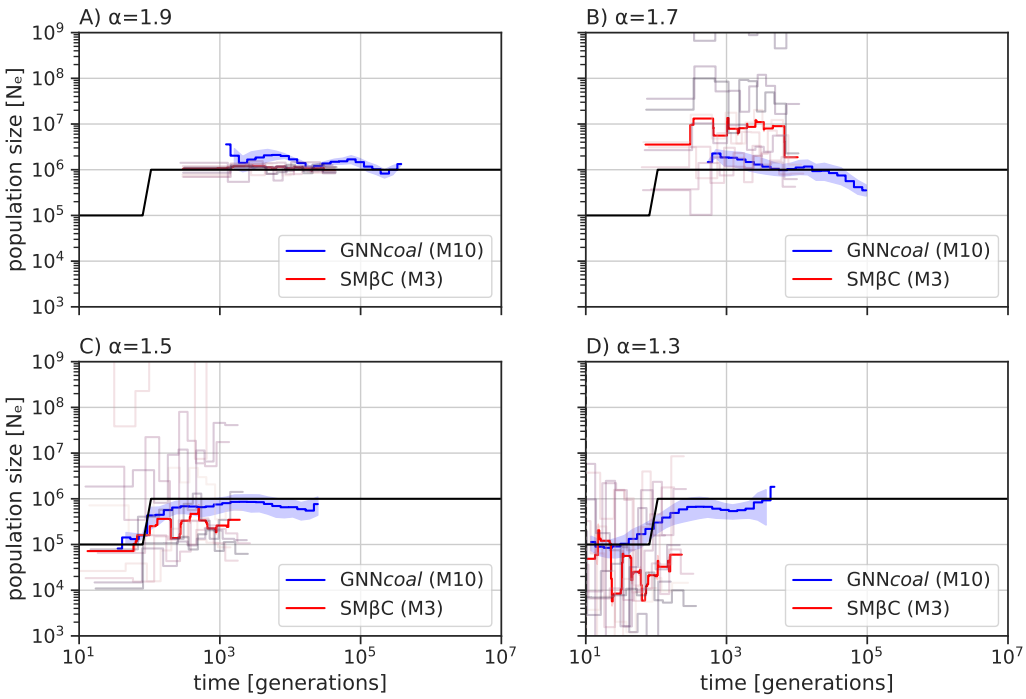

Fig. S7. Best-case convergence estimations of SMβC and GNN*coal* under a Beta coalescent. Estimations of past demographic history by SMβC using 10 sequences of 100 Mb in red (median) and by GNN using 10 sequences and 500 trees in blue (mean and CI95) when population undergoes a "decrease" demographic scenario (black) under 4 different α values A) 1.9, B) 1.7, c) 1.5 and D) 1.3. The recombination and mutation rate are set to 1×10^{-8} per generation per bp.

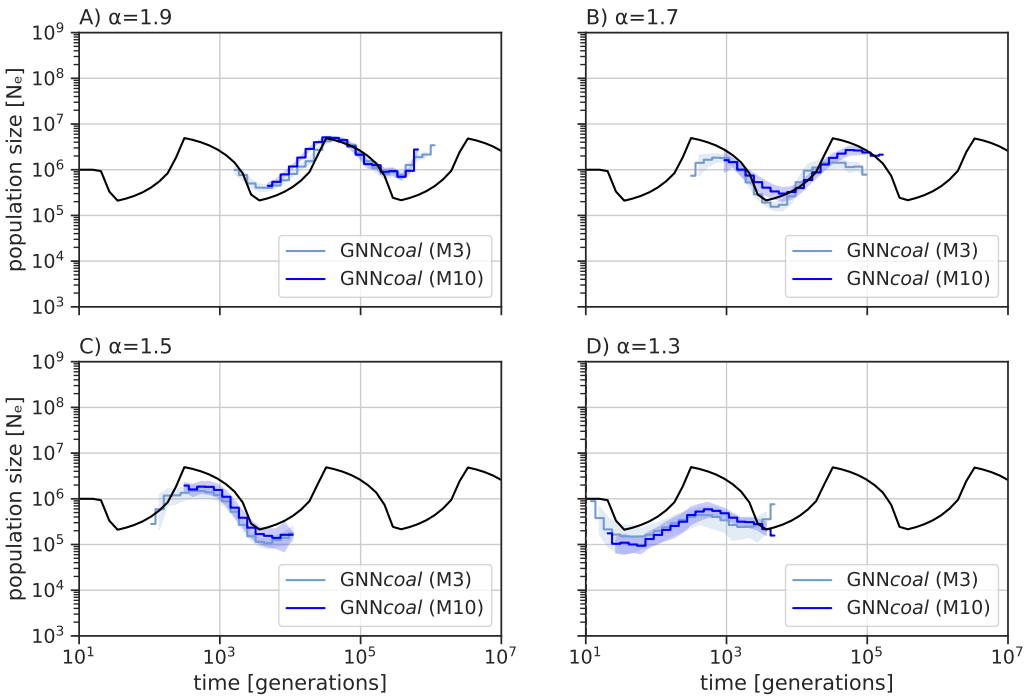

Fig. S8. Best-case convergence estimations of two GNNs under a Beta coalescent Estimations of past demographic history by GNN*coal* using 10 sequences in blue and by a GNN*coal* using 3 sequences in ice blue when population undergoes a "Sawtooth" demographic scenario (black) under 4 different α values A) 1.9, B) 1.7, c) 1.5 and D) 1.3 (mean and CI95). The recombination and mutation rate are set to 1×10^{-8} per generation per bp.

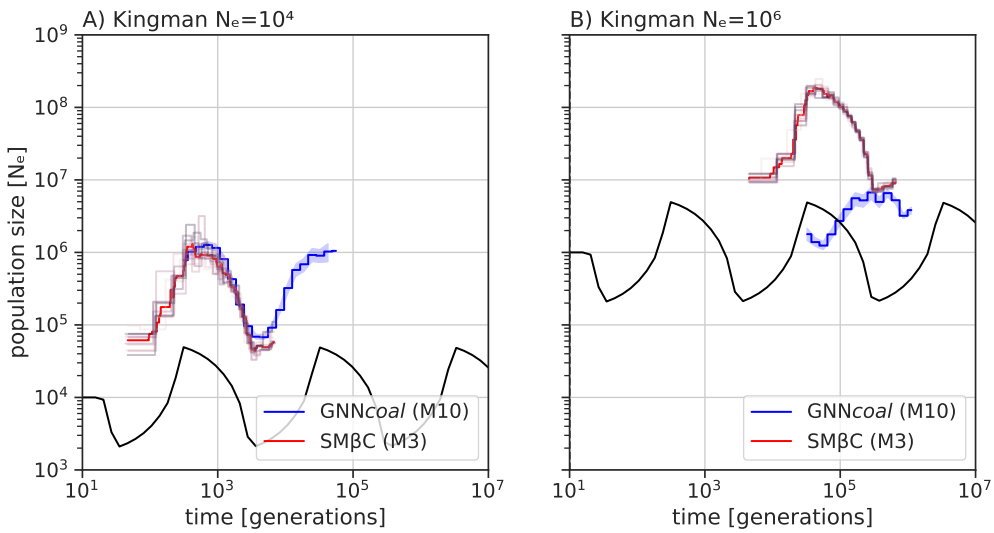

Fig. S9. Best-case convergence estimations of SMβC and GNN*coal* under a Kingman coalescent. Estimations of past demographic history by SMβC using 10 sequences and 10 Mb in red (median) and by GNN using 10 sequences and 500 trees in blue (mean and CI95) when population undergoes a "sawtooth" demographic scenario (black) under 2 different population sizes A) $N_e = 10^4$ and B) $N_e = 10^6$. The recombination and mutation rate are set to 1×10^{-8} per generation per bp.

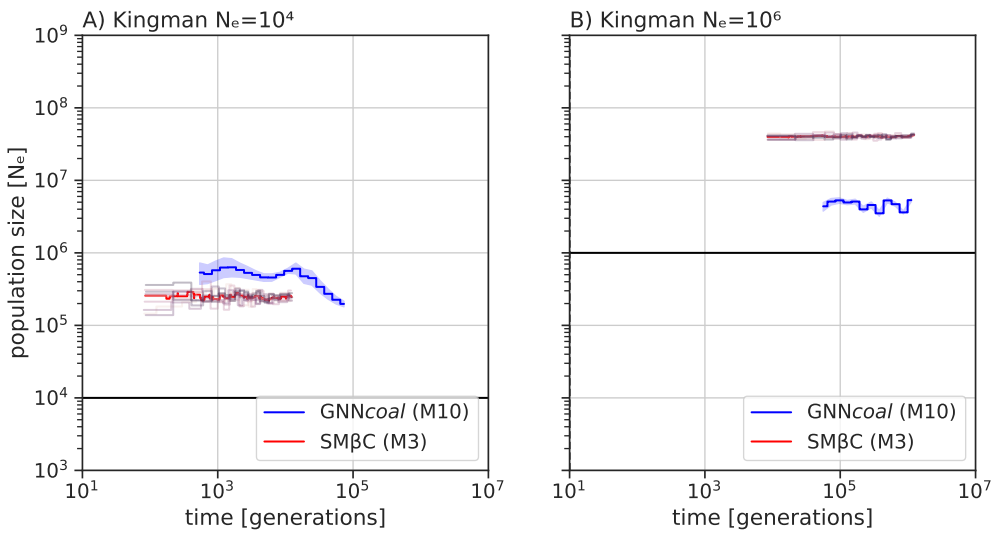

Fig. S10. Best-case convergence estimations of SMβC and GNN*coal* under a Kingman coalescent. Estimations of past demographic history by SMβC using 10 sequences and 10 Mb in red (median) and by GNN using 10 sequences and 500 trees in blue (mean and CI95) when population undergoes a "constant" demographic scenario (black) under 2 different population sizes A) $N_e = 10^4$ and B) $N_e = 10^6$. The recombination and mutation rate are set to 1×10^{-8} per generation per bp.

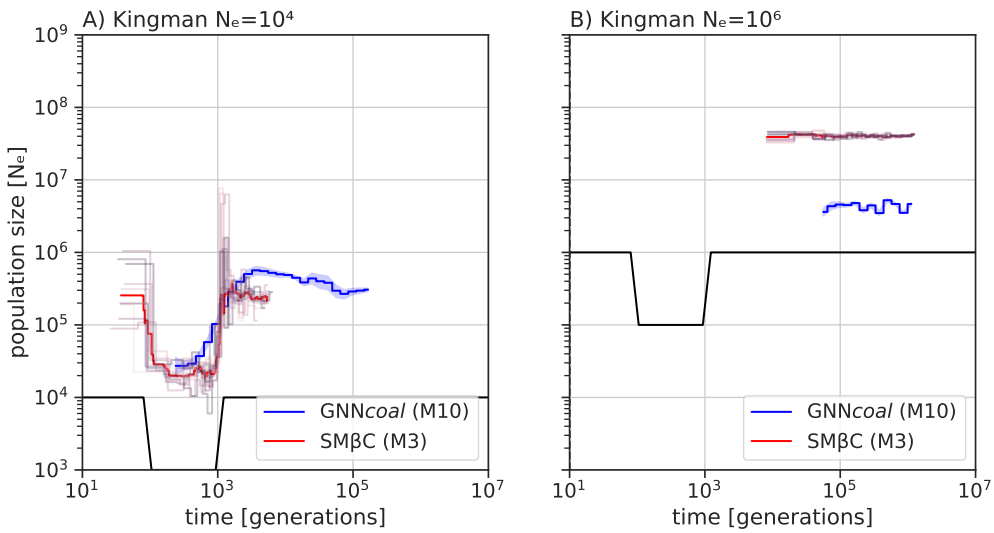

Fig. S11. Best-case convergence estimations of SMβC and GNN*coal* under a Kingman coalescent. Estimations of past demographic history by SMβC using 10 sequences and 10 Mb in red (median) and by GNN using 10 sequences and 500 trees in blue (mean and CI95) when population undergoes a "bottleneck" demographic scenario (black) under 2 different population sizes A) $N_e = 10^4$ and B) $N_e = 10^6$. The recombination and mutation rate are set to 1×10^{-8} per generation per bp.

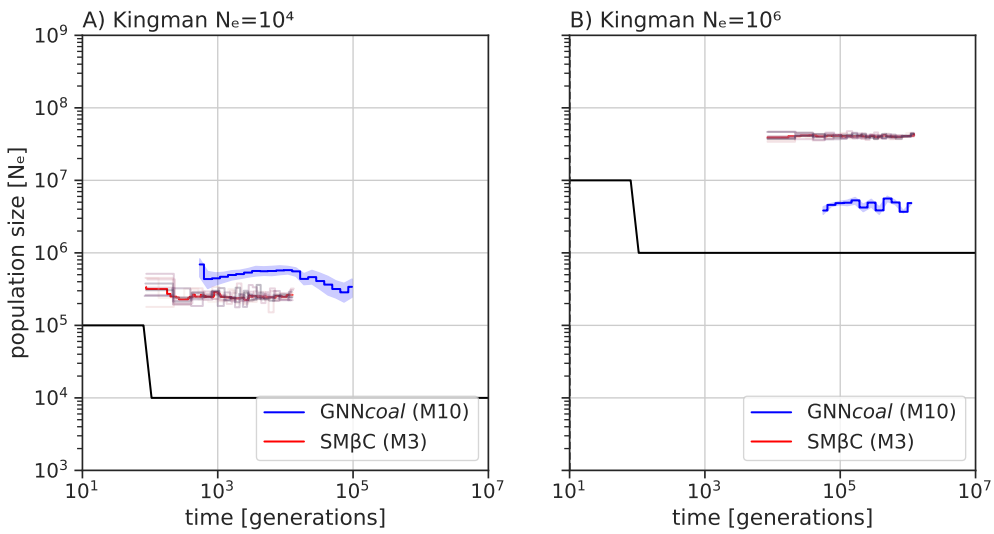

Fig. S12. Best-case convergence estimations of SMβC and GNN*coal* under a Kingman coalescent. Estimations of past demographic history by SMβC using 10 sequences and 10 Mb in red (median) and by GNN using 10 sequences and 500 trees in blue (mean and CI95) when population undergoes a "increase" demographic scenario (black) under 2 different population sizes A) $N_e = 10^4$ and B) $N_e = 10^6$. The recombination and mutation rate are set to 1×10^{-8} per generation per bp.

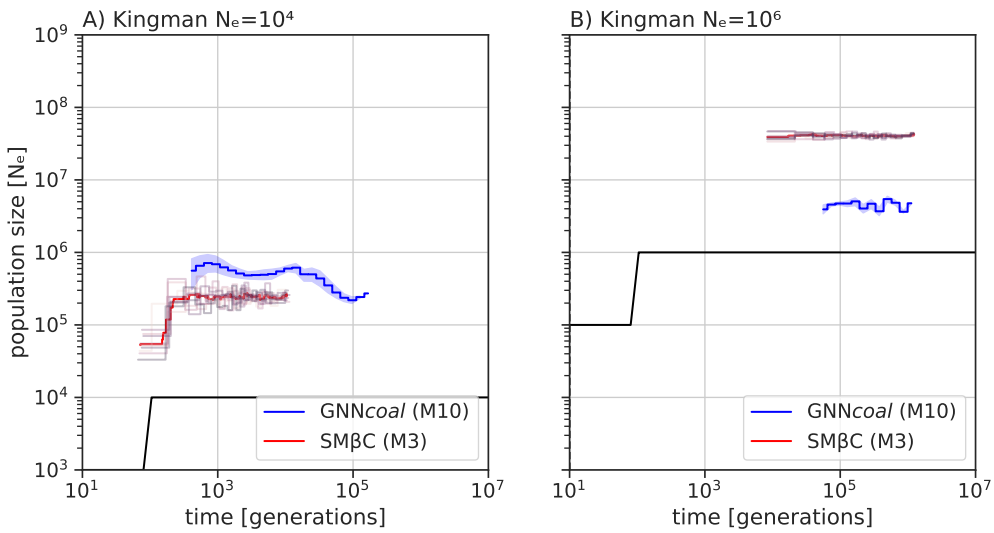

Fig. S13. Best-case convergence estimations of SMβC and GNN*coal* under a Kingman coalescent. Estimations of past demographic history by SMβC using 10 sequences and 10 Mb in red (median) and by GNN using 10 sequences and 500 trees in blue (mean and CI95) when population undergoes a "decrease" demographic scenario (black) under 2 different population sizes A) $N_e = 10^4$ and B) $N_e = 10^6$. The recombination and mutation rate are set to 1×10^{-8} per generation per bp.

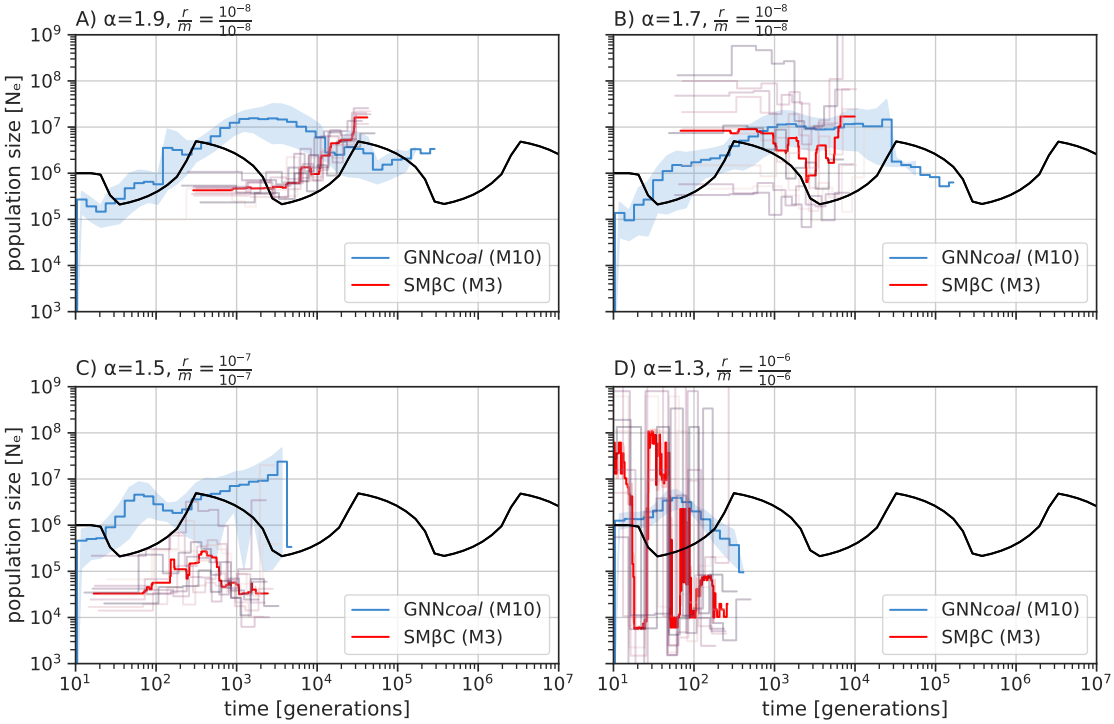

Fig. S14. Demographic inference estimations of SMβC and GNN*coal* using ARGweaver output under a sawtooth scenario.. Estimations of the ARG is first performed by ARGweaver using 10 sequences of10 Mb. Estimations of past demographic history is then performed on the inferred ARG by SMβC in red (median) and by GNN using 10 sequences and 500 trees in blue (mean and CI95) when population undergoes a "sawtooth" demographic scenario (black). The recombination and mutation rate per generation per bp are set to 1×10^{-8} for α equal to 1.9 and 1.7 , 1×10^{-7} for α equal to 1.5 and set to 1×10^{-6} for α equal to 1.3.

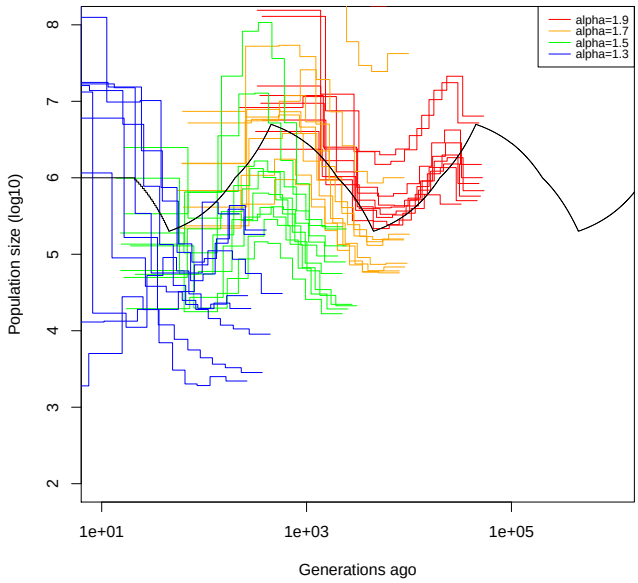

Fig. S15. Demographic inference estimations of SMβC on simulated data under a sawtooth demographic scenario.Estimated population size by SMβC from simulated sequence data under a sawtooth demographic scenario (black) and the Beta coalescent. Estimations of past demographic history by SMβC using 10 sequences of 10 Mb under different α values (1.9 in red, 1.7 in orange, 1.5 in green and 1.3 in blue. The recombination and mutation rate per generation per bp are set to 1×10^{-8} for α equal to 1.9 and 1.7 , 1×10^{-7} for α equal to 1.5 and set to 1×10^{-6} for α equal to 1.3 .

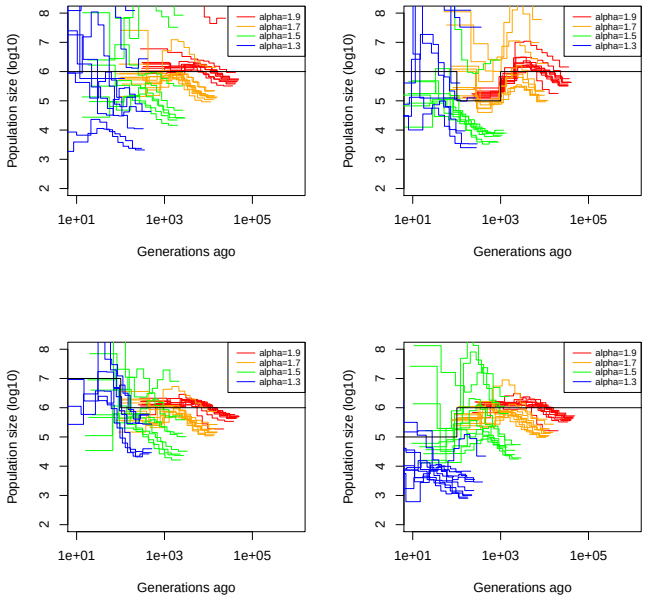

Fig. S16. Demographic inference estimations of SMβC on simulated data under 4 difference demographic scenarios. Estimated population size by SMβC from simulated sequence data under 4 demographic scenarios (black) (Constant population size in A, Bottleneck in B, Sudden increase of population size in C, and sudden decrease in D). Sequences are simulated under the Beta coalescent. Estimations of past demographic history by SMβC using 10 sequences of 10 Mb under different α values (1.9 in red, 1.7 in orange, 1.5 in green and 1.3 in blue. The recombination and mutation rate per generation per bp are set to 1×10^{-8} for α equal to 1.9 and 1.7 , 1×10^{-7} for α equal to 1.5 and set to 1×10^{-6} for α equal to 1.3 .

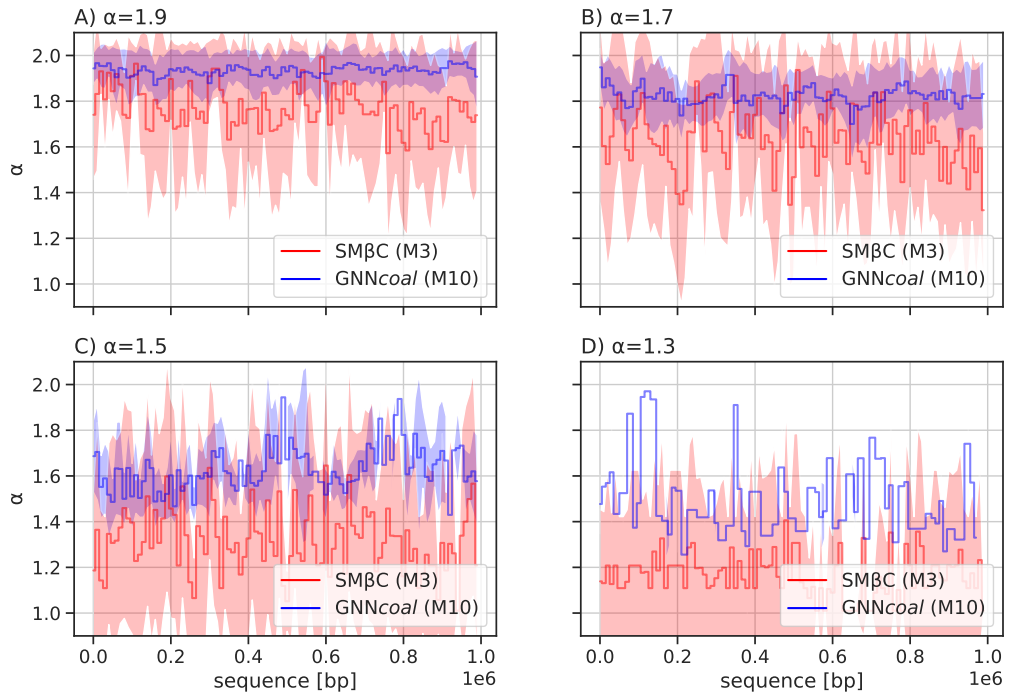

Fig. S17. Averaged estimations of alpha by the GNN*coal* approach and the SMβC along the sequence Estimations of α by SMβC using 20 sequences 1 Mb in red and blue by GNN*coal* using 10 sequences and in blue under 4 different α values 1.9,1.7,1.5 and 1.3 for a constant demography of size 10^6 in A),B),C) and D) (mean and standard deviation for both methods). The GNN*coal* used at most 20 coalescent trees per batch along the sequence. Both methods used a window size of 10^3 bp. The recombination and mutation rate are set to 1×10^{-8} per generation per bp.

2 Supplementary Tables

scenario	True α	α^*:SMβC,M=3	α^*:SMβC,M=4	α^* : GNN, M=3	α^* : GNN, M=10
Constant	1.9	1.88 (0.04)	1.84 (0.1)	1.90 (0.03)	1.88 (0.03)
Bottleneck	1.9	1.86 (0.07)	1.82 (0.16)	1.91 (0.03)	1.91 (0.02)
Increase	1.9	1.88 (0.05)	1.86 (0.12)	1.91 (0.03)	1.88 (0.02)
Decrease	1.9	1.89 (0.02)	1.88 (0.07)	1.93 (0.03)	1.88 (0.04)
Sawtooth	1.9	1.84 (0.06)	1.74 (0.11)	1.96 (0.02)	1.94 (0.02)
Constant	1.7	1.53 (0.12)	1.63 (0.09)	1.78 (0.03)	1.77 (0.03)
Bottleneck	1.7	1.67 (0.14)	1.57 (0.09)	1.78 (0.04)	1.76 (0.03)
Increase	1.7	1.57 (0.14)	1.63 (0.17)	1.77 (0.01)	1.74 (0.03)
Decrease	1.7	1.54 (0.15)	1.57 (0.14)	1.80 (0.04)	1.78 (0.04)
Sawtooth	1.7	1.67 (0.14)	1.60 (0.19)	1.81 (0.02)	1.74 (0.03)
Constant	1.5	1.58 (0.10)	1.56 (0.14)	1.61 (0.02)	1.56 (0.02)
Bottleneck	1.5	1.53 (0.11)	1.55 (0.19)	1.58 (0.02)	1.53 (0.03)
Increase	1.5	1.50 (0.12)	1.52 (0.14)	1.64 (0.05)	1.61 (0.03)
Decrease	1.5	1.57 (0.17)	1.55 (0.14)	1.60 (0.03)	1.55 (0.03)
Sawtooth	1.5	1.47 (0.11)	1.55 (0.19)	1.61 (0.04)	1.56 (0.02)
Constant	1.3	1.40 (0.13)	1.45 (0.14)	1.39 (0.03)	1.33 (0.03)
Bottleneck	1.3	1.43 (0.13)	1.48 (0.20)	1.34 (0.02)	1.28 (0.04)
Increase	1.3	1.45 (0.14)	1.51 (0.24)	1.42 (0.02)	1.38 (0.02)
Decrease	1.3	1.44 (0.16)	1.50 (0.12)	1.4 (0.02)	1.30 (0.03)
Sawtooth	1.3	1.39 (0.09)	1.39 (0.07)	1.41 (0.03)	1.36 (0.05)

Table S1: Average estimated values of α by SMβC and the GNN*coal* approach over ten repetitions using the true ARG of 10 sequences of 100 Mb with recombination and mutation rate set to 1×10^{-8} per generation per bp under a Beta coalescent process (with different α parameter). The analysis were run on five different demographic scenarios (Constant population size, Bottleneck, Sudden increase, Sudden decrease and a Sawtooth demography). The standard deviation is indicated in brackets. These results complement the demographic estimates from Figures 4 and 5, as well as Supplementary Figure 4 to 7

scenario	True α	α^*:SMβC,M=3	α^*:SMβC,M=4
Constant	1.7	1.64 (0.12)	1.56 (0.11)
Bottleneck	1.7	1.59 (0.15)	1.55 (0.12)
Increase	1.7	1.65 (0.10)	1.61 (0.15)
Decrease	1.7	1.71 (0.10)	1.67 (0.12)
Sawtooth	1.7	1.60 (0.18)	1.51 (0.20)
Constant	1.5	1.46 (0.11)	1.51 (0.16)
Bottleneck	1.5	1.50 (0.14)	1.45 (0.16)
Increase	1.5	1.56 (0.13)	1.47 (0.15)
Decrease	1.5	1.51 (0.11)	1.47 (0.12)
Sawtooth	1.5	1.40 (0.13)	1.44 (0.17)
Constant	1.3	1.38 (0.15)	1.37 (0.08)
Bottleneck	1.3	1.34 (0.10)	1.34 (0.06)
Increase	1.3	1.34 (0.08)	1.46 (0.17)
Decrease	1.3	1.41 (0.15)	1.44 (0.09)
Sawtooth	1.3	1.35 (0.07)	1.34 (0.05)

Table S2: Average estimated values of α by SMβC over ten repetitions using the true ARG of 10 sequences of 100 Mb with recombination and mutation rate set to 5×10^{-8} for $\alpha = 1.7$ per generation per bp and 5×10^{-7} for α equal to 1.5 and 1.3 under a Beta coalescent process (with different α values). The coefficient of variation is indicated in brackets..

scenario	True α	α^*:SMβC,M=3	α^*:SMβC,M=4	α^* : GNN, M=3	α^* : GNN, M=10
Constant	2	1.98 (0.023)	1.97 (0.023)	1.99 (0.002)	1.99 (0.005)
Sawtooth	2	1.96 (0.027)	1.87 (0.064)	1.98 (0.003)	1.99 (0.005)
Bottleneck	2	1.96 (0.047)	1.96 (0.04)	1.99 (0.002)	1.99 (0.003)
Decrease	2	1.98 (0.022)	1.99 (0.012)	1.98 (0.009)	1.99 (0.005)
Increase	2	1.97 (0.020)	1.97 (0.025)	1.99 (0.002)	1.99 (0.004)

Table S3: Average estimated values of α by SMβC and the GNN*coal* approach over ten repetitions using the true ARG of 10 sequences of 100 Mb for SMβC and at 500 trees for the GNN*coal* with recombination and mutation rate set to 1×10^{-8} per generation per bp under a Beta coalescent process (with different α parameter) with $N_e = 10^4$ at generation 0. The standard deviation is indicated in brackets. These results complement the estimations of demography from Supplementary Figure 9-13 .

scenario	True α	α^*:SMβC,M=3	α^*:SMβC,M=4	α^* : GNN, M=3	α^* : GNN, M=10
Sawtooth	1.9	1.77 (0.05)	1.86 (0.05)	1.55 (0.08)	2.05 (0.01)
Sawtooth	1.7	1.56 (0.13)	1.79 (0.1)	1.7 (0.11)	2.00 (0.06)
Sawtooth	1.5	1.66 (0.11)	1.92 (0.07)	1.61 (0.07)	1.96 (0.04)
Sawtooth	1.3	1.41 (0.13)	1.53 (0.20)	1.30 (0.06)	1.78 (0.08)

Table S4: Average estimated values of α by SMβC and the GNN*coal* approach using ARG inferred by ARGweaver over ten repetitions using 10 simulated sequences of 10 Mb with recombination and mutation rate set to 1×10^{-8} per generation per bp under a Beta coalescent process (with different α parameter). The analysis were run under Sawtooth demographic scenario. The standard deviation is indicated in brackets. These results complement the demographic estimates from Supplementary Figures 14

Supplementary Text S1: The Sequentially Markovian β Coalescent

Kevin Korfmann, Thibaut Sellinger, Fabian Freund,
Matteo Fumagalli, Aurélien Tellier

1 SMβC

The Sequentially Markovian β Coalescent is a Hidden Markov Model based on the Multiple Sequentially Markovian Coalescent (MSMC) where Multiple Merger event are allowed to occur under the β coalescent.

To define our Hidden Markov Model (HMM) we need to define :

- Hidden States
- The signal (observed data)
- A Transition matrix (Probability of passing from one state to another)
- An Emission matrix (Probability of observing the signal conditional to the hidden state)
- An Initial probability (Probability of hidden states at the first position of the sequence)

1.1 Notations and Assumptions

We here define the different notations used and their meaning:

- r : recombination rate per nucleotide
- μ : Mutation rate per nucleotide
- u : recombination time, follows a continuous uniform distribution between 0 and first coalescent time.
- ξ_t : Scaled population size at time t ($N_t = \xi_t N_0$)
- $\chi_t = \xi_t^{\beta-1}$
- M : Number of analyzed haploid sequences (or haploid individuals)
- β : The multiple merger parameter

The model's assumptions are :

- $(\chi_t)_{t\geq 0}$ is piece-wise constant (intervals are specified in the following)

We first define the transition rates of the Beta n-coalescent [3]. The rate of transition from a state with b lineages to $b-n+1$ lineages, i.e. a merger of n lineages is

$$\lambda_{b,\beta,b-n+1} = \frac{B(n-\beta, b-n+\beta)}{\Gamma(2-\beta)\Gamma(\beta)}. \tag{1}$$

$$\Lambda_{b,\beta,b-n+1} = \frac{\binom{b}{n} B(n-\beta, b-n+\beta)}{\Gamma(2-\beta)\Gamma(\beta)}. \tag{2}$$

Thus, the total rate is

$$\lambda_{b,\beta} = \sum_{k=2}^{b} \frac{\binom{b}{k} B(k-\beta, b-k+\beta)}{\Gamma(2-\beta)\Gamma(\beta)} \tag{3}$$

Waiting times are exponentially distributed in the coalescent for population size constant in time. For time-varying population sizes, we define the time-changed Λ-n-coalescent as the (rescaled) genealogy limit from a Wright-Fisher type Cannings model with skewed offspring distributions as introduce in [5], which leads to a time-change waiting time for coalescence events: If a waiting time has rate λ in the standard Beta n-coalescent (started at some time t_0), it has a waiting time density of

$$f(t) = \frac{\lambda}{\chi(t)} e^{-\int_{t_0}^{t} \frac{\lambda}{\chi(s)} ds}, \tag{4}$$

which follows as described in [2].

1.2 Hidden States

Our hidden states at one position are the first coalescent time $t > 0$ at that position and which individuals $i := (i_1, \ldots, i_n)$ coalesce in the corresponding coalescence. A transition from coalescent time s to time t or a change in the index i can only occur when a recombination happens.

1.3 Observations

The observation signal is the comparison of the M analyzed sequences. Thus the signal is a series of number indicating the allelic state of the sequences at each position. For M=3, under the infinite site model hypothesis, only 4 different state can be observed along the sequence. All sequences are the same at this position (indicated by a 0), or one of the three sequences is different from the two other (indicated by i, if the sequence i is different from the two other).

1.4 Transition Matrix

Five transitions are possible, we transition from (s, j) to (t, i). Here we assume that t and s are in interval time α and γ. At indices i and j, n and m individuals coalesce. In addition, recombination occurs with probability :

$$P(rec|s) = 1 - e^{-rMs} \tag{5}$$

We assume that only a single recombination event can happen before a merger. A recombination event of one of M lineages splits one ancestral lineage in two (backwards in time). The additional lineage is not yet described by the coalescent without the recombination event, we call this free. It can merge with any of the remaining M lineages, but also with the second parental ancestral lineage (i.e. the second split lineage from the recombination event). The transition probabilities/rates conditional on the (known) behaviour of the other lineages are as described in [1, Sect. 5]: Conditional on the mergers of the M other lineages, a binary merger of the "freed" lineage appears with rate $M\lambda_{M+1,\beta,M}$ and it joins an existing merger of m lineages at some time t with probability $1-\frac{\lambda_{M+1,\beta,M+1-m+1}}{\lambda_{M,\beta,M-m+1}}=\frac{\lambda_{M+1,\beta,M+1-m}}{\lambda_{M,\beta,M-m+1}}$, where the second equation is due to the consistency of rates in Λ-n-coalescents. In the following, we derive conditional probabilities and/or conditional densities for certain events.

1.4.1 t<s

For this to happen, a recombination must occur before time t. The new number of individuals first coalescing is now $n=2$, and the recombination event needs to affect one of these two individuals $i=\{i_1,i_2\}$ (which happens with probability $2/M$), splitting one lineage in two. Then, we just multiply the density of a binary merger of the free lineage with any of the other M lineages in the time-changed coalescent, which is Eq. (4) with rate $M\lambda_{M+1,\beta,M}$ and the probability that the first merger is indeed merging i, which is $\frac{1}{M}$ (we pick the second lineage of i at random from M lineages):

$$f(t,i|s,j,u)=\frac{2\lambda_{M+1,\beta,M}}{M\chi_t}e^{-\int_u^t\frac{M\lambda_{M+1,\beta,M}}{\chi_v}dv} \tag{6}$$

1.4.2 t=s

<u>Case 1</u>: a non coalescing individual joins the coalescent event $j=\{j_1,\dots,j_n\}$. For this to happen, the recombination event must occur before time s in a non coalescing branch, which happens with probability $\frac{M-n}{M}$. Then, the newly split second ancestral lineage of i need to not coalesce in a binary collision until time s, which equals $\exp(-\int_u^s\frac{M\lambda_{M+1,\beta,M}}{\chi_v}dv)$ (by integrating Eq. (4) with rate $M\lambda_{M+1,\beta,M}$). Finally, it then needs to join in the coalescent event j, which happens with probability $\frac{\lambda_{M+1,\beta,M+1-m}}{\lambda_{M,\beta,M-m+1}}$. This shows that

$$P(s,i|s,j,u)=\frac{(M-n)\lambda_{M+1,\beta,M+1-m}}{M\lambda_{M,\beta,M-m+1}}e^{-\int_u^s\frac{M\lambda_{M+1,\beta,M}}{\chi_v}dv} \tag{7}$$

for $i=j\cup\{i\}$.

<u>Case 2</u>: Recombination occurs in a coalescing individual $i\in j=\{j_1,\dots,j_n\}$ of a multiple merger event with $n>2$ (happens with probability $\frac{1}{n}$). The new lineage then coalesces higher in time, i.e. it does neither coalesce in a binary merger before s (as in case 1) nor in the collision at s (which is the complementary event from case 1). As above, this leads to

$$P(s,i|s,j,u)=\frac{\lambda_{M+1,\beta,M+1-m+1}}{n\lambda_{M,\beta,M-m+1}}e^{-\int_u^s\frac{M\lambda_{M+1,\beta,M}}{\chi_v}dv} \tag{8}$$

for $i=j\setminus\{i\}$

Case 3: Nothing changes. This happens if a) there is no recombination event (so $u > s$), b) the lineage split makes a binary merger between the two lineages resulting from the split ("self-coalesce") before s, c) recombination splits a lineage merged at the coalescence event at s, but that the second ancestral lineage from the split joins the merger.
a) happens with probability 1 if $u > s$, b) has conditional density as in Eq. 6 without the factor $2/M$, integrating over $[u, s]$ yields the probability $(1-(1/M)\exp(-\int_u^t \frac{M\lambda_{M+1,\beta,M}}{\chi_v}dv)$ and c) follows as case 2, only we need the recombination event on a lineage already participating in the merger at time s (so just replacing $M-n$ with n in Eq. (8)).

1.4.3 t>s

For this to happen, a recombination must occur before time s and break a coalescent event of only two individuals (j_1, j_2) (w. probability $2/M$). Assume without restriction j_1 was affected by recombination. For (the new ancestral lineage of) j_1 to coalesce at time t, it must not coalesce until time s, then not coalesce in the former coalescent event and then the next coalescence event happens at time t. The next coalescence event can take any form and does not need to merge j_1. Additionally, we just need to keep track of the $M-1$ non-recombining lineages and j_1, since we are not conditioning on the behaviour after s and thus both self-coalescence and the coalescence of the second split lineage (not j_1) can be ignored. Thus, to compute the conditional rate for merging into i, we first compute the probability that the new ancestral lineage representing j_1 in the new DNA segment does not coalesce until time s, given by Eq. (8) with $n = 2$, and multiply this by the conditional density for merging into any i of M lineages afterwards. Thus, this is just Eq. (4) with rate $\lambda_{M,\beta}$ multiplied with $\frac{\lambda_{M,\beta,M-n+1}}{\lambda_{M,\beta}}$. This leads to

$$f(t, i|s, j, u) = \frac{2\lambda_{M+1,\beta,M}}{M\lambda_{M,\beta,M-1}} e^{-\int_u^s \frac{M\lambda_{M+1,\beta,M}}{\chi_v}dv} \frac{\lambda_{M,\beta,M-n+1}}{\chi_t} e^{-\int_s^t \frac{\lambda_{M,\beta}}{\chi_v}dv} \tag{9}$$

for $i = \{i_1, \ldots, i_n\}$.

1.4.4 Full transition probability

$$p(t, i|s, j, u) =$$

$$\begin{cases} (1-e^{-Mrs})\frac{2\lambda_{M+1,\beta,M}}{\chi_t M} e^{-\int_u^t \frac{M\lambda_{M+1,\beta,M}}{\chi_v}dv} & u < t < s \\ e^{-Mrs} + (1-e^{-Mrs})(\int_u^s \frac{e^{\int_u^k -\frac{M\lambda_{M+1,\beta,M}}{\chi_v}dv}}{\chi_k}dk + \frac{(M-n)\lambda_{(n+1),\beta,2}}{M\lambda_{(n+1),\beta,2}+\lambda_{(n+1),\beta,1}} e^{\int_u^t -\frac{M\lambda_{M+1,\beta,M}}{\chi_v}dv}) & t = s, m = n \\ (1-e^{-Mrs})\frac{(M-n)\lambda_{(n+1),\beta,1} e^{-\int_u^s \frac{M\lambda_{M+1,\beta,M}}{\chi_v}dv}}{M(\lambda_{(n+1),\beta,2}+\lambda_{(n+1),\beta,1})} & t = s, m = n+1 \\ (1-e^{-Mrs})\frac{1}{n}\frac{\lambda_{(n+1),\beta,2} e^{-\int_u^s \frac{M\lambda_{M+1,\beta,M}}{\chi_v}dv}}{s(\lambda_{(n+1),\beta,2}+\lambda_{(n+1),\beta,1})} & t = s, m+1 = n \\ (1-e^{-Mrs})\frac{\lambda_{M,\beta,(M-m+1)}}{\binom{M}{m}\chi_\alpha} e^{-\int_s^t \frac{\lambda_{M,\beta}}{\chi_v}dv} e^{-\int_u^s \frac{M\lambda_{M+1,\beta,M}}{\chi_v}dv} \frac{2\lambda_{(n+1),\beta,2}}{M(\lambda_{(n+1),\beta,2}+\lambda_{(n+1),\beta,1})} & t > s, i = l, j = k \end{cases} \tag{10}$$

As explained before, the state space must be finite. We therefore discretized time in k intervals. At one point the hidden state is α if $t \in [T_\alpha, T_{\alpha+1}]$, where $\alpha \in [0, (n-1)]$. We define T_α :

$$T_\alpha = \frac{-\ln(1-\frac{\alpha}{n})}{\lambda_{M,\beta}} \tag{11}$$

We therefore have:

$$p(\alpha, i|s, j) = \int_{T_\alpha}^{T_{\alpha+1}} p(t, i|s, j)dt \tag{12}$$

Note: Because time is discretized, if the first coalescent time is bigger than $T_{(n-1)}$, then all individual coalesce.

1.4.5 Initial Probability

We use the equilibrium probability as initial probability while assuming m individual coalesce. The equilibrium probability is given by :

$$\begin{aligned} q_o(\alpha, i) &= \int_{T_\alpha}^{T_{\alpha+1}} \frac{\lambda_{M,\beta,(M-m+1)}}{\chi_\alpha \binom{M}{m}} e^{-\int_0^t \frac{\lambda_{M,\beta}}{\chi_v} dv} dt \\ &= \frac{\lambda_{M,\beta,(M-m+1)} e^{\sum_{\eta=0}^{\alpha-1} \frac{\lambda_{M,\beta}}{\chi_\eta} \Delta_\eta}}{\binom{M}{m} \lambda_{M,\beta}} (1 - e^{-\Delta_\alpha \frac{\lambda_{M,\beta}}{\chi_t}}) \end{aligned} \tag{13}$$

Where :

$$\Delta_\gamma = T_{\gamma+1} - T_\gamma \tag{14}$$

1.4.6 Calculation of $t_{\gamma,j}$

Assuming n individual coalesces.

$$\begin{aligned} t_{\gamma,j} = E[\text{Coalescent time}|\gamma, j] = \frac{E[\text{Coalescent time} \cap \gamma, j]}{P(\gamma, j)} &= \frac{\int_{T_\gamma}^{T_{\gamma+1}} t \lambda_{M,\beta,(M-n+1)} e^{-\int_0^t \frac{\lambda_{M,\beta}}{\chi_v} dv}}{\binom{M}{n} q_0(\gamma, j)} dt \\ &= \frac{T_\gamma - T_{\gamma+1} e^{-\Delta_\gamma \frac{\lambda_{M,\beta}}{\chi_\gamma}}}{(1 - e^{-\Delta_\gamma \frac{\lambda_{M,\beta}}{\chi_\gamma}})} + \frac{\chi_\gamma}{\lambda_{M,\beta}} \end{aligned} \tag{15}$$

Where :

$$\Delta_\gamma = T_{\gamma+1} - T_\gamma \tag{16}$$

We note that $t_{\gamma,j}$ is independent of j, thus $t_{\gamma,j} = t_\gamma$.

1.4.7 Calculation of $p(\alpha, i|\gamma, j)$

$\alpha < \gamma$

We here calculate the transition probability from the state γ to a time t in the time interval

α.

$$
\begin{aligned}
P(t,i|t_\gamma,j) &= \frac{P_\gamma}{t_\gamma}\int_0^t \frac{2\lambda_{M+1,\beta,M}}{\chi_v M} e^{-\int_u^t \frac{M\lambda_{M+1,\beta,M}}{\chi_v}dv} du \\
&= \frac{P_\gamma}{t_\gamma}\Big(\sum_{\eta=0}^{\alpha-1}\int_{T_\eta}^{T_{\eta+1}} \frac{2\lambda_{M+1,\beta,M}}{\chi_v M} e^{-\int_u^t \frac{M\lambda_{M+1,\beta,M}}{\chi_v}dv} du + \int_{T_\alpha}^t \frac{2\lambda_{M+1,\beta,M}}{\chi_v M} e^{-\int_u^t \frac{M\lambda_{M+1,\beta,M}}{\chi_v}dv} du\Big) \\
&= \frac{P_\gamma}{t_\gamma}\Big(\sum_{\eta=0}^{\alpha-1}\int_{T_\eta}^{T_{\eta+1}} \frac{2\lambda_{M+1,\beta,M}}{\chi_\alpha M} e^{-\int_{T_{\eta+1}}^{T_\alpha} \frac{M\lambda_{M+1,\beta,M}}{\chi_v}dv} e^{-\int_{T_\alpha}^t \frac{M\lambda_{M+1,\beta,M}}{\chi_v}dv} e^{-\int_u^{T_{\eta+1}} \frac{M\lambda_{M+1,\beta,M}}{\chi_v}dv} du + \\
&\qquad \int_{T_\alpha}^t \frac{2\lambda_{M+1,\beta,M}}{\chi_\alpha M} e^{-(t-u)\frac{M\lambda_{M+1,\beta,M}}{\chi_\alpha}dv} du\Big) \\
&= \frac{P_\gamma}{t_\gamma}\frac{2\lambda_{M+1,\beta,M}}{\chi_\alpha M}\Big(\sum_{\eta=0}^{\alpha-1}\int_{T_\eta}^{T_{\eta+1}} e^{-\sum_{\zeta=\eta+1}^{\alpha-1}\Delta_\zeta \frac{M\lambda_{M+1,\beta,M}}{\chi_\zeta}} e^{-(t-T_\alpha)\frac{M\lambda_{M+1,\beta,M}}{\chi_\alpha}} e^{-(T_{\eta+1}-u)\frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}} du \\
&\qquad + \frac{(1-e^{-(t-T_\alpha)\frac{M\lambda_{M+1,\beta,M}}{\chi_\alpha}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\alpha}}\Big) \\
&= \frac{P_\gamma}{t_\gamma}\frac{2\lambda_{M+1,\beta,M}}{\chi_\alpha M}\Big(\sum_{\eta=0}^{\alpha-1} e^{-\sum_{\zeta=\eta+1}^{\alpha-1}\Delta_\zeta \frac{M\lambda_{M+1,\beta,M}}{\chi_\zeta}} e^{-(t-T_\alpha)\frac{M\lambda_{M+1,\beta,M}}{\chi_\alpha}} \frac{(1-e^{-\Delta_\eta\frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}} \\
&\qquad + \frac{(1-e^{-(t-T_\alpha)\frac{M\lambda_{M+1,\beta,M}}{\chi_\alpha}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\alpha}}\Big)
\end{aligned}
\tag{17}
$$

We then have to integrate of the time interval alpha to have the transition probability from the state γ to the state α.

$$
\begin{aligned}
P(\alpha,i|\gamma,j) &= \int_{T_\alpha}^{T_{\alpha+1}} \frac{P_\gamma}{t_\gamma}\frac{2\lambda_{M+1,\beta,M}}{\chi_\alpha M}\Big(\sum_{\eta=0}^{\alpha-1} e^{-\sum_{\zeta=\eta+1}^{\alpha-1}\Delta_\zeta \frac{M\lambda_{M+1,\beta,M}}{\chi_\zeta}} e^{-(t-T_\alpha)\frac{M\lambda_{M+1,\beta,M}}{\chi_\alpha}} \frac{(1-e^{-\Delta_\eta\frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}} \\
&\qquad + \frac{(1-e^{-(t-T_\alpha)\frac{M\lambda_{M+1,\beta,M}}{\chi_\alpha}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\alpha}}\Big)dt \\
&= \frac{P_\gamma}{t_\gamma}\frac{2\lambda_{M+1,\beta,M}}{\chi_\alpha M}\Big(\sum_{\eta=0}^{\alpha-1} e^{-\sum_{\zeta=\eta+1}^{\alpha-1}\Delta_\zeta \frac{M\lambda_{M+1,\beta,M}}{\chi_\zeta}} \frac{(1-e^{-\Delta_\alpha\frac{M\lambda_{M+1,\beta,M}}{\chi_\alpha}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\alpha}} \frac{(1-e^{-\Delta_\eta\frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}} \\
&\qquad + \frac{(\Delta_\alpha - \frac{(1-e^{-(\Delta_\alpha\frac{M\lambda_{M+1,\beta,M}}{\chi_\alpha}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\alpha}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\alpha}}\Big) \\
&= \frac{P_\gamma}{t_\gamma}\frac{2}{M^2}\Big(\sum_{\eta=0}^{\alpha-1} e^{-\sum_{\zeta=\eta+1}^{\alpha-1}\Delta_\zeta \frac{M\lambda_{M+1,\beta,M}}{\chi_\zeta}} (1-e^{-\Delta_\alpha\frac{M\lambda_{M+1,\beta,M}}{\chi_\alpha}}) \frac{(1-e^{-\Delta_\eta\frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}} \\
&\qquad + (\Delta_\alpha - \frac{(1-e^{-(\Delta_\alpha\frac{M\lambda_{M+1,\beta,M}}{\chi_\alpha}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\alpha}}))
\end{aligned}
\tag{18}
$$

Where the recombination probability if defined as:

$$P_\gamma = (1 - e^{-Mrt_\gamma}) \tag{19}$$

$\gamma < \alpha$ We here calculate the transition probability from the state γ to a time t in the time interval α.

$$\begin{aligned}
P(t,i|t_\gamma,j) =& \int_0^{t_\gamma} \frac{P_\gamma}{t_\gamma} \frac{\lambda_{M,\beta,(M-m+1)}}{\binom{M}{m}\chi_\alpha} e^{-\int_{t_\gamma}^{t} \frac{\lambda_{M,\beta}}{\chi_v} dv} e^{-\int_u^{t_\gamma} \frac{M\lambda_{M+1,\beta,M}}{\chi_v} dv} \frac{2\lambda_{(n+1),\beta,2}}{M(\lambda_{(n+1),\beta,2} + \lambda_{(n+1),\beta,1})} du \\
=& \frac{P_\gamma}{t_\gamma} \frac{\lambda_{M,\beta,(M-m+1)}}{\binom{M}{m}\chi_\alpha} (\sum_{\eta=1}^{\gamma-1} \int_{T_\eta}^{T_{\eta+1}} e^{-\int_{t_\gamma}^{t} \frac{\lambda_{M,\beta}}{\chi_v} dv} e^{-\int_{T_{\eta+1}}^{T_\gamma} \frac{M\lambda_{M+1,\beta,M}}{\chi_v} dv} e^{-\int_{T_\gamma}^{t_\gamma} \frac{M\lambda_{M+1,\beta,M}}{\chi_v} dv} \\
& \times e^{-\int_u^{T_{\eta+1}} \frac{M\lambda_{M+1,\beta,M}}{\chi_v} dv} \frac{2\lambda_{(n+1),\beta,2}}{M(\lambda_{(n+1),\beta,2} + \lambda_{(n+1),\beta,1})} du \\
& + \int_{T_\gamma}^{t_\gamma} e^{-\int_{t_\gamma}^{t} \frac{\lambda_{M,\beta}}{\chi_v} dv} e^{-\int_u^{t_\gamma} \frac{M\lambda_{M+1,\beta,M}}{\chi_v} dv} \frac{2\lambda_{(n+1),\beta,2}}{M(\lambda_{(n+1),\beta,2} + \lambda_{(n+1),\beta,1})} du) \\
=& \frac{P_\gamma}{t_\gamma} \frac{2\lambda_{(n+1),\beta,2} e^{-\int_{t_\gamma}^{t} \frac{\lambda_{M,\beta}}{\chi_v} dv}}{M(\lambda_{(n+1),\beta,2} + \lambda_{(n+1),\beta,1})} \frac{\lambda_{M,\beta,(M-m+1)}}{\binom{M}{m}\chi_\alpha} (\sum_{\eta=1}^{\gamma-1} e^{-\sum_{\zeta=\eta+1}^{\gamma-1} \Delta_\zeta \frac{M\lambda_{M+1,\beta,M}}{\chi_\zeta}} \\
& \times e^{-(t_\gamma - T_\gamma)\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}} \frac{(1 - e^{-\Delta_\eta \frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}} \\
& + \frac{(1 - e^{-(t_\gamma - T_\gamma)\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}})
\end{aligned} \tag{20}$$

We then have to integrate of the time interval alpha to have the transition probability from the state γ to the state α.

$$\begin{aligned}
P(\alpha,i|\gamma,j) =& \int_{T_\alpha}^{T_{\alpha+1}} \frac{P_\gamma}{t_\gamma} \frac{2\lambda_{(n+1),\beta,2} e^{-\int_{t_\gamma}^{t} \frac{\lambda_{M,\beta}}{\chi_v} dv}}{M(\lambda_{(n+1),\beta,2} + \lambda_{(n+1),\beta,1})} \frac{\lambda_{M,\beta,(M-m+1)}}{\binom{M}{m}\chi_\alpha} \\
& (\sum_{\eta=1}^{\gamma-1} e^{-\sum_{\zeta=\eta+1}^{\gamma-1} \Delta_\zeta \frac{M\lambda_{M+1,\beta,M}}{\chi_\zeta}} e^{-(t_\gamma - T_\gamma)\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}} \frac{(1 - e^{-\Delta_\eta \frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}} + \frac{(1 - e^{-(t_\gamma - T_\gamma)\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}}) dt \\
=& \int_{T_\alpha}^{T_{\alpha+1}} \frac{P_\gamma}{t_\gamma} \frac{2\lambda_{(n+1),\beta,2} e^{-\int_{t_\gamma}^{T_\alpha} \frac{\lambda_{M,\beta}}{\chi_v} dv} e^{-\int_{T_\alpha}^{t} \frac{\lambda_{M,\beta}}{\chi_v} dv}}{M(\lambda_{(n+1),\beta,2} + \lambda_{(n+1),\beta,1})} \frac{\lambda_{M,\beta,(M-m+1)}}{\binom{M}{m}\chi_\alpha} \\
& (\sum_{\eta=1}^{\gamma-1} e^{-\sum_{\zeta=\eta+1}^{\gamma-1} \Delta_\zeta \frac{M\lambda_{M+1,\beta,M}}{\chi_\zeta}} e^{-(t_\gamma - T_\gamma)\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}} \frac{(1 - e^{-\Delta_\eta \frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}} + \frac{(1 - e^{-(t_\gamma - T_\gamma)\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}}) dt \\
=& \frac{P_\gamma}{t_\gamma} \frac{2\lambda_{(n+1),\beta,2} e^{-\int_{t_\gamma}^{T_\alpha} \frac{\lambda_{M,\beta}}{\chi_v} dv} (1 - e^{-\Delta_\alpha \frac{\lambda_{M,\beta}}{\chi_\alpha}})}{M(\lambda_{(n+1),\beta,2} + \lambda_{(n+1),\beta,1})} \frac{\lambda_{M,\beta,(M-m+1)}}{\binom{M}{m}\lambda_{M,\beta}} \\
& (\sum_{\eta=1}^{\gamma-1} e^{-\sum_{\zeta=\eta+1}^{\gamma-1} \Delta_\zeta \frac{M\lambda_{M+1,\beta,M}}{\chi_\zeta}} e^{-(t_\gamma - T_\gamma)\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}} \frac{(1 - e^{-\Delta_\eta \frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}} + \frac{(1 - e^{-(t_\gamma - T_\gamma)\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}})
\end{aligned} \tag{21}$$

$\gamma = \alpha, m = n + 1$

For a multiple merger event to happen, there are three possibilities. A non coalescing branch join the coalescent event, or it coalesces in the same hidden state (before or after the coalescent event).

$$
\begin{aligned}
P(\gamma, i|\gamma, j) =& \frac{P_\gamma}{t_\gamma}\int_0^{t_\gamma}\frac{(M-n)\lambda_{(n+1),\beta,1}e^{-\int_u^{t_\gamma}\frac{M\lambda_{M+1,\beta,M}}{\chi_v}dv}}{M(\lambda_{(n+1),\beta,2}+\lambda_{(n+1),\beta,1})}du + Pc_1 + Pc_2 \\
=& \frac{P_\gamma}{t_\gamma}\frac{(M-n)\lambda_{(n+1),\beta,1}}{M(\lambda_{(n+1),\beta,2}+\lambda_{(n+1),\beta,1})}(\sum_{\eta=1}^{\gamma-1}\int_{T_\eta}^{T_{\eta+1}} e^{-\int_u^{T_{\eta+1}}\frac{M\lambda_{M+1,\beta,M}}{\chi_v}dv}e^{-\int_{T_{\eta+1}}^{t_\gamma}\frac{M\lambda_{M+1,\beta,M}}{\chi_v}dv}du \\
&+\int_{T_\gamma}^{t_\gamma} e^{-\int_u^{t_\gamma}\frac{M\lambda_{M+1,\beta,M}}{\chi_v}dv}du) + Pc_1 + Pc_2 \\
=& \frac{P_\gamma}{t_\gamma}\frac{(M-n)\lambda_{(n+1),\beta,1}}{M(\lambda_{(n+1),\beta,2}+\lambda_{(n+1),\beta,1})}(\sum_{\eta=1}^{\gamma-1}\frac{(1-e^{-\Delta_\eta\frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}}e^{-\int_{T_{\eta+1}}^{t_\gamma}\frac{M\lambda_{M+1,\beta,M}}{\chi_v}dv} \\
&+\frac{(1-e^{-(t_\gamma-T_\gamma)\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}}) + Pc_1 + Pc_2 \\
=& \frac{P_\gamma(M-n)\lambda_{(n+1),\beta,1}}{t_\gamma M(\lambda_{(n+1),\beta,2}+\lambda_{(n+1),\beta,1})}(\sum_{\eta=1}^{\gamma-1}\frac{(1-e^{-\Delta_\eta\frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}}e^{-\sum_{\zeta=\eta+1}^{\gamma-1}\frac{\Delta_\zeta M\lambda_{M+1,\beta,M}}{\chi_\zeta}}e^{-(t_\gamma-T_\gamma)\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}} \\
&+\frac{(1-e^{-(t_\gamma-T_\gamma)\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}}) + Pc_1 + Pc_2 \qquad (22)
\end{aligned}
$$

Pc_1 is the probability that a recombination happens before the first coalescent event in the non coalescing branch, it then coalesce before the current first coalescent event but in the same hidden state (resulting in a multiple merger coalescent because of the discretized time)

Pc_2 is the probability that a recombination happens before $T_{\gamma+1}$ in the non coalescing branch, it then coalesce after the current first coalescent event but in the same hidden state (resulting

in a multiple merger coalescent because of the discretized time)

$$
\begin{aligned}
Pc_1 = \int_{T_\gamma}^{t_\gamma} \frac{P_\gamma}{t_\gamma} \frac{\lambda_{2,\beta}}{\chi_\gamma M} \Big(\sum_{\eta=0}^{\gamma-1} e^{-\sum_{\zeta=\eta+1}^{\gamma-1} \Delta_\zeta \frac{M\lambda_{M+1,\beta,M}}{\chi_\zeta}} e^{-(t-T_\gamma)\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}} \frac{(1-e^{-\Delta_\eta \frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}} \\
+ \frac{(1-e^{-(t-T_\gamma)\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}} \Big) dt \\
= \frac{P_\gamma}{t_\gamma} \frac{\lambda_{2,\beta}}{\chi_\gamma M} \Big(\sum_{\eta=0}^{\gamma-1} e^{-\sum_{\zeta=\eta+1}^{\gamma-1} \Delta_\zeta \frac{M\lambda_{M+1,\beta,M}}{\chi_\zeta}} \frac{(1-e^{-(t_\gamma-T_\gamma)\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}} \frac{(1-e^{-\Delta_\eta \frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}} \\
+ \frac{((t_\gamma - T_\gamma) - \frac{(1-e^{-(t_\gamma-T_\gamma)\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}} \Big) \\
= \frac{P_\gamma}{t_\gamma} \frac{1}{M^2} \Big(\sum_{\eta=0}^{\gamma-1} e^{-\sum_{\zeta=\eta+1}^{\gamma-1} \Delta_\zeta \frac{M\lambda_{M+1,\beta,M}}{\chi_\zeta}} (1-e^{-(t_\gamma-T_\gamma)\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}}) \frac{(1-e^{-\Delta_\eta \frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}} \\
+ ((t_\gamma - T_\gamma) - \frac{(1-e^{-(t_\gamma-T_\gamma)\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}}) \Big)
\end{aligned}
\tag{23}
$$

$$
\begin{aligned}
Pc_2 = \int_{t_\gamma}^{T_{\gamma+1}} \frac{P_\gamma}{t_\gamma} \frac{\lambda_{(n+1),\beta,2} e^{-\int_{t_\gamma}^{t} \frac{\lambda_{2,\beta}}{\chi_\gamma} dv}}{M(\lambda_{(n+1),\beta,2} + \lambda_{(n+1),\beta,1})} \frac{\lambda_{2,\beta}}{\chi_\gamma} \\
\Big(\sum_{\eta=1}^{\gamma-1} e^{-\sum_{\zeta=\eta+1}^{\gamma-1} \Delta_\zeta \frac{M\lambda_{M+1,\beta,M}}{\chi_\zeta}} e^{-(t_\gamma-T_\gamma)\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}} \frac{(1-e^{-\Delta_\eta \frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}} + \frac{(1-e^{-\Delta_\gamma \frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}} \Big) dt \\
= \frac{P_\gamma}{t_\gamma} \frac{\lambda_{(n+1),\beta,2}(1-e^{-(T_{\gamma+1}-t_\gamma)\frac{\lambda_{2,\beta}}{\chi_\gamma}})}{M(\lambda_{(n+1),\beta,2} + \lambda_{(n+1),\beta,1})} \\
\Big(\sum_{\eta=1}^{\gamma-1} e^{-\sum_{\zeta=\eta+1}^{\gamma-1} \Delta_\zeta \frac{M\lambda_{M+1,\beta,M}}{\chi_\zeta}} e^{-(t_\gamma-T_\gamma)\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}} \frac{(1-e^{-\Delta_\eta \frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}} + \frac{(1-e^{-\Delta_\gamma \frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}} \Big)
\end{aligned}
\tag{24}
$$

$\gamma = \alpha, m = n-1$

$$P(\gamma, i|\gamma, j) = \frac{P_\gamma}{t_\gamma}\int_0^{t_\gamma} \frac{\lambda_{(n+1),\beta,2} e^{-\int_u^{t_\gamma}\frac{M\lambda_{M+1,\beta,M}}{\chi_v}dv}}{n(\lambda_{(n+1),\beta,2}+\lambda_{(n+1),\beta,1})}du$$

$$= \frac{P_\gamma}{t_\gamma}\frac{\lambda_{(n+1),\beta,2}}{n(\lambda_{(n+1),\beta,2}+\lambda_{(n+1),\beta,1})}(\sum_{\eta=1}^{\gamma-1}\int_{T_\eta}^{T_{\eta+1}} e^{-\int_u^{T_{\eta+1}}\frac{M\lambda_{M+1,\beta,M}}{\chi_v}dv} e^{-\int_{T_{\eta+1}}^{t_\gamma}\frac{M\lambda_{M+1,\beta,M}}{\chi_v}dv}du$$

$$+\int_{T_\gamma}^{t_\gamma} e^{-\int_u^{t_\gamma}\frac{M\lambda_{M+1,\beta,M}}{\chi_v}dv}du)$$

$$= \frac{P_\gamma}{t_\gamma}\frac{\lambda_{(n+1),\beta,2}}{n(\lambda_{(n+1),\beta,2}+\lambda_{(n+1),\beta,1})}(\sum_{\eta=1}^{\gamma-1}\frac{(1-e^{-\Delta_\eta\frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}} e^{-\int_{T_{\eta+1}}^{t_\gamma}\frac{M\lambda_{M+1,\beta,M}}{\chi_v}dv} + \frac{(1-e^{-(t_\gamma-T_\gamma)\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}})$$

$$= \frac{P_\gamma}{t_\gamma}\frac{\lambda_{(n+1),\beta,2}}{n(\lambda_{(n+1),\beta,2}+\lambda_{(n+1),\beta,1})}(\sum_{\eta=1}^{\gamma-1}\frac{(1-e^{-\Delta_\eta\frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\eta}} e^{-\sum_{\zeta=\eta+1}^{\gamma-1}\Delta_\zeta\frac{M\lambda_{M+1,\beta,M}}{\chi_\zeta}} e^{-(t_\gamma-T_\gamma)\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}}$$

$$+\frac{(1-e^{-(t_\gamma-T_\gamma)\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}})}{\frac{M\lambda_{M+1,\beta,M}}{\chi_\gamma}}) \tag{25}$$

$\gamma = \alpha, m = n$

$$P(\gamma, j|\gamma, j) = 1 - \sum_{\alpha\neq\gamma, i\neq j} P(\alpha, i|\gamma, j) \tag{26}$$

1.5 Emission Matrix

To compute the emission probabilities, we need the external branch length at each position of the genome (noted as Ts). Ts is estimated with an additional HMM as described in [4].

1.5.1 M=3

One can only observe 4 possibilities. Observation 0, no mutation within the 3 individual. Observation 1 (2,3), individual 1 (2,3) is different from individual 2 and 3 (1 and 3 , 1 and 2).

if the first coalescent event involves 2 individuals we have :

$$P(0|\gamma) = e^{-\mu(Ts)} \tag{27}$$

$i \in 1, 2, 3$. Mutation occurred and did not occurred in the first coalescent event.

$$P(i|\gamma, i) = (1 - e^{-\mu(Ts-2t_\gamma)}) \tag{28}$$

$i \in 1, 2, 3; \bar{i} \neq i$. Mutation occurred and is in the first coalescent event.

$$P(i|\gamma, \bar{i}) = (1 - e^{-\mu(2t_\gamma)}) \tag{29}$$

if the first coalescent event involves 3 individuals we have :

$$P(0|\gamma) = e^{-\mu(3t_\gamma)} \tag{30}$$

$i \in 1, 2, 3$

$$P(i|\gamma) = 1 - e^{-\mu(3t_\gamma)} \tag{31}$$

1.5.2 M=4

One can only observe 8 possibilities as we only focus on SNPs. Observation 0, no mutation within the 4 individual. Observation 1 (2,3,4), individual 1 (2,3,4) is different from all other individual. Observation 5 to 7 , two individual are different from the other two.

if the first coalescent event involves 4 individuals:

$$P(0|\gamma) = e^{-\mu(4t_\gamma)} \tag{32}$$

$i \in 1, 2, 3, 4$

$$P(i|\gamma) = 1 - e^{-\mu(4t_\gamma)} \tag{33}$$

if the first coalescent event involves 3 individuals:

$$P(0|\gamma) = e^{-\mu(Ts)} \tag{34}$$

$i \in 1, 2, 3, 4$

$$P(i|\gamma, \bar{i}) = (1 - e^{-\mu(t_\gamma)}) \tag{35}$$

$i \in 1, 2, 3, 4$

$$P(i|\gamma, i) = (1 - e^{-\mu(Ts-3t_\gamma)}) \tag{36}$$

if the first coalescent event involves 2 individuals:

No mutation occurred.

$$P(0|\gamma) = e^{-\mu(Ts)} \tag{37}$$

$i \in 1, 2, 3, 4$ and mutation is within on one of the individual that coalesce.

$$P(i|\gamma) = (1 - e^{-\mu(t_\gamma)}) \tag{38}$$

$i \in 1, 2, 3, 4$ and mutation is not on one of the individual that coalesce.

$$P(i|\gamma) = (1 - e^{-\mu(\frac{(Ts-4t_\gamma)}{2}+t_\gamma)}) \tag{39}$$

$i \in 5, 6, 7$ and the two individual coalescing are identical.

$$P(i|\gamma) = e^{-\mu(Ts)} \tag{40}$$

$i \in 5, 6, 7$ and the two individual coalescing are different,then two mutation must occur, which has probability 0.

$$P(i|\gamma) = 0 \tag{41}$$

References

[1] Jean-Stephane Dhersin, Fabian Freund, Arno Siri-Jegousse, and Linglong Yuan. On the length of an external branch in the Beta-coalescent. *Stochastic Processes and their Applications*, 123(5):1691–1715, MAY 2013.

[2] Fabian Freund. Cannings models, population size changes and multiple-merger coalescents. *Journal of mathematical biology*, 80(5):1497–1521, 2020.

[3] Götz Kersting, Jason Schweinsberg, and Anton Wakolbinger. The evolving beta coalescent. *Electronic Journal of Probability*, 19:1–27, 2014.

[4] Stephan Schiffels and Richard Durbin. Inferring human population size and separation history from multiple genome sequences. *Nature Genetics*, 46(8):919–925, AUG 2014.

[5] J Schweinsberg. Coalescent processes obtained from supercritical Galton-Watson processes. *Stochastic Processes and their Applications*, 106(1):107–139, JUL 2003.

Supplementary Text S2: Description of the Graph Neural Network Approach (GNN*coal*)

Kevin Korfmann, Thibaut Sellinger, Fabian Freund,
Matteo Fumagalli, Aurélien Tellier

1 Brief Introduction to neural network

We wish to define a neural network that has some learnable parameters (i.e. weights) to predict parameters of interest. In this study, we focus on the past variation of population size and on the α parameter of the Beta distribution shaping the offspring distribution [8]. To define a neural network, we first generate a training dataset, *i.e.* a dataset containing information to learn from (the ancestral recombination graph) and predictive variables of interest (past demography and α parameter). We then process inputs through the neural network and compute the loss, which evaluates the distance between predictions and ground truth. Afterwards, we propagate gradients back into the network's parameters and update its weights. We now describe further details of this procedure.

2 Datasets

2.1 Set 1: Variable demography and MMC

To build our training data set, we rely on the efficient coalescent and sequence simulator *msprime* [1, 4]. We simulated 10 haploid genomes for a total of 2×10^4 demographic scenarios with 100 replicates each. We followed the procedure outlined in Figure S1 in Supplementary text S2 (see below) to define our scenarios. To ensure that each simulation contains sufficient data points, we imposed that 95% of the 100 replicates had to contain at least 500 trees. This procedure resulted in a training data set containing between 9.5×10^8 and 1×10^9 coalescent trees. Each simulated data point was stored as a tree sequence and converted into the graph format (*i.e.* *Pytorch Geometric* data object for the forward pass. This procedure drastically decreased the storage space requirements at the marginal cost of training speed. Furthermore, we trained our network over a range of the multiple merger parameter α from 1.01 to 1.99.

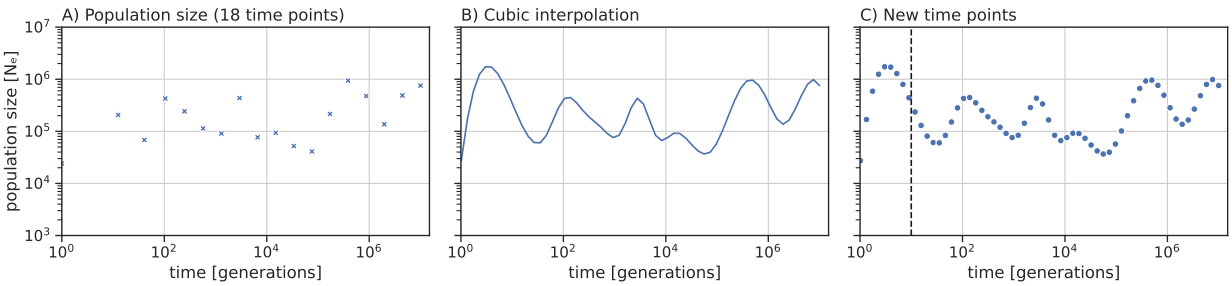

Fig. S1. Demography sampling process Population size (y-axis) changes through time in generations (x-axis). Demography sampling process according to [2, 7] (A) and cubic interpolation (B), and re-sampling for 60 uniformly spaced points. The sampling process is repeated if any population size is outside the 10^4 and 10^7 population size window.

2.2 Set 2: MMC versus selection under constant population size

To discern between multiple merger and selection, we simulated 1000 replicates under constant population size ($N_e = 10^5$) by randomly choosing between Kingman and β-coalescent models. Additionally, we randomly chose one of the selection regimes when a

Kingman coalescent scenario was drawn. For each scenario, we simulated 100 replicates with 99% of each replicate containing at least 2,000 trees (*i.e.* a batch size of 2000). This procedure resulted in a training data set of more than 1.98×10^8 coalescent trees. An additional validation dataset was generated consisting of 1,000 simulations under the same scenarios. A script for replicating these simulations is available at `https://github.com/kevinkorfmann/GNNcoal-analysis`.

3 Training

The demography inference model was trained for two epochs, an epoch being defined by iterating one time over the training dataset. *i.e.* the data is processed twice, once by the untrained neural network and a second time after the first epoch. The model to infer the α parameter was trained on 10% of the dataset used for demography inference as we found the procedure to be sufficient for accurate estimates. We set the batch size to 500 (*i.e.* number of trees processed before updating the neural network parameters). Jupyter notebooks to perform the training are accessible at `https://github.com/kevinkorfmann/GNNcoal-analysis`.

4 Neural network

The neural network is implemented in *PyTorch* [6] using the extension *Pytorch Geometric* for the graph convolution operation and differential hierarchical pooling strategy [3, 5, 9]. For training and inference, each coalescent tree was interpreted as an undirected graph, with each ancestral node or leaf having edges to their parent or children, and vice versa. Each node i contains a feature vector (a feature vector being the predicted values by the neural network) $\mathbf{x_i}$ of size 60 (number of desired estimated parameters), which are horizontally stacked into a feature vector matrix (*i.e.* each node/coalescent event has feature vector) $\mathbf{X}$ and initialized with ones on the diagonal and zeros everywhere else. Each feature vector are unique. As mentioned above, we chose a feature vector of size 60, which corresponds to the final output dimension of the demography network. A convolution of an individual feature vector $\mathbf{x_i}$ of node i is computed following [5]:

$$\mathbf{x}_i' = \mathbf{\Theta}^\top \sum_{j \in \mathcal{N}(v) \cup \{i\}} \frac{e_{j,i}}{\sqrt{\hat{d}_j \hat{d}_i}} \mathbf{x}_j \tag{1}$$

With $\hat{d}_i = 1 + \sum_{j \in \mathcal{N}(i)} e_{j,i}$ and $e_{j,i}$ being the pivot edge from j to i. The node features $\mathbf{x_i}$ are transformed by a learnable weight matrix Θ, normalized according to the degrees of the respective nodes, and summed over the neighboring nodes. Alongside this node-wise formulation, the same computation can be written as matrix-operations. The latter will be the notation used to explain the following pooling strategy:

$$\mathbf{X}' = \hat{\mathbf{D}}^{-1/2} \hat{\mathbf{A}} \hat{\mathbf{D}}^{-1/2} \mathbf{X} \mathbf{\Theta}, \tag{2}$$

Where $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ represents the adjacency matrix and $\hat{D}_{ii} = \sum_{j=0} \hat{A}_{ij}$, represent the diagonal degree matrix. Here, the edge information enters the convolution as part of the values of the adjacency matrix. Both representations are valid descriptions of the same convolution [3].

We defined our GNN as a series of three consecutive graph convolutions, with input, hidden and output dimensions of size 60, each convolution separated by a one-dimensional batch normalization step. Increasing the number of convolutions above a certain value will lead to a *smoothing* of the feature vectors, as the information of more distant neighbours will be taken into account for each node [5]. The output of each convolution is then concatenated and passed through-linear activation and ReLu-function ($Relu(z) = max(0, z)$). The complete implementation is available at `https://github.com/kevinkorfmann/GNNcoal`.

Because each node has a feature vector (*i.e.* each coalescent event of a coalescent tree lead to a specific prediction), the feature vector matrix (*i.e.* the collection of estimated demography parameters from a coalescent tree) needs to be processed to obtain a consensus predicted demography. This step is called the pooling strategy. A detailed description of the pooling strategy used in our study is formulated in [9] and short description can be found below.

We compress the number of nodes hierarchically in a step-wise manner, from the l state to $l + 1$ state. Close nodes in the original graph are summarized by a supernode in a new smaller graph, ideally having only one supernode at the end with one feature vector, containing our variables of interest. Thus, this downsizing can be formulated as decreasing the original adjacency matrix $\mathbf{A}$ of a given genealogy to a smaller subgraph $\mathbf{A}^{l+1}$ ($\mathbf{A} \in \mathbb{R}^{n\times n} \rightarrow \mathbf{A}^{l+1} \in \mathbb{R}^{m\times m}$ with n, m being the number of nodes in each graph and $\mathbf{Z} \in \mathbb{R}^{n\times d} \rightarrow \mathbf{Z}^{l+1} \in \mathbb{R}^{m\times d}$ with condition of $m < n$; here d being the length of the feature vector). To achieve this, we cluster a set of nodes in the original genealogy into a smaller subset of supernodes. A cluster assignment matrix $\mathbf{S} \in \mathbb{R}^{n^l\times n^{l+1}}$ (with n^l is the number of clusters in the current step and n^{l+1} the number of clusters in the next step). In the first iteration, the number of clusters n^l is equal to the number of nodes, and it is *learned* using an GNN_{pool}. This means that the criterion to compress individual nodes is dependent on the overall objective of network, which is to reduce the loss function with respect to predicting parameters of interest. The specific computation using two GNNs ((GNN_{pool} and GNN_{embed})) is explained in the following:

1. We compute the feature vectors $\mathbf{Z^l}$ using GNN_{embed} at step l (equation (3)).

$$\mathbf{Z}^l = GNN_{l,embed}(\mathbf{A}^l, \mathbf{X}^l) \tag{3}$$

2. Next we compute the cluster assignment matrix, by computing "feature vectors" and applying the row-wise softmax function ($\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{d} e^{z_j}} \quad for\ i = 1, 2, \ldots, d$) as normalization step. This results is a probability distribution of each node to be part of the next cluster.

$$\mathbf{S}^l = softmax(GNN_{l,pool}(\mathbf{A}^l, \mathbf{X}^l)) \tag{4}$$

3. Lastly, the new adjacency matrix $\mathbf{A}^{l+1}$ and corresponding feature vector matrix $\mathbf{X}^{l+1}$ are calculated ((5) and (6)). Notably, in equation $\mathbf{A^l}$ is multiplied twice with $\mathbf{S}$ to ensure that $\mathbf{A}^{l+1}$ is square (*i.e.* the number of columns is equal to the number of rows).

$$\mathbf{X}^{l+1} = \mathbf{S}^{l^T}\mathbf{Z}^l \rightarrow \mathbb{R}^{n^{l+1}\times d} \tag{5}$$

$$\mathbf{A}^{l+1} = \mathbf{S}^{l^T}\mathbf{A^l}\mathbf{S}^l \rightarrow \mathbb{R}^{n^{l+1}\times n^{l+1}} \tag{6}$$

In our approach, we used three consecutive pooling iterations $l = 1, ..., 3$. At each iteration, a graph is compressed down to 30% of the remaining nodes. In total, six GNNs (three embedding GNNs and three pooling GNNs) contribute to the number of learnable parameters, followed by two linear layers with interspaced ReLU-activation to compress the output into the desired demographic-time window dimension. The demography model implementation can be found at `https://github.com/kevinkorfmann/GNNcoal`.

We now focus on the α inference and the classification. The model for inferring the α parameter adds three linear layers to the demography network model, while the classification model adds two linear layers. To illustrate the simplicity to estimate α and classify models, we show the code for reducing the number of dimensions below.

α inference model:

```
class AlphaInferenceModel(nn.Module):

    def __init__(self, DemographyNet, time_window=60):
        super().__init__()
        self.l1 = nn.Linear(time_window, time_window//2)
        self.l2 = nn.Linear(time_window//2, time_window//4)
        self.l3 = nn.Linear(time_window//4, 1)
        self.DemographyNet = DemographyNet

    def forward(self, batch):
        x = self.DemographyNet(batch)
        return self.l3(F.relu(self.l2(F.relu(self.l1(x)))))
```

Classification model:

```
class ClassificationModel(nn.Module):

    def __init__(self, DemographyNet, num_classes, time_window=60):
        super().__init__()
        self.l1 = nn.Linear(time_window, time_window//2)
        self.l2 = nn.Linear(time_window//2, num_classes)
        self.DemographyNet = DemographyNet

    def forward(self, batch):
        x = self.DemographyNet(batch)
        return self.l2(F.relu(self.l1(x)))
```

5 Time window

The output of the neural network is a fixed-size vector containing the values of inferred population size. However, not all trees contain coalescent events falling into these pre-specified time-bins. As a consequence, the population size can only be inferred for simulations where trees contain coalescence events in those specific bins. A multi-step heuristic has been developed to obtain an estimate of the inference window for each simulation, respectively. The procedure is summarized as:

```
    1. Simulate 100 repetitions for a demographic parameter set.
    2. Count number of coalescent events for first 500 trees for each simulation.
    3. Smoothen coalescent event vectors with sliding window*.
    4. Only retain the largest consecutive coalescent time-window per repetition.
    5. Create Boolean mask by retaining the time-windows with at least X coalescent events
    along repetition axis.

    *Sliding window: If left and right of pivot element up to Y coalescent events
    are present, retain the pivot element, move one step to the right and repeat.
```

This heuristically constructed mask is used to zero out all inferences outside of the coalescent events before computing the RMSE-loss function and back-propagating the weights (avoiding under- and over-fitting). In practice, a single row of the mask, is multiplied into a matrix of dimensions equal to the number of trees times the number of windows (here: 500x60). With this procedure, each tree receives the same masked loss based on all repetitions (to balance the training data set). Coalescent events and masks for constant demography of size 10^6 are provided in Figure S2 below.

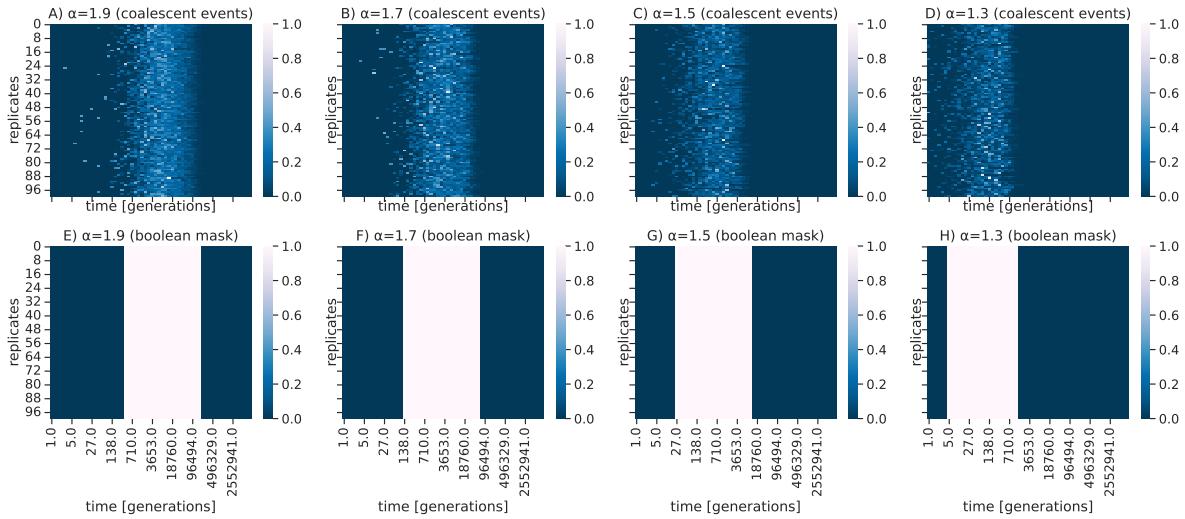

Fig. S2. Coalescent events and boolean masks A)-D) The number of coalescent events for 100 repetitions (y-axis) across the first 500 trees of a simulation for 60 time windows (x-axis). E)-H): Boolean mask after applying a heuristic to determine a suitable time-window for demography inference.

During the inference, when only one sample is available, the masking procedure was approximated by calculating the mean of the log-scaled node times of the first 500 trees and by taking two standard deviations at both sides from the mean to form a similar time-window even if only one repetition is available (see below).

```
def alternative_coalescent_mask(ts, population_time, x_times_std=2, n_trees=500):

    trees = ts.aslist()[0:n_trees]
    nodes_n_trees = []
    for tree in trees:
        nodes_n_trees += list(tree.nodes())

    node_times = [ts.get_time(node.id) for node in ts.nodes() if node.id >= ts.num_samples and
        node.id in nodes_n_trees]

    log_node_times = np.log(node_times)
    mean = log_node_times.mean()
    std = log_node_times.std()
    lowerbound = np.exp(mean-x_times_std*std)
    upperbound = np.exp(mean+x_times_std*std)
    mask1 = population_time > lowerbound
    mask2 = population_time < upperbound
    mask = np.logical_and(mask1, mask2)
    return mask
```

References

[1] Franz Baumdicker, Gertjan Bisschop, Daniel Goldstein, Graham Gower, Aaron P. Ragsdale, Georgia Tsambos, Sha Zhu, Bjarki Eldon, E. Castedo Ellerman, Jared G. Galloway, Ariella L. Gladstein, Gregor Gorjanc, Bing Guo, Ben Jeffery, Warren W. Kretzschumar, Konrad Lohse, Michael Matschiner, Dominic Nelson, Nathaniel S. Pope, Consuelo D. Quinto-Cortes, Murillo F. Rodrigues, Kumar Saunack, Thibaut Sellinger, Kevin Thornton, Hugo van Kemenade, Anthony W. Wohns, Yan Wong, Simon Gravel, Andrew D. Kern, Jere Koskela, Peter L. Ralph, and Jerome Kelleher. Efficient ancestry and mutation simulation with msprime 1.0. *GENETICS*, 220(3), MAR 3 2022.

[2] Simon Boitard, Willy Rodríguez, Flora Jay, Stefano Mona, and Frédéric Austerlitz. Inferring population size history from large samples of genome-wide molecular data - an approximate bayesian computation approach. 12(3):e1005877.

[3] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with PyTorch geometric.

[4] Jerome Kelleher, Kevin R. Thornton, Jaime Ashander, and Peter L. Ralph. Efficient pedigree recording for fast population genetics simulation. 14(11):e1006581.

[5] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. 2016.

[6] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. October 2017.

[7] Théophile Sanchez, Jean Cury, Guillaume Charpiat, and Flora Jay. Deep learning for population size history inference: Design, comparison and combination with approximate bayesian computation. 21(8):2645–2660.

[8] J Schweinsberg. Coalescent processes obtained from supercritical Galton-Watson processes. *Stochastic Processes and their Applications*, 106(1):107–139, JUL 2003.

[9] Rex Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L. Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling.