Coalescence and Translation:

A Language Model for Population Genetics

Kevin Korfmann[1], Nathaniel S. Pope[1], Melinda Meleghy[2],
Aurélien Tellier[2], Andrew D. Kern[1]

[1]University of Oregon, Institute of Ecology and Evolution, Eugene, USA
[2]Technical University of Munich, Department of Life Science Systems, Munich, Germany

Abstract

Probabilistic models such as the sequentially Markovian coalescent (SMC) have long provided a powerful framework for population genetic inference, enabling reconstruction of demographic history and ancestral relationships from genomic data. However, these methods are inherently specialized, relying on predefined assumptions and/or limited scalability. Recent advances in simulation and deep learning provide an alternative approach: learning directly to generalize from synthetic genetic data to infer specific hidden evolutionary processes. Here we reframe the inference of coalescence times as a problem of translation between two biological languages: the sparse, observable patterns of mutation along the genome and the unobservable ancestral recombination graph (ARG) that gave rise to them. Inspired by large language models, we develop `cxt`, a decoder-only transformer that autoregressively predicts coalescent events conditioned on local mutational context. We show that `cxt` performs on par with state-of-the-art MCMC-based likelihood models across a broad range of demographic scenarios, including both in-distribution and out-of-distribution settings. Trained on simulations spanning the `stdpopsim` catalog, the model generalizes robustly and enables efficient inference at scale, producing over a million coalescence predictions in minutes. In addition `cxt` produces a well calibrated approximate posterior distribution of its predictions, enabling principled uncertainty quantification. Our work moves towards a foundation model for population genetics, bridging deep learning and coalescent theory to enable flexible, scalable inference of genealogical history from genomic data.

Introduction

The stochastic process of genetic lineages tracing back to common ancestors over time gives rise to a complex structure known as the Ancestral Recombination Graph (ARG), with branches that merge through coalescence or divide through recombination (Kingman 1982; Hudson 1983; R. C. Griffiths and Paul Marjoram 1997; R. C. Griffiths 1991). In the absence of mutations, these branches remain entirely concealed. However, mutations along the branches of the genealogy offer a noisy window into the hidden ARG, potentially enabling us to infer the underlying set of topologies across the genome, as well as date the timing of coalescent events and mutations. Currently the field of population genetics is undergoing a revolution in our ability to infer the ARG that underlies a sample, and over the past few years we have obtained our first glimpses into reconstructions of whole genome genealogies (Kelleher et al. 2019; Speidel et al. 2019; B. C. Zhang et al. 2023; Deng, Nielsen, and Yun S. Song 2024).

Inferring the ARG (or approximations of it) has long been a central goal in population genetics. One of the earliest attempts was by Griffiths and Marjoram 1996, who used Markov chain simulations of the coalescent process for ancestral inference. Subsequent work conceptualizing recombination as a point process along the genome (Wiuf and Hein 1999), shifted focus toward emphasizing the spatial aspects of recombination, and laid the groundwork for more efficient approximations of the coalescent with recombination such as the sequentially Markovian coalescent (SMC) (McVean and Cardin 2005). The SMC approximation has been pivotal in the development of numerous inference methods, beginning with approaches requiring only a single diploid genome to investigate demographic changes, such as PSMC (H. Li and Durbin 2011) (which effectively infers an ARG for two haploid samples). Since, SMC-type methods have been extended in numerous ways including: to multiple genomes (Schiffels and Durbin 2014; Malaspinas et al. 2016), integration of site-frequency spectrum information (Terhorst, Kamm, and Yun S. Song 2017), the inclusion of species-specific life-history traits (e.g. T. P. P. Sellinger et al. 2020; Korfmann et al. 2024; T. Sellinger, Johannes, and Tellier 2024), and as well as algorithms that more efficiently scale as a function of sample size (Palamara et al. 2018; Schweiger and Durbin 2023; Terhorst 2024).

While the SMC approximation leads to tractable likelihoods for a limited set of models, these typically omit processes that are ubiquitous in nature - most notably population structure, selective pressures, or complex demographic events. Extending the SMC (and inference methods that rely upon it) to accommodate such features often requires substantial re-engineering. In contrast, simulation-based inference offers a flexible alternative: it bypasses the need for tractable likelihoods by estimating directly from simulated data, thus enabling inference under richer, more realistic models without the burden of analytical reformulation. Methods such as approximate Bayesian computation (ABC), are a key approach in population genetics (e.g. Beaumont, W. Zhang, and Balding 2002), and recent work has brought modern machine learning approaches to bear on simulation-based inference (as reviewed in Schrider and Kern 2018; Korfmann, Gaggiotti, and Fumagalli 2023). Broadly put, the approach is to associate simulated data with the parameters of the model that produced the simulation, either through rejection sampling, regression, or a combination of the two and to subsequently apply the model on empirical data. ABC has proven very successful in a variety of population genetic inference tasks including demographic inference (e.g. Boitard et al. 2016; Jay, Boitard, and Austerlitz 2019). However, ABC has known limitations, perhaps most importantly that the data must be reduced to a compact representation to avoid the curse of dimensionality, losing information relevant to inference in the process. This constraint is particularly problematic in sequence-based inference tasks - such as those targeted by PSMC-like models - where the input is inherently high-dimensional and information loss during compression can hinder predictive accuracy.

A powerful alternative to standard ABC is to utilize deep learning (DL) for simulation-based inference,

which has proven its adaptability across diverse domains (Schmidhuber 2022). In population genetics specifically, the utility of deep learning for model-based inference has been well-documented (as reviewed in Korfmann, Gaggiotti, and Fumagalli 2023). DL models may be rapidly trained (often within minutes), by leveraging widely available hardware suitable for common neural network architectures including fully connected networks (Sheehan and Yun S Song 2016; Battey, Ralph, and Kern 2020), convolutional neural networks (Kern and Schrider 2018; Flagel, Brandvain, and Schrider 2019; Torada et al. 2019; Sanchez et al. 2021; Whitehouse and Schrider 2023; Smith et al. 2023; Smith and Kern 2023; Nait Saada et al. 2023), recurrent neural networks (Adrion, Galloway, and Kern 2020), and variational autoencoders (Battey, Coffing, and Kern 2021). Traditional deep learning architectures in population genetics are typically trained for narrow, task-specific inference–such as estimating a particular demographic parameter–and often fail to generalize beyond the distribution of their training data. This limitation becomes particularly pronounced in sequence-based settings, where inputs are high-dimensional and modeling general dependencies is critical.

Transformer-based large language models (LLMs) offer a promising alternative, by enhancing prediction capabilities through the learning of stochastic processes as opposed to classical and scenario-constrained point estimations (Nait Saada et al. 2023, e.g.,). Transformer LLMs are perhaps best known in the context of pretrained generative models for human language such as GPT (Radford and Narasimhan 2018). These models are pretrained in that they are first trained on a "pretext" task, such as next token prediction, that requires the LLM to learn context among observations (tokens) in a sequence locally and their relation to higher-level associations in the space of language. Modern LLMs are in part wildly successful due to their massive numbers of parameters (e.g. hundreds of billions) that enable them to effectively act as databases for associations among many tokens and thus to generalize across contextual settings. Here we describe a pretrained generative model for population genetics that leverages a small language model (10-20M parameters) to effectively translate from mutational patterns across chromosomes to coalescent time estimates. This opens the door to a foundation model paradigm: pretraining a flexible transformer on a diverse range of coalescent simulations, followed by fine-tuning for specific evolutionary tasks, if necessary.

Recently, pretrained foundation models have been developed for genomics yielding powerful DNA language models. Notably, models such as GPN-MSA and Evo 2 are trained directly on genomic sequence data to learn rich, evolution-informed representations that support downstream tasks like variant effect prediction and sequence design (Brixi et al. 2025; Benegas et al. 2025). These models treat DNA as a textual corpus, applying token-based objectives such as masked language modeling to extract statistical structure from naturally occurring sequences. In this work, we introduce `cxt` the first language model for coalescent estimation. `cxt` is a decoder-only transformer model inspired by GPT-2 and designed for pairwise time to the most recent common ancestor (TMRCA) prediction - a core component of genealogical inference. In contrast to DNA language models that are pretrained on the task of masked base prediction from a collection of sequenced genomes, our pretext task leverages synthetic data simulated under explicit population genetic models to predict the *next* pairwise TMRCA along a sequence. Rather than modeling the sequence itself, `cxt` aims to recover latent genealogical structure, providing a flexible and generalizable model for downstream inference. While we borrow the terminology of foundation models to emphasize extensibility and modularity with respect to the vast training corpus, `cxt` is more precisely a foundational model for a key subcomponent of population genetic inference, that is pretrained on a broad distribution of demographic scenarios and designed for efficient fine-tuning.

To enable this inference, we define a novel task, analogous to *next-token prediction* in language modeling, which we term *next-coalescence prediction*. In this formulation, the model predicts the next coalescence time along a sequence of previously predicted events, across discrete sequence space, conditioned on local mutation densities within a fixed context window. While this does not reconstruct full genealogical

topologies, it enables a structured prediction task that translates between observed mutation patterns and pairwise coalescence times - akin to approaches such as PSMC. In this sense, the model offers a way to navigate between the observed distribution of mutation densities and the latent structure of the ARG by means of a sequence of coalescence event prediction.

To evaluate this approach, we focused on three criteria: (1) **accuracy**– we aim for performance comparable to or better than recent theory-driven methods such as `Singer` and `Gamma-SMC`; (2) **robustness**– particularly under model misspecification and out-of-distribution scenarios; and highlighted specifically (3) **capacity**– we wish to assess a model's ability to generalize across a range of diverse simulations. Using extensive simulations and application to human genomic variation data, we demonstrate that `cxt` provides state-of-the-art performance for the task of inferring coalescence times locally across recombining chromosomes, and that inference can be done very efficiently in parallel with GPU acceleration. The generative nature of `cxt` enables rapid sampling of the learned probability distribution for uncertainty estimation which we show to be well calibrated. Further, simulation experiments demonstrate that `cxt` performs exceptionally well across a wide range of models, including the entire `stdpopsim` catalog. `cxt` is generally robust to out-of-sample prediction, but offers substantial gains when fine-tuned. Finally, in an application to human 1000 Genome data (Consortium et al. 2010; Auton and Salcedo 2015), we demonstrate how `cxt` can recover known outliers in time to the most recent common ancestor (e.g., the HLA region) along the genome.

Results

We begin by presenting a conceptual schematic that adapts the general language modeling paradigm to population genetics, and draw parallels with SMC-based coalescent modeling to define our next-coalescence prediction task (Figure 1). We then benchmark `cxt`'s accuracy against current fast inference methods, specifically `Singer` and `Gamma-SMC` (Deng, Nielsen, and Yun S. Song 2024; Schweiger and Durbin 2023). Next, we evaluate the model's database-like capabilities, training and evaluating using the nearly complete `stdpopsim` v0.2 and v0.3 catalogs across diverse species, demographic scenarios, and genetic maps (excluding only bacteria and two species with extreme recombination rates that preclude large-scale simulation). We assess the global accuracy of TMRCA prediction by comparing true and estimated coalescence rates across simulations. Finally, we apply `cxt` to real genomic data, estimating coalescent times across two chromosomes from the 1000 Genomes Project, using pairs of haploid sequences of British ancestry as our focal samples.

Fast and Accurate Inference

We developed `cxt`, a decoder-only transformer architecture adapted for the task of inferring local coalescence times from mutation patterns across recombining genomes. Unlike language models that embed discrete tokens, `cxt` projects continuous mutation densities into a latent space using feed-forward networks, followed by attention-based contextualization. Rotary positional embeddings encode genomic distance, allowing the model to reason over local genealogical structure. Trained using a next-coalescence prediction objective, `cxt` captures the stochasticity of evolutionary processes while achieving rapid and accurate inference.

We show a descriptive schematic of the model in Figure 1A. `cxt` follows a general language modeling paradigm in which the model predicts a discretized coalescent time for each genomic window based on preceding predictions and the local mutational context. Specifically, we divide the genome into fixed-size positional windows and discretize coalescent times into bins on a logarithmic scale. Given a sequence of ground truth discretized pairwise TMRCA values $E = (e_1, e_2, \ldots, e_n)$, the model autoregressively learns

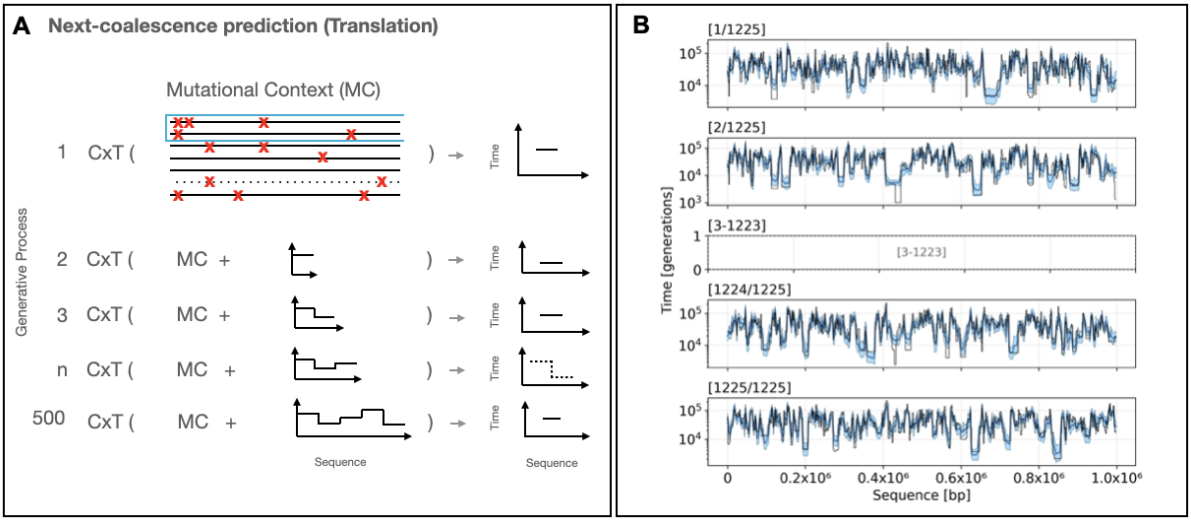

Figure 1: `cxt` introduces the notation of *next-coalescence prediction* (A). `cxt` is a language model that conditions on a chosen "pivot" haplotype pair and predicts the pair's time to the most recent common ancestor (TMRCA) for each window. `cxt` ingests mutational tensors constructed using the pivot pair and site–frequency–spectrum (SFS) values computed in 2kb windows across a 1Mb region. The model works autoregressively: after each window is predicted, that estimate is appended to the context and supplied to the next step, yielding a step-wise reconstruction of the entire coalescent history. Because every haplotype pair can be processed in parallel, the approach scales efficiently. In panel (B) all $\binom{50}{2}$ pairwise coalescence curves for a sample of 50 haplotypes were inferred simultaneously in under five minutes on a single NVIDIA A100 GPU. True TMRCAs are plotted in black, individual replicate predictions in light blue, and the prediction mean in dark blue.

the conditional distribution:

$$P(E \mid \mathrm{MC}) = \prod_{i=1}^{n} P(e_i \mid \hat{e}_{<i}, \mathrm{MC}),$$

where $\hat{e}_{<i}$ represents the model's stochastic predictions for the preceding windows and MC denotes the mutational context (the observed data) across all windows. Concretely, the mutational context in a given window is the site frequency spectrum calculated across all samples, partitioned into the contributions from sites that are heterozygous and sites that are homozygous in the pivot pair.

At each genomic window i, the model outputs a probability distribution over discretized TMRCA bins:

$$\hat{e}_i \sim P_\theta(e_i \mid \hat{e}_{<i}, \mathrm{MC}),$$

with P_θ parameterized by a neural network with weights θ. The training objective is to maximize the log probability of the true sequence given the (preceding) predicted sequence:

$$\mathcal{L}(\theta) = \sum_{i=1}^{n} \log P_\theta(e_i \mid \hat{e}_{<i}, \mathrm{MC}),$$

where e_i denotes the ground truth label in window i, and $\hat{e}_{<i}$ are the model's own predictions up to (but not including) window i. The full sequence of predictions is then shifted such that expected nucleotide diversity matches observed nucleotide diversity for the pivot pair.

This objective encourages the model to learn a mapping from local mutation patterns to genealogical structure, enabling accurate sequential inference of coalescent events across the genome.

Table 1: Comparison of mean squared error across different demographic scenarios, here lower is better. `cxt` (1-3) being the result of the narrow model and (4-5) the values after fine-tuning. For `Singer` (2-3) are results the method by itself, (4-5) for `Singer`/Polegon and (1) for both `Singer` and `Singer+Polegon`. `Gamma-SMC` results are presented in last column.

Scenario	`cxt` **(A)**	`Singer`/**Polegon (B)**	`Gamma-SMC` **(C)**
Constant (1)	0.**2298**	0.2717/0.2404	0.3326
Sawtooth (2)	0.2995	**0.2830**	0.5241
Island (3)	**0.2442**	0.2890	0.3740
Sawtooth-Finetuned (4)	**0.1764**	0.2243	–
Island-Finetuned (5)	**0.2426**	0.2601	–

Figure 1B shows a concrete example of the model's output. Leveraging our GPU-enabled method, we infer pairwise coalescence times for all $\binom{50}{2}$ pairs of a sample of 50 haploid chromosomes in parallel by placing them in a batch of size 1225. We then run the generative process multiple times, typically 15 replicates, and average the results to obtain stable and robust estimates. This generative process provides samples from an approximate posterior distribution of pair coalescence times across the sequence (where the prior is implicit via the simulated training set; see Calibration of approximate posteriors). In Figure 1B, we apply our "narrow" model to a constant population size scenario with a population size of 2×10^4 and approximately equal mutation and recombination rates (see `msprime` Training Datasets Description below). In that figure true coalescent times are shown in black, the replicate predictions in light blue, and the average prediction in dark blue. In this simple setting, where the model is well-specified, `cxt` produces highly accurate predictions. Importantly, `cxt` can predict coalescent times for *all* pivot pairs concurrently in approximately five minutes of computation on an NVIDIA A100 GPU with 80GB memory.

Benchmark Comparisons

To evaluate `cxt`'s performance, we benchmarked its accuracy relative to existing methods. We selected two recent approaches, `Singer` and `Gamma-SMC` (Deng, Nielsen, and Yun S. Song 2024; Schweiger and Durbin 2023), chosen for their state-of-the-art accuracy and fast run-times respectively. The speed of inference was particularly important, as we sought to move beyond evaluating a single distinguished pair (as in `SMC++` (Terhorst, Kamm, and Yun S. Song 2017)) to instead compare all pairwise combinations.

In Figure 2, we aim to demonstrate two main results. First, in Figure 2 (**A1**–**A3**, **B1**–**B3**, and **C1**–**C3**), we compare our narrow model against `Singer` and `Gamma-SMC`, respectively. For the constant demography scenario (**A1**, **B1**, and **C1**), `cxt` achieved mean squared error (MSE) across the genome of as low as 0.2298, compared to 0.2717 for `Singer` and 0.3326 for `Gamma-SMC`.

Second, we assessed **robustness** by evaluating `cxt` on two scenarios it was not explicitly trained on: a sawtooth demographic model and an island model (**A2** and **A3**; see 4). In both cases, `cxt` again achieved similar or lower errors than comparators, with MSE=0.2995 for the sawtooth scenario (**A2**) and 0.2442 for the island model (**A3**), compared to 0.2830 (**B2**) and 0.2890 (**B3**) for `Singer`, and 0.5241 (**C2**) and 0.3740 (**C3**) for `Gamma-SMC`. Notably, `Gamma-SMC` struggled the most under non-constant population size.

One advantage of the language model approach we take with `cxt` is that a base model can be subsequently fine-tuned for specific scenarios not covered in the original training set. To demonstrate this, we fine-tuned the narrow model separately on the sawtooth (**A4**) and island (**A5**) scenarios. Fine-tuning further improved performance, yielding mean squared errors as low as 0.1747 for the sawtooth model (**A4**) and 0.2410 for the island model (**A5**), with the latter being comparable to the non-fine-tuned version.

During the course of this work, an algorithmic extension to `Singer`, called `Polegon`, was released,

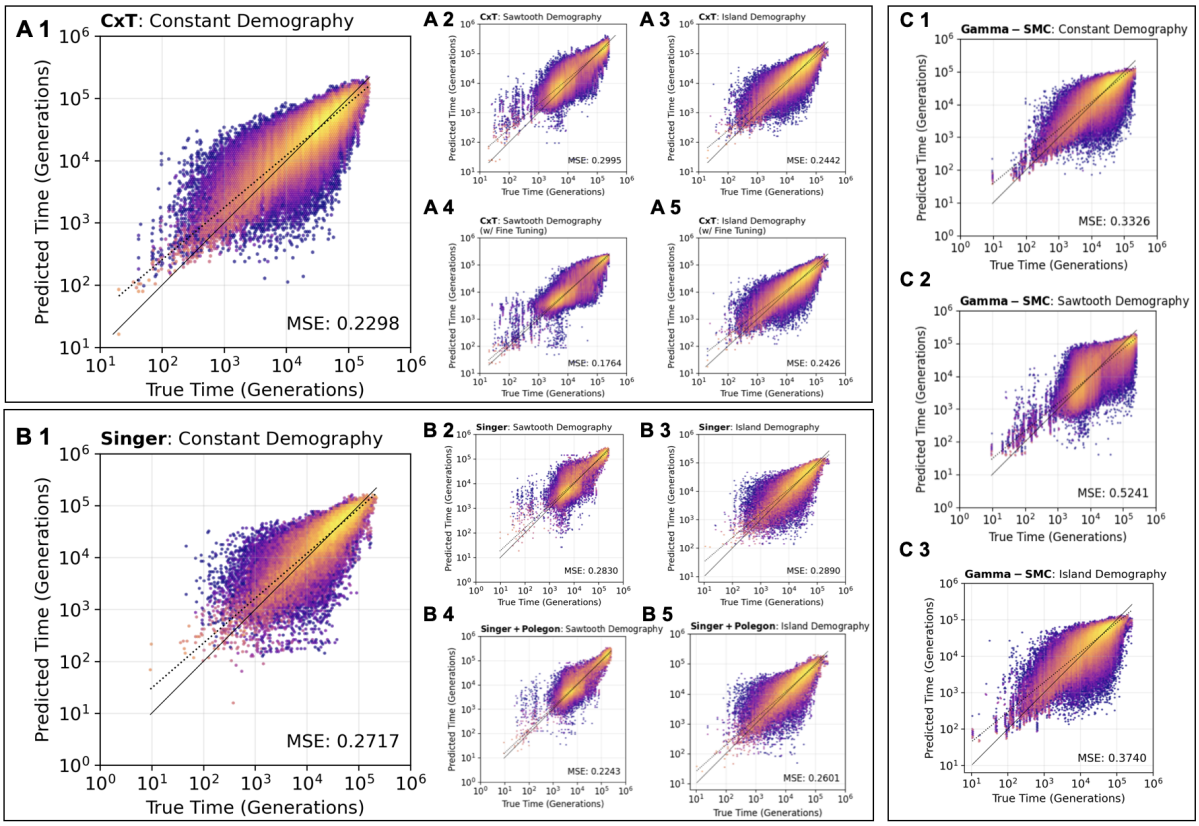

Figure 2: `cxt` (**A**) comparison against two recent methods, `Singer` (ARG-method, **B**) and `Gamma-SMC` (SMC-method, **C**), both methods were chosen due to their fast inference and high accuracy. In all plots (**A1-C3**), true versus predicted coalescence times are plotted. Specifically, we tested `cxt` narrow model in (**A1**), for the scenario it was explicitly trained on and on two scenarios (**A2**) and (**A3**): a sawtooth scenario and an island model (see parameters in methods), where it was not trained on to test its robustness. In (**A4**) and (**A5**), a version of the model fine-tuned to each scenario is tested. Analogously, `Singer` is evaluated on the same scenarios, with `Singer+Polegon` serving as a fine-tuned variant. This is achieved by incorporating reduced prior assumptions via *Polegon*, shown in (**B4**) and (**B5**).
Lastly, the performance of `Gamma-SMC` is displayed in (**C1**–**C3**). Three different random seeds were used, producing three replicates per plot.

which improved `Singer`'s standalone performance. Nevertheless, even with `Polegon`, `cxt` outperformed `Singer+Polegon` achieving errors of 0.2298 and 0.2442 for the constant (**B4**) and island (**B5**) models, respectively. While that is so, `Polegon` improved the performance under the original constant sized scenario, achieving a MSE of 0.2404 (see Figure S7).

In conclusion, the language modeling approach implemented in `cxt` is highly competitive with the most recent theory-driven methods, whether ARG-based (as in `Singer`) or SMC-based (as in `Gamma-SMC`), particularly for models that do not conform to model assumptions. A full summary of error estimates is provided in Table 1 and Figure 2.

Towards a foundation model

In the previous section, we investigated the case where `cxt` has been trained with a single demographic model, effectively encoding the prior into the simulated training data, and optionally finetuned. The strength of language modeling does not lie in the narrow case, but in our ability to condition on the model's behavior based on the context. Thus, our model can be loosely regarded as a sort of approximate database that, depending on the context, can yield different distributions.

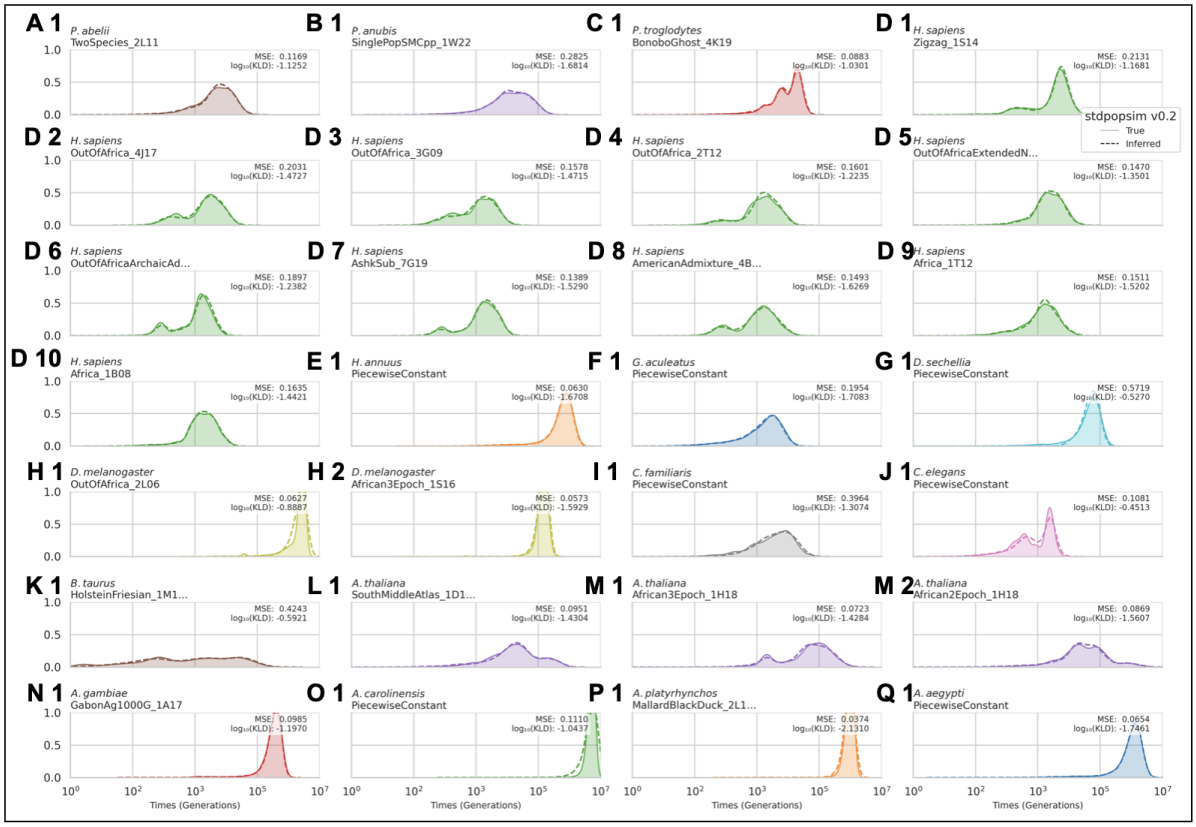

Figure 3: Evaluation of marginal coalescence distribution inference (dashed line) against the true distribution (shaded line), probing the model's capacity to distinguish among many scenarios based on context alone. The broad model's ability to infer a `stdpopsim` v0.2 coalescence distribution is tested, which includes diverse mutation and recombination rates, as well as demographic scenarios (A1–P1). This distribution is obtained by aggregating pairwise inferences and visualizing the result as a kernel density for each scenario. All results shown are from new simulations, not included in the original training dataset and generated without the use of a genetic map.

In Figure 3 we show the inferred coalescence time distributions for nearly all `stdpopsim` v0.2 species, including their respective demographic models, or assuming a constant population size when no demography was available. Predictions on simulations using fine-scale recombination maps are shown separately in Supplementary Figure S5. All results shown here are based on the broad model (see `msprime` Training Datasets Description), which was trained across all scenarios. The evaluation presented in Figures 3 and S5 is performed on new simulation realizations that were not part of the training set, thus there is no opportunity for data leakage. Notably, in nearly every case, the inferred distributions (dashed lines) closely match the true distributions (shaded lines).

Out-of-sample tests for the broad model

During the course of this project, `stdpopsim` v0.3 was released, expanding the catalog with additional species (*S. sucrofa, R. norvegicus, P. sinus, M. musculus*, and *G. gorilla*). We took advantage of this release to evaluate the generalization ability of our broad model, testing performance without retraining. Tests on these species thus represent out-of-sample examples for the broad model.

The v0.3 set includes species with very large effective population sizes, such as *S. scrofa* ($N_e = 270000$) and *R. norvegicus* ($N_e = 124000$), as well as species with very small population sizes, such as *P. sinus* ($N_e = 3500$). In addition, most new species exhibit high recombination relative to mutation rates, e.g. $m = 5.83 \times 10^{-9}$ and $r = 1 \times 10^{-8}$ per bp per generation ($r/m = 1.715$) in the case of *P. sinus*, with the

notable exception of *G. gorilla* ($r = 1.193 \times 10^{-8}$ and $m = 1.235 \times 10^{-8}$ leading to $r/m = 0.966$).

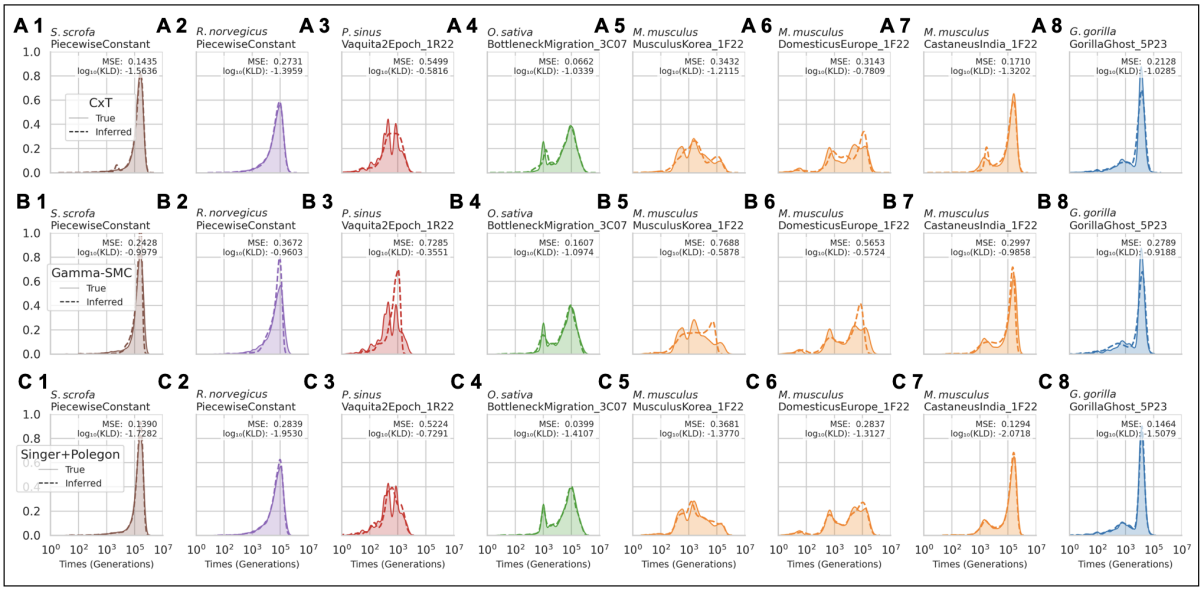

Figure 4: Out-of-sample evaluation of the broad model on `stdpopsim` v0.3. Each panel shows inferred marginal coalescence distributions (dashed) against true distributions (shaded). We simulated data from the novel species / demographic histories added in `stdpopsim` v0.3, aggregating pairwise inferences into kernel density plots. All results are from simulations previously unseen during training (**A1**–**A8**), while **C1**–**C8** present outcomes for `Singer+Polegon` and `Gamma-SMC`. Both `Singer` and `Gamma-SMC` require the mutation rate and recombination rates as input, while `cxt` only requires the mutation rate.

The results from this out-of-sample evaluation can be seen in Figure 4. Notably, `cxt` is able to generalize quite well. Even when presented with data produced by previously unseen parameter regimes of recombination, mutation rates, population sizes and structures. `cxt` still performs on par with `Singer+Polegon` and substantially better than `Gamma-SMC`.

Generalization across parameter regimes

With this project, we aimed to build a model that encompasses `stdpopsim`, motivated by the rationale that as `stdpopsim` expands, we can incrementally incorporate new species. However, when a sample from a new species — differing in mutation and recombination rates, effective population size, or population structure — uses `cxt`, we assume that the accuracy of inference depends on how closely that species aligns with any point in the training data manifold.

In Supplementary Figure S9 we investigate the model's ability to interpolate and extrapolate to data that is outside of the training data manifold with respect to mutation and recombination ratios for a constant N_e of 20000. We start by defining a grid of mutation and recombination rates that encompasses most `stdpopsim` species (panel **A** in Figs S9) which is determined based on where most species fall in Figure S9 (**B**). Figure S9 (**A**) displays various recombination-to-mutation rate ratios; Figure S9 (**B**) shows the distribution of species across the grid; and Figures S9 (**C**) and (**D**) present the MSE and KL divergence, respectively. Consistent with expectations, the highest MSE appears in the low mutation, high recombination region (bottom right of Figure S9 (**C**)) where the signal-to-noise ratio is the least favorable to TMRCA inference. Figure S10 shows examples of TMRCA predictions from the four corners of this parameter grid. Together, these demonstrate that `cxt` is able to successfully generalize outside of the specific regions of parameter space that align with training data.

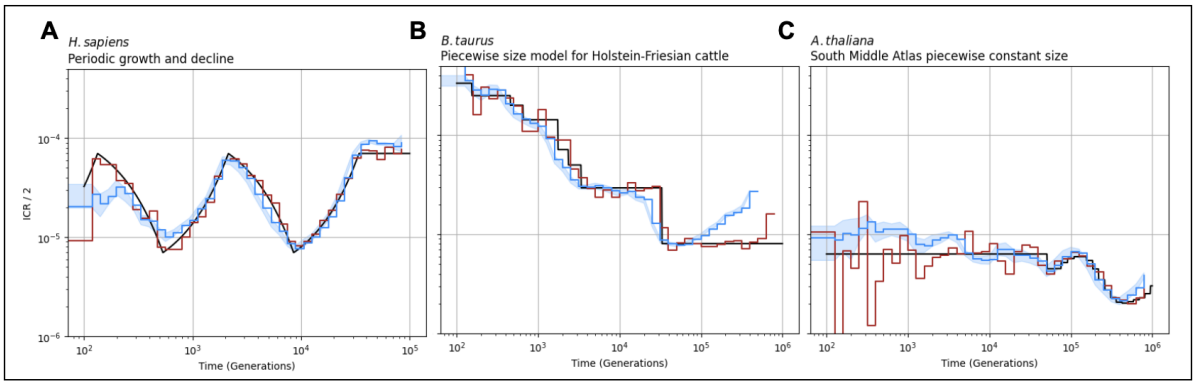

Figure 5: Inverse-instantaneous coalescence rate calculation for the piecewise-constant demography of (**A**) *H. sapiens*, (**B**) *B. taurus* and (**C**) *A. thaliana*. The inference of pairwise-coalescence events leads to the implicit inference of demography estimates through the marginal coalescence distribution assuming coalescence occurs as a Poisson process (see methods). For each scenario 10 Mb and 25 diploid samples have been used to achieve resolution throughout the specified time-windows.

Calibration of approximate posteriors

`cxt` samples TMRCA trajectories from an approximate posterior, that may be used to provide uncertainty estimates around local predictions as well as point estimates (e.g. the posterior mean). To this end, we empirically tested whether the marginal approximate posteriors are well-calibrated (in the sense of providing asymptotically correct coverage of the true TMRCA) and consistent (in the sense that increasing the amount of mutational information causes the posterior to concentrate).

To do so, we first simulated 1000 completely independent pivot pairs for three demographic scenarios – including one from the out-of-training `stdpopsim` v0.3 set (*Oryza sativa*) – then calculated posterior intervals for the TMRCA in each 2Kb window using 100 sampled `cxt` trajectories per pair. Exact posterior distributions have the property that an interval containing a proportion α of the posterior mass should also contain the true value with probability α. Thus, we empirically calculate the proportion of true TMRCAs falling within intervals of different widths from the marginal approximate posteriors. Second, we calculate the average posterior variance as a function of mutation rate (scaled by 0.5, 1, 2 relative to the species' mutation rate from the `stdpopsim` catalog). Both of these measures are calculated with respect to the position of the TMRCA in the full 1Mb prediction window, as we expect performance to decline near the boundaries where the local mutational context is truncated.

Across three distinct parameter regimes, we find that `cxt` generates generally well-calibrated posterior samples regardless of the location of the 2Kb window (Supplementary Figure S6, top row). These approximate posteriors are typically slightly over-concentrated for the largest intervals (e.g. an interval containing 95% of the posterior mass contains the true TMRCA roughly 92% of the time), but remarkably stationary: the empirical coverage only declines significantly for the last 2Kb window. The reason for this decline is a tendency of the trained model to predict extreme jumps in the final window, which can be dealt with in practice by omitting this window prior to centering the predictions. Further, as we increase/decrease mutational density, the approximate posterior intervals concentrate/dilate reflecting an increase/decrease in information (Supplementary Figure S6, bottom row). As expected, predictions for 2Kb windows that are close to the boundaries have greater uncertainty due to truncation of the local mutational context. These findings indicate that the approximate posteriors produced by `cxt` reliably quantify uncertainty in predicted coalescence times, regardless of demographic scenario and mutational density. This is a real strength of the method, all the more so because sampling from this posterior is so computationally efficient.

Demography estimation through coalescence rates

Estimation of pair coalescence rates provides a direct window into how effective population size changes over time, perhaps most famously typified by the original `PSMC` model of Li and Durbin (2011). To evaluate our ability to estimate effective population size trajectories from `cxt` predictions, we calculated pair coalescence time distributions from simulated data from models of *H. sapiens*, *B. taurus*, and *A. thaliana*, and then estimated coalescence rates from these distributions to estimate historical effective population size (Figure 5; see methods). In Figure 5, three lines are plotted: a black "expectation" line, which directly reflects the true rate from the simulation, and notably, a line labeled "inference limit". This latter line is produced sampling the true TMRCAs every two Kb across the sequence, and then calculating coalescence rates as was done for the `cxt` predictions. These results show that the historical effective population size estimates derived from `cxt` are globally accurate (e.g. the genome wide distribution of pair coalescence times matches that of the generative model).

We also inferred the cross-coalescence rate (i.e., the rate of coalescence between two samples drawn with respect to deme) under the two-population OutOfAfrica_2T12 model from `stdpopsim`. Generally, inference of cross-coalescence rates is informative of the divergence time between populations within a species. Figure S8 presents these results: panels (**A**) and (**C**) show the within-population rates for African and European samples, respectively, while panel (**B**) shows the cross-population rate estimated from one African and one European haplotype. Panels (**B**) and (**C**) closely recover the expected trajectories; however, in the African analysis panel (**A**), `cxt` exhibits a modest deviation. During creation of the training dataset, 50 samples were drawn with random population assignments for the structured models. Thus, in any given training instance it is highly unlikely that only individuals from one population were present, which is likely to be the source of this bias (and could be mitigated by adjusting the training scheme for structured models).

It is worth noting the demographic inferences we are presenting in Figures 5 and S8 are generated from a relatively small simulation with respect to sequence length – we are limiting ourselves to 10 Mb of simulated genome simply for speed and for efficiency of our demonstrations. As smaller sequence lengths increase the variance of coalescence rate estimates in recent time due to effects of truncation of the longest tracts of recent ancestry, we expect that application to full length chromosomes would lead to significant improvements in accuracy.

Application to 1000 Genomes data

Having demonstrated the accuracy and robustness of `cxt` on simulated data, we next turned our attention to applying it to human genetic variation data from the 1000 Genomes dataset (Consortium et al. 2010). We randomly selected 50 individuals of indicated British descent (GBR), and estimated coalescent times for 10 focal pairs, holding out the remaining $n = 48$ samples as the AFS input to `cxt`. We choose this sample size of 10 focal pairs because we can conveniently fit all chromosome-wide comparisons on to a single A100 GPU in a single batch, allowing us to decode whole chromosome data quickly. In practice, the user could do larger number of comparisons, and is ultimately limited by the available GPU memory when considering batch size. By way of example we show landscapes from two chromosomes, human chromosomes 2 and 6, in Figure 6 (top and bottom are chr2 and chr6, respectively) in which we show both median and mean summaries of coalescent time along with individual focal pair estimates. As a sanity check, we plotted the observed TMRCAs against the number of recombination events predicted by `cxt` for both chromosomes in Figure S11. This confirmed our expectation that older TMRCA segments tend to be smaller due to recombination. In Figure 6 we see fine scale noise in the time estimate across each chromosome, but with larger scale features that locally distort the coalescent history. For instance on chromosome 2 we highlight in red the *LCT* region, which is a local trough of coalescent time (Fig. 6A),

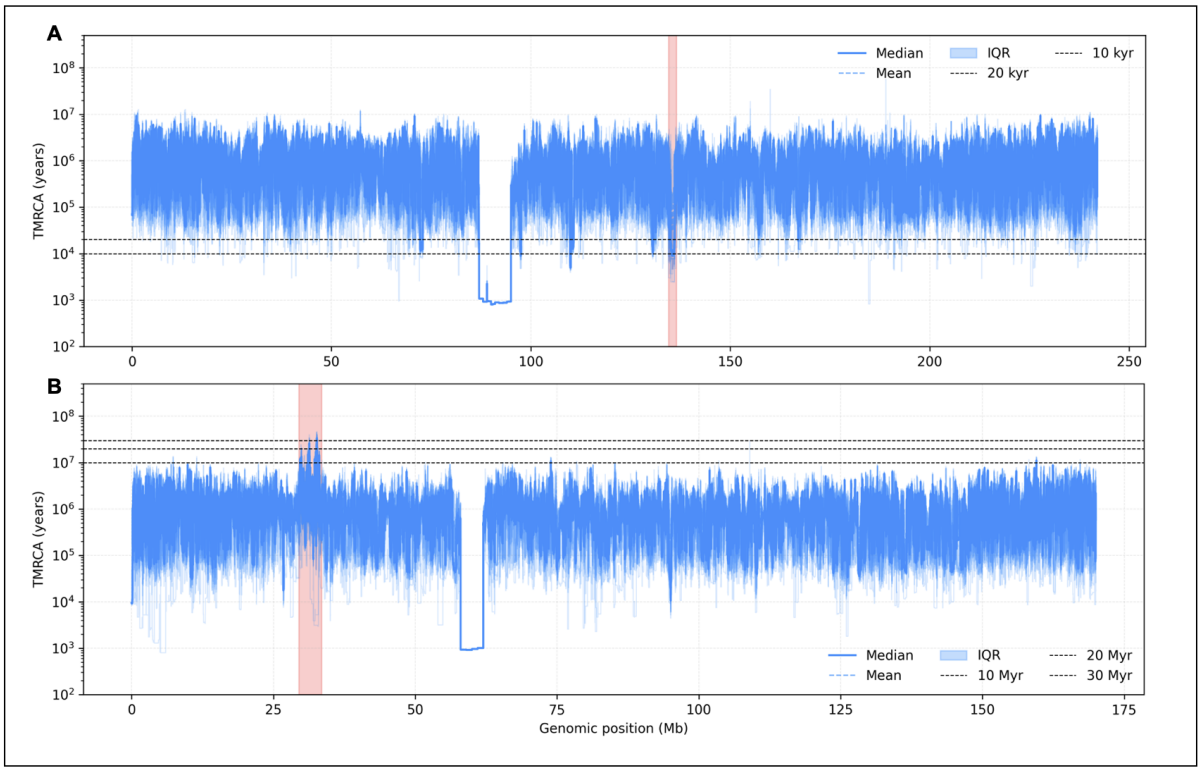

Figure 6: Inference of coalescent times from ten focal pairs of British individuals of the 1000 Genomes project. In (A) the chromosome 2 and in (B) chromosome 6 are portrayed. The LCT region (A) and HLA region (B) are highlighted respectively. Each pivot pair has been inferred 15 times to get a reliable mean, median and variance estimates. Additionally, light blue lines show the respected inferences of the ten pairs.

as might be expected from its history of adaptive evolution in European populations. On chromosome 6 we highlight the *HLA* region, which is a clear outlier in the genome, as harboring some of the most ancient evolutionary histories.

In Figure 7 we focus on the *LCT* and *HLA* regions, shown in Figs 7A & 7B respectively. There is a clear dip in coalescent time proximal to *LCT*, reflecting the well-characterized sweep dynamics in this region. Dates for common ancestry are slightly older than 10 thousand years ago near *LCT*. It is important to note that this date reflects not the onset of selection of this haplotype, but instead the date in which it first arose in the population. Notably, not all the pivot pairs date that recent (see light blue lines), highlighting variation in evolutionary history at this locus.

Figure 7B shows a similar view of coalescent dates near the *HLA* region on chromosome 6. We observe three distinct ancient peaks as reported by (Deng, Yun S Song, and Nielsen 2025) —although our estimates appear slightly younger. We see coalescent times at *HLA* genes of roughly 30 millions years– an exceptionally ancient estimate that likely is a strong signature of long term balancing selection at this locus.

Discussion

In this study, we introduce a novel application of language modeling to population genetics by training a modified GPT-2 model to sequentially learn patterns of the coalescent with recombination. Inspired by SMC-based methods, our approach replaces the traditional HMM, which enforces a first-order dependence structure via the Markov property with a high-parameter neural network that leverages an attention

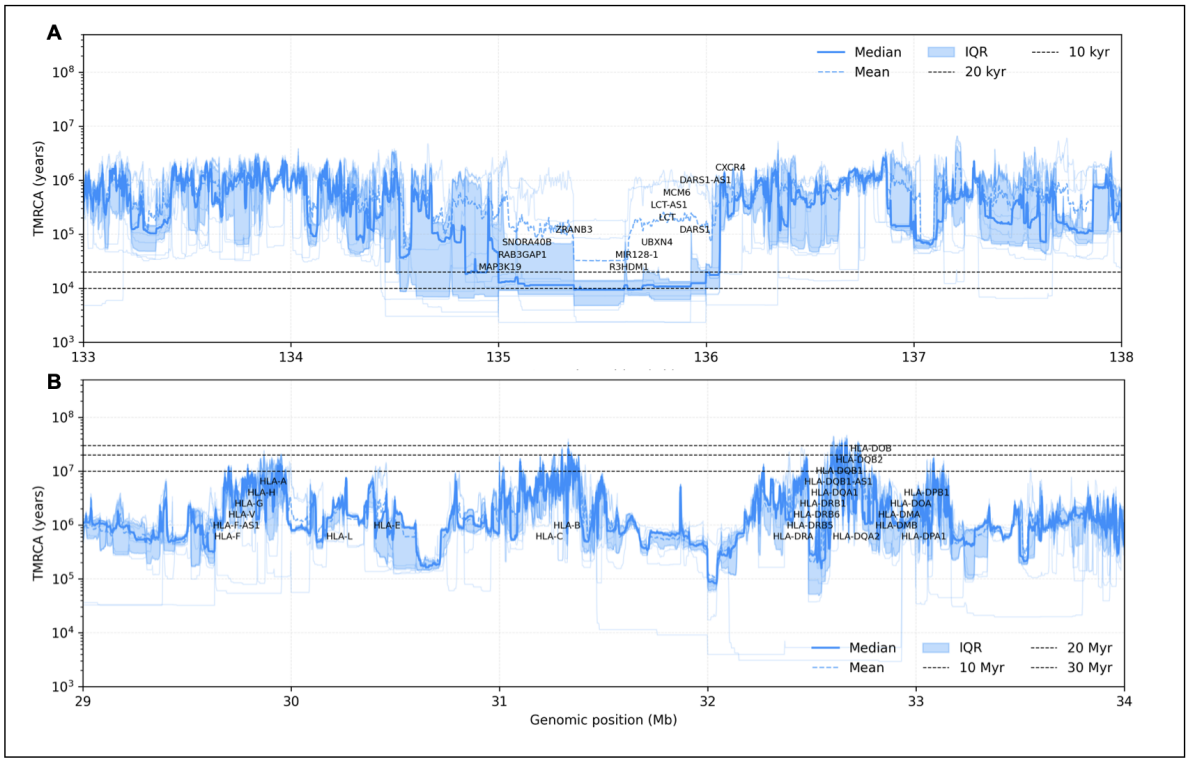

Figure 7: Inference of coalescent times from ten focal pairs of British individuals of the 1000 Genomes project. In (A) the LCT sweep is shown (chromosome 2) and in (B) the HLA region is portrayed (chromosome 6). Each focal pair has been inferred 15 times to get a reliable mean, median and variance estimates. Additionally, light blue lines show the respected inferences of the ten pairs.

mechanism for sequence analysis, which does not have this constraint. This shift removes the need for an explicit likelihood function; instead, assumptions and priors are implicitly encoded in the training data.

A key innovation of our method is its autoregressive structure: we iteratively and generatively predict coalescence events conditioned on observed mutations, progressively expanding the predictions up to a specified length size. Within each window, the model samples coalescence times from a predicted discrete probability distribution, and such samples can form the basis of an approximate posterior distribution. This approach differs fundamentally from prior deep learning methods such as `ReLERNN` (Adrion, Galloway, and Kern 2020) or `CoalNN` (Nait Saada et al. 2023), which treat inference along the chromosome as a supervised regression task - learning a single point estimate from sequence data using a sequential architecture that does not model the underlying generative process. In this work, we train a language model to learn the *neutral coalescent with recombination* as a **conditional stochastic process**, not to predict a point estimate. Rather than predicting, for instance, recombination rates directly, `cxt` models how pairwise TMRCA values evolve across the genome. It does so by (A) sharing information across samples through a site frequency spectrum representation, (B) training across a broad range of demographic scenarios to expose the language model to a wide variety of mutational patterns, and (C) focusing attention on local changes in relative TMRCA rather than on absolute times. This formulation enables the model to produce distributions over coalescent outcomes, which can be sampled efficiently in parallel.

In this report we demonstrate the speed and accuracy of `cxt`, which can produce estimates of an entire diploid genome in minutes - utilizing one GPU. Comparisons to `Gamma-SMC` and `SINGER` demonstrated that `cxt` consistently outperforms `Gamma-SMC` while matching the performance of the state-of-the-art ARG-based inference method `SINGER`, which is a particularly strong benchmark because it uses all samples

jointly. This places `cxt` among the most competitive methods for coalescent inference at scale - offering an amortized, generative alternative to traditional MCMC sampling over ARG likelihoods. In the same way that PSMC introduced a base framework that could later be extended in many directions, we envision `cxt` and its pipeline as an immediately extensible foundation for future work. There are at least two conceptual advantages of the deep learning based LM over the traditional SMC-based modeling: 1) DL-based language models can take arbitrary inputs and 2) simulation based training allows for learning of nearly any process (we aren't limited to those with analytic solutions). Thus, some clear future directions for `cxt` include incorporation of selection, multi-lineage coalescence (e.g., larger sample sizes), and threading of newly sampled haplotypes onto an existing ARG. Some of these goals may be achievable simply by fine-tuning on top of the neutral base process learned here.

One notable shortcoming of `cxt` is the rigidity introduced by discretizing the sequence into fixed-length windows (here, two Kb). This "one-size-fits-all" approach is not ideal for every recombination scenario. However, given that variability in recombination rates across `stdpopsim` species is relatively modest (see Supplementary Figure S9B), the impact may be negligible in many cases. That said, for very ancient coalescent events—beyond 5–6 million generations— where ancestral segments become extremely short, this fixed window size may introduce skew and may affect species with large population sizes, the effective recombination rate (which scales with population size) may require smaller window sizes - for example, in certain mosquito species. Looking ahead, one natural direction is to reduce the window size for finer resolution. However, smaller windows require denser temporal grids and longer input sequences, which increase the context window - and since our model uses standard attention with pairwise comparisons, this leads to a quadratic rise in computational cost. One solution is to restrict attention to local genomic neighborhoods. Increasing the context window itself, independent of sequence length, may also help the model distinguish among haplotypes by conditioning on more mutations. A parallel trade-off involves model size, speed, and generalizability: while larger models are slower to train and deploy, they can absorb a broader range of demographic scenarios and perform better when interpolation is required - making them more robust across diverse evolutionary settings.

To summarize, this work introduces a language model that functions as a sort of inference database, designed to accompany the growing catalog of species in `stdpopsim`. Even though it requires only a mutation rate for calibration, in the likely case that the data tends to deviate from the training distribution, its core advantage is deeper: this model can absorb prior knowledge directly through training, without explicit mathematical derivation. This makes it not just a fast and accurate tool for TMRCA inference, but a flexible framework - adaptable to novel evolutionary scenarios.

Methods

We start by providing a detailed overview of the methods introduced in our study. First, we briefly introduce the concept of **next-coalescence prediction**. Second, we describe the parameterization of coalescent simulation and the way we format the retrieved mutational data in order to present it to our network. Third, we outline the architecture of the network itself, which is derived from the GPT-2 decoder-only model. Finally, we describe the various experiments conducted to benchmark our method according to the objectives described above.

Next-coalescence Prediction

We use as our prediction task, *next-coalescence* prediction, which we define to mean the next point in time in which a focal (or pivot) pair of sequences find common ancestry as we move across (say from left to right) a sequence. Our method processes one megabase (Mb) segments of sequence, subdivided

into five hundred two kilobase (Kb) windows. For the purposes of this paper we use a sample size of 25 diploid individuals, but this is arbitrary. Our model focuses on a pivot pair of chromosomes from which we predict pairwise coalescence events, or more specifically a probability distribution from which the next coalescence event can be sampled for that pivot pair along the sequence for each of the 500 windows. This describes an iterative process by which the predicted coalescence time is concatenated with the mutational context, forming the next input to the model to predict the next coalescence event (see Figure 1). We refer to this procedure as *next-coalescence prediction*. Throughout this paper we trained using 25 diploid individuals, however it can be applied to more samples without retraining and fewer if training is performed. We continue by describing the processing of the genotype matrix as obtained from coalescent simulation before presenting the architecture and listing the specific simulation parameterization next.

Processing of Coalescent Simulations

We start from a genotype matrix G, where $G \in \mathbb{Z}^{N \times M}$ with N diploid samples and M sites, which represents phased data directly accessible for a coalescent simulator (here `msprime`).

We describe the processing of that matrix in two ways:

1. Algorithmically in Algorithm 1
2. Schematically in the Appendix (see Figure S1)

Algorithmically, the processing of the genotype matrix and associated positions involves two key steps:

1. Collapsing mutation patterns between pivot samples using logical operations:
   - Heterogeneous mutations via `XOR`: `xor`(00)→0, `xor`(01)→1, `xor`(11)→0
   - Homogeneous mutations via `XNOR` (logical complement of `XOR`)
2. Scaling the extracted patterns by their site frequency spectrum (SFS) values: This scaling step simply refers to multiplying each mutation either heterozygous or homozygous of the pivot pair, by the absolute frequency of that mutation (SFS value: singleton, doubleton, tripleton and so on ...). Absolute frequencies being derived across all 25 diploid samples. Focusing on pivot pair mutations is a necessary conditioning, such that the inferred pairwise coalescence time matches with the informative mutations. Thus, scaling each mutation, which would be otherwise just binary can be thought of as attaching a crude estimation of the age to each mutation and making use of all 25 diploid samples.

This processing is performed using sliding windows of 4, 16, 64, 128 Kb each with a 2 Kb step size, and is repeated in parallel across progressively larger fixed window sizes. The complete algorithmic details are provided in Algorithm 1 below.

The input consists simply of the genotype matrix, accompanied by positional information for each variant. Notably, as described above, we operate on estimating average coalescence times per window of two Kb from relevant pivot pairs. These inputs are readily obtained via simulation (here we use the coalescent simulator `msprime` (Baumdicker et al. 2022)) and can be generated from genotype matrices derived from empirical data, e.g. VCF files.

Implementation of the Decoder-Only Architecture

We next describe the architecture of the decoder-only transformer and outline the modifications made to adapt it to our genetic translation task. While the original transformer described an encoder-decoder architecture (Vaswani et al. 2017), our approach aligns closely with the principles and implementation of

Algorithm 1 ProcessGenotypeMatrixAndPositions (Vectorized)

Require: $G \in \mathbb{Z}^{N \times M}$ ▷ Genotype matrix (N samples, M sites)
Require: $P \in \mathbb{R}^{M}$ ▷ Site positions ($P[j]$ is the position of site j)
Require: L ▷ Sequence length (total genomic range)
Require: $W_m = [2, 8, 32, 64]$ ▷ Window multipliers
Require: $w = 4000$ ▷ Base window size
Require: $s = 2000$ ▷ Step size
Require: A, B ▷ Pivot sample indices
Require: `xor_ops` ▷ Logical operation: XOR or XNOR
Ensure: $X \in \mathbb{Z}^{2 \times |W_m| \times K \times N}$ ▷ Feature matrix, with $K = \lceil L/s \rceil$

// Step 1: Compute Logical Frequencies (Vectorized)
$F \leftarrow \mathrm{sum}(G, \mathrm{axis} = 0)$ ▷ Vectorized sum over samples; $F \in \mathbb{Z}^{1 \times M}$
$L_val_{XOR} \leftarrow G[A, :] \oplus G[B, :]$
$L_val_{XNOR} \leftarrow \neg\big(G[A, :] \oplus G[B, :]\big)$

$F_logic[0, :] \leftarrow F \odot L_val_{XOR}$ ▷ For XOR
$F_logic[1, :] \leftarrow F \odot L_val_{XNOR}$ ▷ For XNOR

// Step 2: Define Sliding Windows (Vectorized Boundaries)
$K \leftarrow \lceil L/s \rceil$
start $\leftarrow s \cdot [\,0, 1, \ldots, K-1\,]$ ▷ Vector of window start positions
for $i \leftarrow 1$ **to** $|W_m|$ **do**
  $W_{\mathrm{end}}[i] \leftarrow w \cdot W_m[i]$
end for
end $\leftarrow \min\Big(\mathbf{start} + W_{\mathrm{end}},\ L\Big)$ ▷ Broadcasted over multipliers

For each window index k and multiplier index i, define

$$W[k, i] = \{\, j \mid P[j] \in [\,\mathbf{start}[k],\ \mathbf{end}[i]\,]\,\}$$

// Step 3: Compute Site Frequency Spectrum (Vectorized Bincount)
for $d \in \{0, 1\}$ **do**
  for $i \leftarrow 1$ **to** $|W_m|$ **do**
    for $k \leftarrow 0$ **to** $K - 1$ **do**
      $X[d, i, k, :] \leftarrow \mathrm{bincount}\Big(F_logic[d, W[k, i]],\ \mathrm{minlength} = N\Big)$
    end for
  end for
end for

return X

GPT-2 (Radford et al. 2019), incorporating a set of modifications to address the unique requirements of our population genetic data.

In traditional natural language processing tasks, text is tokenized and represented as input $x \in \mathbb{R}^{B \times T}$ (with B: batch size and T: the number of tokens), which is then further processed using an embedding table of dimension $V \times E$ (V: number of tokens, E: embedding dimension) as sort of a trainable lookup table for each latent representation per token. We initially considered developing a similar tokenizer tailored for population genetics, however instead of discretizing the unique mutation patterns observed, which could potentially be an enormous space, we opted to directly project mutations into latent space using SFS values per window. This approach bypasses the limitations of discretization while capturing the variability of mutation patterns. Consequently, we implemented a fully connected projection to map the SFS values per window, per window size, and heterozygous or homozygous site distinction directly into the token latent space, having as input and output:

$$\boldsymbol{X} \in \mathbb{Z}^{B \times 2 \times |\boldsymbol{W}_m| \times K \times N} \quad \text{and} \quad \boldsymbol{X} \in \mathbb{Z}^{B \times K \times E}.$$

Here, B denotes the batch size; "2" distinguishes heterozygous vs. homozygous states; $|\mathbf{W}_m|$ is the number of distinct window sizes; K is the number of windows; and N is the number of samples and E is the embedding dimension.

Algorithm 2 ProjectionReplacingEmbedding

Require: $x \in \mathbb{R}^{B \times |\boldsymbol{W}_m| \times WS \times K \times IE}$, parameters W_1, b_1, W_2, b_2, weights w_1, w_2
Ensure: $\hat{x} \in \mathbb{R}^{B \times K \times d}$

```
x[..., 0] ← 0
φ1(x) ← GELU(W1 x + b1)
φ2(x) ← W2 x + b2
x̂ ← w1 φ1(x) + w2 φ2(x)
x̂ ← reshape(x̂, [B, K, d])
return x̂
```

The projection module is described in Algorithm 2. It conditions the language model by mapping SFS values into a latent space, while the model's predictions remain discretized. Specifically, we partition the log-scaled coalescence times into 256 equally spaced intervals between 3 and 14 (or 324 intervals between 3 and 17 for our 20-million-parameter model) and represent these intervals using a small embedding table. Consequently, the model concatenates the mutation and coalescence latent embeddings, ensuring that the context window consistently reflects the mutation patterns produced by the underlying process. This design enables the inference of diverse evolutionary models without retraining or fine-tuning, as the model captures the critical contextual information directly from a representative segment of the genomic process. Moreover, our approach uses rotary embeddings instead of learned positional embeddings. Since the method operates on windows shifted uniformly by 2 kb, there is no need to adjust positional embeddings as required for variant-level resolution. For further details, please refer to the implementation at https://github.com/kr-colab/cxt.

Regarding the model itself, we introduce a single architecture, parameterized to encompass both a narrow model, trained exclusively on a single scenario, and a broad model, trained on a diverse range of scenarios. The parameterization of the models is described in Table 2.

The output dimension is expanded by two values, corresponding to start and padding tokens, with the latter not being used. The combined latent space has a maximum length of 1001, consisting of 500 mutational latent vectors, a start token vector, and coalescence vectors. A dropout rate of **0.1** is applied to promote regularization.

Table 2: Token-Free Decoder Model Configuration

Parameter	**Narrow Model**	**Broad Model**
Number of Layers (n_{layer})	6	10
Number of Attention Heads (n_{head})	4	4
Embedding Dimension (n_{embd})	400	400
Dropout Rate	0.1	0.1
Bias in Linear Layers	False	False
Number of Samples per Forward Pass	50	50
Sample Scale Embedding Factor	2	2
Output Dimension ($\text{output}_{\text{dim}}$)	258	326
Combined Feature Dimension ($\text{combined}_{\text{dim}}$)	1001	1001
Total parameters in millions	≈ 11.7	≈ 19.5

Calibration of the Molecular Clock

`cxt` output is run in replicates (default 15), that together provide a distribution over TMRCA trajectories. Although the LLM is agnostic to mutation rate (in the sense that this is not provided as an explicit input, and the model sees data generated under many mutation rates during training), it will generally be desirable to condition on a particular mutation rate/molecular clock. This is done by scaling the raw predictions such that expected nucleotide diversity matches the observed nucleotide diversity across the full context (e.g. the 1Mb sequence). Empirically, this first-order correction is most helpful when the model is applied to data that is atypical relative to the training simulations (see Figures S2 and S3). Crucially, the relative TMRCA is highly accurate even in out-of-distribution scenarios, so that the first order correction is generally sufficient to make bias and mean squared error constant across a wide range of scenarios. When the input data are from a demographic scenario used in the training simulations, then the impact of this correction becomes negligible (Figure S4, compared to Figure 2).

One issue is that a deterministic first-order correction will reduce variability in `cxt`'s "posterior" samples– for example, TMRCA trajectories that are constant across the 1Mb window but have distinct values across replicates, would be all collapsed to the same value. Thus, we model the correction factor probabilistically to preserve replicate variability. For a given pivot pair i, we assume the observed mutation count y_i follows a Poisson model:

$$y_i \sim \text{Pois}\left(2\mu c_i s \sum_j t_{ij}\right),$$

where s is the window size, μ is the mutation rate, t_{ij} is the predicted TMRCA in window j of replicate i, and c_i is the per-replicate correction factor. Under an improper constant prior, this yields a posterior:

$$c_i \sim \text{Gamma}\left(y_i + 1,\ 2\mu s \sum_j t_{ij}\right).$$

We then sample an independent c_i for each replicate and apply it to all windows in that replicate. This propagates uncertainty from diversity estimation into the corrected predictions. In practice, the difference is negligible when mutation counts are large, as the posterior concentrates around the maximum likelihood estimate. However, when mutation counts are sparse or when the model produces flat TMRCA profiles, this stochastic correction helps avoid degeneracies (e.g., where there are no mutations and the MLE is undefined).

Next, we continue describing the datasets obtained through various coalescence parameter initiations used for training both models.

`msprime` Training Datasets Description

We organized our training data into four distinct sets. The first is a large base dataset, used to train the narrow model, with parameters chosen to approximate human simulations. This set consists of a single demographic scenario with constant population size (see Table 3). The also performed fine tuning of the narrow model for a few auxiliary models (Table 4). The third dataset is the `stdpopsim` v0.2 dataset (see Table 5), the main dataset used in this study that we call the broad model. The final dataset comprises `stdpopsim` v0.3 models, which are used for evaluation both before and after retraining (see Table 6).

Training details and inference description

We implemented our models using PyTorch Lightning to streamline code management, including multi-GPU training and loss tracking. Matrix multiplication precision was set to float32 (medium), relying on PyTorch Lightning's internal precision handling, and we did not use additional manual autocasting with 'bf16-mixed' during training. To optimize memory efficiency and storage, datasets were stored in float16 format.

The model was trained for five epochs using the AdamW optimizer with a learning rate of 3×10^{-4}, which was gradually decreased during training via cosine annealing. Fine-tuned models were trained for only two epochs on their respective datasets, using a learning rate reduced by a factor of ten.

Each training sample of a batch consists of 500 windows, followed by a start token, which is the second row of the embedding table for time discretization (the first row is reserved for potential padding, but is not used in this implementation). This is then followed by 500 coalescent times. The target sequence is simply a right-shifted version of the coalescent times, beginning with the first coalescent event. Accordingly, the model is trained in the same manner as standard language models, using negative log-likelihood loss and a causal mask. The causal mask has been adapted for translation within the relevant attention layers, as described below.

The fused-causal attention mask: In standard next-token prediction and causal language modeling, the attention mask is typically a square matrix with dimensions equal to the number of tokens, applied before the softmax normalization. However, since translation is performed using a decoder-only architecture, the attention mechanism must simultaneously process both mutation and coalescence events. Because mutation events remain fully available during generation, the model could, in principle, condition the generation of coalescence events step by step on the mutations, which persist in the context. This would allow the model to attend freely to mutation embeddings while restricting causal attention specifically to coalescence predictions. We show our fused-causal attention mask in Figure S12.

Estimating Population Size via Instantaneous Inverse Coalescence Rates

We estimate the instantaneous coalescence rate from a set of ancestor coalescence times by discretizing time into logarithmically spaced windows ($T_{\max} = 40$). Specifically, we define the time grid using

Table 3: Base Dataset (Approx. for Human Parameters)

Parameter	**Value**
Population Model	Constant Demography
Population Size	2×10^4
Mutation Rate	1.29×10^{-8} per Base
Recombination Rate	1.28×10^{-8} per Base
Simulations	2×10^6
Sequence Length	1 Mb

Table 4: Base Dataset Extension (auxiliary Dataset): Model Parameters *

Model	Abbreviation	Values	Description
`ne.constant`	N_e	$10^4,\ 2\times10^4,\ 4\times10^4$	Effective population size
	μ	$1\times10^{-8},\ 5\times10^{-8}$	Mutation rate per base
	r	$1\times10^{-8},\ 5\times10^{-8}$	Recombination rate per base
`ne.sawtooth`	N_e	$10^4,\ 2\times10^4,\ 4\times10^4$	Effective population size
	Magnitude	`strong, medium, weak`	Amplitude of N_e oscillations
	μ	$1\times10^{-8},\ 5\times10^{-8}$	Mutation rate per base
	r	$1\times10^{-8},\ 5\times10^{-8}$	Recombination rate per base
`island.3pop`	N_e	$10^4,\ 2\times10^4,\ 4\times10^4$	Effective population size
	m	$0.05,\ 0.2$	Migration rate between islands
	μ	1×10^{-8}	Mutation rate per base
	r	1×10^{-8}	Recombination rate per base
`hard_sweeps`	N_e	$10^4,\ 2\times10^4,\ 4\times10^4$	Effective population size
	p_s	$2.5\times10^{-5},\ 1\times10^{-5},\ 7.5\times10^{-5}$	Selected site position or selection coefficient
	μ	$1\times10^{-8},\ 5\times10^{-8}$	Mutation rate per base
	r	$1\times10^{-8},\ 5\times10^{-8}$	Recombination rate per base

* Within each model, all-vs-all parameter combinations were simulated; the only exception (when $\mu = 5\times10^{-8}$ & $r = 5\times10^{-8}$) was skipped due to long simulation times.

$$\texttt{time_windows} = \text{log-space}\left(2, \lfloor \log_{10}(T_{\max}) \rfloor, N+1\right)$$

where $T_{\max}$ is the maximum simulation time and N is the number of desired windows. We then explicitly set the first window edge to 0 by assigning `time_windows[0] = 0.0`. This creates N non-overlapping intervals $[t_0, t_1), [t_1, t_2), \ldots, [t_{N-1}, t_N)$ that cover the relevant timescale on a logarithmic scale, providing finer resolution for more recent coalescent events.

Let $T_1, T_2, \ldots, T_n$ be the observed coalescence times. For each window $[t_i, t_{i+1})$, we compute the proportion of events falling into that bin, giving a discretized probability density function (PDF), denoted p_i. The corresponding cumulative distribution function (CDF) is $F(t_i) = \sum_{j<i} p_j$. Assuming a memoryless coalescence process (i.e., a Poisson process), the instantaneous rate $\lambda(t)$ satisfies:

$$\lambda(t) = \frac{f(t)}{1 - F(t)}$$

where $f(t)$ is the density and $1 - F(t)$ is the survival function. Integrating this rate over the interval $[t_i, t_{i+1})$ gives:

$$\int_{t_i}^{t_{i+1}} \lambda(t)\, dt = \log\left(\frac{1 - F(t_i)}{1 - F(t_{i+1})}\right)$$

and dividing by the window width $\Delta t = t_{i+1} - t_i$, we estimate the average coalescence rate in the interval as:

$$\hat{\lambda}_i = \frac{1}{\Delta t} \log\left(\frac{1 - F(t_i)}{1 - F(t_{i+1})}\right)$$

For the final window (where the survival function becomes too small), we estimate the rate via the inverse of the mean residual coalescence time:

$$\hat{\lambda}_{\text{last}} = \left(\mathbb{E}[T - t_{\text{last}} \mid T \geq t_{\text{last}}]\right)^{-1}$$

This approach provides a nonparametric, smoothed estimate of the coalescence rate as a function of time, suitable for demographic inference.

Table 5: `stdpopsim` dataset (v0.2): Simulations configuration for training

Species	**Demographic Model**	**Genetic Map**	**Simulations**
Aedes aegypti	PieceWiseConstant		6.0×10^4
Anas platyrhynchos	MallardBlackDuck_2L19		5.0×10^3
Anolis carolinensis	PieceWiseConstant		1.0×10^3
Anopheles gambiae	GabonAg1000G_1A17		2.0×10^4
Arabidopsis thaliana	SouthMiddleAtlas_1D17	SalomeAveraged_TAIR10	1.0×10^5
Arabidopsis thaliana	African2Epoch_1H18	SalomeAveraged_TAIR10	1.0×10^5
Arabidopsis thaliana	African3Epoch_1H18	SalomeAveraged_TAIR10	1.0×10^5
Bos taurus	HolsteinFriesian_1M13		2.0×10^5
Caenorhabditis elegans	PieceWiseConstant	RockmanRIAIL_ce11	2.0×10^5
Canis familiaris	PieceWiseConstant	Campbell2016_CanFam3_1	2.0×10^5
Drosophila melanogaster	African3Epoch_1S16	ComeronCrossoverV2_dm6	1.0×10^3
Drosophila melanogaster	OoA_2L06	ComeronCrossoverV2_dm6	1.0×10^3
Drosophila sechellia	PieceWiseConstant		6.0×10^4
Gasterosteus aculeatus	PieceWiseConstant		2.0×10^5
Helianthus annuus	PieceWiseConstant		6.0×10^4
Homo sapiens	OoAExtNeaAdmixturePulse_3I21	HapMapII_GRCh38	2.0×10^5
Homo sapiens	OoA_3G09	HapMapII_GRCh38	2.0×10^5
Homo sapiens	OoA_2T12	HapMapII_GRCh38	2.0×10^5
Homo sapiens	Africa_1T12	HapMapII_GRCh38	2.0×10^5
Homo sapiens	AmericanAdmixture_4B18	HapMapII_GRCh38	2.0×10^5
Homo sapiens	OoAArchaicAdmixture_5R19	HapMapII_GRCh38	2.0×10^5
Homo sapiens	Zigzag_1S14	HapMapII_GRCh38	2.0×10^5
Homo sapiens	AshkSub_7G19	HapMapII_GRCh38	2.0×10^5
Homo sapiens	OoA_4J17	HapMapII_GRCh38	2.0×10^5
Homo sapiens	Africa_1B08	HapMapII_GRCh38	2.0×10^5
Pan troglodytes	BonoboGhost_4K19		2.0×10^5
Papio anubis	SinglePopSMCpp_1W22	Pyrho_PAnubis1_0	2.0×10^5
Pongo abelii	TwoSpecies_2L11	NaterPP_PonAbe3	2.0×10^5

Table 6: `stdpopsim` dataset (v0.3): Simulations configuration for evaluation and retraining

Species	**Demographic Model**	**Simulations** (for retraining)
Gorilla gorilla	GorillaGhost_5P23	2.0×10^5
Oryza sativa	BottleneckMigration_3C07	2.0×10^5
Rattus norvegicus	PieceWiseConstant	2.0×10^5
Sus scrofa	PieceWiseConstant	2.0×10^5

Acknowledgments

We thank members of the Kern-Ralph co-lab for helpful discussions and feedback. We also thank the authors of `Gamma-SMC` and `Singer` for their implementations and for providing their code. Finally, we thank Negar Rahnamae for designing the logo for the `cxt` method. This work was supported by NIH awards R35148253 and R01HG010774.

References

Kingman, J.F.C. (1982). "The coalescent". In: *Stochastic Processes and their Applications* 13.3, pp. 235–248. DOI: 10.1016/0304-4149(82)90011-4.

Hudson, Richard R. (1983). "Properties of a neutral allele model with intragenic recombination". In: *Theoretical Population Biology* 23.2, pp. 183–201. DOI: 10.1016/0040-5809(83)90013-8.

Griffiths, Robert C. (1991). "The Two-Locus Ancestral Graph". In: *Institute of Mathematical Statistics Lecture Notes - Monograph Series*, pp. 100–117. DOI: 10.1214/lnms/1215459289.

Griffiths, R.C. and P. Marjoram (1996). "Ancestral Inference from Samples of DNA Sequences with Recombination". In: *Journal of Computational Biology* 3.4, pp. 479–502. DOI: 10.1089/cmb.1996.3.479.

Griffiths, Robert C. and Paul Marjoram (1997). "An ancestral recombination graph". In: *Progress in population genetics and human evolution*, pp. 257–270.

Wiuf, C. and J. Hein (1999). "Recombination as a point process along sequences". In: *Theoretical Population Biology* 55.3, pp. 248–259. DOI: 10.1006/tpbi.1998.1403.

Beaumont, Mark A, Wenyang Zhang, and David J Balding (2002). "Approximate Bayesian Computation in Population Genetics". In: *Genetics* 162.4, pp. 2025–2035. DOI: 10.1093/genetics/162.4.2025.

McVean, Gilean A.T and Niall J Cardin (2005). "Approximating the coalescent with recombination". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 360.1459, pp. 1387–1393. DOI: 10.1098/rstb.2005.1673.

Consortium, 1000 Genomes Project et al. (2010). "A map of human genome variation from population scale sequencing". In: *Nature* 467.7319, p. 1061.

Li, Heng and Richard Durbin (2011). "Inference of human population history from individual whole-genome sequences". In: *Nature* 475.7357, pp. 493–496. DOI: 10.1038/nature10231.

Schiffels, Stephan and Richard Durbin (2014). "Inferring human population size and separation history from multiple genome sequences". In: *Nature genetics* 46.8, pp. 919–925. DOI: 10.1038/ng.3015.

Auton, Adam and Tovah Salcedo (2015). "The 1000 genomes project". In: *Assessing rare variation in complex traits: Design and analysis of genetic studies*, pp. 71–85.

Boitard, Simon et al. (2016). "Inferring Population Size History from Large Samples of Genome-Wide Molecular Data - An Approximate Bayesian Computation Approach". In: *PLOS Genetics* 12.3, e1005877. DOI: 10.1371/journal.pgen.1005877.

Malaspinas, Anna-Sapfo et al. (2016). "A genomic history of Aboriginal Australia". In: *Nature* 538.7624, pp. 207–214. DOI: 10.1038/nature18299.

Sheehan, Sara and Yun S Song (2016). "Deep learning for population genetic inference". In: *PLoS computational biology* 12.3, e1004845.

Terhorst, Jonathan, John A. Kamm, and Yun S. Song (2017). "Robust and scalable inference of population history from hundreds of unphased whole-genomes". In: *Nature genetics* 49.2, pp. 303–309. DOI: 10.1038/ng.3748.

Vaswani, Ashish et al. (2017). *Attention Is All You Need*. DOI: 10.48550/ARXIV.1706.03762.

Kern, Andrew D and Daniel R Schrider (2018). "diploS/HIC: an updated approach to classifying selective sweeps". In: *G3: Genes, Genomes, Genetics* 8.6, pp. 1959–1970.

Palamara, Pier Francesco et al. (2018). "High-throughput inference of pairwise coalescence times identifies signals of selection and enriched disease heritability". In: *Nature genetics* 50.9, pp. 1311–1317. DOI: 10.1038/s41588-018-0177-x.

Radford, Alec and Karthik Narasimhan (2018). "Improving Language Understanding by Generative Pre-Training". In: URL: https://api.semanticscholar.org/CorpusID:49313245.

Schrider, Daniel R and Andrew D Kern (2018). "Supervised machine learning for population genetics: a new paradigm". In: *Trends in Genetics* 34.4, pp. 301–312.

Flagel, Lex, Yaniv Brandvain, and Daniel R Schrider (2019). "The Unreasonable Effectiveness of Convolutional Neural Networks in Population Genetic Inference". In: *Molecular Biology and Evolution* 36.2, pp. 220–238. DOI: 10.1093/molbev/msy224.

Jay, Flora, Simon Boitard, and Frédéric Austerlitz (2019). "An ABC Method for Whole-Genome Sequence Data: Inferring Paleolithic and Neolithic Human Expansions". In: *Molecular Biology and Evolution* 36.7, pp. 1565–1579. DOI: 10.1093/molbev/msz038.

Kelleher, Jerome et al. (2019). "Inferring whole-genome histories in large population datasets". In: *Nature Genetics* 51.9, pp. 1330–1338. DOI: 10.1038/s41588-019-0483-y.

Radford, Alec et al. (2019). "Language Models are Unsupervised Multitask Learners". In.

Speidel, Leo et al. (2019). "A method for genome-wide genealogy estimation for thousands of samples". In: *Nature Genetics* 51.9, pp. 1321–1329. DOI: 10.1038/s41588-019-0484-x.

Torada, Luis et al. (2019). "ImaGene: a convolutional neural network to quantify natural selection from genomic data". In: *BMC Bioinformatics* 20.9, p. 337. DOI: 10.1186/s12859-019-2927-x.

Adrion, Jeffrey R, Jared G Galloway, and Andrew D Kern (2020). "Predicting the landscape of recombination using deep learning". In: *Molecular biology and evolution* 37.6, pp. 1790–1808.

Battey, CJ, Peter L Ralph, and Andrew D Kern (2020). "Predicting geographic location from genetic variation with deep neural networks". In: *eLife* 9, e54507. DOI: 10.7554/eLife.54507.

Sellinger, Thibaut Paul Patrick et al. (2020). "Inference of past demography, dormancy and self-fertilization rates from whole genome sequence data". In: *PLOS Genetics* 16.4, e1008698. DOI: 10.1371/journal.pgen.1008698.

Battey, CJ, Gabrielle C Coffing, and Andrew D Kern (2021). "Visualizing population structure with variational autoencoders". In: *G3* 11.1, jkaa036.

Sanchez, Théophile et al. (2021). "Deep learning for population size history inference: Design, comparison and combination with approximate Bayesian computation". In: *Molecular Ecology Resources* 21.8, pp. 2645–2660. DOI: 10.1111/1755-0998.13224.

Baumdicker, Franz et al. (2022). "Efficient ancestry and mutation simulation with msprime 1.0". In: *Genetics* 220.3, iyab229.

Schmidhuber, Juergen (2022). *Annotated History of Modern AI and Deep Learning*. DOI: 10.48550/arXiv.2212.11279.

Korfmann, Kevin, Oscar E Gaggiotti, and Matteo Fumagalli (2023). "Deep Learning in Population Genetics". In: *Genome Biology and Evolution* 15.2, evad008. DOI: 10.1093/gbe/evad008.

Nait Saada, Juba et al. (Oct. 2023). "Inference of Coalescence Times and Variant Ages Using Convolutional Neural Networks". en. In: *Molecular Biology and Evolution* 40.10, msad211. ISSN: 0737-4038, 1537-1719. DOI: 10.1093/molbev/msad211. URL: https://academic.oup.com/mbe/article/doi/10.1093/molbev/msad211/7279051 (visited on 06/20/2025).

Schweiger, Regev and Richard Durbin (2023). *Ultra-fast genome-wide inference of pairwise coalescence times*. DOI: 10.1101/2023.01.06.522935.

Smith, Chris CR and Andrew D Kern (2023). "disperseNN2: a neural network for estimating dispersal distance from georeferenced polymorphism data". In: *BMC bioinformatics* 24.1, p. 385.

Smith, Chris CR et al. (2023). "Dispersal inference from population genetic variation using a convolutional neural network". In: *Genetics* 224.2, iyad068.

Whitehouse, Logan S and Daniel R Schrider (2023). "Timesweeper: accurately identifying selective sweeps using population genomic time series". In: *Genetics* 224.3, iyad084.

Zhang, Brian C et al. (2023). "Biobank-scale inference of ancestral recombination graphs enables genealogical analysis of complex traits". In: *Nature Genetics* 55.5, pp. 768–776.

Deng, Yun, Rasmus Nielsen, and Yun S. Song (2024). *Robust and Accurate Bayesian Inference of Genome-Wide Genealogies for Large Samples*. DOI: 10.1101/2024.03.16.585351.

Korfmann, Kevin et al. (2024). "Simultaneous Inference of Past Demography and Selection from the Ancestral Recombination Graph under the Beta Coalescent". In: *Peer Community Journal* 4. DOI: 10.24072/pcjournal.397.

Sellinger, Thibaut, Frank Johannes, and Aurélien Tellier (2024). "Improved inference of population histories by integrating genomic and epigenomic data". In: *eLife* 12. DOI: 10.7554/eLife.89470.2.

Terhorst, Jonathan (2024). "Accelerated Bayesian inference of population size history from recombining sequence data". In: *bioRxiv: The Preprint Server for Biology*, p. 2024.03.25.586640. DOI: 10.1101/2024.03.25.586640.

Benegas, Gonzalo et al. (2025). "A DNA language model based on multispecies alignment predicts the effects of genome-wide variants". In: *Nature Biotechnology*, pp. 1–6.

Brixi, Garyk et al. (2025). "Genome modeling and design across all domains of life with Evo 2". In: *bioRxiv*, pp. 2025–02.

Deng, Yun, Yun S Song, and Rasmus Nielsen (2025). "A General Framework for Branch Length Estimation in Ancestral Recombination Graphs". In: *bioRxiv*, pp. 2025–02.

Supplementary Material

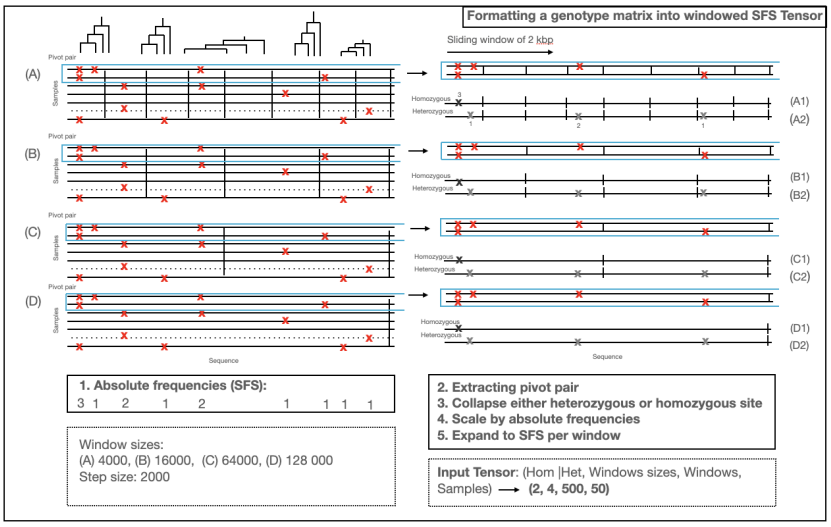

Figure S1: Processing of a phased genotype matrix, as detailed in Algorithm 1. This figure provides a schematic overview. Starting at (**A**), we compute absolute variant frequencies across 50 samples (a fixed hyperparameter) (1). A one-megabase region is divided into four-kilobase windows with a 2 Kb slide (not shown). We focus on a pivot pair (framed in blue), extract it, and classify sites as heterogeneous or homogeneous. Each site is scaled by its SFS class, and each window is expanded into the full sparse SFS dimensions of 50 (see input tensor, bottom right). This process repeats for three additional windows (**B**–**D**), yielding a complete input tensor of shape (2, 4, 500, 50) per pair, with the SFS aggregating information across all samples.

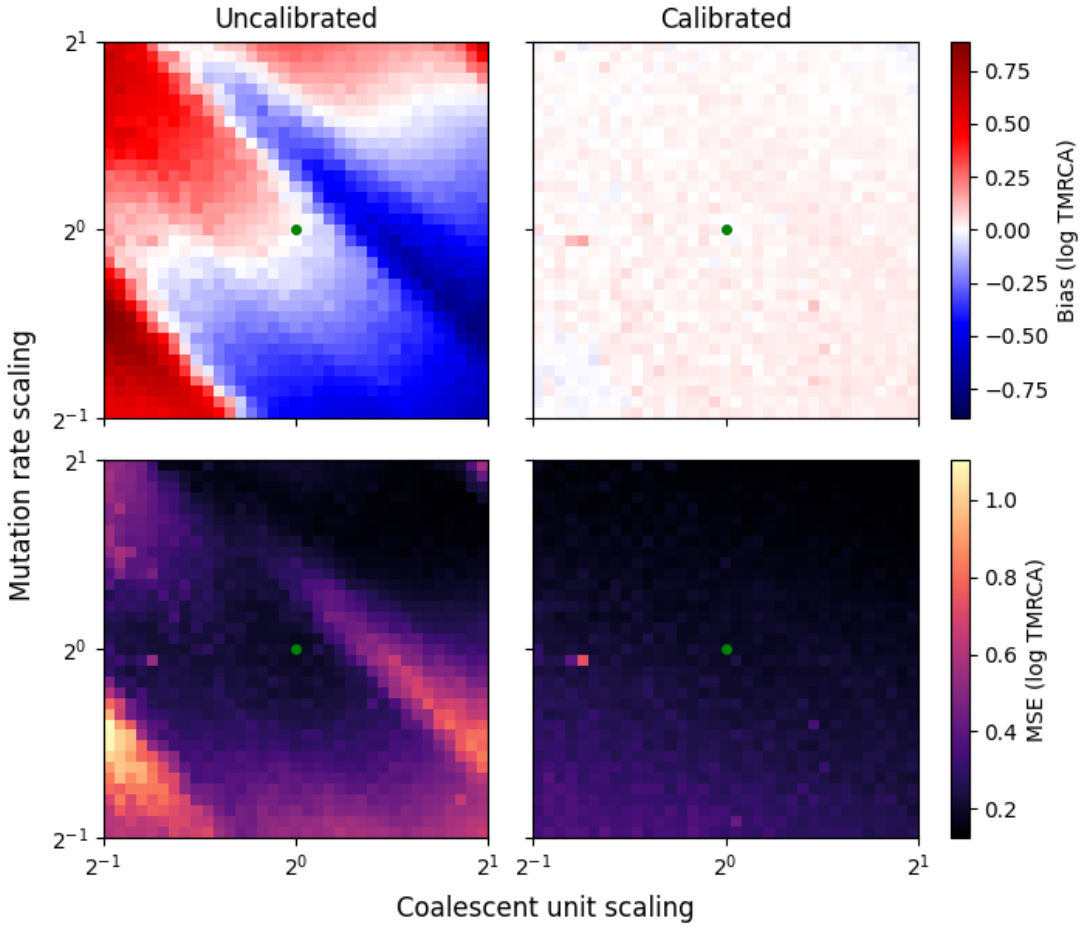

Figure S2: The impact of calibrating the molecular clock using observed nucleotide diversity (see section "Calibration of Molecular Clock" in methods), using simulations from perturbations of the Zigzag model in the `stdpopsim` catalog. The x-axis is a scaling of the Markov generator for the coalescent process, that increases the average TMRCA while keeping the shape of the site frequency spectra constant. The y-axis is a scaling of the mutation rate. The two columns correspond to uncalibrated and calibrated `cxt` predictions (e.g. in the former, the language model is predicting absolute TMRCA; and in the latter relative TMRCA). The two rows are bias (top) and mean squared error (bottom) relative to the true TMRCA, calculated from 30 independent simulations per grid cell. The unperturbed demography (which the language model sees during training) is the green point in the center of each heatmap. As the model is perturbed, the uncalibrated `cxt` predictions become substantially biased in a fashion that is nonlinear with regards to the magnitude of the perturbations (left column); reflecting the fact that the TMRCAs, mutation rate, and recombination rate are not jointly identifiable. However, relative changes in TMRCA are predicted accurately, so that centering the TMRCA trajectories around the observed nucleotide diversity for each pivot pair (e.g. conditioning on the mutation rate) removes the bias regardless of the degree of perturbation. Thus, after calibration (right column) the MSE for `cxt` predictions follows a gradient of mutational density (see Supplementary Figure S3).

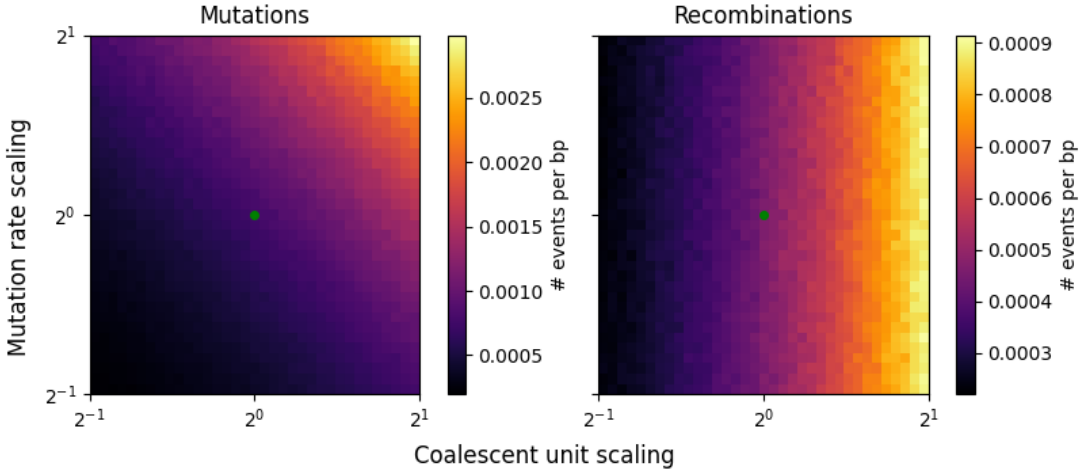

Figure S3: The density of mutations and recombination events across perturbations of the Zigzag `stdpopsim` model, for the simulations shown in Figure S2.

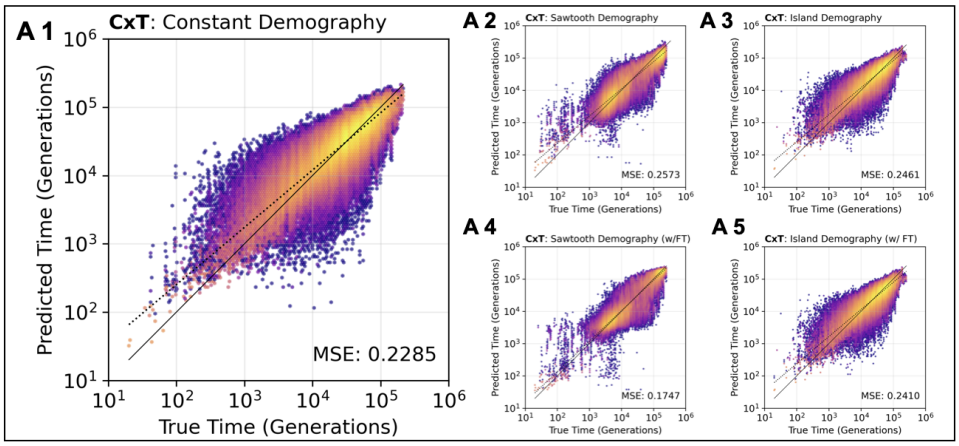

Figure S4: `cxt` (**A**) comparison of true vs. predicted TMRCAs for the (**A1**) constant, (**A2**) sawtooth and (**A3**) island demography, as well as `cxts` results for a fined-tuned variant for (**A4**)) sawtooth and (**A5**) island demography before post-hoc correction.

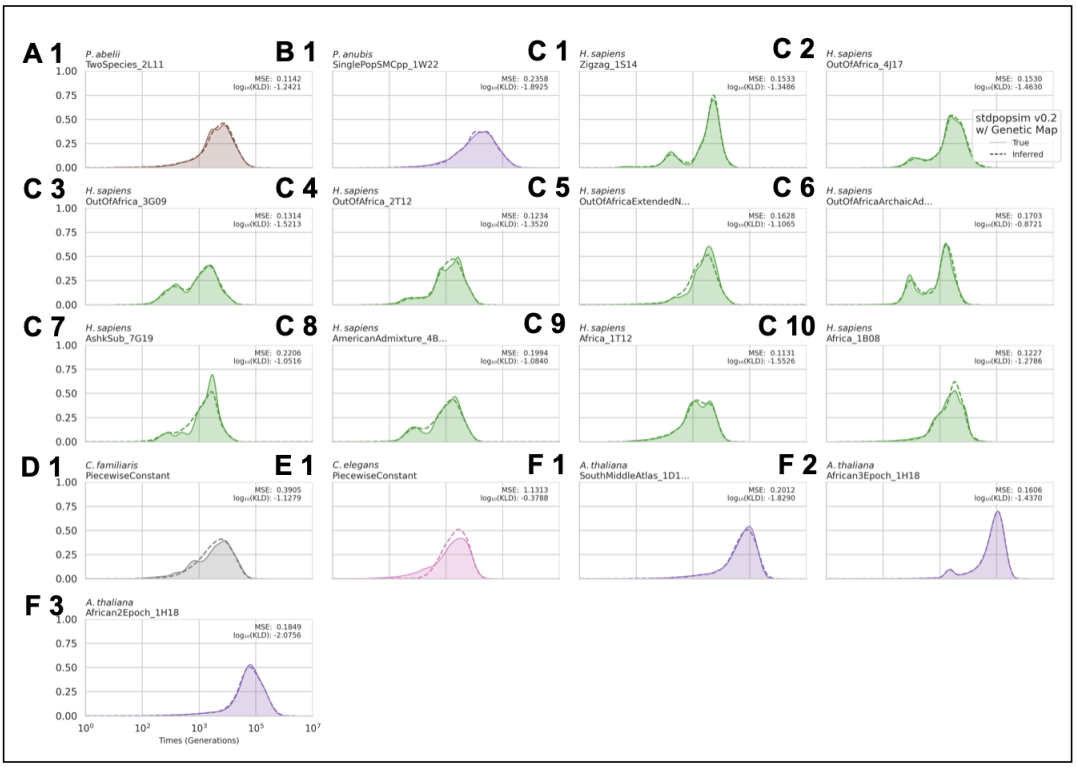

Figure S5: Evaluation of inferred marginal coalescence distributions (dashed line) against the true distributions (shaded line), highlighting the model's **capacity** - its ability to distinguish scenarios based on context alone. The model generalizes to stdpopsim v0.2 simulations (**A1-F3**) with varying mutation, recombination rates, and demography. Distributions are aggregated from pairwise inferences and visualized as kernel densities. All results are from new simulations, excluded from training. Unlike the main figure, this one includes simulations with a genetic map.

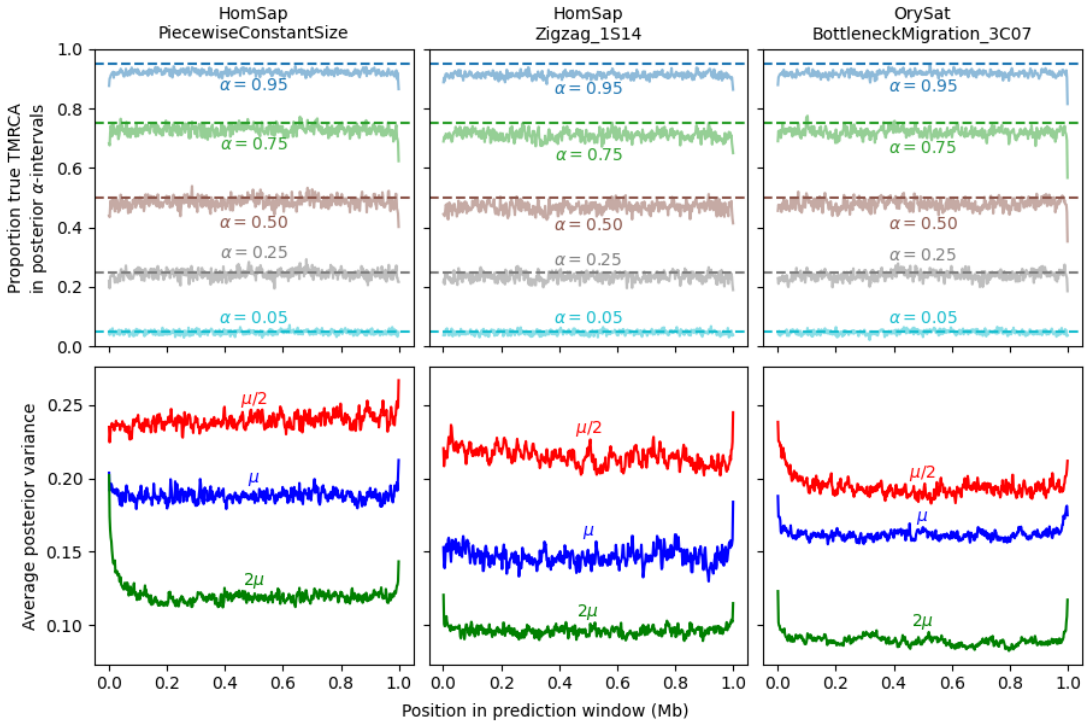

Figure S6: The approximate posterior distributions sampled from by `cxt` give well-calibrated estimates of uncertainty for predicted TMRCAs, across a range of scenarios. Top row: the proportion of true TMRCAs that fall within credibility intervals of a given width (e.g. a given proportion of posterior mass), as a function of position in the 1Mb prediction frame. Credibility intervals are calculated from quantiles of 100 samples from the approximate posterior for each pivot pair, and the proportion covering the true TMRCAs are calculated over 1000 independent pivot pairs (e.g. from distinct genealogical simulations). The expected proportions (for an exact posterior) are shown as dashed lines. Empirically, the approximate posteriors are asymptotically well-calibration, although slightly over-concentrated for larger interval widths (likely due to the time discretization used by `cxt`). The decrease in coverage for the last 2Kb window is due to a tendency of the LLM to introduce large jumps in TMRCA in this window. Bottom row: the posterior variance, averaged over 1000 independent pivot pairs, as a function of position in the prediction frame and mutation rate (where μ stands for the species' mutation rate in `stdpopsim`). As mutational information accrues, the posteriors become more precise; and lose precision towards the boundaries of the prediction frame where local context has been clipped.

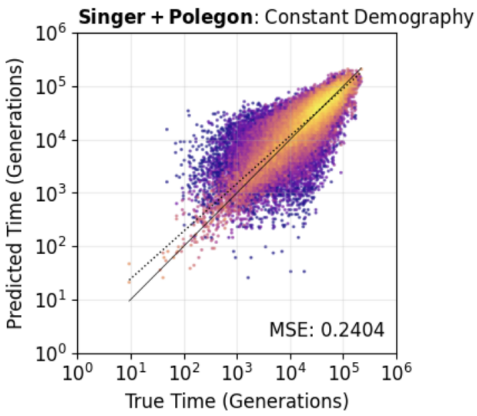

Figure S7: Evaluation of `Singer+Polegon` on the original constant demography scenario leading due an improved MSE evaluation due to changing assumptions of the prior.

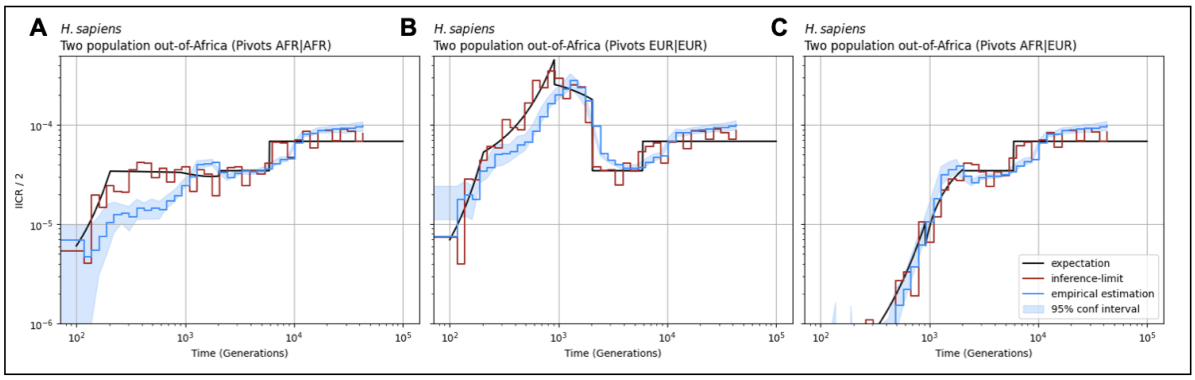

Figure S8: Inverse-instantaneous coalescence rate calculation of a two population out-of-Africa demography for *H. sapiens* using (**A**) only African Samples , (**B**) only European or (**C**) a mixture of African and European samples for cross-coalescence rate estimation. The inference of pairwise-coalescence events leads to the implicit inference of demography estimates through the marginal coalescence distribution assuming coalescence occurs as a Poisson process (see methods). For each scenario 10 Mb and 25 diploid samples have been used to achieve resolution throughout the specified time-windows.

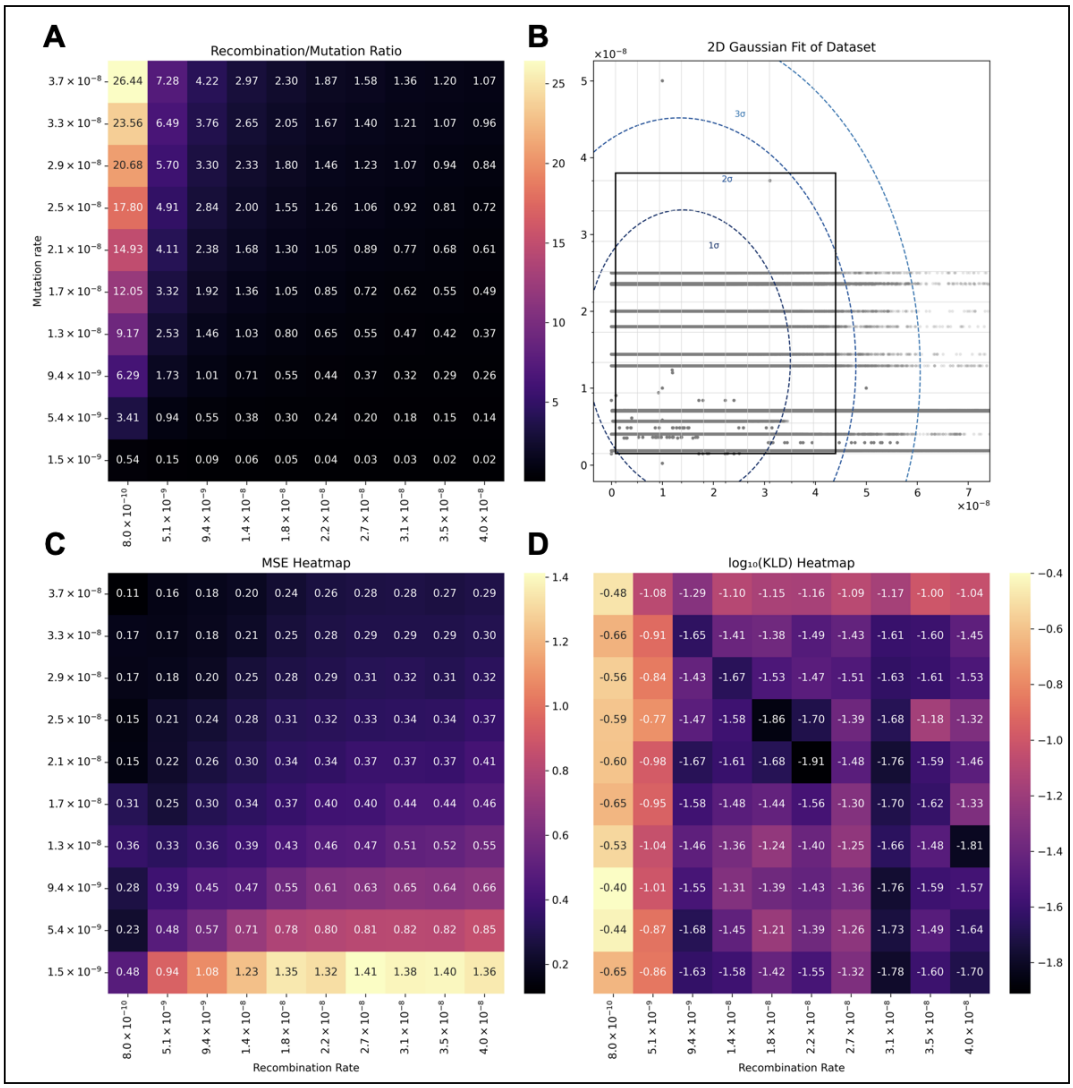

Figure S9: Interpretation and limitations of `cxt` by testing various recombination to mutation ratios (**A**). The ratios have been chosen in such way, that most of stdpopsim v0.2 with the exception of some outliers fall into this grid, while also allowing for some space for the models capacity to interpolate (**B**), particularly the upper corners of (**A-D**). The heatmaps of (**C**) to (**D**), show MSE and KLD values of `cxt` inferences over the entire grid.

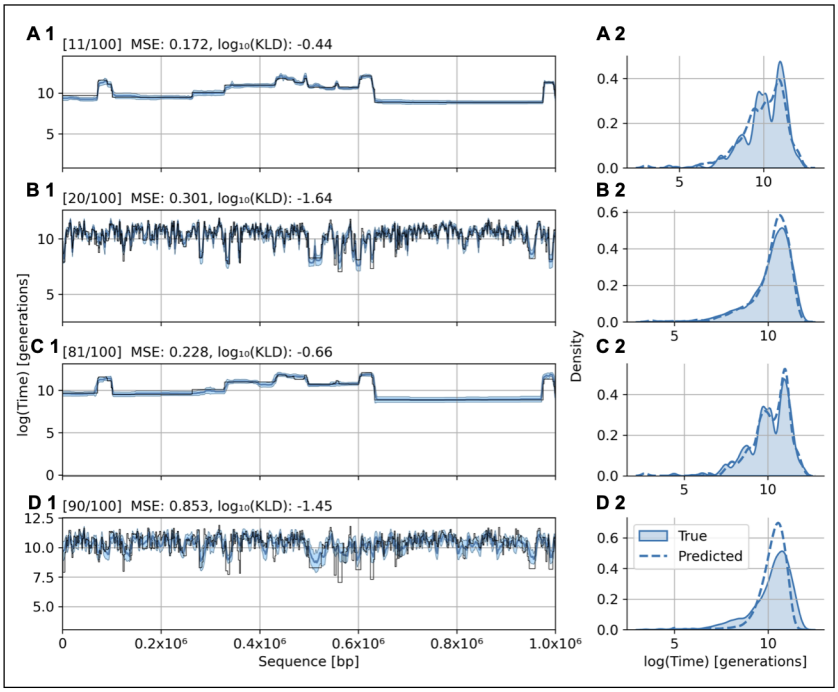

Figure S10: Interpretation and limitations of `cxt` zoomed into four cases of the Supplementary Figures S9. The figures are chosen to be representative for each of the corners of the heatmap of S9 (**A**), starting from top left to bottom right, with MSE and KLD indicated in the subtitles (**A1-D1**). These plots themselves show the TMRCA along the sequence, while (**A2-D2**), show the marginal distribution, true and inferred, respectively.

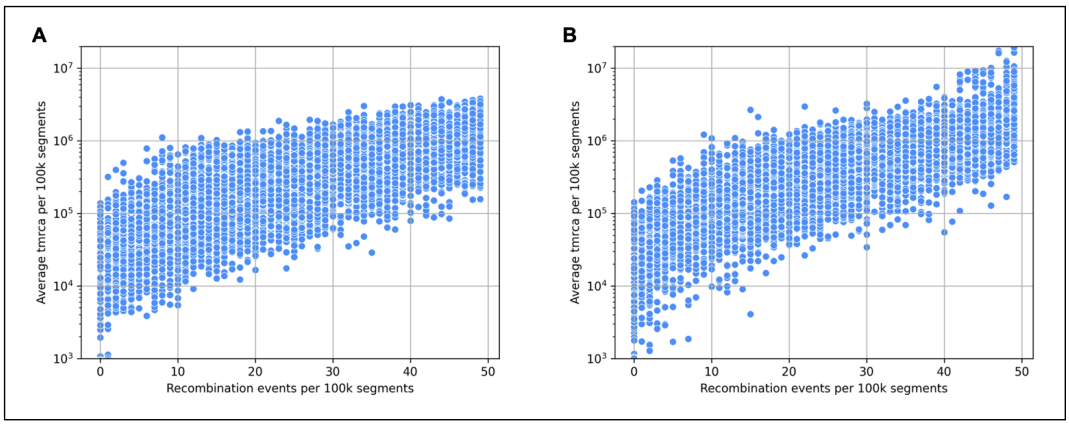

Figure S11: Comparison of the average TMRCA for adjacent 100 Kb windows and the average number of recombination breakpoints for 10 pivot pairs using the inferences from the 1000 Genomes Project for chromosome 2 (**A**) and chromosome 6 (**B**).

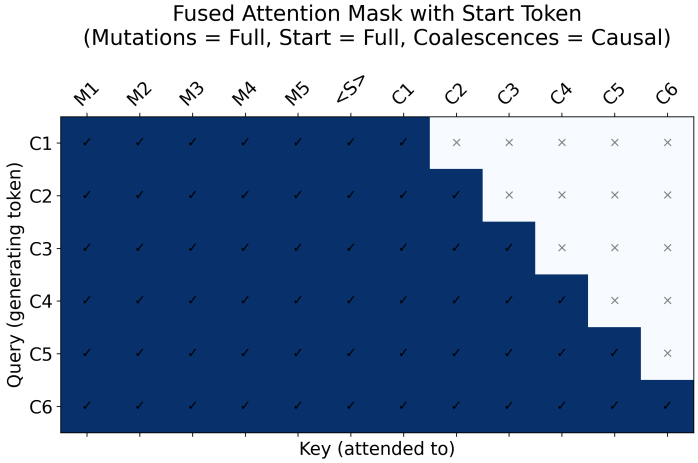

Figure S12: Attention mask modified for translation. The models auto-regressive nature only needs to guaranteed for the target language, being the pairwise coalescence events predicted sequentially. Only five representative mutation or coalescence embeddings are presented, while generally 500 positions are considered, respectively.