Revisiting the out of Africa event with a deep-learning approach

Francesco Montinaro,[1,2,6] Vasili Pankratov,[1,6] Burak Yelmen,[1,3,4] Luca Pagani,[1,5] and Mayukh Mondal[1,*]

Summary

Anatomically modern humans evolved around 300 thousand years ago in Africa. They started to appear in the fossil record outside of Africa as early as 100 thousand years ago, although other hominins existed throughout Eurasia much earlier. Recently, several studies argued in favor of a single out of Africa event for modern humans on the basis of whole-genome sequence analyses. However, the single out of Africa model is in contrast with some of the findings from fossil records, which support two out of Africa events, and uniparental data, which propose a back to Africa movement. Here, we used a deep-learning approach coupled with approximate Bayesian computation and sequential Monte Carlo to revisit these hypotheses from the whole-genome sequence perspective. Our results support the back to Africa model over other alternatives. We estimated that there are two sequential separations between Africa and out of African populations happening around 60-90 thousand years ago and separated by 13-15 thousand years. One of the populations resulting from the more recent split has replaced the older West African population to a large extent, while the other one has founded the out of Africa populations.

Introduction

Recent fossil record analysis suggests that *Homo sapiens* appeared around 300 thousand years ago (kya) in Africa.[1] This hypothesis is corroborated by genetic data,[2] which estimated the deepest split between modern human populations at a similar time point. Although fossil records advocate the occurrence of multiple out of Africa (OOA) events for modern humans,[3] recent genetic studies revealed that all modern non-African populations tested so far fit a model with a single OOA event that happened less than 100 kya.[4–6] This conclusion indicates that earlier OOA migrations, documented by archaeological records, might have not left a substantial contribution to modern human populations, with a possible exception of this model suggested in Papuan populations (two out of Africa model).[7] While the single OOA model is generally supported by both autosomal and uniparental data,[8–10] some observations may reflect a more complex scenario. For example, uniparental data show that most of the haplogroups found in OOA populations have a younger time to the most recent common ancestor (tMRCA) than those found in Africa, with the notable exception of the sister Y haplogroups D and E. Haplogroup D can be found in isolated populations in Asia (i.e., Andamanese, Tibetan, Japanese, etc.), while haplogroup E is ubiquitous in sub-Saharan African populations. These two haplogroups coalesce together before coalescing with any other haplogroup found in OOA populations.[10,11] This observation might be explained by a back to Africa migration involving a population harboring the haplogroup E[10] or a simple out of Africa with haplogroup E remaining in Africa but haplogroup D and other OOA haplogroups coming out of Africa together.[12] Furthermore, some genetic analyses suggest that the separation between Africa and OOA populations might not be a single split event but rather that it was constituted by structure and admixture between populations.[13–16]

Testing these hypotheses (single out of Africa, back to Africa, and two out of Africa) is challenging because of the strong bottleneck of non-African populations,[17–19] differential amount and source of archaic introgression,[4,20,21] and several migrations within Africa.[2,16] The lack of ancient genomic data older than 15 kya[22] from Africa makes it difficult to address this issue from an ancient DNA perspective. However, neural networks (NNs) have been shown to be extremely powerful to disentangle such complex scenarios.[20]

In the last few decades, the development of efficient and powerful computing infrastructure allowed us to gain substantial progress in the machine learning field, especially for computationally demanding algorithms such as NN[23,24] and Bayesian methods.[25,26] NN has been demonstrated to be a useful tool for specific types of tasks, such as classification or natural language processing.[23–25,27] However, NN requires a large amount of data as a training set. In some cases, this limitation can be overcome via the use of simulated data. Given that the modern tools allow for efficient simulation of large sets of synthetic genetic data,[28,29] NN is already adopted in population genomics for interpretation of the genomic data in terms of underlying demography[20,30,31] and positive selection.[32,33]

[1]Institute of Genomics, University of Tartu, Tartu 51010, Estonia; [2]Department of Biology-Genetics, University of Bari, Bari 70124, Italy; [3]Institute of Molecular and Cell Biology, University of Tartu, Tartu 51010, Estonia; [4]Université Paris-Saclay, CNRS UMR 9015, INRIA, Laboratoire Interdisciplinaire des Sciences du Numérique, 91400 Orsay, France; [5]Department of Biology, University of Padova, Padova 35121, Italy
[6]These authors contributed equally
*Correspondence: mondal.mayukh@gmail.com

https://doi.org/10.1016/j.ajhg.2021.09.006.

However, unlike classical approaches, it is still challenging to measure the significance of a prediction performed by NN given that it is a black-box approach. Approximate Bayesian computation (ABC) can be used to weigh the accuracy of a NN-based prediction from the data itself without knowing the maximum likelihood function.[20,30,34] However, ABC may face the "curse of dimensionality" problem when high dimensional input data or summary statistics are used.[35] It performs less efficiently when the harnessed summary statistics do not capture enough information. Here, we used NN to reduce the dimensionality of the original summary statistics and to produce highly informative secondary summary statistics, which is used as input by ABC to perform statistical analysis. Sequential Monte Carlo (SMC, also known as the particle filter method)[36] is used to recursively predict posterior distribution from a prior. In every step, likelihood weight is calculated for each prior sample. Low likelihood samples are discarded, and new samples are resampled from a continuous distribution. In classical SMC, higher probability was given to newer samples, which have higher weights in resampling steps. The likelihood weights are calculated for all these samples and the cycle repeats until convergence is reached. SMC has a strong resemblance to genetic algorithms.[37] It has been shown that ABC estimation can be further improved by the use of SMC,[38–40] but to date they have not been used together with NN.

Here, we present an improved version of ABC-DL (approximate Bayesian computation with deep learning)[20] along with ABC-DLS (approximate Bayesian computation with deep learning and sequential Monte Carlo method), which allows us to infer the most likely scenario among different competing demographic models as well as to estimate their parameter values with high precision. Our approach relies on an NN trained on simulated genetic data under the models being tested. However, it has three key improvements compared to other similar approaches. First, the use of the hdf5[41] data format and tensor flow[42] allows for extremely large training datasets. Second, the conventional NN approach is improved with ABC, which helps to provide statistical support for the NN prediction and to obtain posterior distribution for the model parameter values. Third, inspired by previous works,[38] we applied a modification of the SMC[36] approach to iterate the whole procedure. This algorithm improved the prediction accuracy substantially compared to previously implemented methods.[20,43] We apply this method to test the three OOA models mentioned above.

Material and methods

Real data

We have downloaded the high-coverage 1000 Genomes[44] vcf files produced by the Coriell Institute for Medical Research. We randomly selected five individuals to represent Africa, Europe, and East Asia originating at Yoruba from Ibadan, Nigeria (YRI), Utah residents with European origin (CEU), and Chinese Han from Beijing (CHB) populations (15 individuals in total). As an alternative dataset, we have also downloaded the high-coverage Human Genome Diversity Project (HGDP) vcf files[45] and randomly selected five Yoruba (YRI), five French (FRN), and five Han Chinese (HAN) individuals representing Africa, Europe, and East Asia (Table S1). Unless mentioned otherwise, the results presented here were obtained on the first dataset. In order to avoid introducing any bias by the including only a subset of mixing archaic humans due to availability of such genomes and mixing ancient genome with the modern humans, we decided not to use any ancient genome. Moreover, it is suggested that both Neanderthal and Denisova sequences, which are available, are distantly related with the introgressed populations,[46,47] suggesting an even more complicated model. As our primary objective is with the OOA event and to give all archaic populations the same treatment (because African archaic populations are not yet sequenced), they are treated as ghost populations in our simulations. We treated each dataset independently and kept only bi-allelic single-nucleotide polymorphisms (SNPs) with genotype calls present in every individual and lifted the genomic positions to GRCh37 coordinates by using Picard tools. Next, we used two alternative filtering approaches: (1) filtering out genes and CpG islands (for more details, please see Mondal et al., 2019[20]) and (2) applying a mappability mask coming from SNPable[13] for the 1000 Genomes dataset in Table S2. Filtering strategy 1 will be preferable to 2 in the case that the information of CpG islands and genes are available. In the absence of those information (for example non-human genetic data), our method can still reproduce similar result with filtering strategy 2. All the filtering was performed with vcftools and bcftools.[48,49] The vcf file was converted to joint unfolded site frequency spectrum (SFS) via an inhouse code with scikit allel.[50] For unfolded SFS, we need to know ancestral allele and derived allele for the real data (for simulation, it is already known). We used 1000 Genome ancestral allele alignments of fasta files to polarize the alleles to ancestral and derived. When ancestral allele information is not present for a given SNP, we have removed that SNP from our analysis.

Genotype to site frequency spectrum

The SFS from the three populations (Africa, Europe, and East Asia) was computed as a tridimensional tensor from the sequenced or simulated data and further transformed into an array to be used as input by the NN. SFS is the total number of segregating sites for a given derived allele count present in each population.

$$\mathrm{SFS(i,j,k)} = \sum_{m=1}^{n} \begin{cases} 1, \ \mathit{if}\, s_1 = i \ \mathit{and} \ s_2 = j \ \mathit{and} \ s_3 = k. \\ 0, \ \mathit{otherwise}. \end{cases}$$

Where i, j, and k are the numbers of derived alleles count per SNP in pop1, pop2, and pop3, respectively, and n is the total number of segregating SNPs.

The whole SFS was represented as a row. We multiplied all the elements of the real SFS by a constant (frac) to make it comparable with the length of simulated regions if those do not match. For example, we multiplied the real SFS by 100/647 (or 0.1546) if we simulate a 100 megabase pair (Mb) region per simulation and the real data is coming from 647 Mb region (after filtering). In general, it is possible to use any summary statistics for our approach. We used SFS for our choice of summary statistics because it is straightforward to obtain and informative enough.[17,51]

Simulations

All the simulations were done in msprime.[29] We simulated 100, 500, 1,500 or 3,000 1 Mb genomic regions per run (depending

Table 1. Prior parameters range used for producing the site frequency spectrum (SFS)

Parameters	OOA_S	OOA_B	OOA_M
N_A	5,000–25,000	5,000–25,000	5,000–25,000
N_AF	10,000–150,000	10,000–150,000	10,000–150,000
N_EU	10,000–150,000	10,000–150,000	10,000–15,0000
N_AS	10,000–150,000	10,000–150,000	10,000–150,000
N_EU0	500–5,000	500–5,000	500–5,000
N_AS0	500–5,000	500–5,000	500–5,000
N_B	500–5,000	500–5,000	500–5,000
N_BC	N/A	500–40,000	N/A
N_AF0	N/A	500–40,000	N/A
N_MX	N/A	N/A	500–40,000
N_B0	N/A	N/A	500–40,000
T_FM (ky)	2–5	2–5	2–5
T_FS (ky)	0.1–10	0.1–10	0.1–10
T_DM (ky)	10–50	10–50	10–50
T_EU_AS (ky)	5–30	5–30	5–30
T_NM (ky)	5–50	5–50	5–50
T_XM (ky)	5–120	5–120	5–120
T_Mix (ky)	N/A	5–50	5–50
T_Sep (ky)	N/A	5–50	5–50
T_B (ky)	5–270	5–220	5–220
T_AF (ky)	5–700	5–700	5–700
T_N_D (ky)	330–450	330–450	330–450
T_H_A (ky)	120–250	120– 250	120–250
T_H_X (ky)	450–700	450–700	450–700
Mix (%)	N/A	5–95	5–95
NMix (%)	1–3	1–3	1–3
DMix (%)	0–2	0–2	0–2
XMix (%)	0–10	0–10	0–10
FMix (%)	0–10	0–10	0–10

N/A means not applicable and ky means kilo or thousand years.

on the analysis step) for five individuals per population (African, European, and East Asian). For each run, we used a uniform recombination rate of 10^{-8} per base pair (bp) per generation (because SFS is not affected by the local recombination rate[52]) and a mutation rate of 1.45×10^{-8} per bp per generation[53] while sampling demographic parameters from a uniform distribution within prior ranges shown in Table 1. We also alternatively used a mutation rate of 1.25×10^{-8} per bp per generation (only for Table S3).[54,55] We assumed generation time of 29 years.[56] Most of our simulations for SFS were done on multiple of 1 Mb regions, except Table S4 (which was produced by multiple of 25 Kb regions) and when we created mock SFS to test our DLS approach (simulation parameters coming from Tables 2, S5, and S6). To make the mock SFS similar to real data (which has a length of 647 Mb region after filtering), we simulated three 200 Mb regions and one 47 Mb region together to create a single SFS under a given model. We also simulated on multiple of 25 Kb regions (Table S4) to check whether the length of simulated regions can bias our result. 1 Mb regions were chosen for most of the simulated regions' length because it is faster to produce.

In msprime, admixtures were represented as MassMigration (the fraction of a population replaced by another population in a single generation). In contrast, constant migrations under island models (where applicable) were represented as Migrationrate (the fraction of recipient population replaced by migrants from another population per generation).

The ABC-DLS analysis is efficient enough to be done on a single computer. The main bottleneck of the whole approach is the production of the simulated data. Msprime is fast, but the total amount of data, which needs to be simulated for the NN, is sometimes difficult to produce on a single computer (especially for parameter estimation). Thus, we have used a snakemake pipeline

Table 2. Posterior range for parameters of model B

Parameters	Mean	CI	Events (kya)
N_A	14,526	14,404–14,595	N/A
N_AF	26,436	25,535–28,595	N/A
N_EU	94,437	88,320–104,955	N/A
N_AS	127,071	112,953–138,256	N/A
N_EU0	1,838	1,794–1,922	N/A
N_AS0	760	739–776	N/A
N_BC	16,744	11,108–26,850	N/A
N_B	2,026	1,984–2,096	N/A
N_AF0	35,773	33,296–38,019	N/A
T_DM (ky)	18	17.7–18.4	18 (17.7–18.4)
T_EU_AS (ky)	15.1	14.7–15.6	33.1 (32.5–33.6)
T_NM (ky)	5.2	5–5.8	38.3 (37.6–39)
T_XM (ky)	15.7	14.5–16.3	48.7 (47.6–49.8)
T_Mix (ky)	15.1	14.1–16.5	48.2 (46.8–49.5)
T_Sep (ky)	9.8	8.6–11.3	57.9 (56.1–59.8)
T_B (ky)	13.4	12.7–13.7	71.3 (69.4–73.3)
T_AF (ky)	200.4	197.2–202.1	271.7 (268.6–274.8)
T_N_D (ky)	448.1	442–450.9	448.1 (442–450.9)
T_H_A (ky)	248.1	240.6–252.1	696.2 (688.9–703.4)
T_H_X (ky)	675.9	648.5–700.9	675.9 (648.5–700.9)
Mix (%)	91.14	90.28–91.57	N/A
NMix (%)	2.99	2.95–3	N/A
DMix (%)	0.67	0.61–0.72	N/A
XMix (%)	5.29	4.02–6.22	N/A

CI is the confidence interval of 2.5%–97.5% of respective parameters. ky means kilo years and kya means kilo or thousand years ago from now. ky here represents the time interval of the event and kya represents the estimated time that event happened from now. The relation between events and time intervals can be found in Table S10. N/A, not applicable.

to produce the SFS on the cluster.[57] All the simulations with corresponding parameters were saved in a comma-separated value (CSV) file, which was then used by different approaches (approximate Bayesian computation via random forest [ABC-RF], ABC-DL, and ABC-DLS).

ABC-DL

We have implemented an improved version of ABC-DL[20] (here onward, we call this implementation DL, and the implementation that additionally uses SMC [called DLS] is described below) by using state-of-the-art machine learning packages in python. We used TensorFlow with Keras backend[42] for building the NN and abc package for implementing ABC.[58] The use of hdf5 format[41] enabled us to analyze the whole dataset without loading into the memory. This allows us to train the NN on extremely large simulated datasets.

Parameter estimation with DL

Here, we describe parameter estimation by using NN and ABC. We ran a total of 60,000 different simulations, each producing 3,000 of 1 Mb regions (3 Gb [gigabase pair] in total, roughly equal to the length of the human genome). Every line of the NN input is one such simulation performed under a given demographic model with the columns representing SFS elements and the parameters used for that simulation are used as known output. We ran parameter estimation by using this information and retrieve the parameters predicted on observed data for a given model. We used the known parameters as output for training the NN (y), and we used the SFS as input (x). Thus, we can think of the NN as a non-linear model for predicting the parameters from the SFS. We kept out 10,000 random lines for the testing dataset and ABC analysis, and the rest were used for training the NN. Both SFS and parameter values were normalized per column with MinMax scaler (from scikit-learn)[59] so that each data entry is within 0 and 1.

We used four hidden layers of a dense NN (Figure S1) with activation relu (from Keras package). These are basic building blocks for NNs. We used the linear layer (from Keras package) for the output layer with the same number of units as the number of parameters. We used the masking layer (from Keras package) at the beginning to remove cells with zero values from the learning algorithm. This was done so that NNs do not learn the absence of data.

Then we used Gaussian noise injection (from Keras package) of 0.05 to introduce some noise (Figure S1) to reduce overfitting because this will force the NN learn the true parameters even though the SFS is slightly distorted (or noisy). We used logcosh as a loss function and Nadam for the optimizer (both from Keras package). Although we have used other more classical approaches (i.e., mean square error, stochastic gradient descent, etc.), we found this combination to be better suited for our approach. The NN ran through the training dataset several times (epochs) to increase accuracy. We used EarlyStopping (from Keras package) on loss coming from validation dataset with the patient of 100. This stopped the epoch cycle in case there was no improvement for the last 100 epochs (meaning the NN reached a convergence for the current data). We used the ModelCheckpoint (from Keras package) of the lowest loss result on validation data, as the last epoch is not guaranteed to have the lowest loss on validation data. We also used ReduceLROnPlateau (from Keras package) of factor 0.2 to reduce the learning rate if we reached minima for several epochs (ten by default). Learning rate too low at the start makes the NN learn slow, but in the later stage, high learning rate performs worse than lower learning rate.

After training is done, we used the testing dataset to predict the parameter from the SFS, which was then used for cross-validation tests and parameter prediction via loclinear from ABC[58] with the tolerance of 0.01. The whole approach is presented in a flowchart (Figure S2).

Model selection with DL

Here, we describe model selection by using NN and ABC. Unless stated otherwise, we simulated 2,000 simulations for each of the three demographic models tested (see full description of demographic models below). We simulated all the models for 3 Gb regions to make it equal for different approaches and finally scale the real data accordingly. This approach is preferable because we can train the NN only once for different length of real data. To test whether our scaling approach and the length of simulated region are not creating any biases, we also simulated an only 647 Mb region (equal to our filtering strategy 1) with the multiple of 25 Kb regions without using scaling in Table S4. The resulting CSV files (one per model) are used together as input for model selection. Model selection was repeated ten times both for DL and DLS. We found that if the results have high accuracy (posterior probability on observed data is >95% for the correct model), all the ten runs are essentially equal (which is expected).

We used SFS as input (x) for the NN and the model category as output (y) and removed the parameters values because they are not needed for this step. We used MinMax scaler from sklearn[59] only on the SFS data as above, and the names of the models are converted to one-hot encoding by pandas.Categorical (from pandas package) and Keras.utils.to_categorical (from Keras package). After concatenating files coming from all the competing models, we shuffled the rows by a custom code.[60] We left out around 1,000 random rows per model to test the power of NN (as a testing dataset) and for ABC analysis and used the rest to train the NN (as a training dataset). The rest of the approach is as described above (see parameter estimation with DL). We also tested these models with a much higher number of simulations (50,000 for training and 10,000 for testing and ABC analysis), but because we did not find any substantial differences in the results, we used 2,000 simulations (1,000 for training and 1,000 for testing and ABC analysis) per model throughout this study.

We used two hidden layers of the dense NN with the activation relu (Figure S1). We used the softmax layer (from Keras package) for the output layer with the same number of units as the number of trained models. We added the masking layer and a noise injection layer as above. We used a 1% dropout layer (from Keras package) within every dense layer to make it more robust. We used categorical_crossentropy for the loss function and adam for the optimizer (both from Keras package).

After the training was done, we used the testing dataset to predict models from the simulated SFS, which we then used to perform the cross-validation test by using ABC with the tolerance of 0.033 (which converts to 100 samples) in case of three models and 0.0066 (which converts to 100 samples) in case of 15 models via rejection (from abc package). We applied the model selection by abc.postpr (from abc package) to the real data. See a schematic representation in Figure S2.

ABC-DLS

Our implementation of the ABC-DL method was further improved by using the SMC approach,[36] which opens up a new way to make inferences for similar cases (here onward, we call this implementation DLS). Although SMC has lots of different implementation and can be quite complex, we used a simpler version of SMC here. In our implementation, we only selected the top 5% of best samples coming from every cycle of ABC and discarded the rest. The ranges of the parameter values were estimated from these top samples and new samples were drawn from the uniform distributions for these new posterior ranges. To introduce mutations in the algorithm, we increased the posterior range by 1% in every cycle. As a result, the simulated SFS, which is used as input for the NN, becomes more and more similar to real or observed SFS with every iteration of the algorithm.

Parameter estimation with DLS

This method uses the standard parameter estimation strategy of DL (described above) together with the modified SMC algorithm used for recursion.

For parameter estimation, we used the rejection method in ABC with a tolerance of 0.05 as the rejection method always generates posterior within the prior range. We obtained the posterior range by taking the minimum and the maximum values from the ABC output. This range was then used as a prior range for the next iteration. This cycle is repeated until decrease for all parameters is more than 0.95, suggesting it has reached convergence.

$$decrease = \frac{Posterior_{max} - Posterior_{min}}{Prior_{max} - Prior_{min}}$$

If the decrease is more than 95% for a parameter, the new posterior estimation is rejected for that parameter. Instead, we take the prior range of this step, expand it by 1% of its width and use it as a prior again. We used this strategy to prevent the posterior from shrinking or collapsing unless NN has found some accurate prediction for the parameter (decrease < 95%) and reduces the probability of missing the correct parameter value due to stochastic effect in a single cycle.

$$Posterior_{min} = Prior_{min} - \frac{(Prior_{max} - Prior_{\min}) \times 1\%}{2}$$

$$Posterior_{max} = Prior_{max} + \frac{(Prior_{max} - Prior_{\min}) \times 1\%}{2}$$

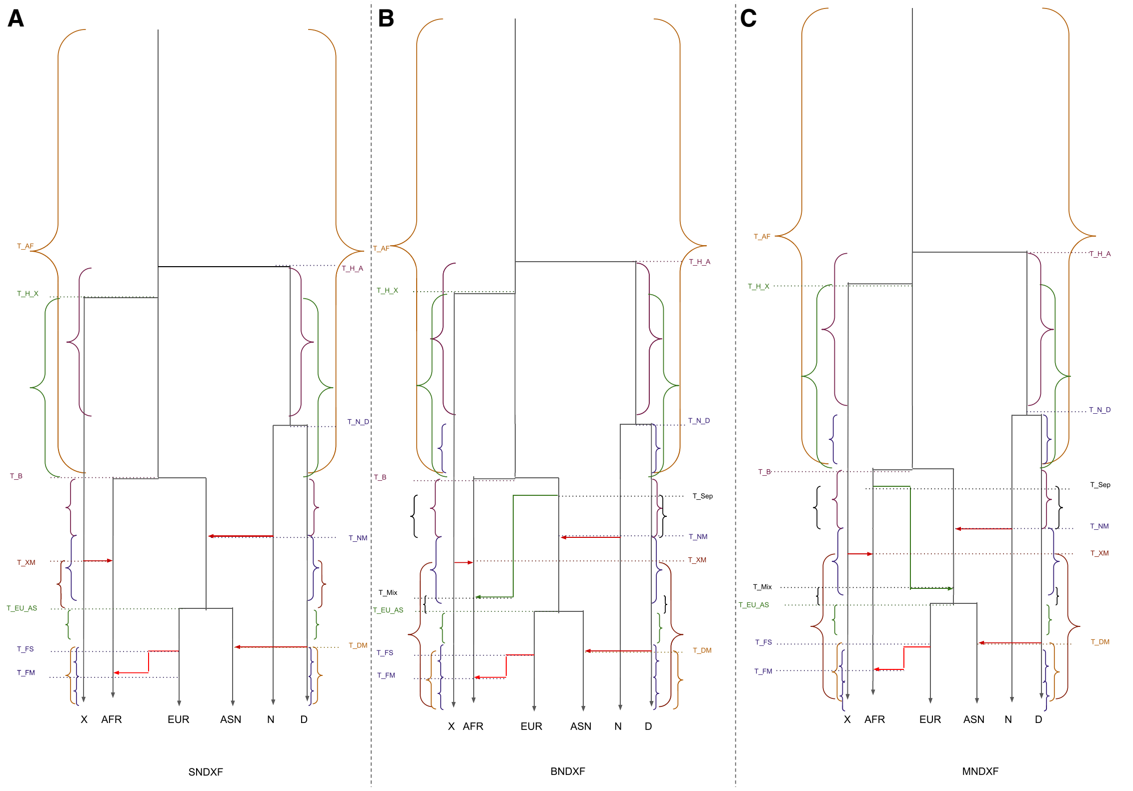


Figure 1. The simplistic schema of the models

(A–C) Simple out of Africa (model S) (A), back to Africa (model B) (B), and out of Africa mixed (model M) (C). Please consult Table 1 for more details of time parameters (in y axis)and Table S10 for the interdependencies of those parameters. AFR is African, EUR is European, and ASN is East Asian. X is African archaic, N is Neanderthal, and D is Denisova.

Simulations from every step were stored and re-used in subsequent cycles if their parameter values fell inside the new prior range. A flow chart of this strategy can be found in Figure S2C.

We used 10,000 simulations as a training dataset and 10,000 simulations for testing. The NN model is exactly as before (as used in DL, Figure S1). To make it more efficient, we started with simulating a total length of 100 Mb (each simulated region being 1 Mb long), and then we increased the total length stepwise (i.e., 0.5, 1.5, and 3 Gb). The priors for 100 Mb regions are the same as presented in Table 1. The final posterior (after convergence reached) of the run with 100 Mb is used as a prior for 0.5 Gb simulation and so on. We multiplied the observed SFS by frac accordingly to scale it to the simulated region length.

After the convergence was reached with 3 Gb in total, we finalized by running 50,000 training and 10,000 testing simulations with the DL method by using loclinear (from abc package) with the tolerance of 0.01. The flowchart of the method is represented in Figure S3.

Model selection with DLS

Here, we describe model selection by using NN, ABC, and SMC together. In principle, we can directly use the final output of the parameter estimation procedure by DLS for every model and then use it for the ABC classification approach. However, this approach would be inefficient given that only one model is true for our real dataset, and thus spending considerable resources to optimize parameters for unlikely scenarios does not make sense. Instead, we used the output of 100 Mb parameter optimizations from the DLS approach as a prior to every model, and then we used the model selection strategy of DL, as mentioned before. In other words, we first optimized the parameters for each model class by using 100 Mb of simulated sequence and then compare the different models between each other. We found out that we already have enough power to distinguish between models by using 100 Mb of total simulated sequence for most of the cases, except Table S7, where we used 500 Mb regions for optimization.

ABC-RF

We tested the real SFS against the three simulated models by using a similar ABC approach but with random forest[61] (here onward, we call this implementation RF) as an inferential tool implemented in the abcrf R package.[43,62] First, we trained our model by using the bagging method applying the function abcrf, with no linear discriminant analysis, and 2,000 decision trees by using 1,000 simulations for each tested model. We then evaluated the performance of ABC-RF through a cross-validation dataset composed of 1,000 simulations for each tested model by using the function predict.abcrf. The same function and settings were used for inferring the best-supported model with the SFS obtained from real data described above. We performed parameter estimation for the most supported scenario applying regression as implemented in the regAbcrf model by using 1,000 decision trees. Each parameter was inferred separately. Similar to DL and DLS approaches, the whole procedure has been repeated ten times for model selection.

Demographic models

Simple out of Africa (model S)

In this simulation model, we have modeled a simple OOA event (Figure 1A) closely following Gravel et al.,[19] except the migration rates are set to zero. All the main results were used without any migration rates, but we also tested models with migrations sometimes. When we simulated a model with constant migrations, we tried both a symmetrical migration matrix as in Gravel et al.[19] (Table S8) and a nonsymmetrical one (Tables S9 and S10).

Back to Africa (model B)

In this model, the basic OOA model still holds and additional changes required for the back to Africa migration (Figure 1B) are added. The basic idea was drawn from Poznik et al.[10] In this scenario, the basal out of Africa population splits into back to Africa and OOA at T_Sep generations ago before the split between European and Asian populations (so that T_Sep is between T_B [the time of separation between Africa and OOA] and T_EU_AS [the time of separation between Europe and Asia]). Next, the back to Africa population migrated to Africa having an effective population size of N_BC and mixed with the ancestral African population at T_Mix generations ago with a mixing proportion of mix (the portion of ancestral African ancestry replaced). After the admixture, the effective population size of the African population is changed from N_AF0 to N_AF.

Mixed out of Africa (model M)

This model (Figure 1C) is similar to model S and has an additional population M separating from the African population T_Sep generations ago (again between T_B and T_EU_AS) and having an effective population size of N_MX. M mixed with OOA at T_Mix generations ago, and mix is the proportion of OOA ancestry being replaced by M. After the admixture, the effective population size of OOA is changed from N_B0 to N_B. The basic idea came from two OOA hypothesis.[7]

Recent admixture in Africa (model R)

As an alternative hypothesis, we also simulated a recent admixture model for the African population. After the OOA population had separated from the African population, the latter splits into two sub-populations in AF1 and AF2 with effective population size of N_AF1 and N_AF2 at T_Sep generations ago. At T_Mix generations ago, these two populations admixed with each other so that a fraction of AF1 equal to mix is replaced by AF2. After the admixture, the effective population size of the African population becomes the modern effective population size of Africa. We have not simulated Neolithic farmer migration from Europe for this scenario.

Other migrations as prior

We also added some pulse migrations or admixtures proposed by different studies on top of these basic models. We simulated OOA to have introgression from Neanderthal[63] at T_NM generations ago with the proportion of NMix. After the separation between Europeans and East Asians, the East Asian population has an introgression from Denisova[47,64] or an unknown archaic population[20] at T_DM and the amount is DMix. Neanderthal separated from Denisova or the unknown population around T_N_D generations ago, and Neanderthal-Denisovan lineage separated from the modern human lineage T_H_A generations ago.[65,66] The African population also has introgression from another unknown archaic population,[16,67,68] which introgressed at T_XM generations ago with the proportion of XMix. This unknown population separated from modern human lineage around T_H_X generations ago. We observed that our method is incapable of finding the effective population size for any archaic population (most likely because we did not use any ancient genome in our real data). Thus, we assumed them to be equal to N_A (ancestral effective population size). We also simulated Neolithic farmers, which separated from Europeans around T_FS generations ago with effective population size of N_A and admixed with the African population around T_FM generations ago with the proportion of Fmix.[69]

For some events, their order is fixed (for example, the separation of European and Asian populations can only happen after the Neanderthal introgression on the basis of our prior assumption) and is described in Table S11.

Relate

We used Relate v.1.1.4,[14] a method for inferring local trees, to validate our parameter estimates. Relate uses branch length of the local trees to estimate coalescent rate through time.[14] Thus, we used it to compare effective population size trajectories and inter-population coalescent rates for the African, European, and East Asian populations between the real and simulated data. We applied Relate to YRI, CEU, and CHB samples (108, 99, and 103 individuals, accordingly) from the high-coverage version of the 1000 Genomes project as well as to genetic data simulated under each of the three models with optimized parameters (Tables 2, S5, and S6). For real data, chromosome 1 was used and a region of the same length was simulated.

We started with 2,054 high-coverage genomes from the 1000 Genomes Project. We kept positions that (1) are bi-allelic SNPs, (2) pass the 1000 Genomes filters and have the value of the QD (quality by depth) parameter above two and (3) have a missing rate below 10%. We phased and imputed the entire dataset by using Eagle version 2.4.1.[70] Next, we ran Relate on chromosome 1 for samples coming from the three focal populations. We used the GRCh38 recombination map, 1000 Genomes strict genomic mask, and a mutation rate of 1.45×10^{-8} per bp per generation. Next, we ran the effective population size estimation module of Relate for each population individually to obtain the effective population size trajectories and for population pairs to obtain the cross-coalescence rates.

For each model, we simulated a region of the same length as chromosome 1 with uniform recombination together for 100 African, 100 European, and 100 East Asian individuals by using msprime.[29] We used the 1000 Genomes strict mask for consistency between real and simulated data in terms of the length of the available sequence. After that, the simulated data were treated as described above.

We estimated effective population size for both real and simulated data as 1/2C, where C is the inferred intra-population coalescence rate. To estimate the relative inter-population coalescence rate, we used the following formula:[13]

$$C\prime_{12} = \frac{2 \times C_{12}}{C_{11} + C_{22}},$$

where C_{11} and C_{22} are intra-population coalescence rates and C_{12} is the inter-population coalescence rate.

Comparing mutation age distribution between the simulations and real data

Relate estimates branch length in generations for the inferred trees and thus allows us to obtain tMRCA for any given mutation mapped to a specific tree branch. This value for a given mutation may differ between populations, reflecting the history of the spread of the mutation. A joined distribution of mutations' tMRCA in two populations may reflect the complex history and interactions between the two populations through time. Thus, we compared such two-dimensional tMRCA distributions obtained from the real genomes and from the genomes simulated under the demographic models tested (Neanderthal, Denisova, and Africa archaic introgression with the above three models [BNDX, MNDX, and SNDX] parameters coming from Tables 2, S5, and S6, respectively) for each of the three population pairs (CEU-YRI, CHB-YRI, and CEU-CHB). We first did a log10 transformation of the tMRCA values (Figure S4) and then did a kernel density estimation for each dataset (real, BNDX, MNDX, and SNDX) and each pair of populations to obtain a matrix of a two-dimensional distribution of allele ages. For the density estimation, we used the kde2d function from the MASS R package setting n (number of grid points) to 100. Next, we subtracted such a matrix obtained for the real data from each of the simulation matrices. Together with the distribution of this difference, we report the root-mean-square deviation between the two matrices.

$$RMSD = \sqrt{\frac{1}{N}\sum_{i=1}^{N} d_i^2},$$

where d_i is the difference between the value in the i^{th} cells of the matrix for the simulated and real distribution and N is the number of cells (100 × 100). Lower root-mean-square deviation (RMSD) value indicates less deviations between the tMRCA distribution of real and simulated data (Figure S5).

We also report the standard error of this value obtained by applying a jack-knife method by iteratively masking out a 50 Mb long region of the sequence (between 1 and 246 Mb of the GRCh38 reference sequence for chromosome 1 to avoid telomere regions with a high fraction of N bases) in a non-overlapping sliding window manner and calculating RMSD on the remaining non-masked sequence, resulting in five values.

Results

ABC-DLS

The general workflow for ABC-DLS (both for model selection and parameters estimation) includes the following steps. First, we simulated[29] multiple genetic datasets for each tested model by using demographic parameters sampled from a uniform distribution within prior ranges (Table 1). Next, we computed the SFS from these data and split the data into a training and a testing subset. We then trained the NN (implemented via TensorFlow with Keras backended[42]) on the former dataset to either select between demographic models or to estimate the demographic parameters. The resulting NN is applied to the testing dataset as well as to the observed summary statistics data (see below as well as material and methods for more details). Next, we applied ABC to estimate support for the NN prediction on the observed data by comparing the NN prediction outcome between the observed data and the testing dataset (see material and methods, Figure S2, and also our previous paper[20]). Finally, in cases when SMC is used, we essentially iterated the parameter estimation step by SMC. We kept the top five percent (equal to the tolerance level) of simulations from the testing dataset that best match the observed data. We then used the parameters of those simulations to update our prior range and sent it for next iteration until convergence was reached (Figures S3 and S4).

Before testing our primary hypothesis on real sequence data, we tested whether our approach (DLS) can reproduce the known results. The predicted parameters for real sequence data (see below for more details) are consistent with previous works from the literature[17,19,71](Table S7). We also tested our method on simulated data with known parameters under various models (model S, B, and M, see below for more information; simulation parameters coming from Tables 2, S5, and S6) and found that our approach with SMC correctly predicted the model in all the cases tested, suggesting it can find the correct model. Also, our method can infer the parameter values with high accuracy when the correct parameter values are known and coming from a mock observed SFS (Table S12).

Model selection

To test our hypothesis, we simulated three OOA models, simple model (model S), back to Africa model (model B), and mix model (model M), and all the models had introgression from Neanderthal to all OOA populations,[63] Denisova or unknown to Asia,[20,47,64] African archaic to Africa,[16,67,68,72] and European Neolithic farmers to Africa[69] (NDXF) (see material and methods for more details, Figure 1 and Table 1). We used the 1000 Genomes Project high-coverage genomes[44] (see material and methods for more details) of five Yoruba (African), five Utah residents with Northern and Western European ancestry (European), and five Han Chinese (East Asian) individuals as our real dataset. Next, we used three different methods to choose between the competing models: (1) RF that combines random forests with ABC;[43] (2) NN and ABC together (DL), which is an analogous but improved version of our previously published method, ABC-DL;[20] and (3) the method introduced here, DLS, which expands the DL method with SMC. Although all three methods identified the back to Africa model as the most probable one, the prediction certainty varied between methods (Tables 3 and 4). Both DL and DLS returned >95% probability for model B although RF gave lower support.

The DLS results were reproduced under different data filtering strategies, different datasets (Table S2), and different total length with different per block length of simulated regions (Table S4). We also tested whether our assumption of pulse migration events (three archaic introgression scenarios and recent migration of Neolithic farmers) could affect our inference. We tested different models with (1) no introgression and no farming migration (NI), (2) Neanderthal and Denisova introgression (ND), (3) Neanderthal, Denisova, and Africa Archaic introgression (NDX), and (4) Neanderthals and Denisova introgression with farming migration (NDF) using only DLS. Except for the no introgression model (Table S13), we always found the back to Africa model to be supported over simple and mix out of Africa models. When we compared all these 15 models together ([B, M, S] × [NI, ND, NDX, NDF, NDXF]) by using DLS, the back to Africa model with Neanderthal, Denisova, and African archaic introgression (BNDX) is

Table 3. Cross-validation via different approaches on simulated data

	OOA_B	OOA_M	OOA_S
RF			
OOA_B	92.85%	1.84%	5.31%
OOA_M	2.69%	84.71%	12.60%
OOA_S	6.17%	15.38%	78.44%
DL			
OOA_B	88.03%	2.54%	9.43%
OOA_M	3.43%	77.78%	18.79%
OOA_S	5.78%	15.69%	78.52%
DLS			
OOA_B	99.09%	0.00%	0.91%
OOA_M	0.00%	99.98%	0.02%
OOA_S	1.25%	0.30%	98.45%

Confusion matrix for misclassification is reported here via RF (random forest), DL (only neural network), and DLS (neural network and sequential Monte Carlo together) for random samples from the models with ABC.

supported over all other possibilities ($P(BNDX|data) = 0.86$) (Table S14). The model with Neolithic farming migration has a lower support ($P(BNDXF|data) = 0.14$). Both RF and DL were incapable of differentiating between these models as precisely as DLS (Tables S15 and S16). We further tested these two models (BNDX and BNDXF) with more precise priors by using more simulated data (see material and methods for more details) via DLS and rejected European Neolithic migration to Africa (Table S10). This result not only demonstrated the robustness of our inference for back to Africa but also independently supported other assumptions (except Neolithic migration) that were reported before, but not all of them were confirmed together.[16,20,47,63,64,68,69] In addition to the 15 models described above, we also tested the recent admixture model (model R), which adds a split and follows admixture within Africa, both happening after the separation of the OOA population on top of the simple out of Africa model. Thus, both our best back to Africa (BNDX) and recent admixture models represent the modern African population as a result of admixture of two components and the main difference is those separations happened before or after OOA population separation. Based on our results model, back to Africa better fits the data compared to recent admixture (Table S17).

Parameter estimation

After demonstrating that back to Africa (BNDX) best explains the real data, we used the three methods described above (RF, DL, and DLS) to estimate the model's parameters. The confidence intervals (CIs) returned by DLS are much narrower than those of the alternative approaches (Tables 2, S18, and S19). Hence, all the results discussed below are the ones obtained with DLS (Figure 2).

Our inference suggests that first there was a separation between the ancient African population and a population ancestral to both back to Africa and the actual out of Africa populations (basal out of Africa) around 71.3 (CI 69.4–73.3) kya. This event was followed by a split between back to Africa and OOA 57.9 (CI 56.1–59.8) kya and an admixture between ancient African and back to Africa 48.2 (CI 46.8–49.5) kya. The Neanderthal introgression to OOA happened much later, 38.3 (CI 37.6–39) kya, suggesting that this back to Africa migration cannot explain the Neanderthal ancestry found in modern African populations.[69] Our method predicted the admixture proportion from back to Africa to be as high as 91% (CI 90.28–91.57), suggesting a massive replacement of the ancient African population.

To independently validate our results, we compared effective population size trajectories and cross-coalescent rates obtained by applying Relate[32] to real data as well as to data simulated under each of the three models with the mean posterior parameters (Tables 2, S5, and S6) predicted by DLS.[14] We observe a close match between the estimates for the real data and our best model (Figure 3), which suggests our parameter estimation to be accurate. This similarity is particularly interesting given that we have not used any linkage disequilibrium (LD)-based summary statistics to optimize those parameters. On the other hand, neither the effective population size trajectory nor the cross-coalescent rate over time is informative enough to differentiate between these three models (data not shown). Specifically, the gradual separation between African and OOA populations, which was shown before with Relate and similar methods,[13,14] cannot be directly explained by the back to Africa or two out of Africa migration, as such gradual separation is also observed in our model S (Figure S6). However, the two-dimensional tMRCA distributions of mutations in population pairs (CEU-YRI and CHB-YRI) coming from Relate analysis best matches the distribution of the back to Africa model with lower RMSD value than other alternatives when comparing with real data (Figures S4 and S5).

Discussion

We here demonstrated that the ABC analysis can be substantially enhanced by using NN coupled with the SMC approach. Our methodology is suitable to test many hypotheses that can be simulated but cannot be extensively tested by other methods, especially for scenarios of admixture from ghost populations where the ancient genomes are unavailable and can accommodate any kind of summary statistics. Our model selection shows it can easily reach close to 100% accuracy in cross-validation steps with few simulations (2,000 samples), suggesting it can be used for testing much more complicated scenarios. We also found that our parameter estimation has high precision (most of the events have confidence interval width

Table 4. Model selection via different approaches on real data

	OOA_B	OOA_M	OOA_S
RF	66.00%	16.40%	17.60%
DL	100.00%	0.00%	0.00%
DLS	100.00%	0.00%	0.00%

Posterior of votes for RF (random forest) and posterior model probabilities for DL (only neural network) and DLS (neural network and sequential Monte Carlo together) are reported here via the real data.

below 5,000 years), which can directly complement with the radiocarbon dating. In this study, we used SFS as summary statistics because it is effortless to calculate and has sufficient information.[17,51] Our results might be further improved by use of some LD-based summary statistics,[67,73] but we opted out because they are computationally demanding to produce and the improvement in the result is minimal (at least for the tested scenario). As we use SFS as our choice of summary statistics, our method is not affected by local recombination rate.[52] Applying two different filtering strategies (see material and methodsfor more details) gave similar results, suggesting that our strategy is quite robust to the choice of the genomic regions to be analyzed.

In our models, we have not adopted any constant migrations between populations, although our approach can incorporate it. This is because we found out that our approach (parameter estimation via DLS) predicted non-zero migration rates when we used mock observed summary statistics data coming from a pulse migration model with no constant migrations and demographic parameters coming from mean values of Table 2 (Tables S9 and S10). This suggests that models including constant migrations may lead to equifinality as proposed by others[74] and/or that our approach is imprecise for estimating them.

Our results also comply with Y chromosome phylogeny and support back to Africa as proposed before.[10] However, our estimated time of separation between populations is much younger than what is reported for the Y chromosomes. One explanation might be that we used a slightly higher mutation rate (1.45 × 10^{-8} per bp per generation)[53] instead of a slightly slower alternative (1.25 × 10^{-8} per bp per generation).[54,55] When we used the slower mutation rate, our estimation for most of the events time increased (Table S3). Indeed, the separation time between back to Africa and OOA populations corresponds to 72.8 (CI 72.4–73.3) kya, which is close to the estimate of tMRCA between haplogroups D and E (72 kya[10]).

Although back to Africa is preferred over other alternatives in most of the cases, considering no introgression as an option (NI) supported mixed out of Africa over other models (Table S13). This result might be a side effect of the Neanderthal introgression in OOA. Under certain conditions (i.e., older separation time between Africa and OOA [T_B]), mixed out of Africa model with no introgression and simple out of Africa with Neanderthal introgression are comparable (Neanderthal population behaves like the first OOA population in this scenario). However, this model was rejected when compared with other more complex models (Table S14). This false result suggests a possible drawback of our method as different demographic histories can give similar SFS patterns, which can bias our interpretation if demographic histories were not incorporated in the model correctly[52] and also advocates for the importance of parameter estimation because it can give insight for the choice of selected model (especially if it does not match with the prior knowledge),

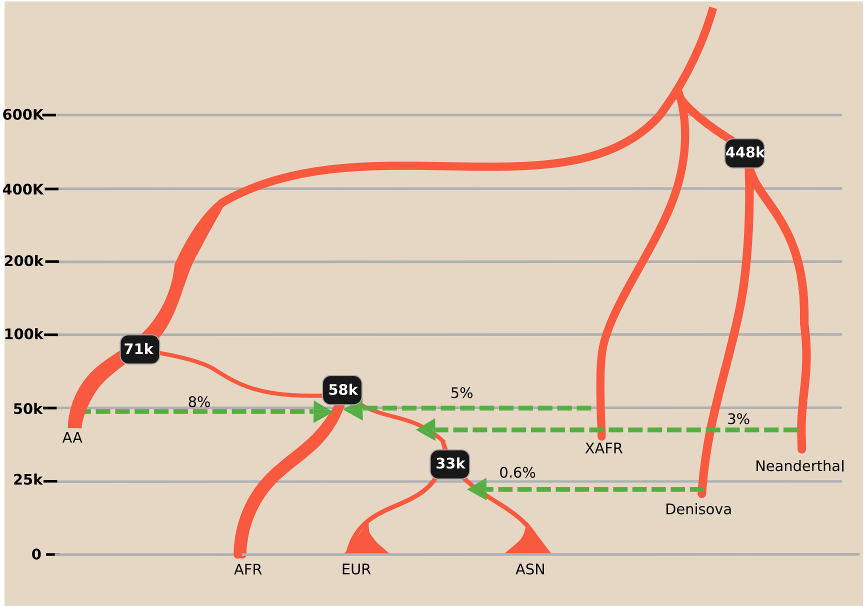


Figure 2. Schematic of inferred demography
Model B with only mean posterior. K is kilo years ago, AFR is African, AA is Ancestral African, EUR is Europeans, ASN is East Asian, and XAFR is African Archaic. Black boxes represent mean posterior of separation time between populations. Thickness of the lines represent effective population size of their respective population although they are not according to scale.

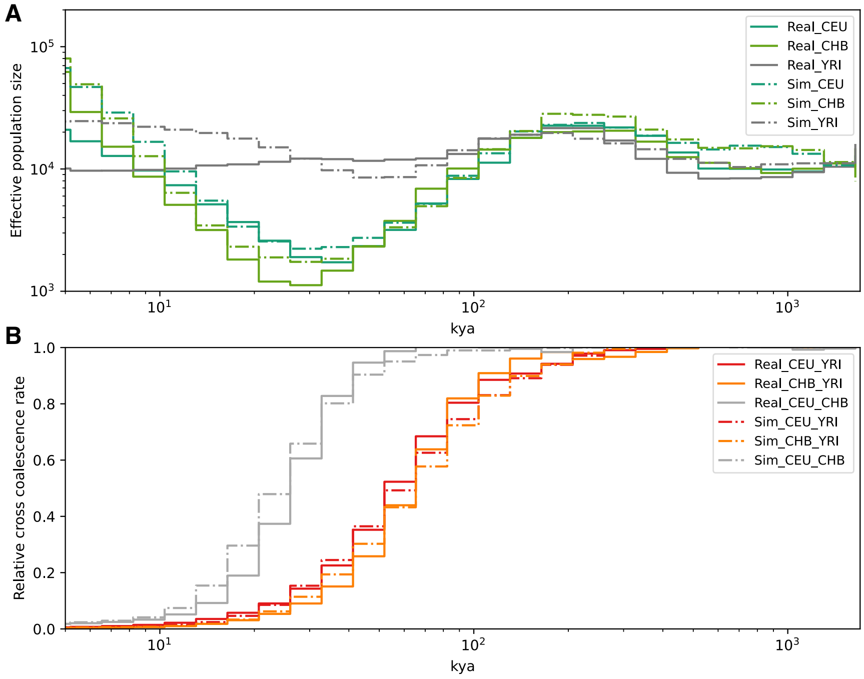


Figure 3. Effective population size trajectories and coalescence rates over time
(A and B) Here, we compare effective population size (A) and relative cross-coalescence rates (B) estimated by using Relate between the real and data simulated under the model B. (A) x axis in kya (kilo years ago) and y axis is the effective population size for corresponding populations presented in the inset. Both axes are in log scale. (B) x axis is in kya and y axis shows the relative cross coalescent rate for corresponding populations pairs presented in the insert. x axis is log scale.

which then can suggest more complicated scenarios to test.

The separation times between *Homo sapiens* and archaic populations are slightly older (877 [CI 773.1–982.3] kya for human and Neanderthal lineage divergence and 1,073.7 [CI 1,032.7–1,117.5] kya for human and African archaic) than those previously inferred[63,66–68] if we used a loose prior of 400–1,600 kya (Table S20). These deviations were not reproduced when we used simulated summary statistics generated under known parameters from Table 2. This may be specific to real sequence data and might be a side effect of some of our assumptions (for example, some unknown interactions between these populations that was not modeled here[65]) or systematic biases due to the use of European reference genome[75] or recent changes of generation time or mutation rate per generation.[76,77] These results were obtained without using the ancient genome at all in our real SFS data to reduce any chances of bias. It would be interesting to revisit these results after incorporating ancient genome directly in our analysis.

We did not find support for a model of European Neolithic migration to West Africa, which was proposed recently.[69] We have to caution that our result is only true for the Yoruba population. As sub-Saharan African populations have quite diverged ancestry, this is not representative of the whole sub-Saharan African population. Nonetheless, our back to Africa model also fails to explain Neanderthal sequence identified in Yoruba. Even if we assume this migration has happened (model BNDXF), our predicted amount of migration is as low as 2.4% (Table S21), which results in the average total length of Neanderthal sequence in Yoruba to be less than 5 Mb as opposed to 17 Mb reported by Chen et al.[69] This suggests that most of the Neanderthal regions found in Yoruba should be explained by some other migration(s) (for example from human to Neaderthal[65]). Alternatively, Neolithic farmers' contribution might be so low on West African populations that our approach with the current format (with SFS as summary statistics) fails to detect it.

We chose Yoruba, European, and East Asian population as a representative population set to test the OOA model because they are quite well studied and show relatively less recent admixture than other populations (for example, South Asian, East African, or Central Asian, etc.), which makes our model relatively simpler than other alternatives. Although back to Africa is better in explaining the real data, there might be more complicated models characterized by additional migrations and admixture that better explain the observed data. We have tested the two out of Africa model under the mixed out of Africa model, but it is not thorough enough because we have not used the populations that are assumed to have a contribution from the first OOA population.[7] It will be interesting to revisit this hypothesis with Papuan populations in the future.

Our back to Africa model can explain the reduction in effective population size of all African populations coinciding with the OOA event.[78] We would like to caution that although we are naming the model "back to Africa," the OOA population did not need to be geographically out of Africa.[21,79] Our estimates, particularly the effective population size of back to Africa (N_BC) (which is more than 10,000), and the time of Neanderthal introgression (T_NM) compared to separation back to Africa population from OOA population (T_Sep) suggest that the split might have happened within Africa itself before the actual out of Africa event. In such a case, our results can be explained by the separation of West and East African population 87.8 kya (T_B) and then later the primary separation of OOA and East African population 72.8 kya (T_Sep) (assuming mutation rate of 1.25 × 10^{-8} per bp per generation[54,55] and generation time of 29 years[56]). In this regard, our model is more akin to the Lipson et al., 2020[16] model rather than what is suggested by Cole et al., 2020.[15] If we assume the model from Lipson et al. to be true, the most parsimonious explanation would be that our back to Africa population represents the basal West African population that separated from OOA populations 72.8 kya (T_Sep). Our ancient African represents ghost modern,[16] which contributed to modern West African population around 10% through admixture around 61.9 kya from our prediction. On the other hand, if we assume true the back to Africa event happened, then most likely the OOA event took place less than 90 kya (T_B). This suggests that most of the older fossils (>100 kya) found outside of Africa[80–82] are unlikely to have contributed to OOA populations (assuming the main lineage of modern humans has not left Africa before that). Geographical location where back to Africa separated from OOA is immensely important for this hypothesis but cannot be estimated from our approach. It will be especially fascinating to test this hypothesis with ancient genomes originating at those areas from that time point when they will be available.

Data and code availability

The code generated during this study are available at https://github.com/mayukhmondal/ABC-DLS. Please contact the corresponding author for further information.

Supplemental information

Supplemental information can be found online at https://doi.org/10.1016/j.ajhg.2021.09.006.

Acknowledgments

We kindly thank Michael Zody of New York Genome Center for giving us access to high-coverage 1000 Genome data. These data were generated at the New York Genome Center with funds provided by NHGRI grant 3UM1HG008901-03S1. This research was supported by the European Union through Horizon 2020 research and innovation program under grant no. 810645, the European Regional Development Fund Project no. MOBEC008, 2014-2020.4.01.16-0030, and 2014-2020.4.01.16-0024. Data analyses were carried out in part in the High-Performance Computing Center of the University of Tartu.

Declaration of interests

The authors declare no competing interests.

Received: March 29, 2021
Accepted: September 9, 2021
Published: October 8, 2021

Web resources

1000 Genome High Coverage, https://www.internationalgenome.org/data-portal/data-collection/30x-grch38

Ancestral allele alignments, http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/phase1/analysis_results/supporting/ancestral_alignments/

HGDP, http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data_collections/HGDP/

Keras Documentation, https://keras.io/

Reference

1. Hublin, J.-J., Ben-Ncer, A., Bailey, S.E., Freidline, S.E., Neubauer, S., Skinner, M.M., Bergmann, I., Le Cabec, A., Benazzi, S., Harvati, K., and Gunz, P. (2017). New fossils from Jebel Irhoud, Morocco and the pan-African origin of Homo sapiens. Nature *546*, 289–292.
2. Schlebusch, C.M., Malmström, H., Günther, T., Sjödin, P., Coutinho, A., Edlund, H., Munters, A.R., Vicente, M., Steyn, M., Soodyall, H., et al. (2017). Southern African ancient genomes estimate modern human divergence to 350,000 to 260,000 years ago. Science *358*, 652–655.
3. Grün, R., Stringer, C., McDermott, F., Nathan, R., Porat, N., Robertson, S., Taylor, L., Mortimer, G., Eggins, S., and McCulloch, M. (2005). U-series and ESR analyses of bones and teeth relating to the human burials from Skhul. J. Hum. Evol. *49*, 316–334.
4. Mondal, M., Casals, F., Xu, T., Dall'Olio, G.M., Pybus, M., Netea, M.G., Comas, D., Laayouni, H., Li, Q., Majumder, P.P., and Bertranpetit, J. (2016). Genomic analysis of Andamanese provides insights into ancient human migration into Asia and adaptation. Nat. Genet. *48*, 1066–1070.
5. Malaspinas, A.S., Westaway, M.C., Muller, C., Sousa, V.C., Lao, O., Alves, I., Bergström, A., Athanasiadis, G., Cheng, J.Y., Crawford, J.E., et al. (2016). A genomic history of Aboriginal Australia. Nature *538*, 207–214.
6. Mallick, S., Li, H., Lipson, M., Mathieson, I., Gymrek, M., Racimo, F., Zhao, M., Chennagiri, N., Nordenfelt, S., Tandon, A., et al. (2016). The Simons Genome Diversity Project: 300 genomes from 142 diverse populations. Nature *538*, 201–206.
7. Pagani, L., Lawson, D.J., Jagoda, E., Mörseburg, A., Eriksson, A., Mitt, M., Clemente, F., Hudjashov, G., DeGiorgio, M., Saag, L., et al. (2016). Genomic analyses inform on migration events during the peopling of Eurasia. Nature *538*, 238–242.
8. Soares, P., Alshamali, F., Pereira, J.B., Fernandes, V., Silva, N.M., Afonso, C., Costa, M.D., Musilová, E., Macaulay, V., Richards, M.B., et al. (2012). The Expansion of mtDNA Haplogroup L3 within and out of Africa. Mol. Biol. Evol. *29*, 915–927.
9. Karmin, M., Saag, L., Vicente, M., Wilson Sayres, M.A., Järve, M., Talas, U.G., Rootsi, S., Ilumäe, A.M., Mägi, R., Mitt, M., et al. (2015). A recent bottleneck of Y chromosome diversity coincides with a global change in culture. Genome Res. *25*, 459–466.
10. Poznik, G.D., Xue, Y., Mendez, F.L., Willems, T.F., Massaia, A., Wilson Sayres, M.A., Ayub, Q., McCarthy, S.A., Narechania, A., Kashin, S., et al. (2016). Punctuated bursts in human male demography inferred from 1,244 worldwide Y-chromosome sequences. Nat. Genet. *48*, 593–599.
11. Mondal, M., Bergström, A., Xue, Y., Calafell, F., Laayouni, H., Casals, F., Majumder, P.P., Tyler-Smith, C., and Bertranpetit, J. (2017). Y-chromosomal sequences of diverse Indian populations and the ancestry of the Andamanese. Hum. Genet. *136*, 499–510.

12. Haber, M., Jones, A.L., Connell, B.A., Asan, Arciero, E., Yang, H., Thomas, M.G., Xue, Y., and Tyler-Smith, C. (2019). A rare deep-rooting D0 African Y-chromosomal haplogroup and its implications for the expansion of modern humans out of Africa. Genetics *212*, 1421–1428.
13. Schiffels, S., and Durbin, R. (2014). Inferring human population size and separation history from multiple genome sequences. Nat. Genet. *46*, 919–925.
14. Speidel, L., Forest, M., Shi, S., and Myers, S.R. (2019). A method for genome-wide genealogy estimation for thousands of samples. Nat. Genet. *51*, 1321–1329.
15. Cole, C.B., Zhu, S.J., Mathieson, I., Prüfer, K., and Lunter, G. (2020). Ancient Admixture into Africa from the ancestors of non-Africans. bioRxiv. https://doi.org/10.1101/2020.06.01.127555.
16. Lipson, M., Ribot, I., Mallick, S., Rohland, N., Olalde, I., Adamski, N., Broomandkhoshbacht, N., Lawson, A.M., López, S., Oppenheimer, J., et al. (2020). Ancient West African foragers in the context of African population history. Nature *577*, 665–670.
17. Gutenkunst, R.N., Hernandez, R.D., Williamson, S.H., and Bustamante, C.D. (2009). Inferring the joint demographic history of multiple populations from multidimensional SNP frequency data. PLoS Genet. *5*, e1000695.
18. Li, H., and Durbin, R. (2011). Inference of human population history from individual whole-genome sequences. Nature *475*, 493–496.
19. Gravel, S., Henn, B.M., Gutenkunst, R.N., Indap, A.R., Marth, G.T., Clark, A.G., Yu, F., Gibbs, R.A., Bustamante, C.D.; and 1000 Genomes Project (2011). Demographic history and rare allele sharing among human populations. Proc. Natl. Acad. Sci. USA *108*, 11983–11988.
20. Mondal, M., Bertranpetit, J., and Lao, O. (2019). Approximate Bayesian computation with deep learning supports a third archaic introgression in Asia and Oceania. Nat. Commun. *10*, 246.
21. Bergström, A., Stringer, C., Hajdinjak, M., Scerri, E.M.L., and Skoglund, P. (2021). Origins of modern human ancestry. Nature *590*, 229–237.
22. van de Loosdrecht, M., Bouzouggar, A., Humphrey, L., Posth, C., Barton, N., Aximu-Petri, A., Nickel, B., Nagel, S., Talbi, E.H., El Hajraoui, M.A., et al. (2018). Pleistocene North African genomes link Near Eastern and sub-Saharan African human populations. Science *360*, 548–552.
23. Ciregan, D., Meier, U., and Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. In 2012 IEEE conference on computer vision and pattern recognition (IEEE), pp. 3642–3649.
24. Graves, A., and Schmidhuber, J. (2009). Offline handwriting recognition with multidimensional recurrent neural networks. In Advances in neural information processing systems, pp. 545–552.
25. Hutter, M. (2007). On universal prediction and Bayesian confirmation. Theor. Comput. Sci. *384*, 33–48.
26. Kurtz, D.M., Esfahani, M.S., Scherer, F., Soo, J., Jin, M.C., Liu, C.L., Newman, A.M., Dührsen, U., Hüttmann, A., Casasnovas, O., et al. (2019). Dynamic risk profiling using serial tumor biomarkers for personalized outcome prediction. Cell *178*, 699–713.e19.
27. Goldberg, Y. (2016). A primer on neural network models for natural language processing. J. Artif. Intell. Res. *57*, 345–420.
28. Hudson, R.R. (2002). Generating samples under a Wright-Fisher neutral model of genetic variation. Bioinformatics *18*, 337–338.
29. Kelleher, J., Etheridge, A.M., and McVean, G. (2016). Efficient Coalescent Simulation and Genealogical Analysis for Large Sample Sizes. PLoS Comput. Biol. *12*, e1004842.
30. Jay, F., Boitard, S., and Austerlitz, F. (2019). An ABC method for whole-genome sequence data: Inferring Paleolithic and Neolithic human expansions. Mol. Biol. Evol. *36*, 1565–1579.
31. Villanea, F.A., and Schraiber, J.G. (2019). Multiple episodes of interbreeding between Neanderthal and modern humans. Nat. Ecol. Evol. *3*, 39–44.
32. Kern, A.D., and Schrider, D.R. (2018). diploS/HIC: an updated approach to classifying selective sweeps. G3 *8*, 1959–1970.
33. Torada, L., Lorenzon, L., Beddis, A., Isildak, U., Pattini, L., Mathieson, S., and Fumagalli, M. (2019). ImaGene: a convolutional neural network to quantify natural selection from genomic data. BMC Bioinformatics *20* (*Suppl 9*), 337.
34. Sanchez, T., Cury, J., Charpiat, G., and Jay, F. (2020). Deep learning for population size history inference: Design, comparison and combination with approximate Bayesian computation. Mol. Ecol. Resour. Published online July 9, 2020. https://doi.org/10.1111/1755-0998.13224.
35. Beaumont, M.A. (2019). Approximate Bayesian Computation. Annu. Rev. Stat. Appl. *6*, 379–403.
36. Liu, J.S., and Chen, R. (1998). Sequential Monte Carlo methods for dynamic systems. J. Am. Stat. Assoc. *93*, 1032–1044.
37. Mitchell, M. (1998). An introduction to genetic algorithms (MIT press).
38. Sisson, S.A., Fan, Y., and Tanaka, M.M. (2007). Sequential Monte Carlo without likelihoods. Proc. Natl. Acad. Sci. USA *104*, 1760–1765.
39. Beaumont, M.A., Cornuet, J.-M., Marin, J.-M., and Robert, C.P. (2009). Adaptive approximate Bayesian computation. Biometrika *96*, 983–990.
40. Toni, T., Welch, D., Strelkowa, N., Ipsen, A., and Stumpf, M.P. (2009). Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. J. R. Soc. Interface *6*, 187–202.
41. Collette, A. (2013). Python and HDF5: unlocking scientific data (O'Reilly Media, Inc.).
42. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). TensorFlow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016.
43. Raynal, L., Marin, J.M., Pudlo, P., Ribatet, M., Robert, C.P., and Estoup, A. (2019). ABC random forests for Bayesian parameter inference. Bioinformatics *35*, 1720–1728.
44. Byrska-Bishop, M., Evani, U.S., Zhao, X., Basile, A.O., Abel, H.J., Regier, A.A., Corvelo, A., Clarke, W.E., Musunuri, R., Nagulapalli, K., Fairley, S., et al. (2021). High coverage whole genome sequencing of the expanded 1000 Genomes Project cohort including 602 trios. bioRxiv. https://doi.org/10.1101/2021.02.06.430068.
45. Bergström, A., McCarthy, S.A., Hui, R., Almarri, M.A., Ayub, Q., Danecek, P., Chen, Y., Felkel, S., Hallast, P., Kamm, J., et al. (2020). Insights into human genetic variation and population history from 929 diverse genomes. Science *367*, eaay5012.

46. Prüfer, K., de Filippo, C., Grote, S., Mafessoni, F., Korlević, P., Hajdinjak, M., Vernot, B., Skov, L., Hsieh, P., Peyrégne, S., et al. (2017). A high-coverage Neandertal genome from Vindija Cave in Croatia. Science *358*, 655–658.
47. Jacobs, G.S., Hudjashov, G., Saag, L., Kusuma, P., Darusallam, C.C., Lawson, D.J., Mondal, M., Pagani, L., Ricaut, F.X., Stoneking, M., et al. (2019). Multiple Deeply Divergent Denisovan Ancestries in Papuans. Cell *177*, 1010–1021.e32.
48. Li, H. (2011). A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. Bioinformatics *27*, 2987–2993.
49. Danecek, P., Auton, A., Abecasis, G., Albers, C.A., Banks, E., DePristo, M.A., Handsaker, R.E., Lunter, G., Marth, G.T., Sherry, S.T., et al. (2011). The variant call format and VCFtools. Bioinformatics *27*, 2156–2158.
50. Miles, A., Murillo, R., Ralph, P., Harding, N., Pisupati, R., Rae, S., and Millar, T. (2020). cggh/scikit-allel: v1.3.2 (Zenodo). https://doi.org/10.5281/zenodo.3976233.
51. Excoffier, L., Dupanloup, I., Huerta-Sánchez, E., Sousa, V.C., and Foll, M. (2013). Robust demographic inference from genomic and SNP data. PLoS Genet. *9*, e1003905.
52. Lapierre, M., Lambert, A., and Achaz, G. (2017). Accuracy of demographic inferences from the site frequency spectrum: The case of the yoruba population. Genetics *206*, 439–449.
53. Scally, A. (2016). The mutation rate in human evolution and demographic inference. Curr. Opin. Genet. Dev. *41*, 36–43.
54. Kong, A., Frigge, M.L., Masson, G., Besenbacher, S., Sulem, P., Magnusson, G., Gudjonsson, S.A., Sigurdsson, A., Jonasdottir, A., Jonasdottir, A., et al. (2012). Rate of de novo mutations and the importance of father's age to disease risk. Nature *488*, 471–475.
55. Tian, X., Browning, B.L., and Browning, S.R. (2019). Estimating the genome-wide mutation rate with three-way identity by descent. Am. J. Hum. Genet. *105*, 883–893.
56. Tremblay, M., and Vézina, H. (2000). New estimates of intergenerational time intervals for the calculation of age and origins of mutations. Am. J. Hum. Genet. *66*, 651–658.
57. Köster, J., and Rahmann, S. (2012). Snakemake–a scalable bioinformatics workflow engine. Bioinformatics *28*, 2520–2522.
58. Csilléry, K., François, O., and Blum, M. (2012). Approximate Bayesian Computation (ABC) in R: A Vignette. Methods in Ecology and Evolution *3*, 475–479.
59. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, V., et al. (2011). Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. *12*, 2825–2830.
60. Salle, A., Villavicencio, A., and Idiart, M. (2016). Matrix Factorization using Window Sampling and Negative Sampling for Improved Word Representations. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (Association for Computational Linguistics), pp. 419–424.
61. Breiman, L. (2001). Random forests. Mach. Learn. *45*, 5–32.
62. Pudlo, P., Marin, J.M., Estoup, A., Cornuet, J.M., Gautier, M., and Robert, C.P. (2016). Reliable ABC model choice via random forests. Bioinformatics *32*, 859–866.
63. Green, R.E., Krause, J., Briggs, A.W., Maricic, T., Stenzel, U., Kircher, M., Patterson, N., Li, H., Zhai, W., Fritz, M.H., et al. (2010). A draft sequence of the Neandertal genome. Science *328*, 710–722.
64. Browning, S.R., Browning, B.L., Zhou, Y., Tucci, S., and Akey, J.M. (2018). Analysis of Human Sequence Data Reveals Two Pulses of Archaic Denisovan Admixture. Cell *173*, 53–61.e9.
65. Kuhlwilm, M., Gronau, I., Hubisz, M.J., de Filippo, C., Prado-Martinez, J., Kircher, M., Fu, Q., Burbano, H.A., Lalueza-Fox, C., de la Rasilla, M., et al. (2016). Ancient gene flow from early modern humans into Eastern Neanderthals. Nature *530*, 429–433.
66. Meyer, M., Kircher, M., Gansauge, M.T., Li, H., Racimo, F., Mallick, S., Schraiber, J.G., Jay, F., Prüfer, K., de Filippo, C., et al. (2012). A high-coverage genome sequence from an archaic Denisovan individual. Science *338*, 222–226.
67. Ragsdale, A.P., and Gravel, S. (2019). Models of archaic admixture and recent history from two-locus statistics. PLoS Genet. *15*, e1008204.
68. Lorente-Galdos, B., Lao, O., Serra-Vidal, G., Santpere, G., Kuderna, L.F.K., Arauna, L.R., Fadhlaoui-Zid, K., Pimenoff, V.N., Soodyall, H., Zalloua, P., et al. (2019). Whole-genome sequence analysis of a Pan African set of samples reveals archaic gene flow from an extinct basal population of modern humans into sub-Saharan populations. Genome Biol. *20*, 77.
69. Chen, L., Wolf, A.B., Fu, W., Li, L., and Akey, J.M. (2020). Identifying and Interpreting Apparent Neanderthal Ancestry in African Individuals. Cell *180*, 677–687.e16.
70. Loh, P.-R., Palamara, P.F., and Price, A.L. (2016). Fast and accurate long-range phasing in a UK Biobank cohort. Nat. Genet. *48*, 811–816.
71. Jouganous, J., Long, W., Ragsdale, A.P., and Gravel, S. (2017). Inferring the Joint Demographic History of Multiple Populations: Beyond the Diffusion Approximation. Genetics *206*, 1549–1567.
72. Durvasula, A., and Sankararaman, S. (2020). Recovering signals of ghost archaic introgression in African populations. Sci. Adv. *6*, eaax5097.
73. Theunert, C., Tang, K., Lachmann, M., Hu, S., and Stoneking, M. (2012). Inferring the History of Population Size Change from Genome-Wide SNP Data. Mol. Biol. Evol. *29*, 3653–3667.
74. Wall, J.D. (2017). Inferring Human Demographic Histories of Non-African Populations from Patterns of Allele Sharing. Am. J. Hum. Genet. *100*, 766–772.
75. Mondal, M., Casals, F., Majumder, P.P., and Bertranpetit, J. (2018). Reply to 'No evidence for unknown archaic ancestry in South Asia'. Nat. Genet. *50*, 1637–1639.
76. Moorjani, P., Sankararaman, S., Fu, Q., Przeworski, M., Patterson, N., and Reich, D. (2016). Molecular clock helps estimate age of ancient genomes. Proc. Natl. Acad. Sci. USA *113*, 5459–5460.
77. Besenbacher, S., Hvilsom, C., Marques-Bonet, T., Mailund, T., and Schierup, M.H. (2019). Direct estimation of mutations in great apes reconciles phylogenetic dating. Nat. Ecol. Evol. *3*, 286–292.
78. Schlebusch, C.M., Sjödin, P., Breton, G., Günther, T., Naidoo, T., Hollfelder, N., Sjöstrand, A.E., Xu, J., Gattepaille, L.M., Vicente, M., et al. (2020). Khoe-San Genomes Reveal Unique Variation and Confirm the Deepest Population Divergence in Homo sapiens. Mol. Biol. Evol. *37*, 2944–2954.
79. Pagani, L., and Crevecoeur, I. (2018). What is Africa? A human perspective. https://doi.org/10.13140/RG.2.2.16605.10728.
80. Trinkaus, E. (1993). Femoral neck-shaft angles of the Qafzeh-Skhul early modern humans, and activity levels among immature Near Eastern Middle Paleolithic hominids. J. Hum. Evol. *25*, 393–416.

81. Liu, W., Martinón-Torres, M., Cai, Y.J., Xing, S., Tong, H.W., Pei, S.W., Sier, M.J., Wu, X.H., Edwards, R.L., Cheng, H., et al. (2015). The earliest unequivocally modern humans in southern China. Nature *526*, 696–699.

82. Harvati, K., Röding, C., Bosman, A.M., Karakostis, F.A., Grün, R., Stringer, C., Karkanas, P., Thompson, N.C., Koutoulidis, V., Moulopoulos, L.A., et al. (2019). Apidima Cave fossils provide earliest evidence of Homo sapiens in Eurasia. Nature *571*, 500–504.

The American Journal of Human Genetics, Volume *108*

Supplemental information

Revisiting the out of Africa event with a deep-learning approach

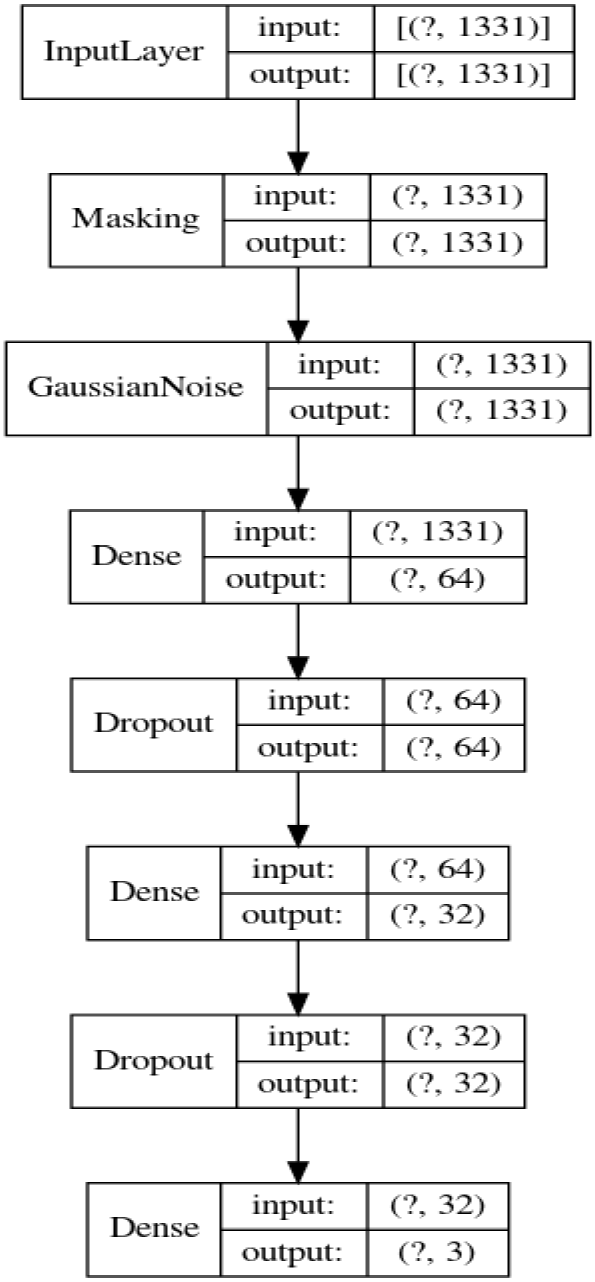
Francesco Montinaro, Vasili Pankratov, Burak Yelmen, Luca Pagani, and Mayukh Mondal

Supplementary Figures

Figure S1: Schematic for used TensorFlow model.

a) Model Selection and b) Parameter Estimation. “?” marks the number of rows which can be variable depending on the simulation and input files. The number denotes the number of elements for input and output per simulation.

a)



b)

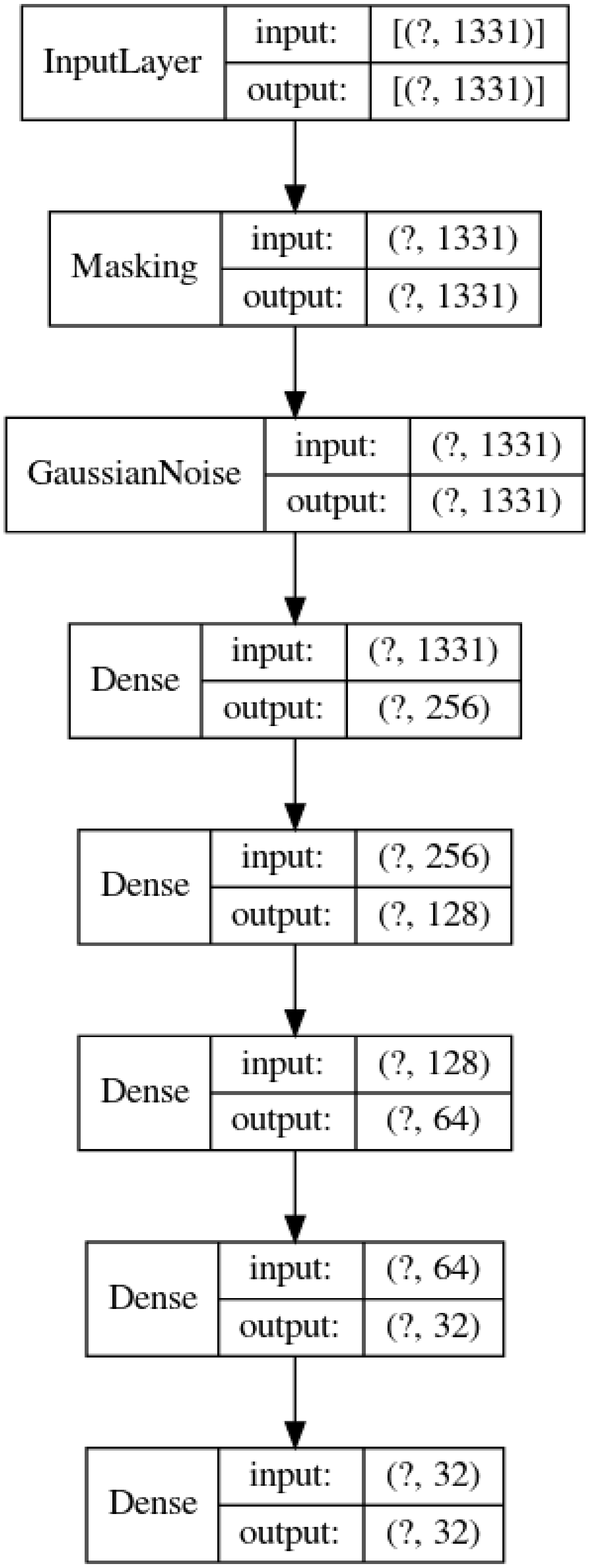
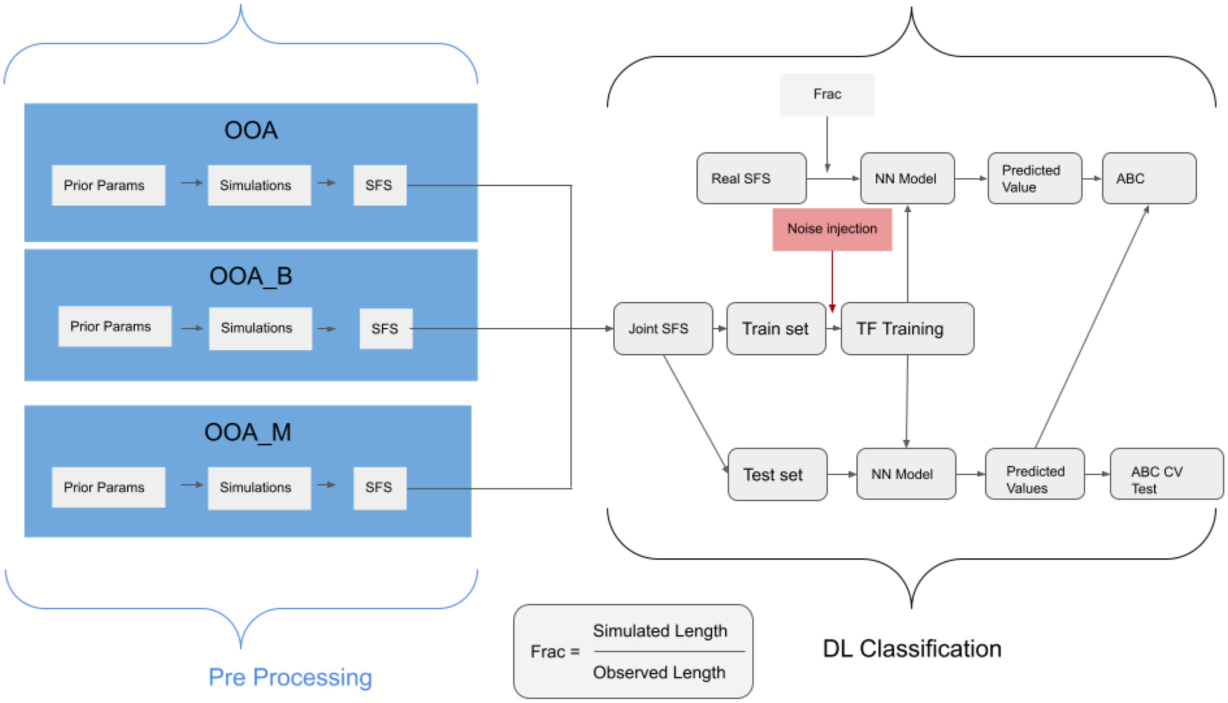

InputLayer
input:
[(?, 1331)]
output:
[(?, 1331)]
Masking
input:
(?, 1331)
output:
(?, 1331)
GaussianNoise
input:
(?, 1331)
output:
(?, 1331)
Dense
input:
(?, 1331)
output:
(?, 256)
Dense
input:
(?, 256)
output:
(?, 128)
Dense
input:
(?, 128)
output:
(?, 64)
Dense
input:
(?, 64)
output:
(?, 32)
Dense
input:
(?, 32)
output:
(?, 32)

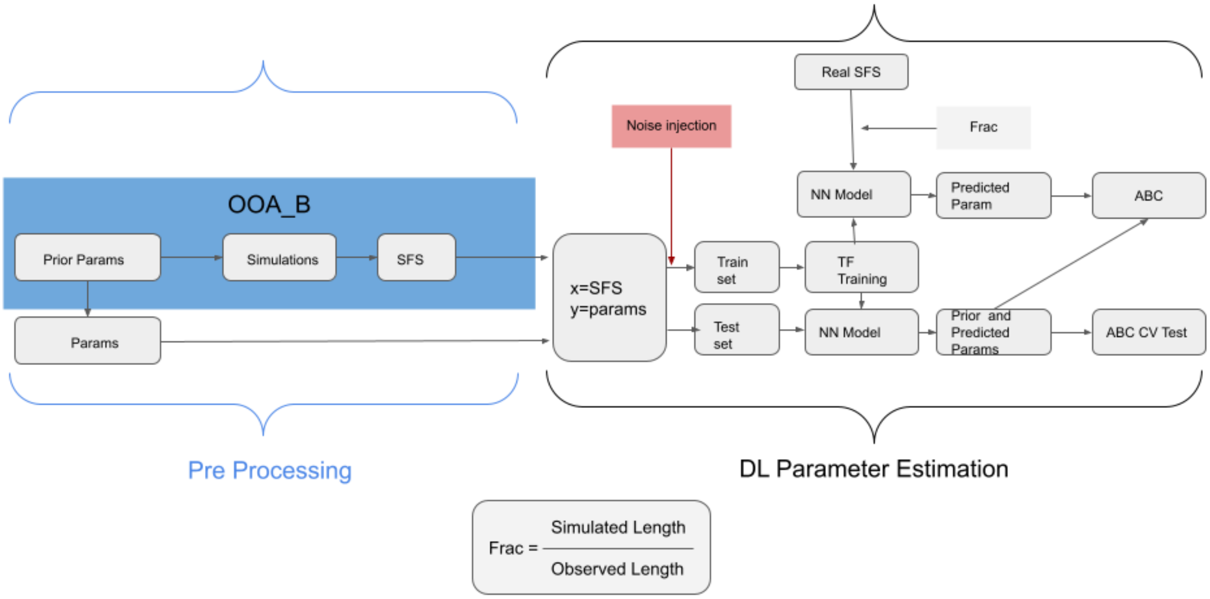
Figure S2: Flowchart of individual ABC-DLS method

a) Model Selection by DL, b) Param Prediction by DL c) SMC

a)



b)



c)

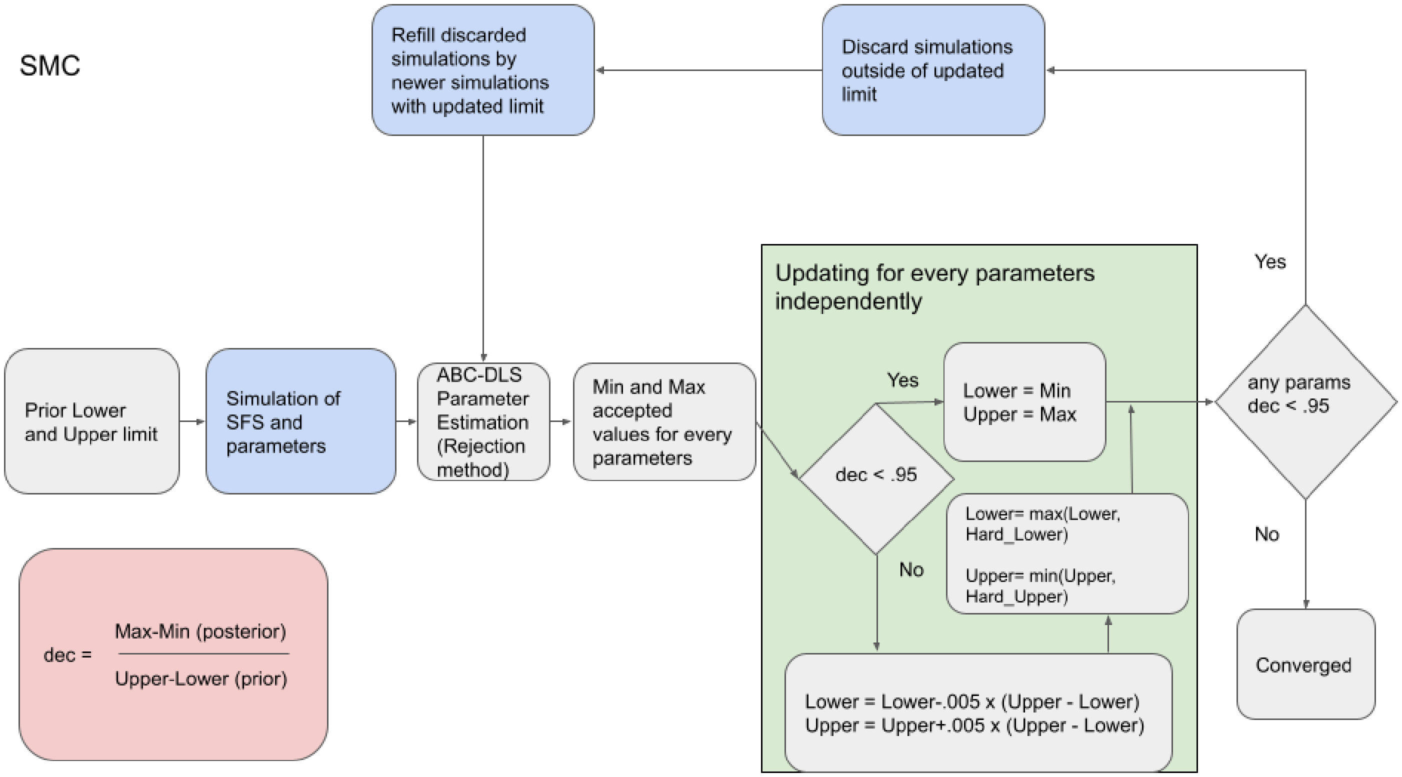

SMC
Refill discarded simulations by newer simulations with updated limit
Discard simulations outside of updated limit
Yes
Updating for every parameters independently
Prior Lower and Upper limit
Simulation of SFS and parameters
ABC-DLS Parameter Estimation (Rejection method)
Min and Max accepted values for every parameters
Yes
Lower = Min
Upper = Max
dec < .95
No
Lower= max(Lower, Hard_Lower)
Upper= min(Upper, Hard_Upper)
any params dec < .95
No
Converged
Lower = Lower-.005 x (Upper - Lower)
Upper = Upper+.005 x (Upper - Lower)
dec = Max-Min (posterior) / Upper-Lower (prior)

Figure S3: Flowchart for the ABC-DLS methods interacting together to get desired results

a) Parameter Estimation using DLS b) Model Selection using DLS

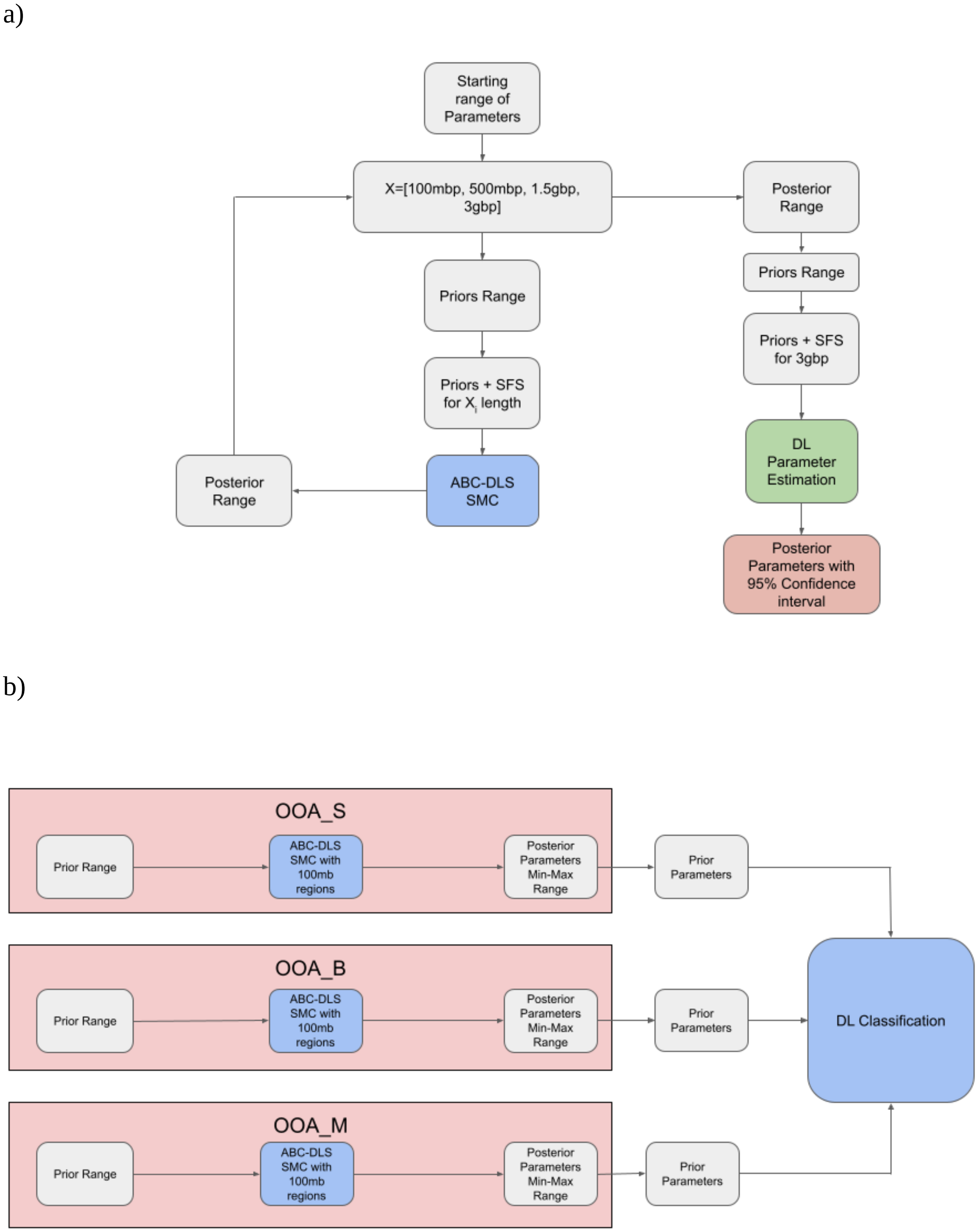

a)
Starting range of Parameters
X=[100mbp, 500mbp, 1.5gbp, 3gbp]
Priors Range
Priors + SFS for X_i length
ABC-DLS SMC
Posterior Range
Posterior Range
Priors Range
Priors + SFS for 3gbp
DL Parameter Estimation
Posterior Parameters with 95% Confidence interval
b)
OOA_S
Prior Range
ABC-DLS SMC with 100mb regions
Posterior Parameters Min-Max Range
Prior Parameters
OOA_B
Prior Range
ABC-DLS SMC with 100mb regions
Posterior Parameters Min-Max Range
Prior Parameters
OOA_M
Prior Range
ABC-DLS SMC with 100mb regions
Posterior Parameters Min-Max Range
Prior Parameters
DL Classification

Figure S4: Comparison between mutation age estimate coming from relate between the simulation and real data.

two-dimensional mutation tMRCA distributions for the real data (a-c) as well as for the three simulation scenarios BNDX. MNDX and SNDX (d-f, g-i and j-l respectively). The tMRCAs are log10 transformed. Each pixel represents a class of mutations with a given combination of age in population A (x axis) and population b (y axis) with the colour showing the number of mutations in each class.

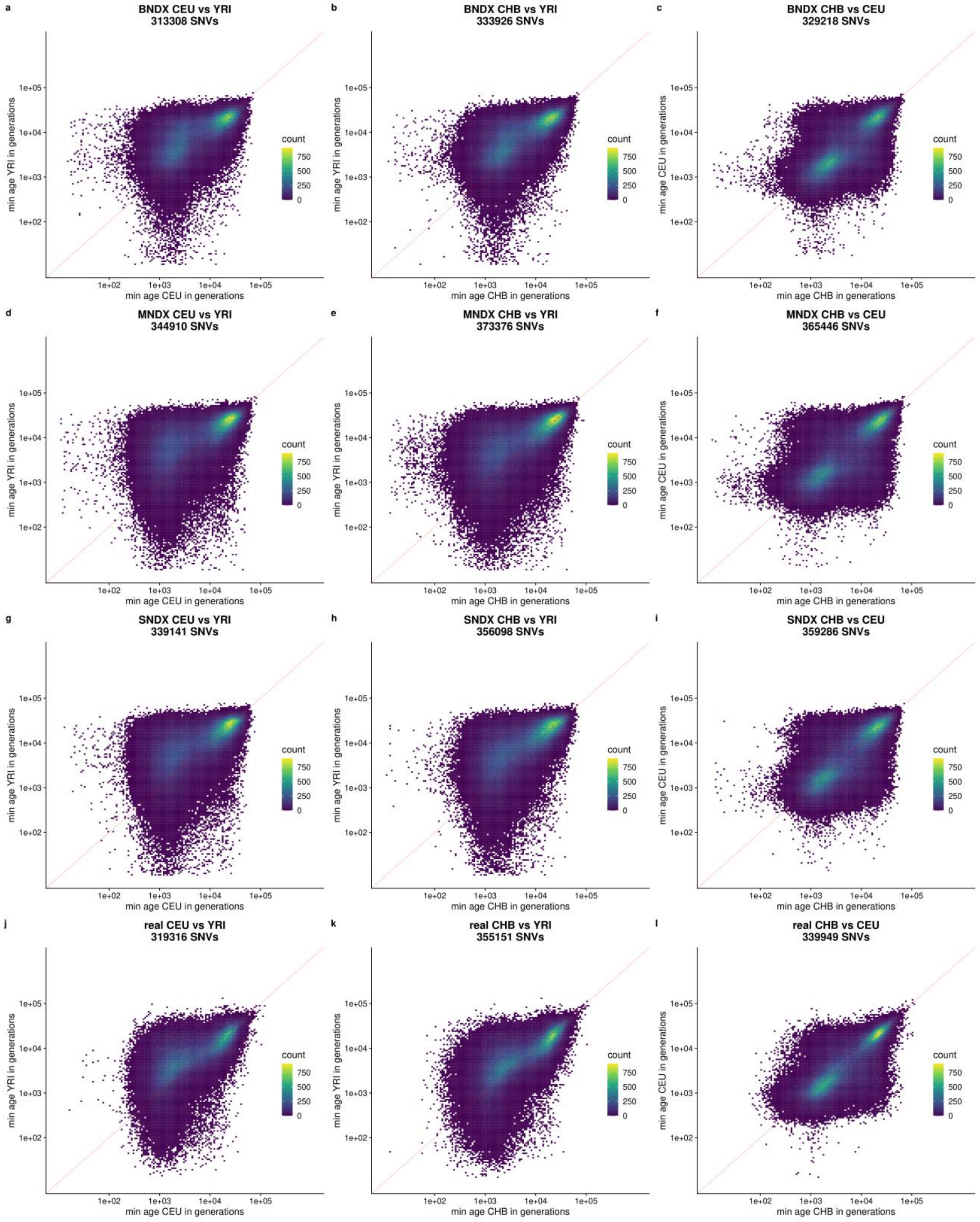

a
BNDX CEU vs YRI
313308 SNVs
b
BNDX CHB vs YRI
333926 SNVs
c
BNDX CHB vs CEU
329218 SNVs
d
MNDX CEU vs YRI
344910 SNVs
e
MNDX CHB vs YRI
373376 SNVs
f
MNDX CHB vs CEU
365446 SNVs
g
SNDX CEU vs YRI
339141 SNVs
h
SNDX CHB vs YRI
356098 SNVs
i
SNDX CHB vs CEU
359286 SNVs
j
real CEU vs YRI
319316 SNVs
k
real CHB vs YRI
355151 SNVs
l
real CHB vs CEU
339949 SNVs
min age YRI in generations
min age CEU in generations
min age CHB in generations
1e+02
1e+03
1e+04
1e+05
count
750
500
250
0

Figure S5: Comparison between mutation age estimate coming from relate between the simulation and real data.

differences between the distributions obtained for the three simulated scenarios and the real distribution from a. Kernel density estimation with a 100 x 100 grid was used after a log10 transformation of the tMRCA values. RMSD stands for “root-mean-square deviation” with the standard error calculated using jack-knife method (see Methods).

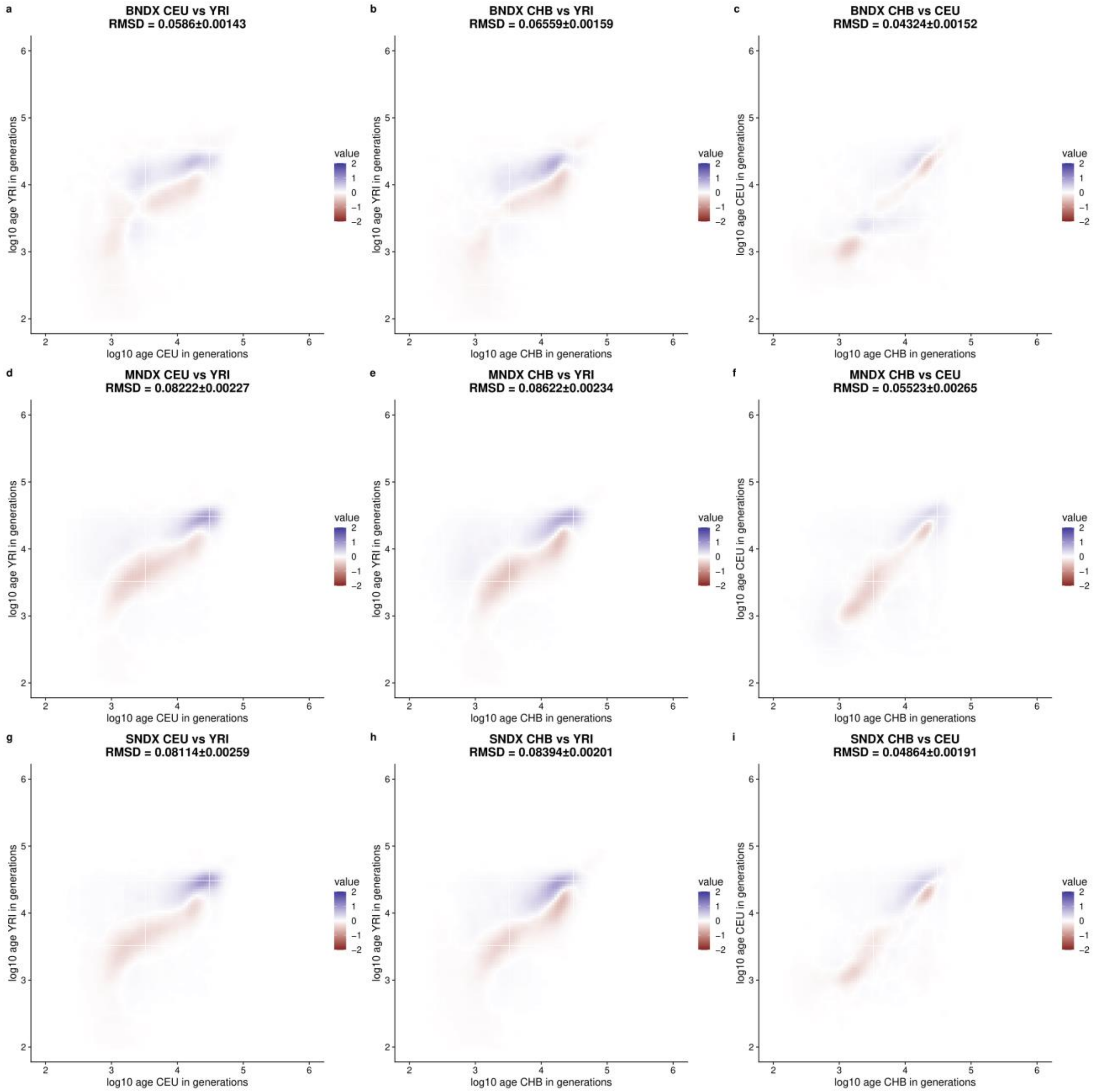
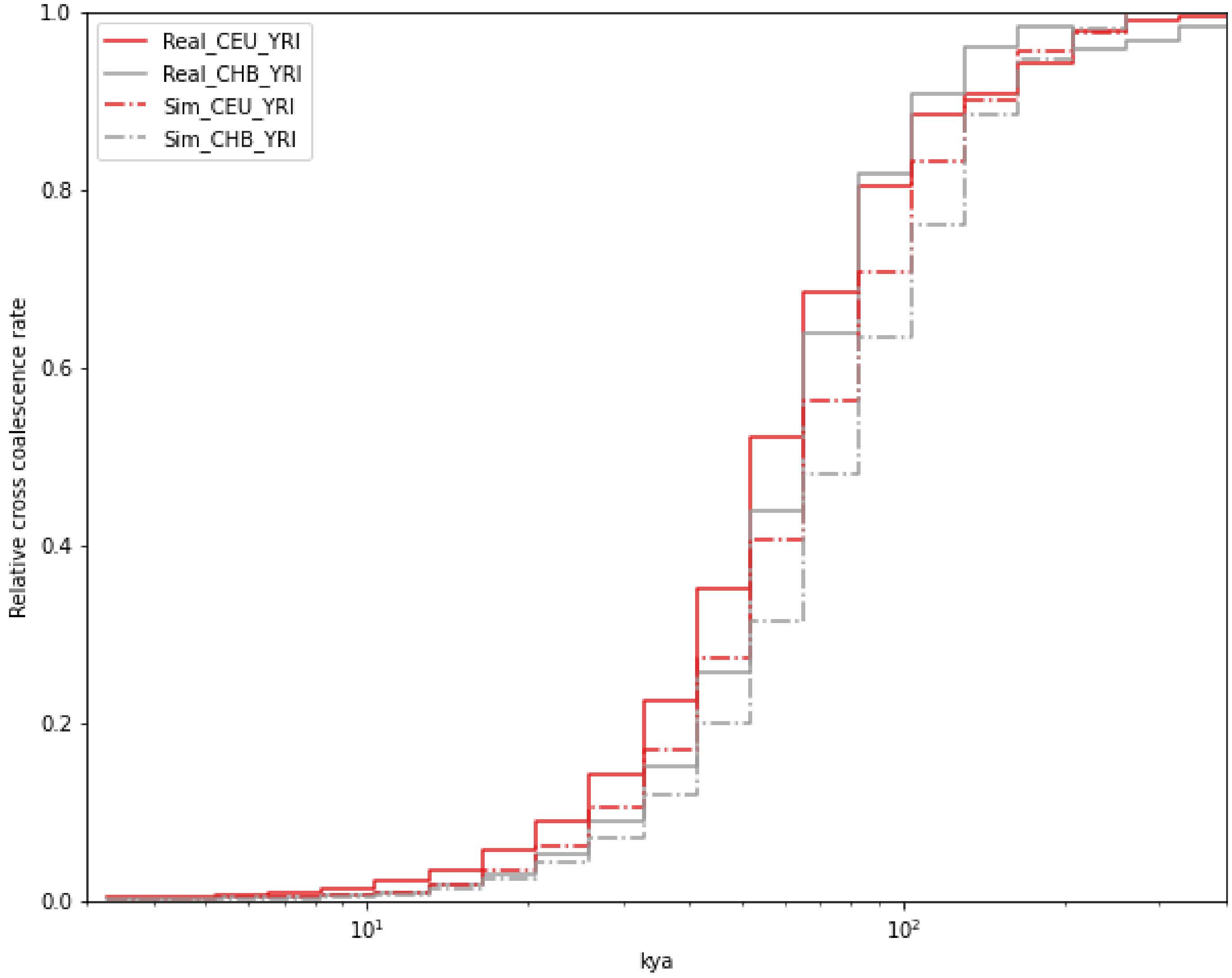


Figure S6: Cross population coalescent rate calculated using relate for Model S



Supplementary Tables

Table S1: The samples that are used to create real SFS.

Population Source	1000 Genome	HGDP
Yoruba	NA18853	HGDP00920
Yoruba	NA18856	HGDP00924
Yoruba	NA18858	HGDP00925
Yoruba	NA18861	HGDP00926
Yoruba	NA18864	HGDP00929
European	NA06984	HGDP00511
European	NA06985	HGDP00512
European	NA06986	HGDP00513
European	NA06989	HGDP00514
European	NA06994	HGDP00515
East Asian	NA18525	HGDP00774
East Asian	NA18526	HGDP00776
East Asian	NA18528	HGDP00777
East Asian	NA18530	HGDP00779
East Asian	NA18531	HGDP00780

Table S2: Cross validation and Model Selection under different strategy for real data.

Filtering Strategy	B	M	S
B	99.65%	0.00%	0.35%
M	0.02%	99.98%	0.00%
S	0.12%	0.04%	99.84%
Posterior model probabilities	100.00%	0.00%	0.00%
Dataset			
B	98.89%	0.06%	1.05%
M	0.17%	99.65%	0.18%
S	2.62%	0.20%	97.18%
Posterior model probabilities	100.00%	0.00%	0.00%

Confusion matrix for misclassification is reported here using SMC for random samples from the models. Posterior model probabilities are final posterior after using the real data.

Table S3: Posterior range for parameters of model B with slower mutation rate.

Parameters	Mean	CI	Events (kya)
N_A	17176	17165 - 17180	
N_AF	47609	46516 - 49576	
N_EU	104191	101782 - 109469	
N_AS	127664	122811 - 135653	
N_EU0	2084	2062 - 2123	
N_AS0	852	842 - 867	
N_BC	13914	13256 - 15235	
N_B	2835	2819 - 2853	
N_AF0	33722	33364 - 34038	
T_DM (ky)	21.88	21.65 - 22.18	21.9 (21.6 - 22.2)
T_EU_AS (ky)	14.4	14.19 - 14.7	36.3 (35.9 - 36.6)
T_NM(ky)	5.27	5.01 - 5.61	41.6 (41.1 - 42)
T_XM (ky)	25.46	25.39 - 25.57	61.7 (61.4 - 62.1)
T_Mix (ky)	25.61	25.46 - 25.68	61.9 (61.5 - 62.3)
T_Sep (ky)	10.94	10.72 - 11.12	72.8 (72.4 - 73.3)
T_B (ky)	14.92	14.77 - 15.14	87.8 (87.3 - 88.2)
T_AF (ky)	257.47	253.21 - 260.14	345.2 (341.7 - 348.7)
T_N_D (ky)	446.28	439.64 - 449.89	446.3 (439.6 - 449.9)
T_H_A (ky)	250.25	248.23 - 251.1	696.5 (691.2 - 701.9)
T_H_X (ky)	715.46	715.39 - 715.49	715.5 (715.4 - 715.5)
Mix (%)	0.9427	0.935 - 0.9485	
NMix (%)	0.0296	0.0288 - 0.03	
DMix (%)	0.0094	0.0089 - 0.0097	
XMix (%)	0.0379	0.0375 - 0.0381	

CI is the confidence interval of 2.5%-97.5% of respective parameters. Ky means kilo years and kya means kilo or thousand years ago from now.

Table S4: Cross validation and Model Selection with total 647 Mb region with the multiple of 25 Kb regions.

	B	**M**	**S**
B	97.41%	0.00%	2.59%
M	0.00%	100.00%	0.00%
S	0.87%	0.00%	99.13%
Posterior model probabilities	97.41%	0.00%	2.59%

Confusion matrix for misclassification is reported here using SMC for random samples from the models. Posterior model probabilities are final posterior after using the real data.

Table S5: Posterior range of model S.

Parameters	**Mean**	**CI**	**Events (kya)**
N_A	14951	14948 - 14952	
N_AF	16644	16591 - 16669	
N_EU	134965	125616 - 140849	
N_AS	123520	118968 - 131342	
N_EU0	1602	1571 - 1620	
N_AS0	678	672 - 691	
N_B	1233	1221 - 1237	
T_DM (ky)	15.4	15.2 - 15.6	15.4 (15.2 - 15.6)
T_EU_AS (ky)	15	14.9 - 15.2	30.4 (30.1 - 30.7)
T_XM (ky)	5.1	5.1 - 5.3	35.5 (35.2 - 35.8)
T_NM (ky)	7.3	7.2 - 7.4	37.7 (37.4 - 37.9)
T_B (ky)	6.4	6.3 - 6.6	44.1 (43.8 - 44.4)
T_AF (ky)	470.8	456.9 - 479.1	514.9 (503.8 - 526)
T_N_D (ky)	455.6	455.6 - 455.6	455.6 (455.6 - 455.6)
T_H_A (ky)	255.6	255.6 - 255.6	711.2 (711.2 - 711.3)
T_H_X (ky)	692.4	687.6 - 695.3	692.4 (687.6 - 695.3)
NMix (%)	2.94	2.84 – 2.99	
DMix (%)	0.85	0.8 – 0.87	
XMix (%)	6.98	6.88 – 7.04	

CI is the confidence interval of 2.5%-97.5% of respective parameters. Ky is kilo years and kya is kilo years ago from now.

Table S6: Posterior range of model M.

Parameters	Mean	CI	Events (kya)
N_A	14720	14675 - 14775	
N_AF	16967	16863 - 17068	
N_EU	50317	39111 - 59103	
N_AS	115455	90962 - 132625	
N_EU0	1552	1423 - 1689	
N_AS0	543	520 - 567	
N_MX	512	500 - 526	
N_B	4697	4136 - 4988	
N_B0	27170	11057 - 39637	
T_DM (ky)	11.9	11.1 - 12.7	11.9 (11.1 - 12.7)
T_EU_AS (ky)	12.2	11.2 - 13.1	24.1 (22.9 - 25.3)
T_NM (ky)	5.2	5 - 5.5	29.3 (28.1 - 30.6)
T_XM (ky)	12.8	12.4 - 13.1	36.8 (35.6 - 38.1)
T_Mix (ky)	5.3	5.1 - 5.6	29.4 (28.2 - 30.7)
T_Sep (ky)	7.2	7 - 7.4	44.1 (42.8 - 45.4)
T_B (ky)	13.6	10.9 - 19.9	57.7 (53 - 62.4)
T_AF (ky)	369.4	344.4 - 393.8	427.1 (401.9 - 452.3)
T_N_D (ky)	422.6	384.3 - 450.4	422.6 (384.3 - 450.4)
T_H_A (ky)	218.4	187.9 - 247.7	641 (596.5 - 685.6)
T_H_X (ky)	686.8	678.1 - 698.2	686.8 (678.1 - 698.2)
Mix (%)	92.15	91.87 – 92.62	
NMix (%)	2.59	2.46 – 2.78	
DMix (%)	0.63	0.55 – 0.68	
XMix (%)	7.1	6.97 – 7.24	

CI is the confidence interval of 2.5%-97.5% of respective parameters. Ky is kilo years and kya is kilo years ago from now.

Table S7: Cross validation and Model Selection with top two models (BNDX and BNDXF).

	BNDX	BNDXF
BNDX	88.92%	11.08%
BNDXF	13.44%	86.56%
Posterior model probabilities	97.83%	2.17%

Confusion matrix for misclassification is reported here using SMC for random samples from the models. Posterior model probabilities are final posterior after using the real data.

Table S8: Prior and Posterior range of Out Of Africa model with migrations.

Parameters	**Prior**	**Posterior**
N_A	5000 - 25000	16479 (16425 - 16515)
N_AF	10000 - 150000	22792 (22676 - 22962)
N_EU	10000 - 150000	112007 (106124 - 116503)
N_AS	10000 - 150000	125069 (118531 - 130843)
N_EU0	500 - 5000	1110 (1092 - 1127)
N_AS0	500 - 5000	588 (580 - 597)
N_B	500 - 5000	3978 (3974 - 3987)
T_EU_AS (ky)	15 - 80	40.6 (40.4 - 40.8)
T_B (ky)	5 - 320	313.9 (283 - 329.2)
T_AF (ky)	5 - 700	186.6 (182.6 - 194.5)
m_AF_B (×10^{-5})	0 - 50	19.49 (19.32 - 19.69)
m_AF_EU (×10^{-5})	0 - 50	0.66 (0.59 - 0.72)
m_AF_AS (×10^{-5})	0 - 50	0.76 (0.7 - 0.82)
m_EU_AS (×10^{-5})	0 - 50	9.31 (9.08 - 9.43)

Posterior column represents Mean and CI (Confidence Interval of 2.5%-97.5%) of respective parameters. Migration rates are denoted by m_pop1_pop2, where pop1 is the population that received the migration, and pop2 is the population from where the migration originated. Here the migration rates are bi-directional and equal.

Table S9: Posterior range of model B with migration rates with known parameters coming from a simulation from Table 2.

Parameters	Mean	2.5P	97.5P
N_A	13990	13793	14099
N_AF	15170	11158	21346
N_EU	71138	17116	136660
N_AS	78385	32652	132758
N_EU0	1932	1309	3495
N_AS0	677	561	857
N_BC	23119	5355	38949
N_B	2134	1797	2457
N_AF0	31272	23403	39594
T_DM (ky)	21.4	17.5	24.5
T_EU_AS (ky)	9.7	7.4	12.4
T_NM (ky)	5.8	5.1	7.2
T_XM (ky)	23.3	4.2	41.1
T_Mix (ky)	14.8	6.2	24.2
T_Sep (ky)	12.5	5.3	20.7
T_B (ky)	15.4	6.0	22.9
T_AF (ky)	315.2	272.9	371.3
T_N_D (ky)	429.6	396.0	448.5
T_H_A (ky)	228.5	200.2	248.7
T_H_X (ky)	589.7	481.4	695.5
Mix (%)	77.18%	64.97%	92.65%
NMix (%)	2.56%	2.12%	2.96%
DMix (%)	1.17%	0.92%	1.32%
XMix (%)	5.77%	1.02%	9.87%
m_AF_B $(\times 10^{-5})$	6.33	0.07	15.24
m_B_AF $(\times 10^{-5})$	6.51	4.47	10.05
m_AF_EU $(\times 10^{-5})$	0.57	0.03	1.38
m_EU_AF $(\times 10^{-5})$	0.47	0.05	0.89
m_AF_AS $(\times 10^{-5})$	2.27	1.31	3.00
m_AS_AF $(\times 10^{-5})$	0.25	-0.02	0.65
m_EU_AS $(\times 10^{-5})$	1.18	0.01	3.81
m_AS_EU $(\times 10^{-5})$	1.86	0.27	4.44

2.5P is 2.5 percentile and 97.5P is 97.5 percentile. Ky is kilo years. Migration rates are denoted by m_pop1_pop2, where pop1 is the population that received the migration, and pop2 is the population from where the migration originated

Table S10: Posterior range of model S with migration rates with known parameters coming from a simulation from Table 2.

Parameters	**Mean**	**2.5P**	**97.5P**
N_A	14,188	14,126	14,213
N_AF	17,013	16,889	17,109
N_EU	84,543	38,042	132,514
N_AS	85,412	51,985	132,469
N_EU0	1,779	1,442	2,236
N_AS0	661	609	735
N_B	1,620	1,446	1,705
T_DM (ky)	22.3	20.5	23.9
T_EU_AS (ky)	7.8	6.6	9.5
T_NM (ky)	5.1	5.0	5.3
T_XM (ky)	12.7	5.8	19.7
T_B (ky)	13.4	5.9	20.2
T_AF (ky)	577.0	548.6	595.3
T_N_D (ky)	439.3	419.0	449.9
T_H_A (ky)	236.3	213.1	249.3
T_H_X (ky)	587.5	471.8	692.9
NMix (%)	3%	2%	3%
DMix (%)	1%	1%	1%
XMix (%)	1%	0%	1%
m_AF_B (×10^{-5})	14.51	12.19	18.52
m_B_AF (×10^{-5})	13.31	12.64	14.38
m_AF_EU (×10^{-5})	1.38	0.80	2.16
m_EU_AF (×10^{-5})	0.07	0.00	0.23
m_AF_AS (×10^{-5})	2.38	1.78	2.78
m_AS_AF (×10^{-5})	0.12	0.00	0.30
m_EU_AS (×10^{-5})	0.86	0.04	2.44
m_AS_EU (×10^{-5})	1.38	0.00	2.91

2.5P is 2.5 percentile and 97.5P is 97.5 percentile. Ky is kilo years. Migration rates are denoted by m_pop1_pop2, where pop1 is the population that received the migration, and pop2 is the population from where the migration originated

Table S11: Relations of events with the time intervals that were used in the simulations.

Events	**Model S**	**Model B**	**Model M**
Farming Admixture	T_FM	T_FM	T_FM
Farming Separation (FS)	T_FM + T_FS	T_FM + T_FS	T_FM + T_FS
Denisova Introgression	T_DM	T_DM	T_DM
Split Europe Asia (E_A)	max (T_DM, FS) + T_EU_AS	max (T_DM, FS) + T_EU_AS	max (T_DM, FS) + T_EU_AS
Neanderthal Introgression (NI)	E_A + T_NM	E_A + T_NM	E_A +T_NM
African Introgression (AI)	E_A + T_XM	E_A + T_XM	E_A + T_XM
Admixture (Mix)	NA	E_A + T_Mix	E_A + T_Mix
Separation (Sep)	NA	Mix + T_Sep	Mix + T_Sep
Split Africa OOA (OOA)	max (NI, AI) + T_B	max (NI, AI, Sep) + T_B	max (NI, AI, Sep) + T_B
Ancestral Size change	OOA + T_AF	OOA + T_AF	OOA + T_AF
Split Neanderthal Denisova	T_N_D	T_N_D	T_N_D
Split Human Neanderthal	T_N_D + T_H_A	T_N_D + T_H_A	T_N_D + T_H_A
Split human African archaic	T_H_X	T_H_X	T_H_X

Table S12: Posterior range of model B with known parameters coming from a simulation from Table 2.

Parameters	Mean	2.5P	97.5P
N_A	14,375	14,257	14,463
N_AF	92,921	41,110	144,818
N_EU	69,243	10,029	146,385
N_AS	83,525	42,971	141,114
N_EU0	2,129	1,434	3,748
N_AS0	756	643	897
N_BC	9,341	3,436	17,813
N_B	2,377	2,142	2,619
N_AF0	32,057	23,728	39,527
T_DM	17.2	12.6	21.6
T_EU_AS	14.3	9.7	19.7
T_NM	5.4	5.0	6.3
T_XM	25.1	7.9	43.2
T_Mix	11.5	4.8	20.7
T_Sep	17.4	9.9	26.4
T_B	13.7	6.2	21.4
T_AF	234.3	188.1	303.6
T_N_D	430.1	396.8	447.5
T_H_A	229.4	200.7	249.0
T_H_X	589.4	483.3	689.6
Mix	92.19%	86.29%	94.93%
Nmix	2.79%	2.49%	2.98%
Dmix	0.79%	0.64%	0.97%
Xmix	3.94%	0.04%	9.08%

2.5P is 2.5 percentile and 97.5P is 97.5 percentile of confidence interval for respective parameters. Ky means kilo or thousand years.

Table S13: Cross validation and Model Selection with no introgression and Neolithic migration models.

	B	**M**	**S**
B	100.00%	0.00%	0.00%
M	0.00%	100.00%	0.00%
S	0.00%	0.00%	100.00%
Posterior model probabilities	0.00%	100.00%	0.00%

Confusion matrix for misclassification is reported here using SMC for random samples from the models. Posterior model probabilities are final posterior after using the real data.

Table S14: Cross validation and Model Selection with all the models together by DLS.

	BND	BNDF	BNDX	BNDXF	BNI	MND	MNDF	MNDX	MNDXF	MNI	SND	SNDF	SNDX	SNDXF	SNI
BND	58.24%	24.08%	11.02%	6.46%	0.00%	0.04%	0.00%	0.00%	0.00%	0.00%	0.16%	0.00%	0.00%	0.00%	0.00%
BNDF	22.42%	57.92%	3.81%	13.94%	0.00%	0.00%	1.02%	0.00%	0.00%	0.00%	0.00%	0.26%	0.00%	0.62%	0.00%
BNDX	10.40%	4.11%	65.13%	19.40%	0.00%	0.02%	0.00%	0.00%	0.00%	0.00%	0.94%	0.00%	0.00%	0.00%	0.00%
BNDXF	8.33%	18.65%	15.66%	56.12%	0.00%	0.04%	0.00%	0.00%	0.90%	0.00%	0.10%	0.00%	0.00%	0.20%	0.00%
BNI	0.00%	0.00%	0.00%	0.00%	100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
MND	0.02%	0.00%	0.02%	0.02%	0.00%	98.80%	0.20%	0.00%	0.00%	0.00%	0.94%	0.00%	0.00%	0.00%	0.00%
MNDF	0.04%	0.67%	0.00%	0.12%	0.00%	0.00%	92.44%	0.00%	0.00%	0.00%	0.00%	6.69%	0.00%	0.04%	0.00%
MNDX	0.00%	0.00%	0.92%	0.14%	0.00%	0.00%	0.00%	98.40%	0.02%	0.00%	0.00%	0.00%	0.52%	0.00%	0.00%
MNDXF	0.00%	0.00%	0.08%	0.26%	0.00%	0.00%	0.00%	0.00%	97.73%	0.00%	0.00%	0.00%	0.00%	1.93%	0.00%
MNI	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%	0.00%	0.00%	0.00%	0.00%	0.00%
SND	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%	0.00%	0.00%	0.00%	0.00%
SNDF	0.00%	0.16%	0.00%	0.00%	0.00%	0.00%	4.42%	0.00%	0.00%	0.00%	0.00%	95.11%	0.00%	0.32%	0.00%
SNDX	0.00%	0.00%	0.20%	0.04%	0.00%	0.00%	0.00%	0.16%	0.00%	0.00%	0.00%	0.00%	99.59%	0.00%	0.00%
SNDXF	0.00%	0.39%	0.00%	1.64%	0.00%	0.00%	0.04%	0.00%	0.41%	0.00%	0.00%	1.06%	0.00%	96.46%	0.00%
SNI	0.00%	0.00%	0.00%	0.00%	0.00%	0.04%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	99.96%
Posterior model probabilities	0.00%	0.00%	85.79%	14.21%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Confusion matrix for misclassification is reported here using DLS for random samples from the models. Posterior model probabilities are final posterior after using the real data. ([B, M, S]x [No Introgression; Neanderthal and Denisova Introgression (ND); Neanderthal, Denisova and Africa archaic introgression (NDX); Neanderthal, Denisova introgression and Farming Migration (NDF); Neanderthal, Denisova, African archaic introgression and Farming Migration (NDXF)]

Table S15: Cross validation and Model Selection with all the models together by RF method.

	BND	BNDF	BNDX	BNDXF	BNI	MND	MNDF	MNDX	MNDXF	MNI	SND	SNDF	SNDX	SNDXF	SNI
BND	37.82%	11.86%	31.89%	9.22%	1.80%	0.42%	1.06%	0.32%	0.85%	0.11%	0.95%	1.38%	0.85%	1.17%	0.32%
BNDF	14.27%	38.13%	10.42%	30.42%	1.35%	0.10%	0.73%	0.10%	0.52%	0.00%	0.10%	1.98%	0.10%	1.67%	0.10%
BNDX	33.04%	10.71%	36.86%	11.10%	0.51%	0.13%	0.77%	0.38%	0.77%	0.00%	1.15%	1.66%	1.66%	1.15%	0.13%
BNDXF	10.18%	30.99%	13.41%	39.59%	0.46%	0.00%	0.56%	0.00%	0.93%	0.00%	0.00%	2.04%	0.00%	1.76%	0.09%
BNI	2.89%	2.26%	3.25%	3.25%	80.65%	0.00%	0.45%	0.09%	0.09%	1.90%	0.36%	0.45%	0.18%	0.27%	3.89%
MND	0.72%	0.00%	0.51%	0.31%	0.00%	32.85%	10.92%	28.22%	7.93%	7.00%	5.66%	0.82%	3.60%	0.72%	0.72%
MNDF	1.94%	0.61%	1.33%	0.97%	0.48%	8.48%	32.36%	7.76%	27.88%	2.06%	1.94%	5.94%	1.82%	4.61%	1.82%
MNDX	0.00%	0.00%	0.11%	0.00%	0.00%	28.99%	7.65%	30.82%	10.24%	7.76%	5.71%	0.65%	6.14%	0.97%	0.97%
MNDXF	0.64%	1.17%	1.38%	0.96%	0.32%	3.83%	31.42%	5.64%	34.19%	2.02%	1.81%	7.56%	1.70%	5.96%	1.38%
MNI	0.15%	0.15%	0.59%	0.15%	1.03%	11.34%	5.67%	11.49%	6.11%	47.94%	1.91%	0.44%	1.03%	0.52%	11.49%
SND	0.63%	0.27%	1.17%	0.00%	0.00%	6.12%	0.99%	6.30%	1.26%	0.36%	34.92%	6.39%	34.29%	4.86%	2.43%
SNDF	2.79%	3.52%	3.40%	2.55%	0.12%	0.61%	5.95%	0.36%	6.31%	0.00%	1.09%	35.92%	1.58%	35.56%	0.24%
SNDX	0.38%	0.00%	0.47%	0.09%	0.00%	5.00%	1.13%	5.66%	1.42%	0.66%	32.92%	6.04%	36.98%	7.45%	1.79%
SNDXF	3.37%	2.69%	3.94%	2.31%	0.67%	0.38%	5.58%	0.48%	6.73%	0.38%	1.54%	33.17%	1.54%	37.12%	0.10%
SNI	0.47%	0.00%	0.37%	0.19%	3.74%	1.50%	1.87%	1.96%	1.22%	12.72%	4.49%	1.12%	3.46%	1.22%	65.67%
Posterior of votes	19.00%	27.15%	15.20%	15.80%	2.35%	1.45%	7.35%	0.85%	3.05%	0.65%	1.00%	1.75%	0.70%	3.45%	0.25%

Confusion matrix for misclassification is reported here using DL for random samples from the models. Posterior model probabilities are final posterior after using the real data. ([B, M, S]x [No Introgression; Neanderthal and Denisova Introgression (ND); Neanderthal, Denisova and Africa archaic introgression (NDX); Neanderthal, Denisova introgression and Farming Migration (NDF); Neanderthal, Denisova, African archaic introgression and Farming Migration (NDXF)]

Table S16: Cross validation and Model Selection with all the models together by DL method.

	BND	BNDF	BNDX	BNDXF	BNI	MND	MNDF	MNDX	MNDXF	MNI	SND	SNDF	SNDX	SNDXF	SNI
BND	37.78%	14.20%	31.12%	10.84%	0.10%	0.10%	0.42%	0.12%	0.38%	0.26%	1.47%	0.79%	1.09%	0.73%	0.60%
BNDF	13.64%	38.78%	11.51%	28.17%	0.00%	0.02%	0.77%	0.02%	0.54%	0.00%	0.28%	3.54%	0.18%	2.56%	0.00%
BNDX	31.49%	13.12%	35.11%	14.54%	0.26%	0.04%	0.70%	0.10%	0.78%	0.00%	1.06%	1.10%	0.65%	1.04%	0.00%
BNDXF	11.05%	30.56%	14.23%	39.00%	0.02%	0.00%	0.53%	0.00%	0.76%	0.00%	0.06%	2.09%	0.00%	1.71%	0.00%
BNI	1.52%	1.45%	1.30%	1.03%	91.02%	0.02%	0.21%	0.00%	0.16%	0.24%	0.00%	0.12%	0.00%	0.24%	2.70%
MND	0.20%	0.02%	0.12%	0.02%	0.00%	34.22%	7.54%	30.92%	5.94%	12.21%	4.65%	0.23%	2.51%	0.27%	1.13%
MNDF	0.39%	1.08%	0.24%	0.60%	0.56%	4.74%	39.77%	3.81%	33.00%	1.14%	1.23%	7.47%	0.43%	5.11%	0.44%
MNDX	0.12%	0.00%	0.04%	0.00%	0.00%	32.55%	4.98%	34.29%	5.57%	8.29%	5.95%	0.29%	5.62%	0.29%	1.99%
MNDXF	0.20%	0.39%	0.45%	0.08%	0.00%	4.65%	31.90%	4.17%	34.11%	1.85%	2.71%	8.93%	2.77%	7.56%	0.24%
MNI	0.04%	0.06%	0.06%	0.00%	0.58%	13.72%	1.50%	11.69%	1.19%	58.40%	1.67%	0.12%	1.36%	0.02%	9.61%
SND	1.09%	0.12%	1.01%	0.14%	0.02%	4.61%	2.31%	4.80%	2.18%	1.42%	39.06%	2.41%	35.80%	2.55%	2.48%
SNDF	1.53%	2.11%	1.85%	1.86%	0.00%	0.23%	6.16%	0.22%	6.48%	0.00%	2.98%	37.98%	2.29%	36.28%	0.00%
SNDX	0.30%	0.00%	0.20%	0.14%	0.00%	2.86%	1.65%	3.84%	1.59%	0.71%	38.02%	1.97%	43.32%	2.73%	2.67%
SNDXF	0.49%	1.27%	1.33%	2.02%	0.00%	0.04%	5.18%	0.10%	8.38%	0.00%	3.23%	36.20%	3.73%	38.04%	0.00%
SNI	0.06%	0.00%	0.00%	0.00%	3.48%	0.86%	0.27%	0.86%	0.27%	11.63%	1.05%	0.12%	1.39%	0.18%	79.83%
Posterior model probabilities	44.19%	15.55%	34.35%	5.91%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Confusion matrix for misclassification is reported here using DL for random samples from the models. Posterior model probabilities are final posterior after using the real data. ([B, M, S]x [No Introgression; Neanderthal and Denisova Introgression (ND); Neanderthal, Denisova and Africa archaic introgression (NDX); Neanderthal, Denisova introgression and Farming Migration (NDF); Neanderthal, Denisova, African archaic introgression and Farming Migration (NDXF)]

Table S17: Cross validation and Model Selection with model SNDX and R.

	BNDX	R
BNDX	99.33%	0.67%
R	0.06%	99.94%
Posterior model probabilities	100.00%	0.00%

Confusion matrix for misclassification is reported here using SMC for random samples from the models. Posterior model probabilities are final posterior after using the real data.

Table S18: Posterior range of model B by DL.

Parameters	Mean	2.5P	97.5P
N_A	16,846	14,956	23,610
N_AF	75,043	28,330	145,659
N_EU	65,155	25,858	139,288
N_AS	89,766	35,529	147,301
N_EU0	2,132	1,286	3,416
N_AS0	1,181	829	1,809
N_BC	22,215	7,162	38,199
N_B	3,060	2,557	3,371
N_AF0	19,484	7,372	37,748
T_DM (ky)	16.0	10.1	24.4
T_EU_AS (ky)	19.8	8.8	29.6
T_NM(ky)	6.9	4.9	10.2
T_XM (ky)	28.0	6.9	63.4
T_Mix (ky)	11.3	4.8	19.5
T_Sep (ky)	14.4	6.3	24.5
T_B (ky)	16.9	5.5	36.2
T_AF (ky)	194.8	12.4	624.5
T_N_D (ky)	403.5	331.1	446.2
T_H_A (ky)	213.1	132.9	251.6
T_H_X (ky)	568.2	450.6	699.5
Mix (%)	48.05%	19.19%	61.19%
NMix (%)	2.72%	2.22%	3.03%
DMix (%)	0.34%	0.08%	0.58%
XMix (%)	1.65%	0.01%	7.37%

2.5P is 2.5 percentile and 97.5P is 97.5 percentile of confidence interval for respective parameters. Ky means kilo or thousand years.

Table S19: Posterior range of model B using RF.

Parameters	Mean	2.5P	97.5P
N_A	16,233	12,072	22,045
N_AF	43,664	15,126	136,542
N_EU	75,454	16,242	143,463
N_AS	83,959	20,726	147,852
N_EU0	3,154	1,469	4,892
N_AS0	1,156	571	2,511
N_BC	20,421	1,816	38,900
N_B	4,049	2,580	4,944
N_AF0	22,495	7,938	38,551
T_DM (ky)	21.4	10.7	38.5
T_EU_AS (ky)	15.2	5.5	28.9
T_NM(ky)	15.2	5.4	34.5
T_XM (ky)	44.3	5.9	110.9
T_Mix (ky)	23.8	5.8	47.5
T_Sep (ky)	23.1	5.4	47.4
T_B (ky)	32.8	5.6	99.8
T_AF (ky)	347.1	17.6	684.8
T_N_D (ky)	394.3	332.0	447.8
T_H_A (ky)	188.0	121.6	247.9
T_H_X (ky)	571.5	454.8	695.4
Mix (%)	61.45%	16.33%	93.80%
NMix (%)	2.33%	1.14%	2.99%
DMix (%)	0.97%	0.08%	1.89%
XMix (%)	5.49%	0.32%	9.80%

CI is the confidence interval of 2.5%-97.5% of respective parameters. Ky means kilo or thousand years.

Table S20: Posterior range for parameters of model B with bigger priors for archaic separation.

Parameters	Mean	CI	Events (kya)
N_A	14214	14200 - 14220	
N_AF	38815	36275 - 43345	
N_EU	44547	39591 - 51198	
N_AS	111147	101701 - 126733	
N_EU0	2459	2296 - 2674	
N_AS0	821	799 - 851	
N_BC	16451	12632 - 21066	
N_B	2296	2242 - 2347	
N_AF0	35353	34396 - 35875	
T_DM (ky)	13.6	12.8 - 14.5	13.6 (12.8 - 14.5)
T_EU_AS (ky)	20.6	19.8 - 21.7	34.3 (33 - 35.5)
T_NM (ky)	7.9	7.8 - 8.2	42.2 (40.9 - 43.5)
T_XM (ky)	18.1	17.6 - 18.6	52.4 (51.1 - 53.8)
T_Mix (ky)	18	17.5 - 18.5	52.3 (50.9 - 53.6)
T_Sep (ky)	10.3	9.7 - 11	62.7 (61.2 - 64.2)
T_B (ky)	14.5	14 - 14.8	77.2 (75.7 - 78.8)
T_AF (ky)	215.4	214.5 - 215.8	292.7 (291 - 294.4)
T_N_D (ky)	504.1	435.6 - 584.4	504.1 (435.6 - 584.4)
T_H_A (ky)	373.6	309.5 - 456.6	877.7 (773.1 - 982.3)
T_H_X (ky)	1073.7	1032.7 - 1117.5	1073.7 (1032.7 - 1117.5)
Mix (%)	94.4	93.26 – 94.99	
NMix (%)	2.48	2.34 – 2.71	
DMix (%)	0.45	0.4 – 0.48	
XMix (%)	4.48	4.24 – 4.68	

CI is the confidence interval of 2.5%-97.5% of respective parameters. Ky means kilo years and kya means kilo or thousand years ago from now.

Table S21: Posterior range for parameters of model B with Farming migration (BNDXF).

Parameters	**Mean**	**CI**	Events (kya)
N_A	14,570	14545 - 14580	
N_AF	22,592	22288 - 23440	
N_EU	91,774	83804 - 99522	
N_AS	111,255	104593 - 119038	
N_EU0	1,811	1782 - 1863	
N_AS0	771	758 - 788	
N_BC	17,880	13522 - 19990	
N_B	2,176	2165 - 2198	
N_AF0	30,920	30380 - 31153	
T_FM (ky)	3.4	2 - 4.9	3.4 (2 - 4.9)
T_FS (ky)	4.6	0.4 - 9.7	8 (3.1 - 12.9)
T_DM (ky)	15.0	14.6 - 15.4	15 (14.6 - 15.4)
T_EU_AS (ky)	17.2	16.8 - 17.7	32.2 (31.6 - 32.7)
T_NM (ky)	5.4	5.2 - 5.7	37.5 (36.9 - 38.2)
T_XM (ky)	17.1	16.7 - 17.3	49.3 (48.6 - 49.9)
T_Mix (ky)	16.9	16.6 - 17.5	49.1 (48.4 - 49.9)
T_Sep (ky)	10.0	9.7 - 10.6	59.2 (58.4 - 60)
T_B (ky)	12.5	12 - 13	71.7 (70.7 - 72.6)
T_AF (ky)	215.6	213.3 - 216.7	287.3 (285.4 - 289.2)
T_N_D (ky)	449.8	444.3 - 452.3	449.8 (444.3 - 452.3)
T_H_A (ky)	254.4	253.7 - 254.6	704.2 (700.1 - 708.2)
T_H_X (ky)	693.9	683.4 - 698.5	693.9 (683.4 - 698.5)
Mix	93.11	92.1 – 93.63	
NMix	2.97	2.9 – 3	
DMix	0.64	0.62 – 0.69	
XMix	5.55	5.01 – 5.99	
FMix	2.42	2.19 – 2.53	

CI is the confidence interval of 2.5%-97.5% of respective parameters. Ky means kilo years and kya means kilo or thousand years ago from now.

Supplementary Methods

To make our results easier to reproduce, we are giving three sample codes which were used to produce three demographic models using msprime. These codes are also available in https://github.com/mayukhmondal/ABC-DLS/blob/master/src/SFS/Demography.py.

Model S:

```
def SNDX(params, inds, length=1e6, mutation_rate=1.45e-8,
recombination_rate=1e-8, replicates=300):
    """
    This is the simple out of Africa model with Neanderthal to OOA
population, Denisova or        Unknown to East Asia and
        African archaic to African populations.
    :param params: all the parameters necessary for this model in a list or
array format
        N_A: The ancestral effective population size
        N_AF: Modern effective population size of Africa population
        N_EU: Modern effective population size of European population
        N_AS: Modern effective population size of East Asian population
        N_EU0: Effective population size of European population before
exponential growth.
        N_AS0: Effective population size of East Asian population before
exponential growth.
        N_B: Effective population size of Out of Africa (OOA) populations
        T_DIntro: Time interval for introgression in East Asian from Denisova
or Unknown from now. in kilo year ago
            (kya)
        T_EU_AS: Time interval for separation of European and East Asian from
T_DIntro. in kya
        T_NIntro: Time interval for introgression in OOA from Neanderthal
from T_EU_AS. in kya
        T_XIntro: Time interval for introgression in African population from
African archaic from T_EU_AS. in kya
        T_B: Time interval for separation between Africa and OOA populations
from maximum between T_NIntro and T_XIntro.
            in kya
        T_AF: Time interval for decrease of effective population size of
African population to ancestral effective
            population size from T_B. in kya
        T_N_D: Time interval for separation between Neanderthal and Denisova
or Unknwon from now. in kya
        T_H_A: Time interval for separation between Neanderthal and modern
humans from T_N_D. in kya
        T_H_X: Time interval for separation between African archaic and
modern humans from now. in kya
        nintro: the fraction of introgression happened to OOA populations.
        dintro: the fraction of introgression happened to East Asians
        xintro: the fraction of introgression happened to African populations
    :param inds: the number of haplotypes per populations for example
(10,10,10)
```

```
    :param length: the length of chromosome that has to be simulated. default
is 1mb region
    :param mutation_rate: the amount of mutation rate. default is 1.45x10^-8
per generation per base
    :param recombination_rate: the amount of recombination rate. default is
10^-8 per generation per base
    :param replicates: the number of replicated of length chromosome. default
is 300
    :return: will return the msprime simulations. which then can be used to
extract SFS
    """
    (N_A, N_AF, N_EU, N_AS, N_EU0, N_AS0, N_B, T_DIntro, T_EU_AS, T_NIntro,
T_XIntro, T_B,
     T_AF, T_N_D, T_H_A, T_H_X, nintro, dintro, xintro) = params
    (n1, n2, n3) = inds
    T_DIntro, T_EU_AS, T_NIntro, T_XIntro, T_B, T_AF, T_N_D, T_H_A, T_H_X =
numpy.array(
        [T_DIntro, T_EU_AS, T_NIntro, T_XIntro, T_B, T_AF, T_N_D, T_H_A,
T_H_X]) * (1e3 / 29.0)
    AFR, EUR, ASN, NEA, DEN, XAF = 0, 1, 2, 3, 4, 5
    events = {}
    r_EU = (math.log(N_EU / N_EU0) / (T_DIntro + T_EU_AS))
    r_AS = (math.log(N_AS / N_AS0) / (T_DIntro + T_EU_AS))
    population_configurations = [
        msprime.PopulationConfiguration(
            sample_size=n1, initial_size=N_AF),
        msprime.PopulationConfiguration(
            sample_size=n2, initial_size=N_EU, growth_rate=r_EU),
        msprime.PopulationConfiguration(
            sample_size=n3, initial_size=N_AS, growth_rate=r_AS),
        msprime.PopulationConfiguration(
            sample_size=0, initial_size=N_A),
        msprime.PopulationConfiguration(
            sample_size=0, initial_size=N_A),
        msprime.PopulationConfiguration(
            sample_size=0, initial_size=N_A)]
    # Denisova or unknown admixture
    events['deni_intro_asn'] = T_DIntro
    deni_intro_asn = [msprime.MassMigration(
        time=events['deni_intro_asn'], source=ASN, destination=DEN,
proportion=dintro)]
    # CEU and CHB merge into B with rate changes at T_EU_AS
    events['split_eu_as'] = events['deni_intro_asn'] + T_EU_AS
    split_eu_as = [
        msprime.MassMigration(
            time=events['split_eu_as'], source=ASN, destination=EUR,
proportion=1.0),
        msprime.PopulationParametersChange(
            time=events['split_eu_as'], initial_size=N_B, growth_rate=0,
population_id=EUR)]
    # introgression Nean
    events['nean_intro_eur'] = T_NIntro + events['split_eu_as']
    nean_intro_eur = [msprime.MassMigration(
        time=events['nean_intro_eur'], source=EUR, destination=NEA,
proportion=nintro)]
    # introgression XAFR
    events['xafr_intro_afr'] = T_XIntro + events['split_eu_as']
    xafr_intro_afr = [msprime.MassMigration(
        time=events['xafr_intro_afr'], source=AFR, destination=XAF,
proportion=xintro)]
    # Population B merges into YRI at T_B
    events['split_afr_ooa'] = max(events['nean_intro_eur'],
events['xafr_intro_afr']) + T_B
    split_afr_ooa = [
        msprime.MassMigration(
            time=events['split_afr_ooa'], source=EUR, destination=AFR,
proportion=1.0)]
    events['ancestral_size'] = events['split_afr_ooa'] + T_AF
    # Size changes to N_A at T_AF
    ancestral_size = [msprime.PopulationParametersChange(
        time=events['ancestral_size'], initial_size=N_A, population_id=AFR)]
    # Denisova or Unknwon merging with Neanderthal
    events['neanderthal_denisova'] = T_N_D
    neanderthal_denisova = [msprime.MassMigration(
        time=events['neanderthal_denisova'], source=DEN, destination=NEA,
proportion=1.0),
        msprime.PopulationParametersChange(
            time=events['neanderthal_denisova'], initial_size=N_A,
population_id=NEA)]
    # Neanderthal merging with humans
    events['human_neanderthal'] = T_N_D + T_H_A
    human_neanderthal = [msprime.MassMigration(
        time=events['human_neanderthal'], source=NEA, destination=AFR,
proportion=1.0)]
    # XAFR merging with humans
    events['human_xafr'] = T_H_X
    human_xafr = [msprime.MassMigration(
        time=events['human_xafr'], source=XAF, destination=AFR,
proportion=1.0)]
    demographic_events = []
    for event in collections.OrderedDict(sorted(events.items(), key=lambda x:
x[1])).keys():
        demographic_events = demographic_events + eval(event)
    geno = msprime.simulate(
        population_configurations=population_configurations,
        demographic_events=demographic_events, length=length,
mutation_rate=mutation_rate,
        num_replicates=replicates,
        recombination_rate=recombination_rate)
    return geno
```

Model B:

```
def BNDX(params, inds, length=1e6, mutation_rate=1.45e-8,
recombination_rate=1e-8, replicates=300):
    """
    This is the back to Africa model with Neanderthal to OOA population,
Denisova or Unknown to East Asia and
        African archaic to African populations.
    :param params: all the parameters necessary for this model in a list or
array format
        N_A: The ancestral effective population size
        N_AF1: Modern effective population size of Africa population
        N_EU: Modern effective population size of European population
        N_AS: Modern effective population size of East Asian population
        N_EU0: Effective population size of European population before
exponential growth.
        N_AS0: Effective population size of East Asian population before
exponential growth.
        N_BC: Effective population size of Back to Africa migrated
population.
        N_B: Effective population size of Out of Africa (OOA) populations
        N_AF2: Effective population size of African populations before Back
to Africa migration
        T_DIntro: Time interval for introgression in East Asian from Denisova
or Unknown from now. in kilo year ago
            (kya)
        T_EU_AS: Time interval for separation of European and East Asian from
T_DIntro. in kya
        T_NIntro: Time interval for introgression in OOA from Neanderthal
from T_EU_AS. in kya
        T_XIntro: Time interval for introgression in African population from
African archaic from T_EU_AS. in kya
        T_Mix: Time interval for mixing with Back to Africa population from
T_EU_AS. in kya
        T_Sep: Time interval for separation of Back to Africa population from
OOA from T_Mix. in kya
        T_B: Time interval for separation between Africa and OOA populations
from maximum between T_NIntro, T_XIntro and
            T_Sep. in kya
        T_AF: Time interval for decrease of effective population size of
African population to ancestral effective
            population size from T_B. in kya
        T_N_D: Time interval for separation between Neanderthal and Denisova
or Unknwon from now. in kya
        T_H_A: Time interval for separation between Neanderthal and modern
humans from T_N_D. in kya
        T_H_X: Time interval for separation between African archaic and
modern humans from now. in kya
        mix: the fraction of African genome is replaced but Back to Africa
population
        nintro: the fraction of introgression happened to OOA populations.
        dintro: the fraction of introgression happened to East Asians
        xintro: the fraction of introgression happened to African populations
    :param inds: the number of haplotypes per populations for example
(10,10,10)
    :param length: the length of chromosome that has to be simulated. default
is 1mb region
    :param mutation_rate: the amount of mutation rate. default is 1.45x10^-8
per generation per base
    :param recombination_rate: the amount of recombination rate. default is
10^-8 per generation per base
    :param replicates: the number of replicated of length chromosome. default
is 300
    :return: will return the msprime simulations. which then can be used to
extract SFS
    """
    (N_A, N_AF1, N_EU, N_AS, N_EU0, N_AS0, N_BC, N_B, N_AF2, T_DIntro,
T_EU_AS, T_NIntro,
     T_XIntro, T_Mix, T_Sep, T_B, T_AF, T_N_D, T_H_A, T_H_X, mix, nintro,
dintro,
     xintro) = params
    (n1, n2, n3) = inds
    T_DIntro, T_EU_AS, T_NIntro, T_XIntro, T_Mix, T_Sep, T_B, T_AF, T_N_D,
T_H_A, T_H_X = numpy.array(
        [T_DIntro, T_EU_AS, T_NIntro, T_XIntro, T_Mix, T_Sep, T_B, T_AF,
T_N_D, T_H_A, T_H_X]) * (1e3 / 29.0)
    events = {}
    AFR, EUR, ASN, GST, NEA, DEN, XAF = 0, 1, 2, 3, 4, 5, 6
    r_EU = (math.log(N_EU / N_EU0) / (T_DIntro + T_EU_AS))
    r_AS = (math.log(N_AS / N_AS0) / (T_DIntro + T_EU_AS))
    # N_B=N_EU0+N_AS0
    population_configurations = [
        msprime.PopulationConfiguration(
            sample_size=n1, initial_size=N_AF1),
        msprime.PopulationConfiguration(
            sample_size=n2, initial_size=N_EU, growth_rate=r_EU),
        msprime.PopulationConfiguration(
            sample_size=n3, initial_size=N_AS, growth_rate=r_AS),
        msprime.PopulationConfiguration(
            sample_size=0, initial_size=N_BC),
        msprime.PopulationConfiguration(
            sample_size=0, initial_size=N_A),
        msprime.PopulationConfiguration(
            sample_size=0, initial_size=N_A),
        msprime.PopulationConfiguration(
            sample_size=0, initial_size=N_A)
    ]
    # Denisova or unknown admixture
    events['deni_intro_asn'] = T_DIntro
    deni_intro_asn = [msprime.MassMigration(
        time=events['deni_intro_asn'], source=ASN, destination=DEN,
proportion=dintro)]
    # CEU and CHB merge into B with rate changes at T_EU_AS
    events['split_eu_as'] = events['deni_intro_asn'] + T_EU_AS
    split_eu_as = [msprime.MassMigration(
        time=events['split_eu_as'], source=ASN, destination=EUR,
proportion=1.0),
        msprime.PopulationParametersChange(
            time=events['split_eu_as'], initial_size=N_B, growth_rate=0,
population_id=EUR),
        msprime.MigrationRateChange(time=events['split_eu_as'], rate=0)]
    # introgression
    events['nean_intro_eur'] = T_NIntro + events['split_eu_as']
    nean_intro_eur = [msprime.MassMigration(
        time=events['nean_intro_eur'], source=EUR, destination=NEA,
proportion=nintro)]
    # introgression XAFR
    events['xafr_intro_afr'] = T_XIntro + events['split_eu_as']
    xafr_intro_afr = [msprime.MassMigration(
        time=events['xafr_intro_afr'], source=AFR, destination=XAF,
proportion=xintro)]
    # back migration
    events['back_migration'] = events['split_eu_as'] + T_Mix
    back_migration = [msprime.MassMigration(time=events['back_migration'],
source=AFR,
                                            destination=GST, proportion=mix),
msprime.PopulationParametersChange(
        time=events['back_migration'], initial_size=N_AF2,
population_id=AFR)]
    # spearation between back and OOA
    events['split_ooa_back'] = events['back_migration'] + T_Sep
    split_ooa_back = [msprime.MassMigration(time=events['split_ooa_back'],
source=GST,
                                            destination=EUR, proportion=1.0)]
    # Population B merges into YRI at T_B
    events['split_afr_ooa'] = max(events['split_ooa_back'],
events['xafr_intro_afr'],
                                  events['nean_intro_eur']) + T_B
    split_afr_ooa = [msprime.MassMigration(
        time=events['split_afr_ooa'], source=EUR, destination=AFR,
proportion=1.0),
        msprime.MigrationRateChange(time=events['split_afr_ooa'], rate=0)]
    # Size changes to N_A at T_AF
    events['ancestral_size'] = events['split_afr_ooa'] + T_AF
    ancestral_size = [msprime.PopulationParametersChange(
        time=events['ancestral_size'], initial_size=N_A, population_id=AFR)]
    # Denisova or Unknwon merging with Neanderthal
    events['neanderthal_denisova'] = T_N_D
    neanderthal_denisova = [msprime.MassMigration(
        time=events['neanderthal_denisova'], source=DEN, destination=NEA,
proportion=1.0),
        msprime.PopulationParametersChange(
            time=events['neanderthal_denisova'], initial_size=N_A,
population_id=NEA)]
    # Neanderthal merging with humans
    events['human_neanderthal'] = T_N_D + T_H_A
    human_neanderthal = [msprime.MassMigration(
        time=events['human_neanderthal'], source=NEA, destination=AFR,
proportion=1.0)]
    # XAFR merging with humans
    events['human_xafr'] = T_H_X
    human_xafr = [msprime.MassMigration(
        time=events['human_xafr'], source=XAF, destination=AFR,
proportion=1.0)]
    demographic_events = []
    for event in collections.OrderedDict(sorted(events.items(), key=lambda x:
x[1])).keys():
        demographic_events = demographic_events + eval(event)
    geno = msprime.simulate(
        population_configurations=population_configurations,
        demographic_events=demographic_events, length=length,
mutation_rate=mutation_rate,
        num_replicates=replicates,
        recombination_rate=recombination_rate)
    return geno
```

Model M:

```
def MNDX(params, inds, length=1e6, mutation_rate=1.45e-8,
recombination_rate=1e-8, replicates=300):
    """
        This is the mix OOA model with Neanderthal to OOA population,
Denisova or Unknown to East Asia and
            African archaic to African populations.
        :param params: all the parameters necessary for this model in a list
or array format
            N_A: The ancestral effective population size
            N_AF: Modern effective population size of Africa population
            N_EU: Modern effective population size of European population
            N_AS: Modern effective population size of East Asian population
            N_EU0: Effective population size of European population before
exponential growth.
            N_AS0: Effective population size of East Asian population before
exponential growth.
            N_MX: Effective population size of Mix to OOA population (assume
second OOA).
            N_B1: Effective population size of Out of Africa (OOA)
populations after admixture with mix population
            N_B2: Effective population size of Out of Africa before admixture
(assume fist OOA).
            T_DIntro: Time interval for introgression in East Asian from
Denisova or Unknown from now. in kilo year ago
                (kya)
            T_EU_AS: Time interval for separation of European and East Asian
from T_DIntro. in kya
            T_NIntro: Time interval for introgression in OOA from Neanderthal
from T_EU_AS. in kya
            T_XIntro: Time interval for introgression in African population
from African archaic from T_EU_AS. in kya
            T_Mix: Time interval for mixing between OOA (OOA_1) and Mix
(OOA_2) from T_EU_AS. in kya
            T_Sep: Time interval for separation of Mix population from Africa
from T_Mix. in kya
            T_B: Time interval for separation between Africa and OOA
populations from maximum between T_NIntro,T_XIntro
                and T_Sep. in kya
            T_AF: Time interval for decrease of effective population size of
African population to ancestral effective
                population size from T_B. in kya
            T_N_D: Time interval for separation between Neanderthal and
Denisova or Unknwon from now. in kya
            T_H_A: Time interval for separation between Neanderthal and
modern humans from T_N_D. in kya
            T_H_X: Time interval for separation between African archaic and
modern humans from now. in kya
            mix: the fraction of OOA (OOA_1) genome is replaced but Mix
population (OOA_2)
            nintro: the fraction of introgression happened to OOA
populations.
```

```
            dintro: the fraction of introgression happened to East Asians
            xintro: the fraction of introgression happened to African
populations
        :param inds: the number of haplotypes per populations for example
(10,10,10)
        :param length: the length of chromosome that has to be simulated.
default is 1mb region
        :param mutation_rate: the amount of mutation rate. default is
1.45x10^-8 per generation per base
        :param recombination_rate: the amount of recombination rate. default
is 10^-8 per generation per base
        :param replicates: the number of replicated of length chromosome.
default is 300
        :return: will return the msprime simulations. which then can be used
to extract SFS
        """
    (N_A, N_AF, N_EU, N_AS, N_EU0, N_AS0, N_MX, N_B1, N_B2, T_DIntro,
T_EU_AS,
     T_NIntro, T_XIntro, T_Mix, T_Sep, T_B, T_AF, T_N_D, T_H_A, T_H_X, mix,
nintro,
     dintro, xintro) = params
    (n1, n2, n3) = inds
    T_DIntro, T_EU_AS, T_NIntro, T_XIntro, T_Mix, T_Sep, T_B, T_AF, T_N_D,
T_H_A, T_H_X = numpy.array(
        [T_DIntro, T_EU_AS, T_NIntro, T_XIntro, T_Mix, T_Sep, T_B, T_AF,
T_N_D, T_H_A, T_H_X]) * (1e3 / 29.0)
    events = {}
    AFR, EUR, ASN, GST, NEA, DEN, XAF = 0, 1, 2, 3, 4, 5, 6
    r_EU = (math.log(N_EU / N_EU0) / (T_DIntro + T_EU_AS))
    r_AS = (math.log(N_AS / N_AS0) / (T_DIntro + T_EU_AS))
    # N_B=N_EU0+N_AS0
    population_configurations = [
        msprime.PopulationConfiguration(
            sample_size=n1, initial_size=N_AF),
        msprime.PopulationConfiguration(
            sample_size=n2, initial_size=N_EU, growth_rate=r_EU),
        msprime.PopulationConfiguration(
            sample_size=n3, initial_size=N_AS, growth_rate=r_AS),
        msprime.PopulationConfiguration(
            sample_size=0, initial_size=N_MX),
        msprime.PopulationConfiguration(
            sample_size=0, initial_size=N_A),
        msprime.PopulationConfiguration(
            sample_size=0, initial_size=N_A),
        msprime.PopulationConfiguration(
            sample_size=0, initial_size=N_A) ]
    # Denisova or unknown admixture
    events['deni_intro_asn'] = T_DIntro
    deni_intro_asn = [msprime.MassMigration(
        time=events['deni_intro_asn'], source=ASN, destination=DEN,
proportion=dintro)]
    # CEU and CHB merge into B with rate changes at T_EU_AS
    events['split_eu_as'] = events['deni_intro_asn'] + T_EU_AS
    split_eu_as = [msprime.MassMigration(
        time=events['split_eu_as'], source=ASN, destination=EUR,
proportion=1.0),
        msprime.PopulationParametersChange(
            time=events['split_eu_as'], initial_size=N_B1, growth_rate=0,
population_id=EUR),
        msprime.MigrationRateChange(time=events['split_eu_as'], rate=0)]
    # NEAN introgression
    events['nean_intro_eur'] = T_NIntro + events['split_eu_as']
    nean_intro_eur = [msprime.MassMigration(
        time=events['nean_intro_eur'], source=EUR, destination=NEA,
proportion=nintro)]
    # introgression XAFR
    events['xafr_intro_afr'] = T_XIntro + events['split_eu_as']
    xafr_intro_afr = [msprime.MassMigration(
        time=events['xafr_intro_afr'], source=AFR, destination=XAF,
proportion=xintro)]
    # mix migration
    events['mix_migration'] = T_Mix + events['split_eu_as']
    mix_migration = [msprime.MassMigration(time=events['mix_migration'],
source=EUR,
                                           destination=GST, proportion=mix),
msprime.PopulationParametersChange(
        time=events['mix_migration'], initial_size=N_B2, population_id=EUR)]
    # spearation between back and OOA
    events['split_afr_mix'] = events['mix_migration'] + T_Sep
    split_afr_mix = [msprime.MassMigration(time=events['split_afr_mix'],
source=GST,
                                           destination=AFR, proportion=1.0)]
    # Population B merges into YRI at T_B
    events['split_afr_ooa'] = max(events['split_afr_mix'],
events['xafr_intro_afr'], events['nean_intro_eur']) + T_B
    split_afr_ooa = [msprime.MassMigration(
        time=events['split_afr_ooa'], source=EUR, destination=AFR,
proportion=1.0),
        msprime.MigrationRateChange(time=events['split_afr_ooa'], rate=0)]
    # Size changes to N_A at T_AF
    events['ancestral_size'] = events['split_afr_ooa'] + T_AF
    ancestral_size = [msprime.PopulationParametersChange(
        time=events['ancestral_size'], initial_size=N_A, population_id=AFR)]
    # Denisova or Unknwon merging with Neanderthal
    events['neanderthal_denisova'] = T_N_D
    neanderthal_denisova = [msprime.MassMigration(
        time=events['neanderthal_denisova'], source=DEN, destination=NEA,
proportion=1.0),
        msprime.PopulationParametersChange(
            time=events['neanderthal_denisova'], initial_size=N_A,
population_id=NEA)]
    # Neanderthal merging with humans
    events['human_neanderthal'] = T_N_D + T_H_A
    human_neanderthal = [msprime.MassMigration(
        time=events['human_neanderthal'], source=NEA, destination=AFR,
proportion=1.0)]
```

```
    # XAFR merging with humans
    events['human_xafr'] = T_H_X
    human_xafr = [msprime.MassMigration(
        time=events['human_xafr'], source=XAF, destination=AFR,
proportion=1.0)]
    demographic_events = []
    for event in collections.OrderedDict(sorted(events.items(), key=lambda x:
x[1])).keys():
        demographic_events = demographic_events + eval(event)
    geno = msprime.simulate(
        population_configurations=population_configurations,
        demographic_events=demographic_events, length=length,
mutation_rate=mutation_rate,
        num_replicates=replicates,
        recombination_rate=recombination_rate)
    return geno
```