The American Journal of Human Genetics, Volume *108*

Supplemental information

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

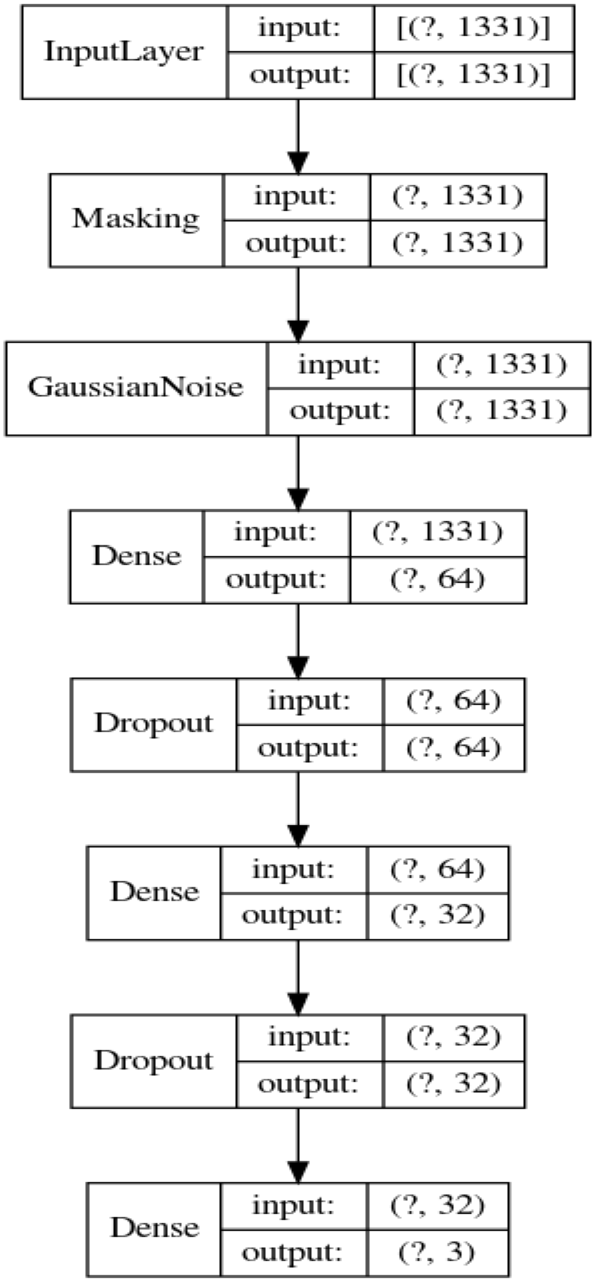
Francesco Montinaro, Vasili Pankratov, Burak Yelmen, Luca Pagani, and Mayukh Mondal

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)

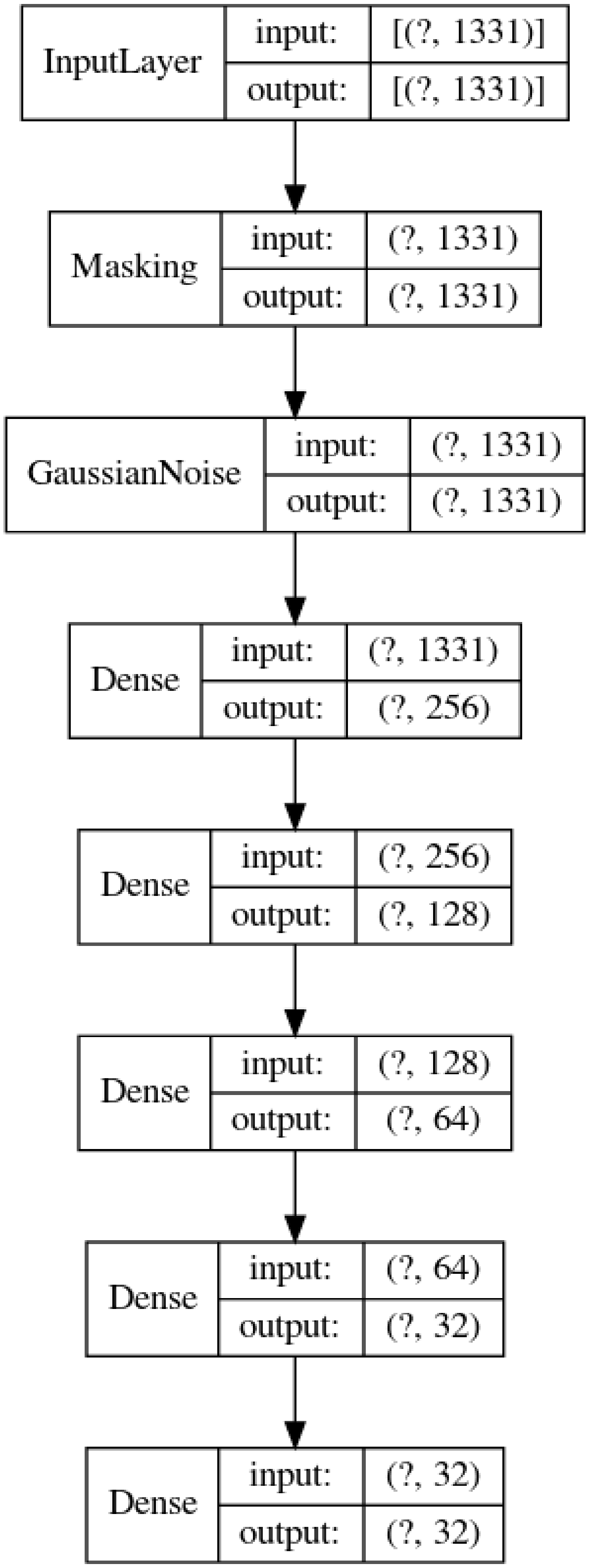
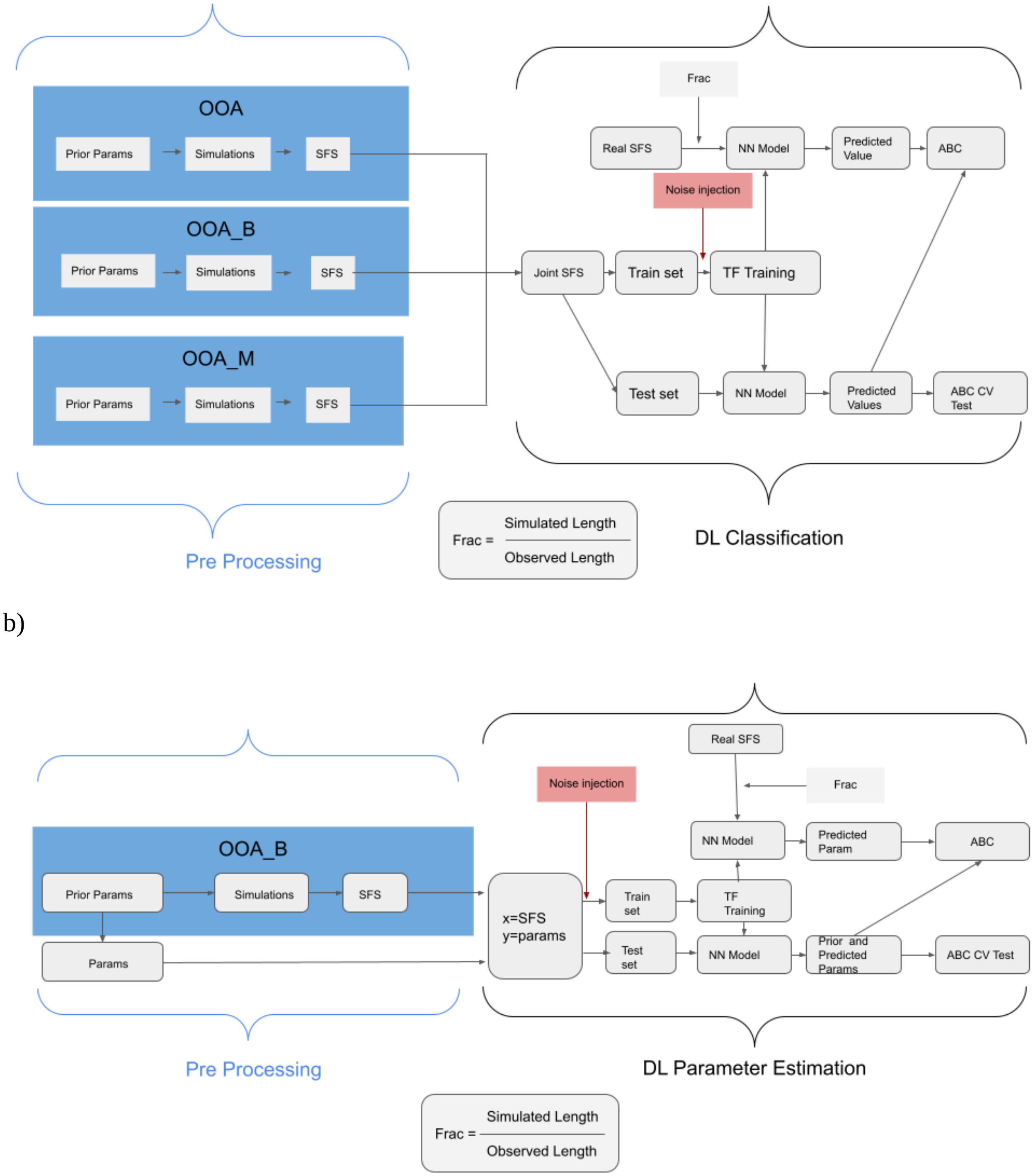

InputLayer
input:
[(?, 1331)]
output:
[(?, 1331)]
Masking
input:
(?, 1331)
output:
(?, 1331)
GaussianNoise
input:
(?, 1331)
output:
(?, 1331)
Dense
input:
(?, 1331)
output:
(?, 256)
Dense
input:
(?, 256)
output:
(?, 128)
Dense
input:
(?, 128)
output:
(?, 64)
Dense
input:
(?, 64)
output:
(?, 32)
Dense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



c)

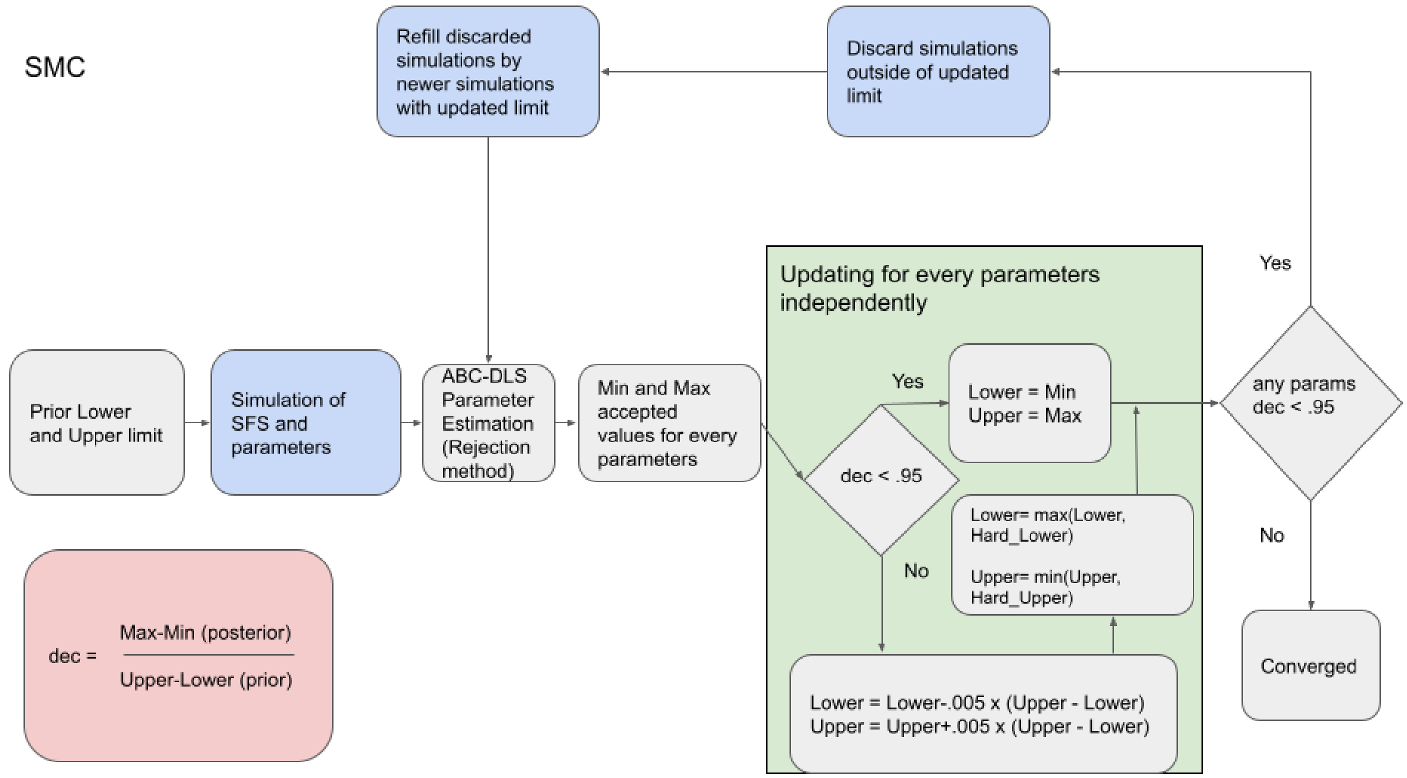
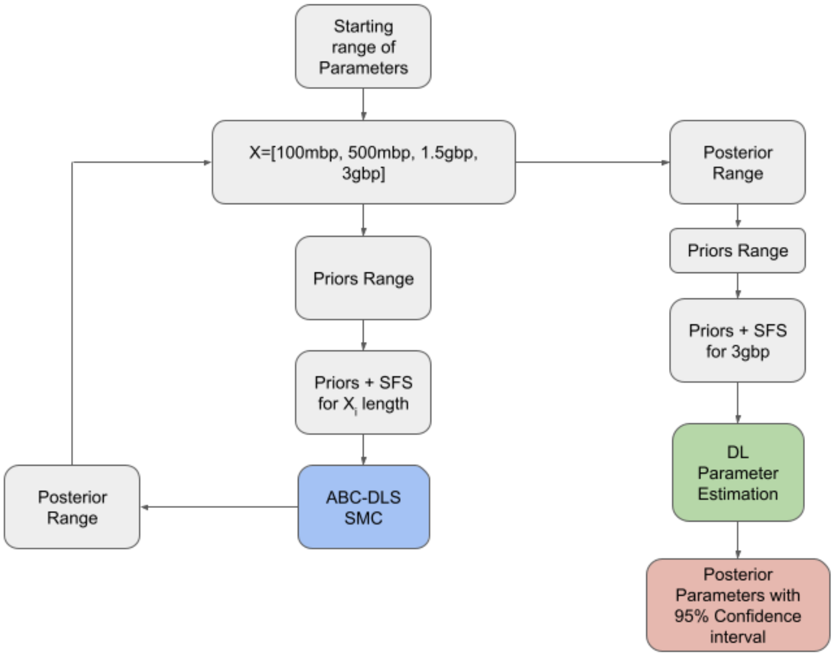

SMC
Prior Lower and Upper limit
Simulation of SFS and parameters
ABC-DLS Parameter Estimation (Rejection method)
Min and Max accepted values for every parameters
Updating for every parameters independently
dec < .95
Yes
Lower = Min
Upper = Max
No
Lower = Lower-.005 x (Upper - Lower)
Upper = Upper+.005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



b)

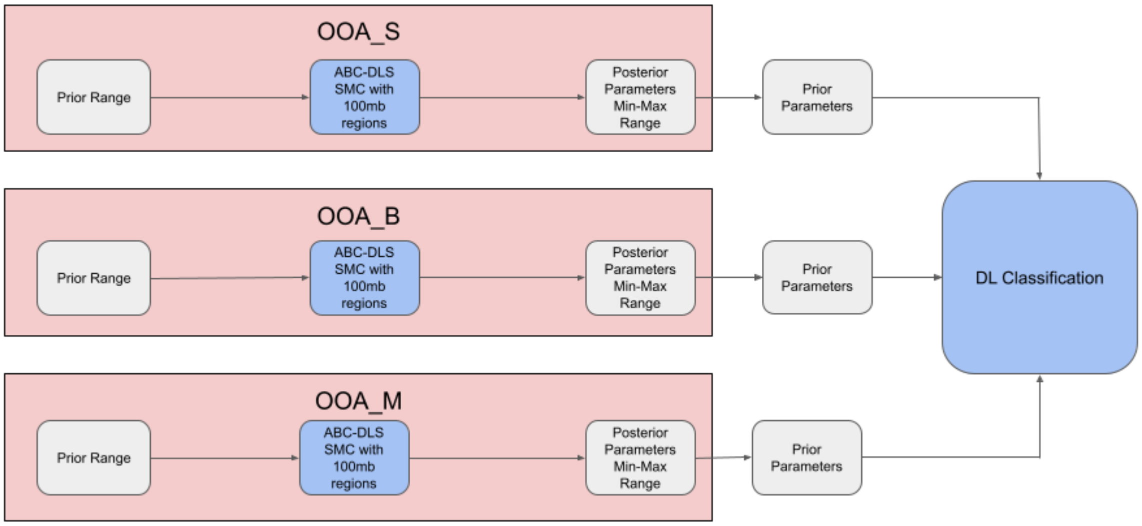


Figure S4: Comparison between mutation age estimate coming from relate between the simulation and real data.

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.

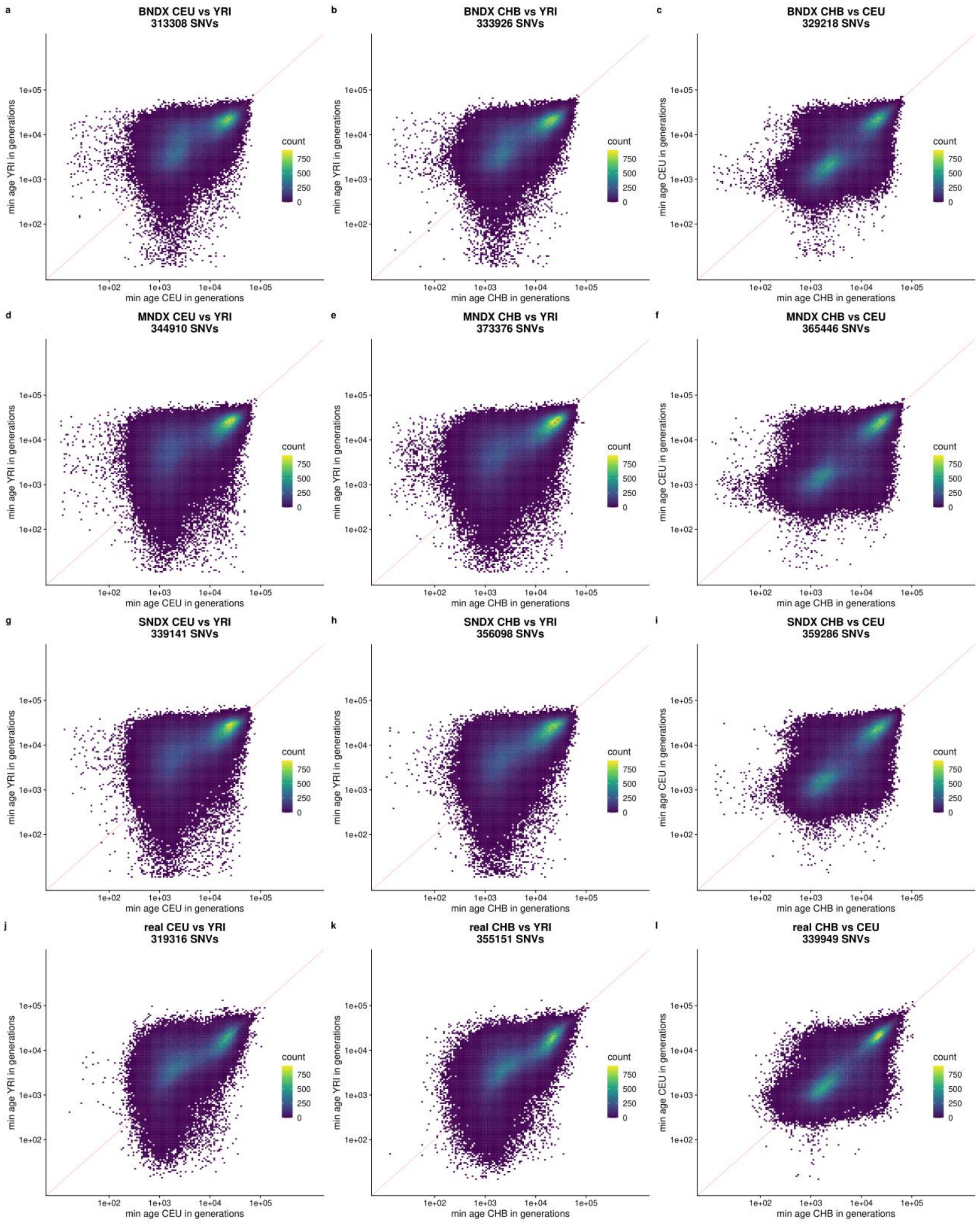

a
BNDX CEU vs YRI
313308 SNVs
b
BNDX CHB vs YRI
333926 SNVs
c
BNDX CHB vs CEU
329218 SNVs
d
MNDX CEU vs YRI
344910 SNVs
e
MNDX CHB vs YRI
373376 SNVs
f
MNDX CHB vs CEU
365446 SNVs
g
SNDX CEU vs YRI
339141 SNVs
h
SNDX CHB vs YRI
356098 SNVs
i
SNDX CHB vs CEU
359286 SNVs
j
real CEU vs YRI
319316 SNVs
k
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).

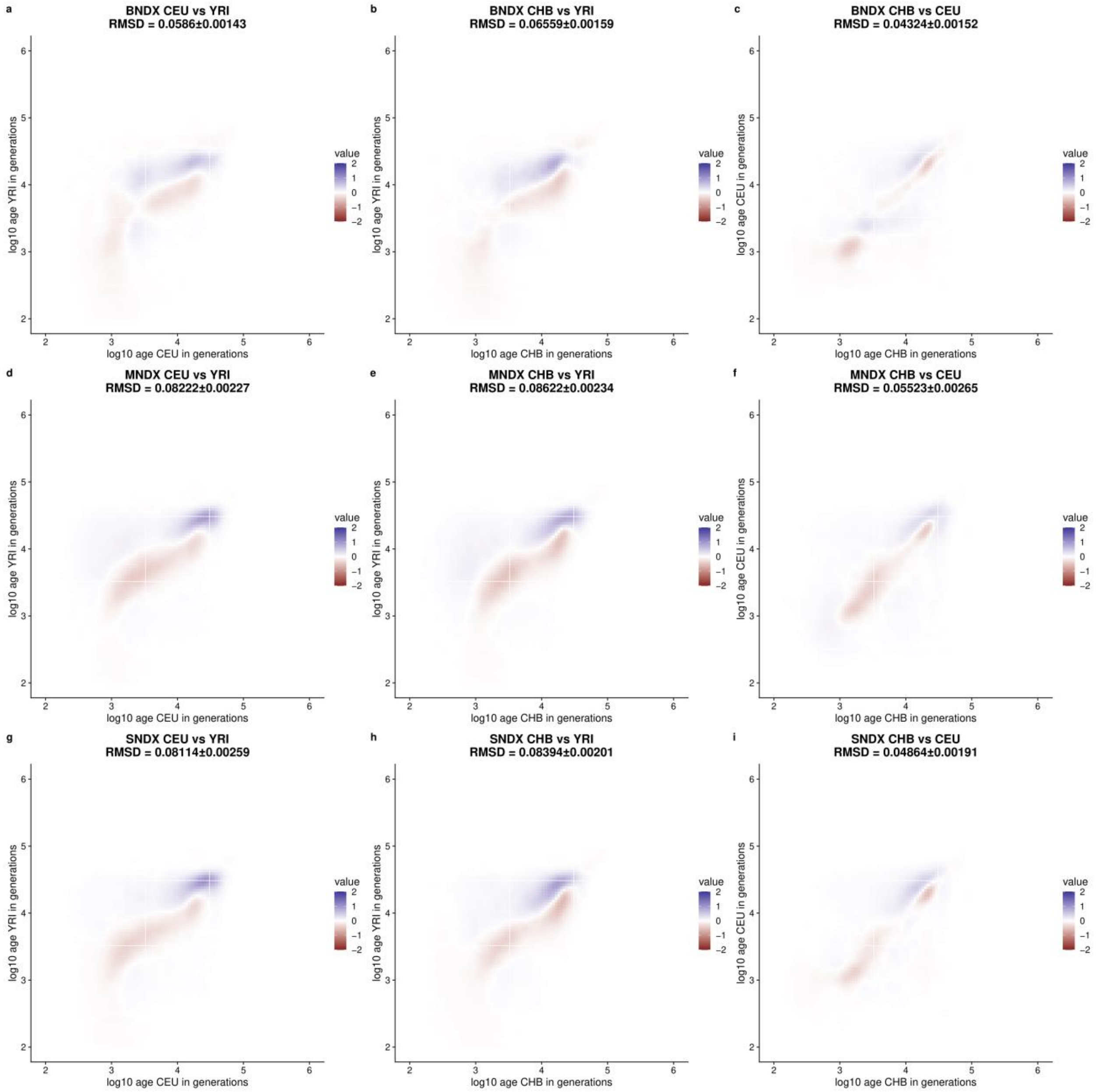
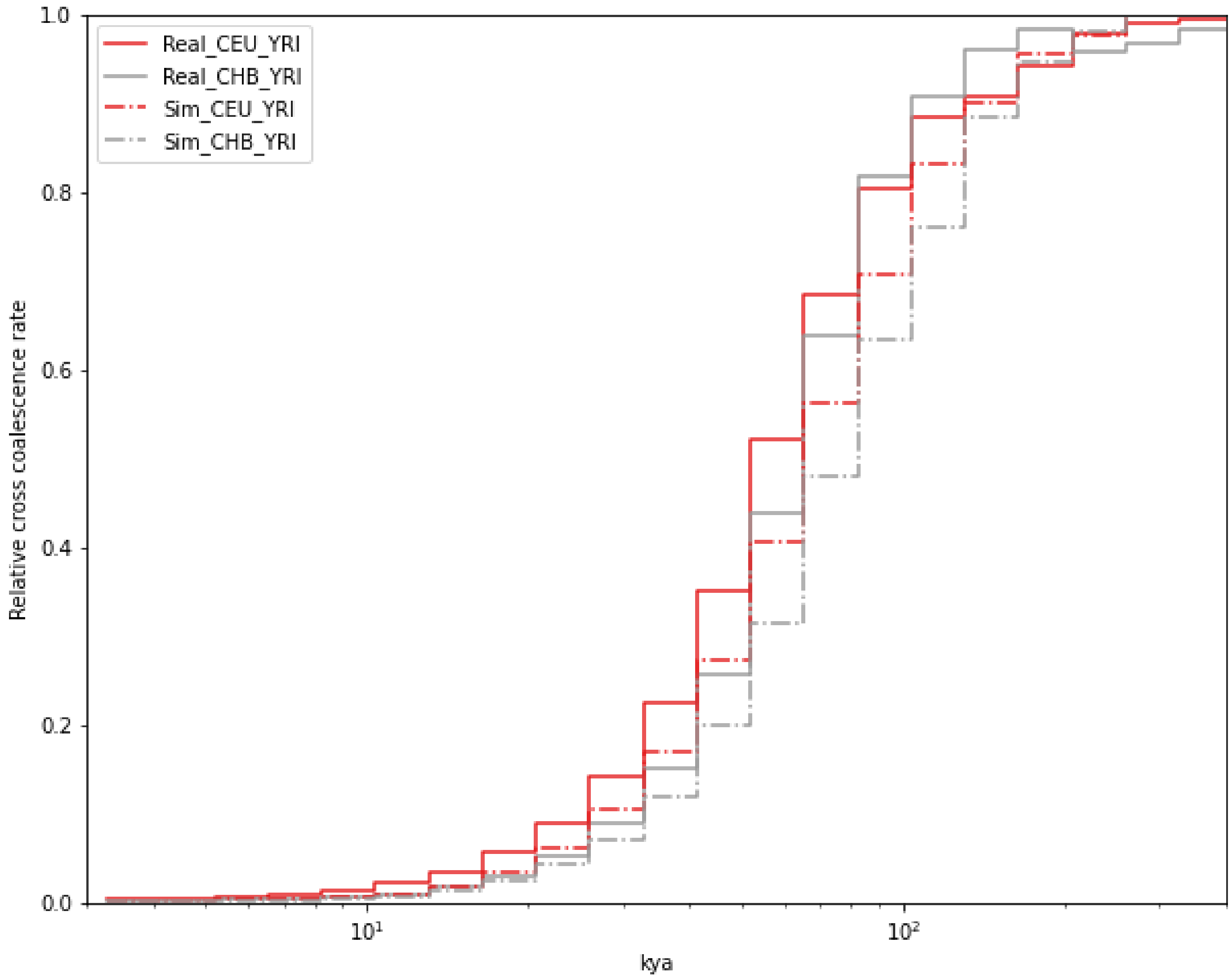


Figure S6: Cross population coalescent rate calculated using relate for Model S



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 ($\times 10^{-5}$)	0 - 50	19.49 (19.32 - 19.69)
m_AF_EU ($\times 10^{-5}$)	0 - 50	0.66 (0.59 - 0.72)
m_AF_AS ($\times 10^{-5}$)	0 - 50	0.76 (0.7 - 0.82)
m_EU_AS ($\times 10^{-5}$)	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 (×10^{-5})	6.33	0.07	15.24
m_B_AF (×10^{-5})	6.51	4.47	10.05
m_AF_EU (×10^{-5})	0.57	0.03	1.38
m_EU_AF (×10^{-5})	0.47	0.05	0.89
m_AF_AS (×10^{-5})	2.27	1.31	3.00
m_AS_AF (×10^{-5})	0.25	-0.02	0.65
m_EU_AS (×10^{-5})	1.18	0.01	3.81
m_AS_EU (×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
```