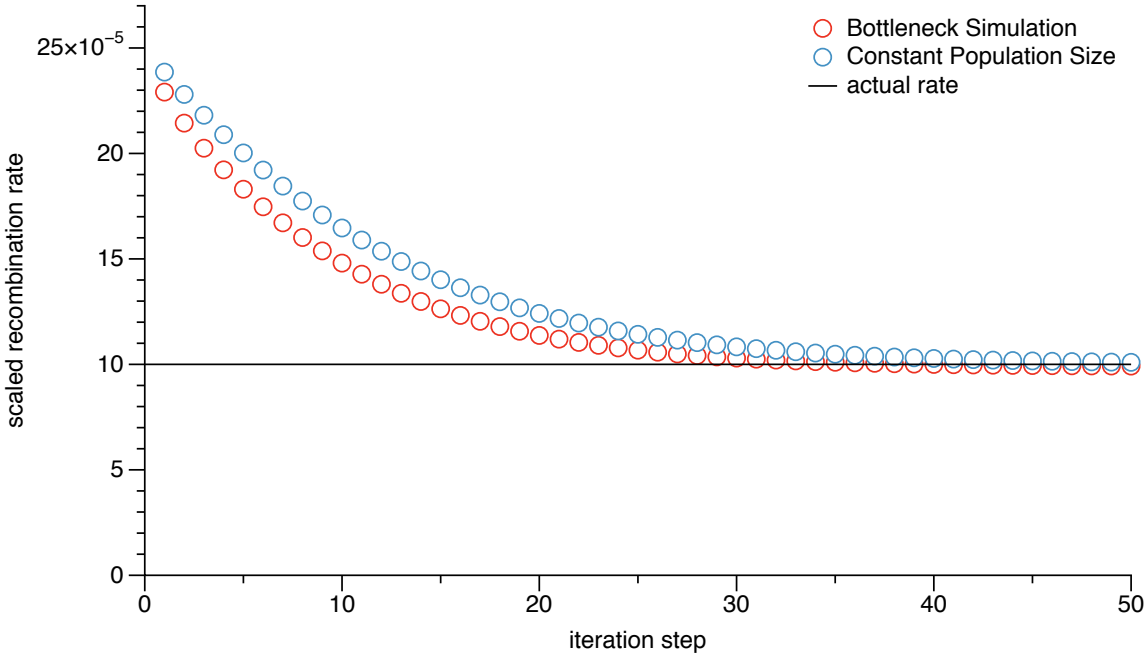
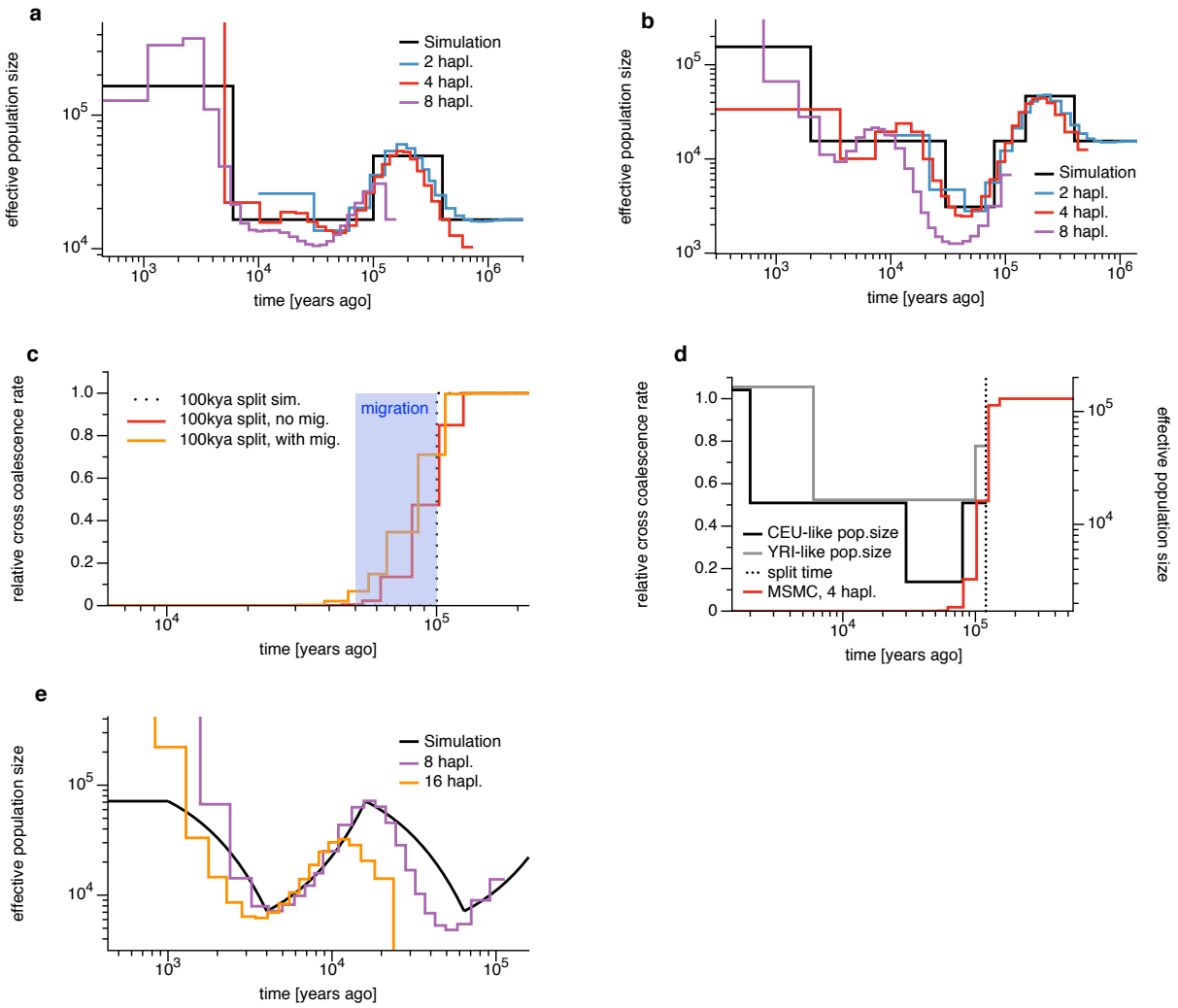
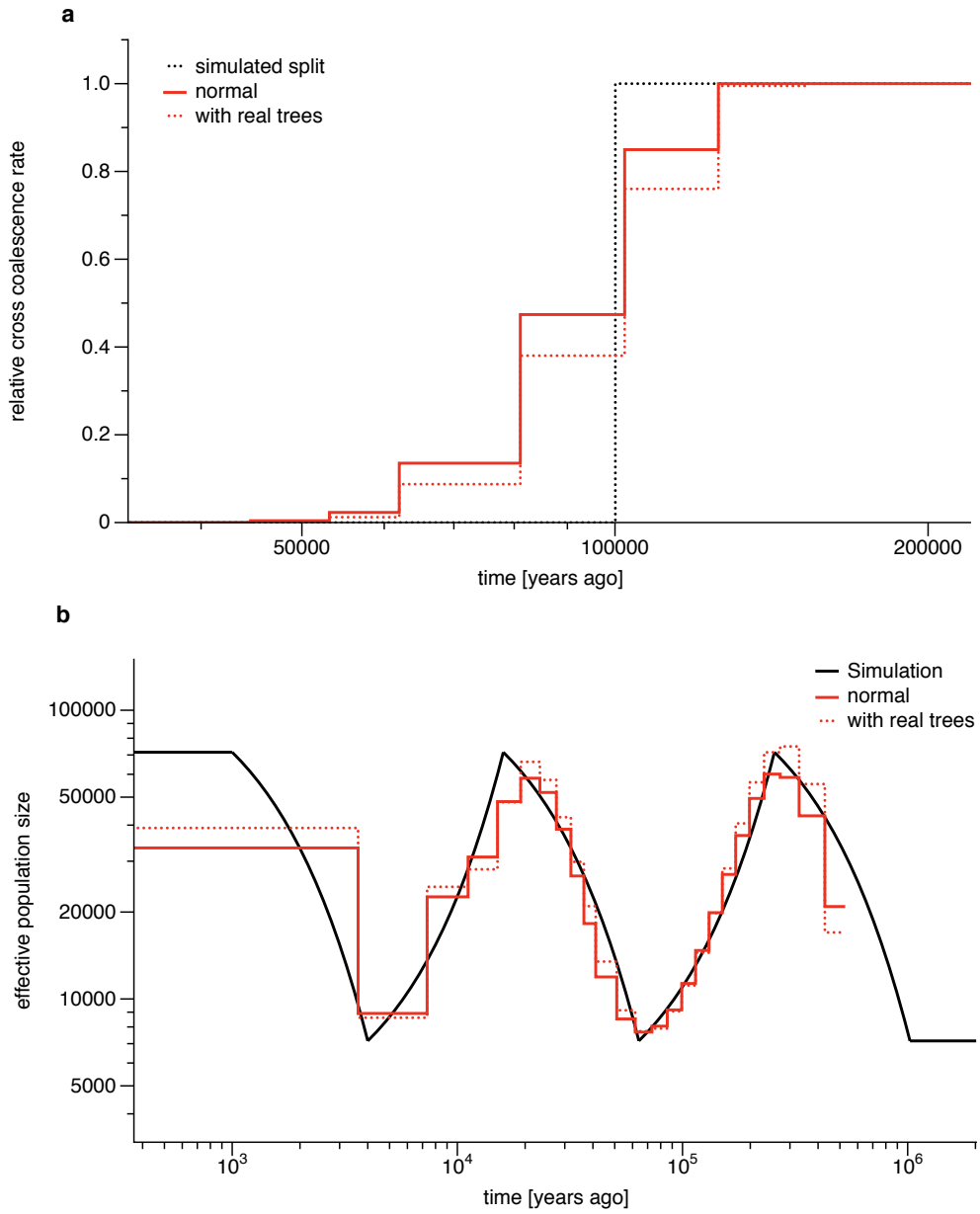


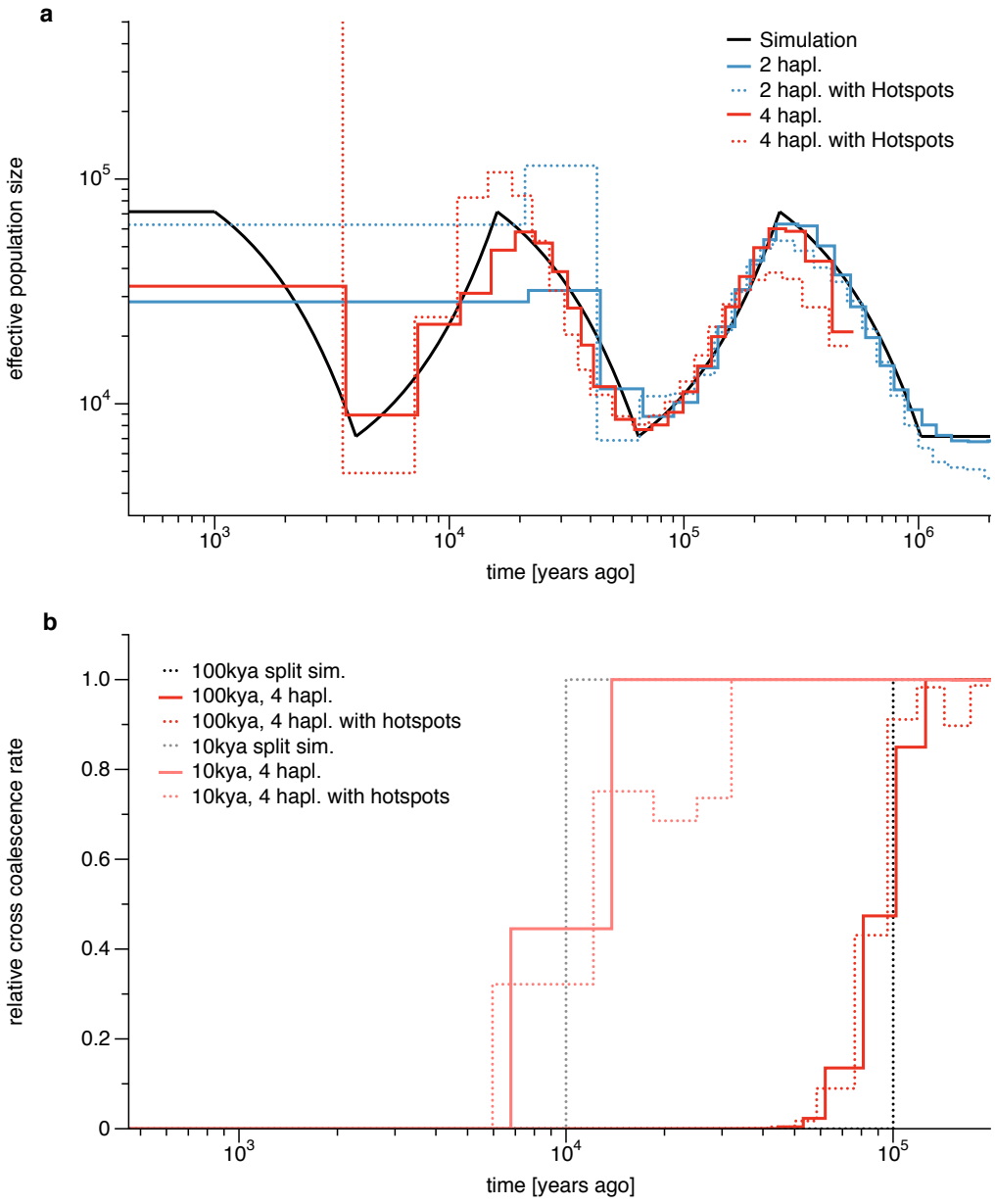
Supplementary Figure 1: Recombination rate inference from PSMC'
MSMC for two haplotypes is a special case that we call PSMC', in contrast to PSMC because we use SMC' (**Supplementary Note**) as underlying coalescent model. Here we show the iterative estimation of the recombination rate for two demographic scenarios: i) a constant population size and ii) a bottleneck in the past. As can be seen, in both cases the estimated recombination rate converges quickly to the true value with very high accuracy.



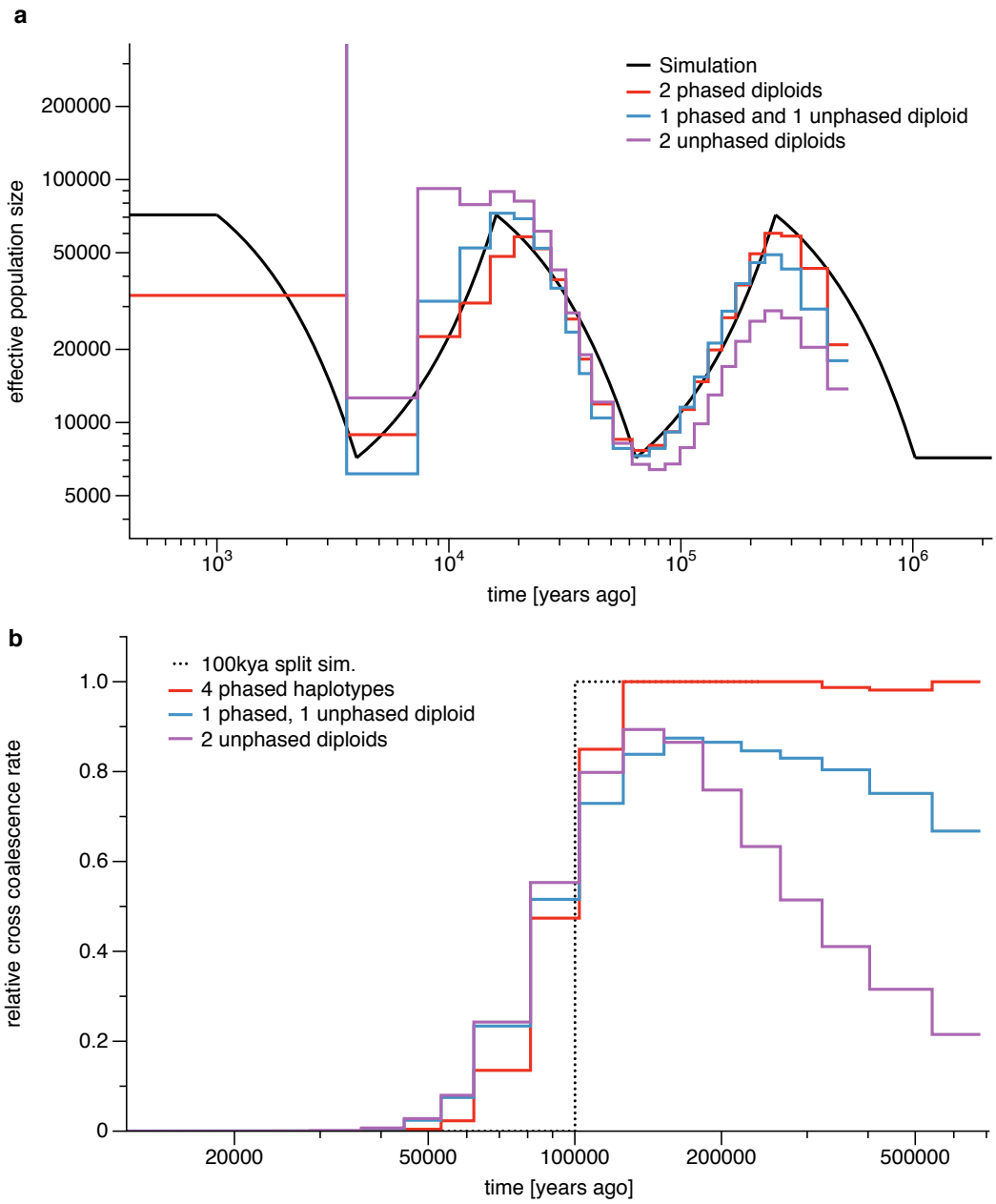
Supplementary Figure 2: SImulations of other scenarios
See the **Supplementary Note** for details on these additional simulations. (**a**) MSMC results from simulated data which represents a much simplified CEU population size history with sharp changes. (**b**) Similar to a) but with a simplified YRI-like history. (**c**) A population split with subsequent migration. (**d**) A population split with subsequent population size changes. (**e**) Inference from 8 and 16 haplotypes. For 16 haplotypes, we needed to reduce the computational complexity by reducing the simulated sequence to 1Gb instead of 3 Gb, and used a coarse-grained set of parameters with 20 time intervals instead of 40.



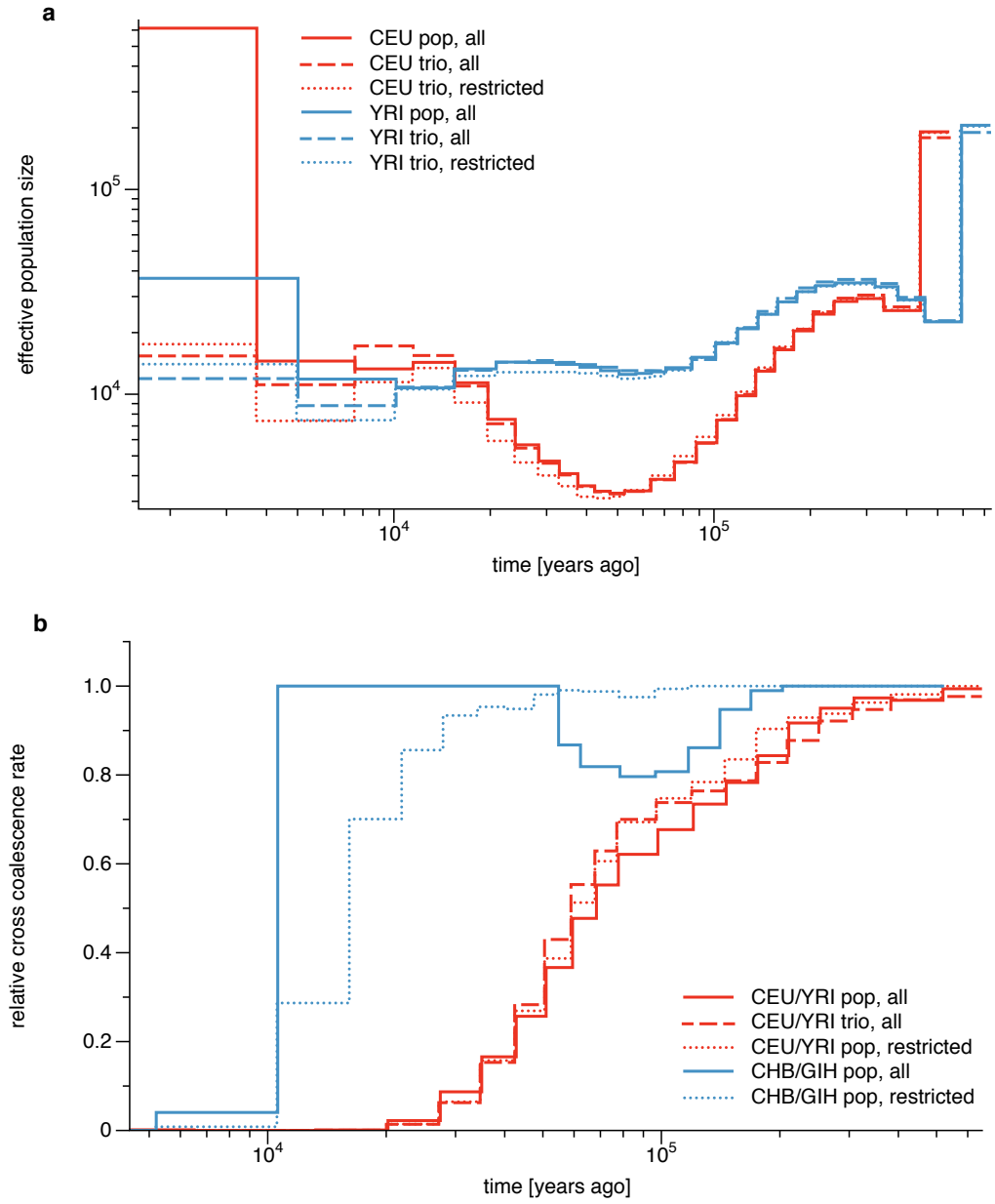
Supplementary Figure 3: Testing Singleton Branch Length Estimates
Here we compare MSMC estimates based on the estimates of T_s obtained via the HMM described in section 7 of the **Supplementary Note** (solid) with the true values of the singleton branch length as output in the simulation (dotted). (**a**) shows population size estimates, (**b**) shows split estimates.



Supplementary Figure 4: Simulations with recombination hotspots
To assess the effect of heterogeneous recombination rates across the genome, we simulated 100 chromosomes with 4 haplotypes of 1Mb each, (**Supplementary Note**). For practical reasons we could not scale up this simulation to 3Gb of total sequence or to more haplotypes. We used as input random chunks of the real human recombination map from the HapMap project. (**a**) This plot shows the effective population size estimates from both the standard simulation and simulations with the human recombination map. We see only small effects of variable recombination rates mostly in the two extreme ends of the estimated time interval. Some of that difference may also be caused by the much smaller total sequence length of the hotspot simulation in comparison to the standard simulation. (**b**) Here we show the two split scenarios at 10kya and 100kya. Again, the differences between the hotspot simulation and the standard simulation are only small.

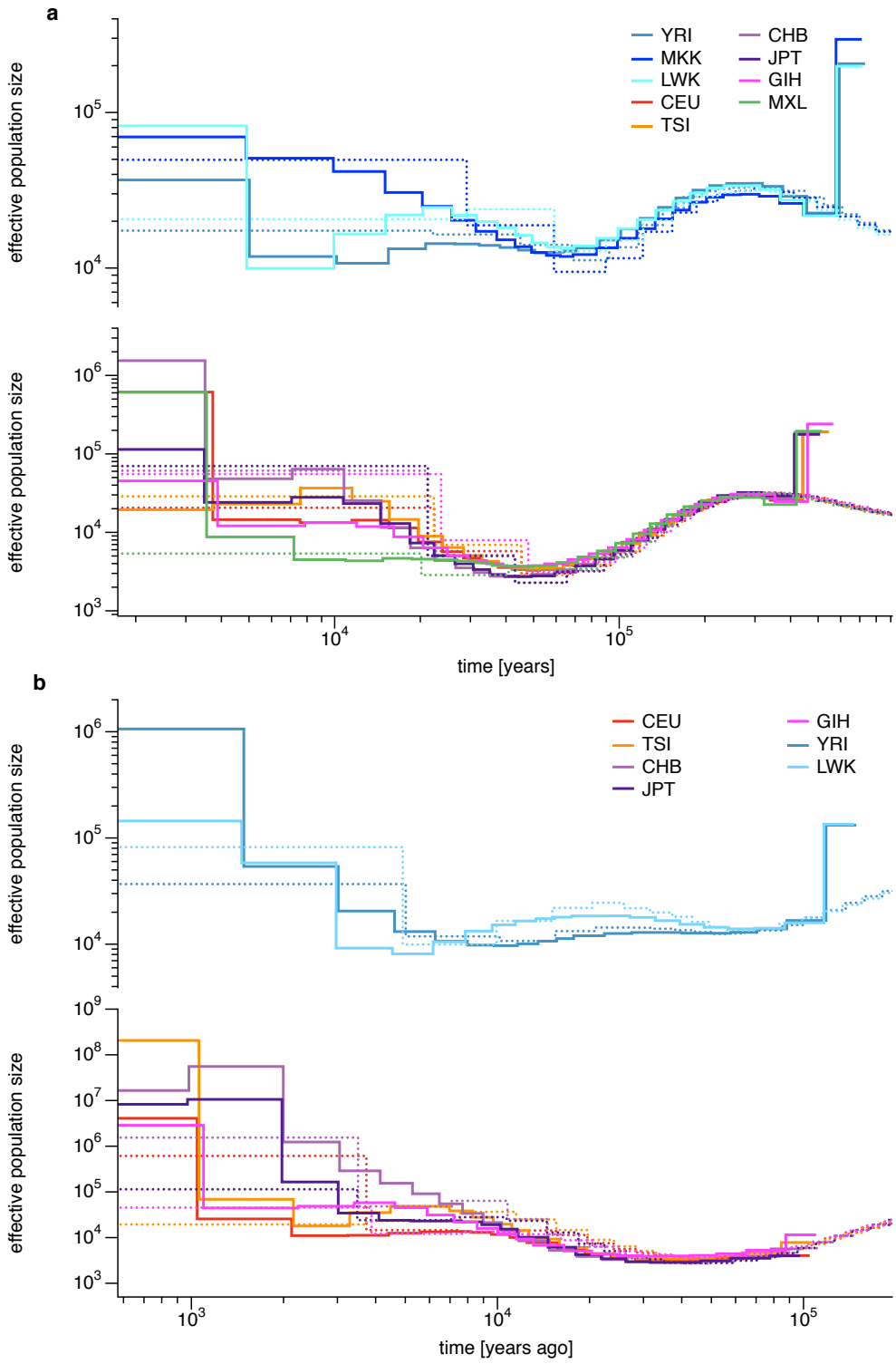


Supplementary Figure 5: Application to unphased data
We generated datasets in which we deliberately "unphased" one or both diploid genomes in a setting of four haplotypes. In (**a**) we plot the population size estimates from two diploid individuals of which both are phased (red), one is unphased (blue) and both unphased (purple). In (**b**) we plot the relative cross coalescence rate estimates based on similarly unphased data.



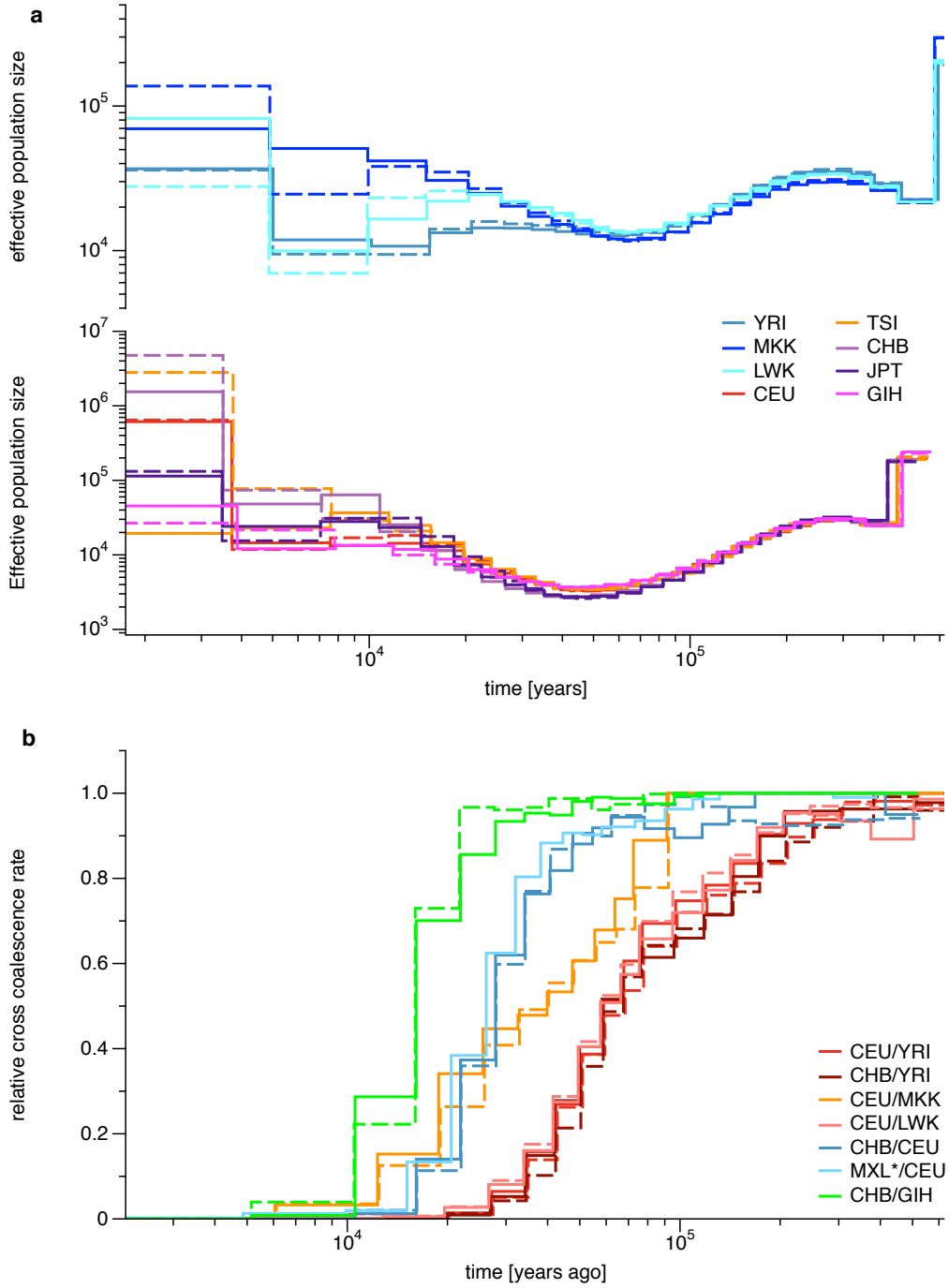
Supplementary Figure 6: Comparison of trio- vs. population-phasing and effect of unphased sites

We tested, whether results from trio-phased data differs from population-phased phasing. We checked this for CEU and YRI, for which we have trio-sequences available. As shown in (**a**) and (**b**), the trio-phased results do not differ strongly from the population phased results (solid vs. dashed lines). When population-phasing our sequences, there are rare sites which are not present in the reference data set and are therefore not phased. There are two possibilities: i) leave them in as unphased sites ("all"), ii) remove them from the analysis ("restricted"). The two cases are shown in a) and b) as solid vs. dotted lines. As shown, for population size inference, removing unphased sites does not appear to improve estimates (in comparison to the trio-phased estimates), but for the population separation analysis, removing unphased sites gives smoother estimates in the most recent times and removes some non-monotonic artifacts (in CHB/GIH).

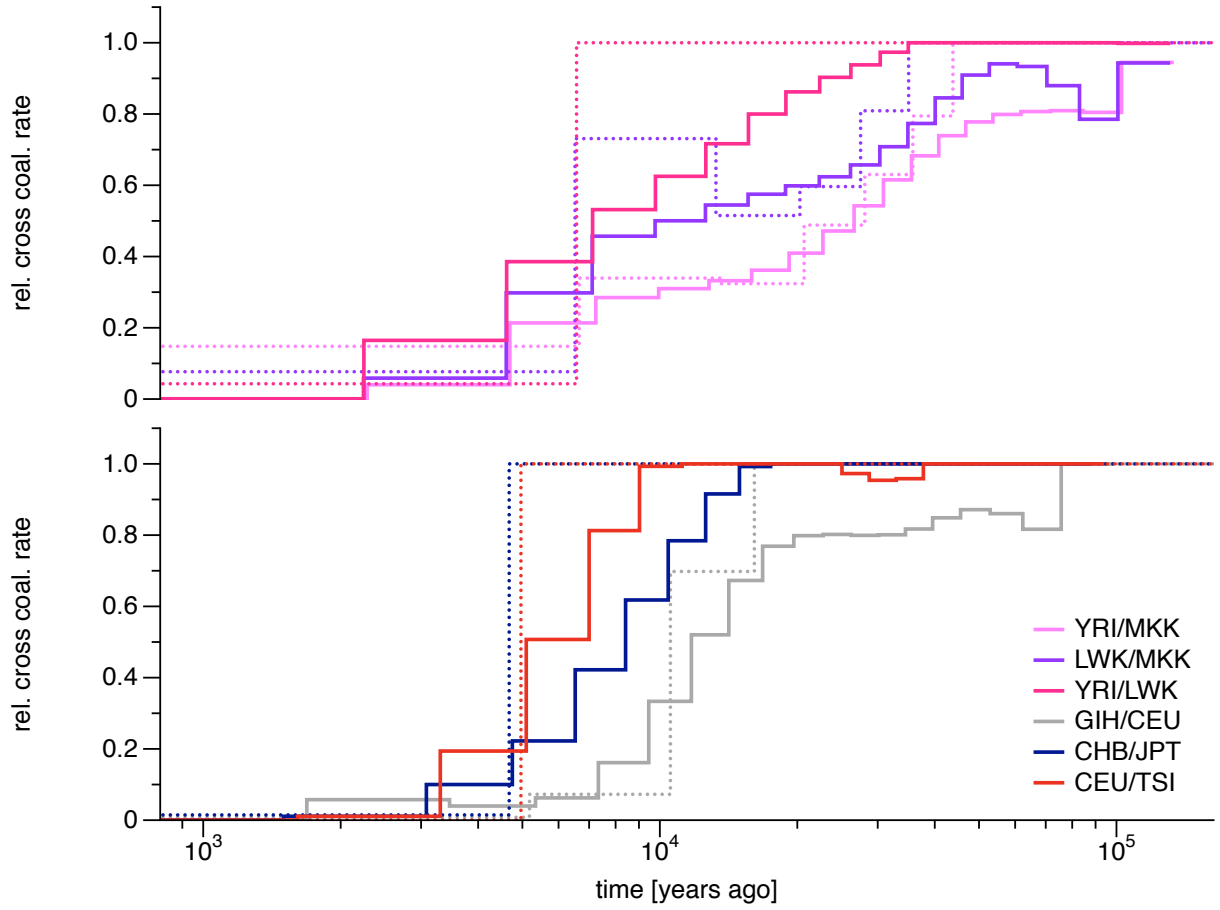


Supplementary Figure 7: Comparisons of population size estimates with two, four and eight haplotypes

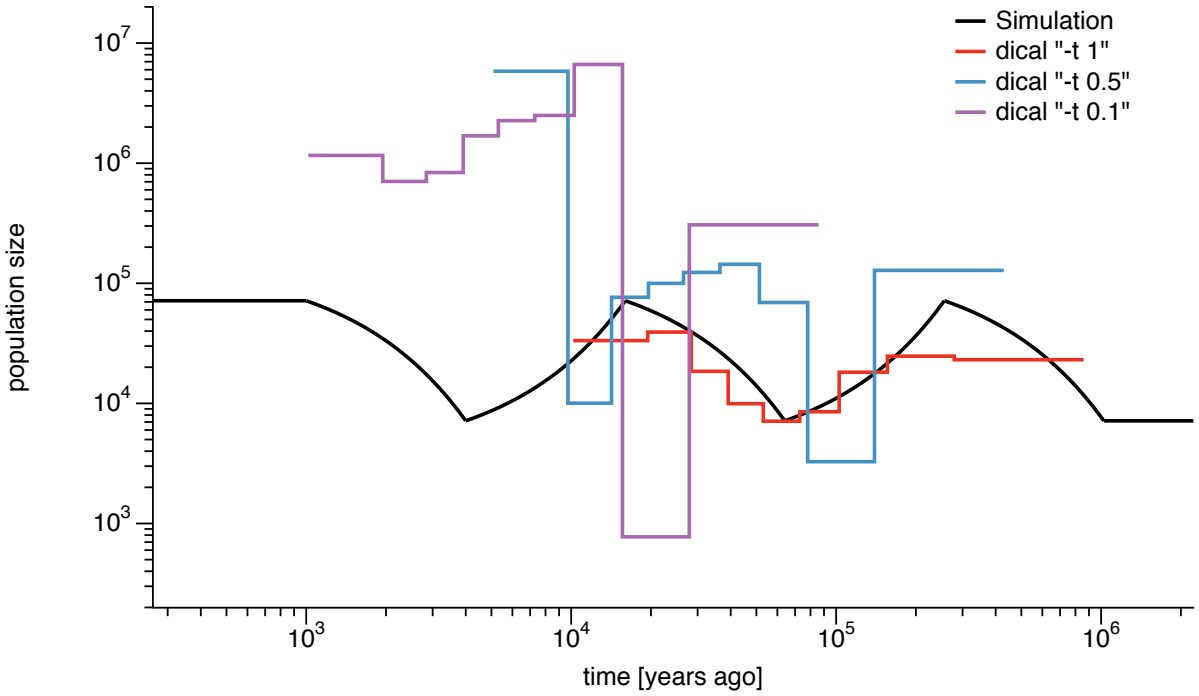
(**a**) We show the estimates based on four haplotypes (solid lines) together with estimates from two haplotypes (dotted lines). For clarity, we separated the curves based on African and Non-African samples. (**b**) This plot shows estimates based on eight haplotypes (thick lines) in comparison with the estimates based on four haplotypes (thin lines).



Supplementary Figure 8: Replicate analysis with four haplotypes
We generated a replicate set of population size and relative cross coalescence rate estimates, based on the two individuals in each population not used for the main analysis, as presented in **Figures 3** and **4**. In both figures, the replicate estimate is shown as dashed line, the original estimate as solid line. For clarity, African and Non-African estimates are separated.



Supplementary Figure 9: Comparison of relative cross coalescence rate estimates with four and eight haplotypes
Here we show relative cross coalescence rate estimates based on eight haplotypes (four haplotypes from each population, in solid lines), with estimates from four haplotypes (two haplotypes from each population, in dotted lines).



Supplementary Figure 10: Comparison with diCal
A different method to estimate historical population sizes from multiple phased haplotypes was recently implemented in the software *diCal* (**Supplementary Note**). Here we have applied *diCal* to 8 haplotypes, each 10Mb long, simulated using the zig-zag population size history as in **Figure 2**. The relatively short length of 10Mb is the same length as used in Sheehan et al. 2013; with the current *diCal* implementation, analysis of larger data sets is not practical. We tested three different time intervals, using the parameter "-t", which sets the left boundary of the last time interval in scaled units. With the "-t 1" option in the plot below, *diCal* obtains correct estimates between 20kya and 200kya (red curve), roughly the same period addressed by MSMC with 2 haplotypes. To explore more recent times that we access with 4 or 8 haplotypes we tried to change the default time interval by using lower "-t" values (see **Supplementary Note** for details), but the resulting population size estimates were not so good (purple and blue lines). The method may be able to perform better if a more efficient implementation allows it to run on whole-genome sized datasets.

Supplementary Table 1

Sample names and populations

Sample ID	Population
NA18526	CHB
NA18537	CHB
NA18555	CHB
NA18558	CHB
NA20845	GIH
NA20846	GIH
NA20847	GIH
NA20850	GIH
NA18940	JPT
NA18942	JPT
NA18947	JPT
NA18956	JPT
NA19017	LWK
NA19020	LWK
NA19025	LWK
NA19026	LWK
NA21732	MKK
NA21733	MKK
NA21737	MKK
NA21767	MKK

Sample ID	Population
NA19735	MXL
NA19649	MXL
NA19669	MXL
NA19670	MXL
NA20502	TSI

Sample ID	Population
NA20509	TSI
NA20510	TSI
NA20511	TSI
NA19238	YRI
NA19239	YRI
NA19240*	YRI
NA18501	YRI
NA18502	YRI
NA12878*	CEU
NA12891	CEU
NA12892	CEU
NA06985	CEU
NA06994	CEU

Supplementary Table 2

D statistic for historic gene flow from CHB to GIH

CEU	GIH	CHB	YRI	# sites	D statistic	-log
A	B	B	A	253926	0.0729	> 549
B	A	B	A	219404		
CHB	GIH	CEU	YRI	# sites		
A	B	B	A	291871	0.1417	> 2240
B	A	B	A	219404		
CEU	GIH	MXL (only Nat. Am.)	YRI	# sites		
A	B	B	A	159754	0.0454	> 138
B	A	B	A	145869		
MXL (only Nat. Am.)	GIH	CEU	YRI	# sites		
A	B	B	A	194481	0.1428	> 1514
B	A	B	A	145869		

Supplementary Table 3

Sample Population Statistics

Population	# called sites	# segregating sites	# unphased	% unphased	Theta
CEU	2,132,078,340	3,973,757	124,972	3.14%	7.19E-04
CHB	2,132,530,333	3,734,161	148,046	3.96%	6.75E-04
GIH	2,137,960,203	4,180,134	279,164	6.68%	7.54E-04
JPT	2,128,798,051	3,689,030	106,160	2.88%	6.68E-04
LWK	2,110,232,589	5,498,240	225,690	4.10%	1.00E-03
MKK	2,132,615,842	5,199,644	334,893	6.44%	9.40E-04
MXL	2,113,376,228	3,958,912	131,396	3.32%	7.22E-04
TSI	2,133,276,458	4,031,944	124,492	3.09%	7.29E-04
YRI	2,133,075,233	5,646,792	197,841	3.50%	1.02E-03

Supplementary Table 4

Inbreeding and cryptic relatedness

Population	heterozygosity within samples π	heterozygosity across samples π	π_c
CEU	7.40E-04	7.39E-04	0.998
CHB	6.98E-04	6.96E-04	0.998
GIH	7.65E-04	7.68E-04	1.004
JPT	6.97E-04	6.91E-04	0.992
LWK	9.58E-04	9.57E-04	1.000
MKK all	9.60E-04	9.22E-04	0.960
MKK (only NA21732 and NA21737)	9.56E-04	7.26E-04	0.760
MXL	7.14E-04	7.36E-04	1.030
TSI	7.42E-04	7.45E-04	1.003
YRI	9.85E-04	9.83E-04	0.998

Supplementary Note

The Multiple Sequentially Markovian Coalescent (MSMC)

Stephan Schiffels and Richard Durbin
Wellcome Trust Sanger Institute

1. MSMC Continuous Time Transition Probability

Each state of MSMC is defined by a triple (i, j, t) with $i < j$, where i and j denote the indices of the two samples that coalesce first, and t is the time of first coalescence. The condition $i < j$ comes from the fact that we consider all *unordered* pairs of two individuals, of which there are $\binom{M}{2}$ in total, where M is the number of haplotypes. We consider the conditional transition probability to switch from the state (k, l, s) to (i, j, t), given a recombination event at time $u < s$ in leaf-branch m.

We distinguish between three types of transitions, determined by how the first coalescence time changes or not: $t < s$, $t = s$, and $t > s$. The three cases are detailed in the following.

■ Definitions

We denote the scaled rate of coalescence between branch i and j over time by $\lambda^{i,j}(t)$ and always assume symmetric rates $\lambda^{i,j}(t) = \lambda^{j,i}(t)$.We define the following convenient marginal rates: the total coalescence rate of branch i to all branches (including itself):

$$\Lambda^i(t) = \sum_{j=1}^{M} \lambda^{i,j}(t),$$

and the overall total coalescence rate of *any* two branches:

$$\Lambda(t) = \sum_{i=1}^{M-1} \sum_{j=i+1}^{M} \lambda^{i,j}(t).$$

We also define these exponentiated integrals:

$$L^m(t_1; t_2) = \exp\left(-\int_{t_1}^{t_2} \Lambda^m(\nu)\, d\nu\right)$$

and

$$L(t_1; t_2) = \exp\left(-\int_{t_1}^{t_2} \Lambda(\nu)\, d\nu\right)$$

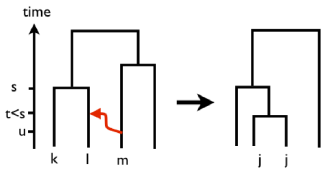
■ Equilibrium probability

The equilibrium probability is then given as

$$q_0(t, i, j) = \lambda^{i,j}(t)\, L(0; t),$$

which is normalized, as shown in section 11.

■ Transitions with $t < s$

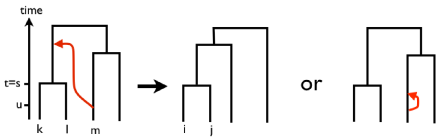


As shown in the picture, for this transition to occur we need to have $m \in \{i, j\}$ and the resulting floating branch coalescence needs to take place between i and j at time t and it must not coalesce with any other branch, including itself before t. This can be summarized as:

Nature Genetics: doi:10.1038/ng.3015

$$q(i,\,j,\,t\,|\,k,\,l,\,s,\,u,\,m)_{t<s} = \begin{cases} \lambda^{i,j}(t)\,L^m(u;\,t) & \text{if} \quad m \in \{i,\,j\} \quad \text{and} \quad u < t. \\ 0 & \text{else} \end{cases}$$

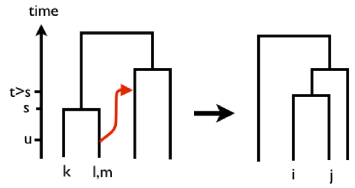
■ Transitions with $t = s$



For this to happen, one of two things must occur: Either, any branch with $m \notin \{k,\,l\}$ does coalesce later than t, or second, any floating branch self-coalesces before time t (see Figure). This results in the following terms:

$$q(i,\,j,\,t\,|\,k,\,l,\,s,\,u,\,m)_{t=s} = \int_u^t \lambda^{m,m}(\kappa)\,L^m(u;\,\kappa)\,d\kappa + \begin{cases} L^m(u;\,t) & \text{if} \quad m \neq k \quad \text{and} \quad m \neq l \\ 0 & \text{else} \end{cases}$$

■ Transitions with $t > s$



Here we require $m \in \{k,\,l\}$, i.e. the recombination needs to break the existing pair of first coalescence. A floating branch is then created and it must not coalesce back with any branch (including itself) before time s. It is not necessary that this particular floating branch will coalesce at time t, since any two other branches could coalesce then and hence make up the new pair $(i,\,j)$. We therefore require that no pair coalesces before time t, resulting in the term:

$$q(i,\,j,\,t\,|\,k,\,l,\,s,\,u,\,m)_{t>s} = \begin{cases} \lambda^{i,j}(t)\,L^m(u;\,s)\,L(s;\,t) & \text{if} \quad m \in \{k,\,l\} \\ 0 & \text{else.} \end{cases}$$

■ Full transition probability

The three cases above can be summarized as:

$$q(i,\,j,\,t\,|\,k,\,l,\,s,\,u,\,m) =$$

$$\delta(t-s)\,\delta_{i,k}\,\delta_{j,l}\left(\int_u^t \lambda^{m,m}(\kappa)\,L^m(u;\,\kappa)\,d\kappa + \left(1-\delta_{m,k}\right)\left(1-\delta_{m,l}\right)L^m(u;\,t)\right) + \begin{cases} \left(\delta_{m,i}+\delta_{m,j}\right)\lambda^{i,j}(t)\,L^m(u;\,t)\,\Theta(t-u) & \text{for} \quad t \le s \\ \left(\delta_{m,k}+\delta_{m,l}\right)\lambda^{i,j}(t)\,L^m(u;\,s)\,L(s;\,t) & \text{for} \quad t > s. \end{cases}$$

We show in section 11 that this conditional probability is normalized, i.e. that

$$\sum_{i<j}^{M}\int_0^\infty q(i,\,j,\,t\,|\,k,\,l,\,s,\,u,\,m)\,dt = 1.$$

for fixed k, l, s, u and m.

The full transition probability is then a sum over both cases with and without recombination, integrating over the parameters s and m with uniform probability density:

$$q(i,\,j,\,t\,|\,k,\,l,\,s) = e^{-M\,r\,s}\,\delta(t-s)\,\delta_{i,k}\,\delta_{j,l} + \left(1-e^{-M\,r\,s}\right)\frac{1}{s}\,\frac{1}{M}\int_0^s \sum_{m=1}^{M} q(i,\,j,\,t\,|\,k,\,l,\,s,\,u,\,m)\,du$$

which results in the expression:

$$q(i,\,j,\,t\,|\,k,\,l,\,s) = \delta(t-s)\,\delta_{i,k}\,\delta_{j,l}\,q_1(k,\,l,\,s) + q_2(i,\,j,\,t\,|\,k,\,l,\,s) \tag{1}$$

with

$$q_1(k,\,l,\,s) = e^{-M\,r\,t} + \left(1-e^{-M\,r\,t}\right)\frac{1}{t}\,\frac{1}{M}\int_0^t \left(\sum_{m=1}^{M}\int_u^t \lambda^{m,m}(\kappa)\,L^m(u;\,\kappa)\,d\kappa + \sum_{m\neq\{i,j\}}^{M} L^m(u;\,t)\right)du \tag{2}$$

and

$$q_2(i,\,j,\,t\,|\,k,\,l,\,s) = \left(1-e^{-M\,r\,s}\right)\frac{1}{s}\,\frac{1}{M}\,\lambda^{i,j}(t)\begin{cases} \int_0^t \sum_{m=\{i,j\}} L^m(u;\,t)\,du & \text{if} \quad t < s \\ L(s;\,t)\int_0^s \sum_{m=\{k,l\}} L^m(u;\,s)\,du & \text{if} \quad t > s. \end{cases} \tag{3}$$

Nature Genetics: doi:10.1038/ng.3015

Note that because of the normalization we always have

$$q_1(k, l, s) = 1 - \sum_{i<j}^{M} \int_0^\infty q_2(i, j, t \mid k, l, s)\, dt. \tag{4}$$

It will therefore suffice to constrain all further derivations to $q_2(i, j, t \mid k, l, s)$ and omit explicit expressions for $q_1(k, l, s)$.

Note that the special case $M = 2$ does *not* reduce to the original PSMC model as derived in [1]. Instead it differs from PSMC in a subtle way, because here we explicitly include the possibility that the floating branch coalesces back onto itself. This modification corresponds to the introduction of SMC' [2] after SMC [3].

■ Piecewise constant coalescence rate

We introduce discrete ordered time boundaries T_α for $\alpha = 0 \ldots n$ with $T_0 = 0$ and $T_n = \infty$. We define piecewise constant population sizes which correspond to piecewise constant coalescence rates:

$$\lambda^{i,j}(t) = \lambda_\alpha^{i,j} \text{ for } T_\alpha \le t < T_{\alpha+1}. \tag{5}$$

We define the discretized marginal rates

$$\Lambda_\alpha = \sum_{i=1}^{M-1} \sum_{j=i+1}^{M} \lambda_\alpha^{i,j}$$

and

$$\Lambda_\alpha^m = \sum_{i=1}^{M} \lambda_\alpha^{i,j}$$

In practice, we use quantile boundaries of the exponential distribution with mean $\binom{M}{2}^{-1}$ (in time units of $2 N_0$ generations):

$$T_\alpha = -\log\left(1 - \frac{\alpha}{n}\right) \Big/ \binom{M}{2}. \tag{6}$$

Let the next *lower* time boundary from t_1 be β, and the next *lower* time boundary from t_2 be α. We use the shortcut $\Delta_\alpha = T_{\alpha+1} - T_\alpha$. The exponentiated integrals $L(t_1; t_2)$ and $L^m(t_1; t_2)$ are then expressed as:

$$L(t_1; t_2)\,|_{\alpha \ne \beta} = \exp\left(-\left(T_{\beta+1} - t_1\right)\Lambda_\beta - \sum_{\kappa=\beta+1}^{\alpha-1} \Lambda_\kappa\, \Delta_\kappa - (t_2 - T_\alpha)\,\Lambda_\alpha\right).$$

$L(t_1; t_2)\,|_{\alpha=\beta} = \exp(-(t_2 - t_1)\,\Lambda_\alpha)$.

and similar formulas for $L^m(t_1; t_2)$, replacing Λ_α with Λ_α^m.

With these expressions, we can compute all the integrals in the transition probability:

$$\begin{aligned}
q_2(i, j, t; \alpha \mid k, l, s; \beta)\,|_{t<s} &= \left(1 - e^{-M r s}\right) \frac{1}{s} \frac{1}{M} \lambda_\alpha^{i,j}(t) \int_0^t \sum_{m=\{i,j\}} L^m(u; t)\, du \\
&= \left(1 - e^{-M r s}\right) \frac{1}{s} \frac{1}{M} \lambda_\alpha^{i,j} \left(\sum_{\gamma=0}^{\alpha-1} \int_{T_\gamma}^{T_{\gamma+1}} \sum_{m=\{i,j\}} L^m(u; t)\, du + \int_{T_\alpha}^{t} \sum_{m=\{i,j\}} L^m(u; t)\, du \right) \\
&= \left(1 - e^{-M r s}\right) \frac{1}{s} \frac{1}{M} \lambda_\alpha^{i,j} \left(\sum_{\gamma=0}^{\alpha-1} \sum_{m=\{i,j\}} L^m\left(T_{\gamma+1}; t\right) \int_{T_\gamma}^{T_{\gamma+1}} e^{-(T_{\gamma+1}-u)\,\Lambda_\gamma^m}\, du + \sum_{m=\{i,j\}} \int_{T_\alpha}^{t} e^{-(t-u)\,\Lambda_\alpha^m}\, du \right) \\
&= \left(1 - e^{-M r s}\right) \frac{1}{s} \frac{1}{M} \lambda_\alpha^{i,j} \left(\sum_{\gamma=0}^{\alpha-1} \sum_{m=\{i,j\}} \frac{1}{\Lambda_\gamma^m} \left(1 - e^{-\Delta_\gamma \Lambda_\gamma^m}\right) L^m\left(T_{\gamma+1}; t\right) + \sum_{m=\{i,j\}} \frac{1}{\Lambda_\alpha^m} \left(1 - e^{-(t-T_\alpha)\,\Lambda_\alpha^m}\right) \right),
\end{aligned} \tag{7}$$

and

$$\begin{aligned}
q_2(i, j, t; \alpha \mid k, l, s; \beta)\,|_{t>s} &= \left(1 - e^{-M r s}\right) \frac{1}{s} \frac{1}{M} \lambda^{i,j}(t)\, L(s; t) \int_0^s \sum_{m=\{k,l\}} L^m(u; s)\, du \\
&= \left(1 - e^{-M r s}\right) \frac{1}{s} \frac{1}{M} \lambda_\alpha^{i,j}\, L(s; t) \left(\sum_{\gamma=0}^{\beta-1} \sum_{m=\{k,l\}} \int_{T_\gamma}^{T_{\gamma+1}} L^m(u; s)\, du + \sum_{m=\{k,l\}} \int_{T_\beta}^{s} L^m(u; s)\, du \right)
\end{aligned}$$

Nature Genetics: doi:10.1038/ng.3015

$$= \left(1 - e^{-M r s}\right) \frac{1}{s} \frac{1}{M} \lambda_\alpha^{i,j} L(s;t) \left(\sum_{\gamma=0}^{\beta-1} \sum_{m=\{k,l\}} \int_{T_\gamma}^{T_{\gamma+1}} L^m(u;s)\, du + \sum_{m=\{k,l\}} \int_{T_\beta}^{s} L^m(u;s)\, du \right)$$

$$= \left(1 - e^{-M r s}\right) \frac{1}{s} \frac{1}{M} \lambda_\alpha^{i,j} L(s;t) \left(\sum_{\gamma=0}^{\beta-1} \sum_{m=\{k,l\}} L^m\left(T_{\gamma+1};s\right) \int_{T_\gamma}^{T_{\gamma+1}} e^{-(T_{\gamma+1}-u)\Lambda_\gamma^m}\, du + \sum_{m=\{k,l\}} \int_{T_\beta}^{s} e^{-(s-u)\Lambda_\alpha^m}\, du \right)$$

$$= \left(1 - e^{-M r s}\right) \frac{1}{s} \frac{1}{M} \lambda_\alpha^{i,j} L(s;t) \left(\sum_{\gamma=0}^{\beta-1} \sum_{m=\{k,l\}} \frac{1}{\Lambda_\gamma^m} \left(1 - e^{-\Delta_\gamma \Lambda_\gamma^m}\right) L^m\left(T_{\gamma+1};s\right) + \sum_{m=\{k,l\}} \frac{1}{\Lambda_\beta^m} \left(1 - e^{-(s-T_\beta)\Lambda_\beta^m}\right) \right).$$

2. MSMC Discrete Time Transition Probability

We first compute the total weight of the equilibrium probability over each time interval:

$$q_0(i,\, j,\, \alpha) = \int_{T_\alpha}^{T_{\alpha+1}} \lambda_\alpha^{i,j} L(0;t)\, dt = \lambda_\alpha^{i,j} L(0;T_\alpha) \int_{T_\alpha}^{T_{\alpha+1}} e^{-(t-T_\alpha)\Lambda_\alpha}\, dt = \frac{\lambda_\alpha^{i,j}}{\Lambda_\alpha} L(0;T_\alpha) \left(1 - e^{-\Delta_\alpha \Lambda_\alpha}\right)$$

For each time interval β we now compute the average coalescence time:

$$\begin{aligned} \left\langle t_\beta \right\rangle &= \frac{1}{q_0(\beta)} \int_{T_\beta}^{T_{\beta+1}} t\, q_0(t;\beta)\, dt = \frac{\Lambda_\beta}{L\left(0;T_\beta\right)\left(1 - e^{-\Delta_\beta \Lambda_\beta}\right)} \int_{T_\beta}^{T_{\beta+1}} t\, L(0;t)\, dt \\ &= \frac{\Lambda_\beta}{\left(1 - e^{-\Delta_\beta \lambda_\beta}\right)} \int_{T_\beta}^{T_{\beta+1}} t\; e^{-(t-T_\beta)\Lambda_\beta}\, dt \\ &= \frac{1}{\left(1 - e^{-\Delta_\beta \Lambda_\beta}\right)\Lambda_\beta} \left(1 + \Lambda_\beta T_\beta - e^{-\Lambda_\beta\, \Delta_\beta} \left(1 + \Lambda_\beta\; T_{\beta+1}\right)\right) \end{aligned} \tag{9}$$

Note that this expression for $\left\langle t_\beta \right\rangle$ has a numerical instability for $\Lambda_\beta \lesssim 10^{-3}$. We set the following asymptotic values:

$$\left\langle t_\beta \right\rangle = \begin{cases} \left(T_\beta + T_{\beta+1}\right)/2 & \text{for } \Lambda_\beta < 10^{-3} \text{ and } \beta + 1 < \infty \\ T_\beta + \left(\Lambda_\beta\right)^{-1} & \text{for } \Lambda_\beta < 10^{-3} \text{ and } \beta + 1 = \infty\,. \end{cases}$$

To get discrete transition probabilities, we integrate the transition probability for each time interval α through $[T_\alpha; T_{\alpha+1}]$, replacing the time s with $\left\langle t_\beta \right\rangle$:

$$q(i,\, j,\, \alpha \mid k,\, l,\, \beta) = \int_{T_\alpha}^{T_{\alpha+1}} q\left(i,\, j,\, t \mid k,\, l,\, \left\langle t_\beta \right\rangle\right) dt$$

- **First case: t<s**

$$q(i,\, j,\, \alpha \mid k,\, l,\, \beta)\,|_{\alpha<\beta}$$

$$= \int_{T_\alpha}^{T_{\alpha+1}} \left(1 - e^{-M r \left\langle t_\beta \right\rangle}\right) \frac{1}{\left\langle t_\beta \right\rangle} \frac{1}{M} \lambda_\alpha^{i,j} \left(\sum_{\gamma=0}^{\alpha-1} \sum_{m=\{i,j\}} \frac{1}{\Lambda_\gamma^m} \left(1 - e^{-\Delta_\gamma \Lambda_\gamma^m}\right) L^m\left(T_{\gamma+1};t\right) + \sum_{m=\{i,j\}} \frac{1}{\Lambda_\alpha^m} \left(1 - e^{-(t-T_\alpha)\Lambda_\alpha^m}\right) \right) dt$$

$$=$$

$$\left(1 - e^{-M r \left\langle t_\beta \right\rangle}\right) \frac{1}{\left\langle t_\beta \right\rangle} \frac{1}{M} \lambda_\alpha^{i,j} \left(\sum_{\gamma=0}^{\alpha-1} \sum_{m=\{i,j\}} \frac{1}{\Lambda_\gamma^m} \left(1 - e^{-\Delta_\gamma \Lambda_\gamma^m}\right) L^m\left(T_{\gamma+1};T_\alpha\right) \int_{T_\alpha}^{T_{\alpha+1}} e^{-(t-T_\alpha)\Lambda_\alpha^m}\, dt + \sum_{m=\{i,j\}} \frac{1}{\Lambda_\alpha^m} \left(\Delta_\alpha - \int_{T_\alpha}^{T_{\alpha+1}} e^{-(t-T_\alpha)\Lambda_\alpha^m}\, dt \right) \right)$$

$$= \left(1 - e^{-M r \left\langle t_\beta \right\rangle}\right) \frac{1}{\left\langle t_\beta \right\rangle} \frac{1}{M} \lambda_\alpha^{i,j} \left(\sum_{\gamma=0}^{\alpha-1} \sum_{m=\{i,j\}} \frac{1}{\Lambda_\gamma^m} \left(1 - e^{-\Delta_\gamma \Lambda_\gamma^m}\right) L^m\left(T_{\gamma+1};T_\alpha\right) \frac{1}{\Lambda_\alpha^m} \left(1 - e^{-\Delta_\alpha \Lambda_\alpha^m}\right) + \sum_{m=\{i,j\}} \frac{1}{\Lambda_\alpha^m} \left(\Delta_\alpha - \frac{1}{\Lambda_\alpha^m} \left(1 - e^{-\Delta_\alpha \Lambda_\alpha^m}\right) \right) \right)$$

- **Second case: t>s**

$$q(i,\, j,\, \alpha \mid k,\, l,\, \beta)\,|_{\alpha>\beta}$$

$$= \int_{T_\alpha}^{T_{\alpha+1}} \left(1 - e^{-M r \left\langle t_\beta \right\rangle}\right) \frac{1}{\left\langle t_\beta \right\rangle} \frac{1}{M} \lambda_\alpha^{i,j} L\left(\left\langle t_\beta \right\rangle;t\right) \left(\sum_{\gamma=0}^{\beta-1} \sum_{m=\{k,l\}} \frac{1}{\Lambda_\gamma^m} \left(1 - e^{-\Delta_\gamma \Lambda_\gamma^m}\right) L^m\left(T_{\gamma+1};\left\langle t_\beta \right\rangle\right) + \sum_{m=\{k,l\}} \frac{1}{\Lambda_\beta^m} \left(1 - e^{-(\left\langle t_\beta \right\rangle - T_\beta)\Lambda_\beta^m}\right) \right) dt$$

$$=$$

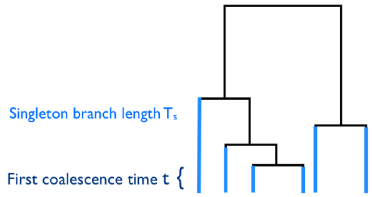
$$\left(1-e^{-M\,r\,\langle t_\beta\rangle}\right)\frac{1}{\langle t_\beta\rangle}\,\frac{1}{M}\,\lambda_\alpha^{i,j}\,L\left(\langle t_\beta\rangle;T_\alpha\right)\frac{1}{\Lambda_\alpha}\left(1-e^{-\Delta_\alpha\,\Lambda_\alpha}\right)\left(\sum_{\gamma=0}^{\beta-1}\sum_{m=\{k,l\}}\frac{1}{\Lambda_\gamma^m}\left(1-e^{-\Delta_\gamma\,\Lambda_\gamma^m}\right)L^m\left(T_{\gamma+1};\langle t_\beta\rangle\right)+\sum_{m=\{k,l\}}\frac{1}{\Lambda_\beta^m}\left(1-e^{-(\langle t_\beta\rangle-T_\beta)\,\Lambda_\beta^m}\right)\right)$$

3. MSMC Emission Probability

An observation consists of an allele string O with alleles denoted as $O^i \in \{1, 0\}$. Real allele strings with letters $\{A, C, T, G\}$ can be normalized to $\{1, 0\}$, approximating rare multi-allelic SNPs by merging of two arbitrary alleles.

■ Singleton branch length

To define the emission probabilities, we need to define the singleton branch length, T_s, a property of the tree that can not be expressed through the hidden state alone. As shown in the following picture, the singleton branch length of the tree is the sum of all those branches which would give rise to singleton mutations, should a mutation occur on that branch. Note that this definition is defined for *minor allele counts*. This means that T_s includes those branches of the tree which result in a derived allele count of $M - 1$ alleles. The singleton branch length can be estimated locally from real data in an approximately independent analysis from MSMC itself, see section 6. For the following, we assume that we have an estimate of T_s at every position along the samples.



Note that the singleton branch length is not strictly independent from the first coalescence time. For MSMC we ignore this interdependency, as justified from simulations, see main text and *Supplementary Figure 3*. Instead, we treat the local estimate T_s as a *soft* constraint. This means that in the emission probabilities below we always consider $\max(T_s, M\,t)$ when we write T_s. This means that we always set $T_s = M\,t$ when we compute a term with $M\,t > T_s$. This is an evident requirement, as seen in the picture above: The sum of all blue branches must be at least as long as M times the first coalescence time t.

■ Emission categories

For more than two haplotypes, i.e. $M > 2$, the possible observations given state (t, i, j) fall into the following categories, corresponding to the list in *Online Methods*:

1. All alleles are the same, no mutation occured within the singleton branches nor in the rest of the tree, which happens with probability

   $$e(O \mid t, i, j) = 1 - \mu\,T_s.$$

   In this formula, we ignore the rest of the tree outside of T_s: since we only know the average branch length of the tree, this would result simply in a constant factor, independent of the state (t, i, j) at that position, and hence does not affect the normalized posterior probability across states.

2. The two alleles in samples i and j differ, $O^i \neq O^j$, and this is the only difference between any two samples. In this case, a mutation occurred in one of the leaf branches i or j, resulting in the emission probability

   $$e(O \mid t, i, j) = \mu\,t$$

3. Both leaves i and j have the same allele, $O^i = O^j$, and exactly one of the other alleles differs from O^i and O^j. In that case, a mutation occurred somewhere within the singleton branch length T_s of the tree, but outside the two leaf branches leading to i and j. Because this includes mutations with resulting allele count $M - 1$, as discussed above, the emission probability consists of two terms. First, the probability that a mutation occurs within T_s but somewhere higher up the tree than time t is $\mu(T_s - M\,t)$. To turn this into the probability to observe a *specific* singleton, this needs to be normalized by the number of possible observations with minor allele count 1 outside the pair of first coalescence, which is $M - 2$. Second, a mutation may have occured more recently than t in the leaf branch carrying the singleton allele, which happens with probability $\mu\,t$:

   $$e(O \mid t, i, j) = \frac{\mu\,(T_s - M\,t)}{M-2} + \mu\,t.$$

4. A higher frequency variant with minor allele count larger than 1 occured, but we have $O^i = O^j$. This means that no mutation occurred anywhere within T_s so this results in the same probability as for category 1:

   $$e(O \mid t, i, j) = 1 - \mu\,T_s.$$

   This ignores any terms from mutations higher up the tree, with the same argument as in category 1 above. For the same reason we do not normalize this term by the number of possible observations, as is necessary in the first term of category 2.

5. A higher frequency variant with minor allele count larger than 1 occured, but we have $O^i \neq O^j$. This requires at least two mutations, one within the pair of first coalescence and one outside, which we assume to occur with zero probability:

$$e(O \mid t, i, j) = 0$$

6. Some samples have no alleles called at a position. As in many other HMM implementations, we model missing data by setting the emission probability to 1 for all states:

$$e(O \mid t, i, j) = 1.$$

4. MSMC Hidden Markov Model

We can now define a Hidden Markov Model, using the above defined transition and emission probabilities. For a given sequence of length L, we define the observations as $O_1 \ldots O_L$.

We define a forward variable $f_1(\alpha, i, j) \ldots f_L(\alpha, i, j)$ by the recursion relation:

- Initialisation

  $f_1(\alpha, i, j) = q_0(\alpha)\, e(O_1 \mid \alpha, i, j)$

- Recursion

  $$f_n(\alpha, i, j) = e(O_n \mid \alpha, i, j) \sum_{\beta,\{k<l\}} q(\alpha, i, j \mid \beta, k, l)\, f_{n-1}(\beta, k, l) \qquad \text{for} \quad 2 < n \le L,$$

where the notation $\{k < l\}$ means that the sum over those indices is performed only through the set of unordered pairs, i.e. with the constraint $k < l$ or $i < j$.

The backwards variable $b_1(\alpha, i, j) \ldots b_L(\alpha, i, j)$ is defined analogously:

- Initialisation

  $b_L(\alpha, i, j) = 1$

- Recursion

  $$b_n(\beta, k, l) = \sum_{\alpha,\{i<j\}} e(O_{n+1} \mid \alpha, i, j)\, q(\alpha, i, j \mid \beta, k, l)\, b_{n+1}(\alpha, i, j) \qquad \text{for} \quad 1 \le n < L.$$

Naively implemented, the transition complexity of the forward- and backward-recursion scales quadratic in the number of time segments, and it scales as M^4 with the number of haplotypes M. As we show in the following, this complexity can be reduced to the second power of M by exploiting symmetries in the transition matrix.

■ Exploiting the transition symmetries for optimization

We normally do not expect all $\binom{M}{2}$ different coalescence rates (all pairs of indices i, j) per time interval to be different. Instead, we normally consider the case in which our samples come in groups, forming subpopulations.

More formally we define index sets $I_1, I_2 \ldots I_{n_I}$, such that each pair index i, j is in exactly one index set. For example, if we consider $M = 4$ haplotypes from only one population we have one index set $I_1 = \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\}$ in which all pair indices are. If the haplotypes come from two different subpopulations, with the first two haplotypes from subpopulation 1 and the other two from subpopulation 2, we have three index sets:

$I_1 = \{(1, 2)\}$ (first coalescence is within population 1)

$I_2 = \{(1, 3), (1, 4), (2, 3), (2, 4)$ (first coalescence is across populations)

$I_3 = \{(3, 4)\}$ (first coalescence is within population 2)

We define the *marginal index* $\varphi_{i,j}$ to yield the index set in which a given pair i, j resides. We set equal coalescence rates for all pairs (i, j) within a given index set, and denote this coalescence rate for index set u by λ_α^u. We then have

$\lambda_\alpha^{i,j} = \lambda_\alpha^u$ with $u = \varphi_{i,j}$

Because the coalescence rates are symmetric within each index set, so are the transition probabilities:

$$q(\alpha, i, j \mid \beta, k, l) = q_1\big(\alpha, \varphi_{i,j}\big)\, \delta_{\alpha,\beta}\, \delta_{i,k}\, \delta_{j,l} + q_2\big(\alpha, \varphi_{i,j} \,\big|\, \beta, \varphi_{k,l}\big)$$

We can exploit this fact to optimize the transition process in the forward- and backward-algorithm of the Hidden Markov Model. Without optimization, we had the forward recursion:

Nature Genetics: doi:10.1038/ng.3015

$$f_n(\alpha, i, j) = e(O_n \mid \alpha, i, j) \sum_{\beta,k<l} f_{n-1}(\beta, k, l)\, q(\alpha, i, j; \beta, k, l),$$

which scales as M^4 (needs to be evaluated for all α, i, j, each of which has a sum over β, k, l).

To optimize, we define the following marginal forward variable:

$$F_n(\alpha, u) = \sum_{i,j \in I_u} f_n(\alpha, i, j)$$

With this definition the recursion reads

$$f_n(\alpha, i, j) = e(O_n \mid \alpha, i, j) \left(f_{n-1}(\alpha, i, j)\, q_1(\alpha, \varphi_{i,j}) + \sum_{\beta,v} F_{n-1}(\beta, v)\, q_2(\alpha, \varphi_{i,j}; \beta, v) \right),$$

which scales as ${n_i}^2$, where n_i is the number of index sets (which is rougly quadratic in the number of subpopulations) since no sum across pair tuples are involved.

Similarly, for the unoptimized version of the backwards recursion we have

$$b_n(\beta, k, l) = \sum_{\alpha,i<j} e(O_{n+1} \mid \alpha, i, j)\, b_{n+1}(\alpha, i, j)\, q(\alpha, i, j; \beta, k, l).$$

We define the marginal backward variable:

$$\tilde{B}_n(\alpha, u) = \sum_{(i,j) \in I_u} b_n(\alpha, i, j)\, e(O_n \mid \alpha, i, j).$$

With this we get

$$b_n(\beta, k, l) = e(O_{n+1} \mid \beta, k, l)\, b_{n+1}(\beta, k, l)\, q_1(\beta, \varphi_{k,l}) + \sum_{\alpha,u} \tilde{B}_{n+1}(\alpha, u)\, q_2(\alpha, u; \beta, \varphi_{k,l}),$$

■ Optimization for SNP-free regions

We now consider the interval $[n_1, n_2]$ with length $n_2 - n_1 = l$ in which sites have no SNPs. We then have trivial emission probabilities of

$$e(O_n \mid \alpha, i, j) = e_0(\alpha) \quad \text{for} \quad n \in [n_1, n_2].$$

■ Forward Recursion

We can then write the forward recursion through the entire region as

$$f_n(\alpha, i, j) = \sum_{\beta,k<l} g^l(\alpha, i, j \mid \beta, k, l)\, f_{n-l}(\beta, k, l)$$

with the matrix power g^l, defined recursively:

$$g^1(\alpha, i, j \mid \beta, k, l) = q(\alpha, i, j \mid \beta, k, l)\, e_0(\alpha)$$

$$g^l(\alpha, i, j \mid \beta, k, l) = e_0(\alpha) \sum_{\gamma,m<n} g^{l-1}(\gamma, m, n \mid \beta, k, l)\, q(\alpha, i, j \mid \gamma, m, n).$$

We assume that we can write the propagation matrix in the form

$$g^l(\alpha, i, j \mid \beta, k, l) = \delta_{\alpha,\beta}\, \delta_{i,k}\, \delta_{j,l}\, g_1^l(\alpha, \varphi_{i,j}) + g_2^l(\alpha, \varphi_{i,j} \mid \beta, \varphi_{k,l}),$$

with $g_1^1(\alpha, u) = q_1(\alpha, u)\, e_0(\alpha)$ and $g_2^1(\alpha, u \mid \beta, v) = q_2(\alpha, u \mid \beta, v)\, e_0(\alpha)$.

We can hence write the recursion equation using this form and the similar form for the transition matrix:

$$\begin{aligned}
&g^l(\alpha, i, j \mid \beta, k, l) \\
&= e_0(\alpha) \sum_{\gamma,m<n} \left(\delta_{\gamma,\beta}\, \delta_{m,k}\, \delta_{n,l}\, g_1^{l-1}(\gamma, \varphi_{m,n}) + g_2^{l-1}(\gamma, \varphi_{m,n} \mid \beta, \varphi_{k,l})\right) \left(\delta_{\alpha,\gamma}\, \delta_{i,m}\, \delta_{j,n}\, q_1(\alpha, \varphi_{i,j}) + q_2(\alpha, \varphi_{i,j} \mid \gamma, \varphi_{m,n})\right) \\
&= e_0(\alpha) \sum_{\gamma,m<n} \left(\delta_{\gamma,\beta}\, \delta_{m,k}\, \delta_{n,l}\, g_1^{l-1}(\gamma, \varphi_{m,n})\, \delta_{\alpha,\gamma}\, \delta_{i,m}\, \delta_{j,n}\, q_1(\alpha, \varphi_{i,j}) + g_2^{l-1}(\gamma, \varphi_{m,n} \mid \beta, \varphi_{k,l})\, \delta_{\alpha,\gamma}\, \delta_{i,m}\, \delta_{j,n}\, q_1(\alpha, \varphi_{i,j}) + \right. \\
&\qquad \left. \delta_{\gamma,\beta}\, \delta_{m,k}\, \delta_{n,l}\, g_1^{l-1}(\gamma, \varphi_{m,n})\, q_2(\alpha, \varphi_{i,j} \mid \gamma, \varphi_{m,n}) + g_2^{l-1}(\gamma, \varphi_{m,n} \mid \beta, \varphi_{k,l})\, q_2(\alpha, \varphi_{i,j} \mid \gamma, \varphi_{m,n})\right)
\end{aligned}$$

Nature Genetics: doi:10.1038/ng.3015

$$= e_0(\alpha)\left(\delta_{\alpha,\beta}\,\delta_{i,k}\,\delta_{j,l}\,g_1^{l-1}(\alpha,\varphi_{i,j})\,q_1(\alpha,\varphi_{i,j}) + g_2^{l-1}(\alpha,\varphi_{k,l}\mid\beta,\varphi_{i,j})\,q_1(\alpha,\varphi_{i,j}) + \right.$$

$$\left. g_1^{l-1}(\beta,\varphi_{k,l})\,q_2(\alpha,\varphi_{i,j}\mid\beta,\varphi_{k,l}) + \sum_{\gamma,w}\|I_w\|\,g_2^{l-1}(\gamma,w\mid\beta,\varphi_{k,l})\,q_2(\alpha,\varphi_{i,j}\mid\gamma,w)\right)$$

where $\|I_u\|$ is the number of pairs in the set I_u.

We read off the two recursion relations:

$$g_1^l(\alpha,u) = e_0(\alpha)\,g_1^{l-1}(\alpha,u)\,q_1(\alpha,u),$$

and $$g_2^l(\alpha,u\mid\beta,v) = e_0(\alpha)\left(g_2^{l-1}(\alpha,u\mid\beta,v)\,q_1(\alpha,u) + g_1^{l-1}(\beta,v)\,q_2(\alpha,u\mid\beta,v) + \sum_{\gamma,w}\|I_w\|\,g_2^{l-1}(\gamma,w\mid\beta,v)\,q_2(\alpha,u\mid\gamma,w)\right)$$

Given the form of the propagation matrix g^l, we can make use of the above defined marginal forward vector $F_n(\alpha)$:

$$f_n(\alpha,i,j) = f_{n-l}(\alpha,i,j)\,g_1^l(\alpha,\varphi_{i,j}) + \sum_{\beta,v} g_2^l(\alpha,\varphi_{i,j}\mid\beta,v)\,F_{n-l}(\beta,v)$$

■ Backward Recursion

Similarly, for the backward recursion, we have

$$b_n(\beta,k,l) = \sum_{\alpha,i<j} h^l(\alpha,i,j\mid\beta,k,l)\,b_{n+l}(a,i,j)$$

with

$$h^1(\alpha,i,j\mid\beta,k,l) = g^1 = q(\alpha,i,j\mid\beta,k,l)\,e_0(\alpha)$$

$$h^l(\alpha,i,j\mid\beta,k,l) = \sum_{\gamma,m<n} h^{l-1}(\alpha,i,j\mid\gamma,m,n)\,q(\gamma,m,n\mid\beta,k,l)\,e_0(\gamma)$$

We try to write h^l in the form

$$h^l(\alpha,i,j\mid\beta,k,l) = \delta_{\alpha,\beta}\,\delta_{i,k}\,\delta_{j,l}\,h_1^l(\alpha,\varphi_{i,j}) + h_2^l(\alpha,\varphi_{i,j}\mid\beta,\varphi_{k,l}),$$

with $h_1^1(\alpha,u) = q_1(\alpha,u)\,e_0(\alpha)$ and $h_2^1(\alpha,u\mid\beta,v) = q_2(\alpha,u\mid\beta,v)\,e_0(\alpha)$.

For the recursion it then follows:

$$h^l(\alpha,i,j\mid\beta,k,l)$$

$$= \sum_{\gamma,m<n} e_0(\gamma)\left(\delta_{\alpha,\gamma}\,\delta_{i,m}\,\delta_{j,n}\,h_1^{l-1}(\alpha,\varphi_{i,j}) + h_2^{l-1}(\alpha,\varphi_{i,j}\mid\gamma,\varphi_{m,n})\right)\left(\delta_{\gamma,\beta}\,\delta_{m,k}\,\delta_{n,l}\,q_1(\gamma,\varphi_{m,n}) + q_2(\gamma,\varphi_{m,n}\mid\beta,\varphi_{k,l})\right)$$

$$= \sum_{\gamma,m<n} e_0(\gamma)\left(\delta_{\alpha,\gamma}\,\delta_{i,m}\,\delta_{j,n}\,h_1^{l-1}(\alpha,\varphi_{i,j})\,\delta_{\gamma,\beta}\,\delta_{m,k}\,\delta_{n,l}\,q_1(\gamma,\varphi_{m,n}) + h_2^{l-1}(\alpha,\varphi_{i,j}\mid\gamma,\varphi_{m,n})\,\delta_{\gamma,\beta}\,\delta_{m,k}\,\delta_{n,l}\,q_1(\gamma,\varphi_{m,n}) + \right.$$

$$\left.\delta_{\alpha,\gamma}\,\delta_{i,m}\,\delta_{j,n}\,h_1^{l-1}(\alpha,\varphi_{i,j})\,q_2(\gamma,\varphi_{m,n}\mid\beta,\varphi_{k,l}) + h_2^{l-1}(\alpha,\varphi_{i,j}\mid\gamma,\varphi_{m,n})\,q_2(\gamma,\varphi_{m,n}\mid\beta,\varphi_{k,l})\right)$$

$$= \delta_{\alpha,\beta}\,\delta_{i,k}\,\delta_{j,l}\,h_1^{l-1}(\alpha,\varphi_{i,j})\,q_1(\alpha,\varphi_{i,j})\,e_0(\alpha) + h_2^{l-1}(\alpha,\varphi_{i,j}\mid\beta,\varphi_{k,l})\,q_1(\beta,\varphi_{k,l})\,e_0(\beta) +$$

$$e_0(\alpha,\varphi_{i,j})\,h_1^{l-1}(\alpha,\varphi_{i,j})\,q_2(\alpha,\varphi_{i,j}\mid\beta,\varphi_{k,l}) + \sum_{\gamma,w}\|I_w\|\,h_2^{l-1}(\alpha,\varphi_{i,j}\mid\gamma,w)\,q_2(\gamma,w\mid\beta,\varphi_{k,l})\,e_0(\gamma),$$

from which we read off the two recursion relations:

$$h_1^l(\alpha,u) = e_0(\alpha)\,h_1^{l-1}(\alpha,u)\,q_1(\alpha,u), \qquad \text{and} \qquad h_2^l(\alpha,u\mid\beta,v) =$$

$$h_2^{l-1}(\alpha,u\mid\beta,v)\,q_1(\beta,v)\,e_0(\beta) + e_0(\alpha)\,h_1^{l-1}(\alpha,u)\,q_2(\alpha,u\mid\beta,v) + \sum_{\gamma,w}\|I_w\|\,h_2^{l-1}(\alpha,u\mid\gamma,w)\,q_2(\gamma,w\mid\beta,v)\,e_0(\gamma)$$

We can again use a marginalized version of the backward variable for speed up:

$$b_n(\beta,k,l) = h_1^l(\beta,\varphi_{k,l})\,b_{n+l}(\beta,k,l) + \sum_{\alpha,u} h_2^l(\alpha,u\mid\beta,\varphi_{k,l})\,B_{n+l}(a,u)$$

with

$$B_n(\alpha, u) = \sum_{k,l \in I_u} b_n(\alpha, k, l),$$

which is different from $\tilde{B}_n(\alpha, u)$, as it does not use the emission probability.

■ Missing data

For contiguous segments of missing data, we can apply the same recipe as for homozygous regions, simply replacing the emission probability $e_0(\alpha, i, j)$ with 1 for all states. All the matrix powers as described above can then be computed with the same recursion formulas.

5. Parameter Re-estimation

The objective function in the Baum-Welch algorithm is defined as

$$F(\theta, \bar{\theta}) = \sum_{\alpha,\beta,\{i<j\},\{k<l\}} \log\left(q(\alpha, i, j \mid \beta, k, l; \bar{\theta})\right) \Xi(\alpha, i, j \mid \beta, k, l; O, \theta).$$

with

$$\Xi(\alpha, i, j \mid \beta, k, l; O, \theta) = \sum_{n=2}^{L} \xi_n(\alpha, i, j \mid \beta, k, l; O, \theta)$$

and

$$\xi_n(\alpha, i, j \mid \beta, k, l; O, \theta) = f_n(\beta, k, l)\, q(\alpha, i, j \mid \beta, k, l; \theta)\, e(O_{n+1} \mid \alpha, i, j)\, b_{n+1}(\alpha, i, j)$$

Here, θ denotes the set of parameters to be learned, which are all coalescence rates $\lambda_\alpha^{i,j}$ and the recombination rate ρ.

The re-estimated parameters are obtained by optimizing over $\bar{\theta}$:

$$\theta' = \underset{\bar{\theta}}{\operatorname{argmax}}\, F(\theta, \bar{\theta}).$$

We can write the objective function as

$$F(\theta, \bar{\theta}) = \sum_{\alpha,\beta,\{i<j\},\{k<l\}} \log\left(q_1(\alpha; \bar{\theta})\, \delta_{\alpha,\beta}\, \delta_{i,k}\, \delta_{j,l} + q_2(\alpha \mid \beta; \bar{\theta})\right) \Xi(\alpha, i, j \mid \beta, k, l; O, \theta).$$

We now separate out the diagonal part of the sum:

$$F(\theta, \bar{\theta}) = \sum_{\alpha} \log\left(q_1(\alpha; \bar{\theta}) + q_2(\alpha \mid \alpha; \bar{\theta})\right) \Xi_{\text{diag}}(\alpha; O, \theta) +$$

$$\sum_{\alpha,\beta} \log\left(q_2(\alpha \mid \beta; \bar{\theta})\right) \Xi_{\text{off}}(\alpha \mid \beta; O, \theta)$$

and define the two separate sums:

$$\Xi_{\text{diag}}(\alpha \mid O, \theta) = \sum_{\{i<j\}} \Xi(\alpha, i, j \mid \alpha, i, j; O, \theta)$$

and

$$\Xi_{\text{off}}(\alpha \mid \beta; O, \theta) = \sum_{\{i<j\},\{k<l\}} \Xi(\alpha, i, j \mid \beta, k, l; O, \theta)\left(1 - \delta_{\alpha,\beta}\, \delta_{i,k}\, \delta_{j,l}\right).$$

Now the objective function reads:

$$F(\theta, \bar{\theta}) = \sum_{\alpha} \log\left(q_1(\alpha; \bar{\theta}) + q_2(\alpha \mid \alpha; \bar{\theta})\right) \Xi_{\text{diag}}(\alpha \mid O, \theta) + \sum_{\alpha \neq \beta} \log\left(q_2(\alpha \mid \beta; \bar{\theta})\right) \Xi_{\text{off}}(\alpha \mid \beta; O, \theta)$$

The advantage with this form is that Ξ_{off} can be efficiently computed using the marginal forward- and backward-variables from above. We first write

$$\Xi_{\text{off}}(\alpha \mid \beta) = \sum_{n=1}^{L-1} \xi^n_{\text{off}}(\alpha \mid \beta) \quad \text{and} \quad \Xi_{\text{diag}}(\alpha) = \sum_{n=1}^{L-1} \xi^n_{\text{diag}}(\alpha)$$

with

Nature Genetics: doi:10.1038/ng.3015

$$\xi^n_{\text{off}}(\alpha \mid \beta) = \sum_{\{i<j\},\{k<l\}} f_n(\beta, k, l)\, q(\alpha, i, j \mid \beta, k, l)\, e(O_{n+1} \mid \alpha, i, j)\, b_{n+1}(\alpha, i, j)\left(1 - \delta_{\alpha,\beta}\,\delta_{i,k}\,\delta_{j,l}\right)$$

$$= \sum_{\{i<j\},\{k<l\}} f_n(\beta, k, l)\, q_2(\alpha \mid \beta)\, e(O_{n+1} \mid \alpha, i, j)\, b_{n+1}(\alpha, i, j)\left(1 - \delta_{\alpha,\beta}\,\delta_{i,k}\,\delta_{j,l}\right)$$

$$= F_n(\beta)\, q_2(\alpha \mid \beta)\, \tilde{B}_n(\alpha) - \delta_{\alpha,\beta} \sum_{\{i<j\}} f_n(\alpha, i, j)\, q_2(\alpha \mid \alpha)\, e(O_{n+1} \mid \alpha, i, j)\, b_{n+1}(\alpha, i, j)$$

and

$$\xi^n_{\text{diag}}(\alpha) = \sum_{\{i<j\},\{k<l\}} f_n(\beta, k, l)\, q(\alpha, i, j \mid \beta, k, l)\, e(O_{n+1} \mid \alpha, i, j)\, b_{n+1}(\alpha, i, j)\, \delta_{\alpha,\beta}\,\delta_{i,k}\,\delta_{j,l}$$

$$= \sum_{\{i<j\}} f_n(\alpha, i, j)\,(q_1(\alpha) + q_2(\alpha \mid \alpha))\, e(O_{n+1} \mid \alpha, i, j)\, b_{n+1}(\alpha, i, j)$$

■ Constrained optimization

The Baum-Welch algorithm in principle lets us re-estimate all coalescence rates and recombination rates. However, in practice we constrain the number of free parameters:

Parameter Constraints:

- For fixed α and u, all coalescence rates $\lambda_\alpha^{i,j}$ with $\varphi_{i,j} = u$ need to be equal, where $\varphi_{i,j}$ denotes the index sub-group as defined in the beginning of chapter 4.
- Parameters for neighbouring time intervals may be constrained to be equal, according to a defined pattern. For human data and one single population, we usually use 40 time intervals, with quantile boundaries as in equation 6, and the most recent 10 intervals left as freel parameters, and the other 30 intervals joined to pairs of 2, leading to 25 free coalescence rate parameters. For samples coming from two subpopulations, we usually use 30 time intervals, with the first 8 being independent, and the latter 22 being joined to pairs of two, leading to 19 free parameters within each population and 19 additional free parameters for the coalescence rate across populations.
- All coalescence rates across populations must be smaller or equal to the mean rate within the two populations, such that the relative cross coalescence rate in interval α, $\gamma_\alpha = 2\,\lambda_\alpha^{12}/(\lambda_\alpha^{11} + \lambda_\alpha^{22})$ remains between 0 and 1.
- All coalescence rates must be positive.
- The recombination rate must be positive.

These constraints can be incorporated into numerical optimization techniques using logarithmic and tangential variable transforms (see [4]).

6. Estimating the singleton branch length

The local singleton branch length of the tree can be estimated approximately from the data. For that we build an HMM inspired from PSMC or PSMC' (MSMC with 2 haplotypes), with observations being a sequence of "1", "0" and ".". Here, "1" denotes a position with minor allele count 1 (a singleton), "0" denotes any other called position and "." denotes missing data. The hidden state of this HMM is the singleton branch length T_s. It is straight forward to see that the emission probability of that model is:

$$e(1 \mid T_s) = 1 - \mu\, T_s, \quad e(0 \mid T_s) = \mu\, T_s, \quad e(. \mid T_s) = 1.$$

This is very similar to the PSMC/PSMC' emission probability, with the only difference being a missing factor 2 in front of T_s, see below.

It is difficult to derive an analytically exact transition probability for that model, since the recombination process that changes one tree to another does not only depend on the singleton branch length. However, we find that simple heuristics are good enough: We know that most recombination events that change the singleton branch length occur within the singleton branch length itself, which happens with probability $\rho\, T_s$. For the probability *conditional* on there being a recombination event, we now simply approximate the process by the analogous expression from PSMC', equations 1, 2 and 3, using a constant population size (to avoid having to re-estimate T_s every iteration of MSMC) and replacing $t = T_s/2$. The factor $1/2$ is important because PSMC considers half the total branch length as its hidden state, whereas here we use the branch length T_s itself as hidden state.

We again discretize the state space similarly to MSMC using quantile boundaries:

$$T_{s,i} = -\log\left(1 - \frac{i}{n_{T_s}}\right) \Big/ (2 + 1/(M-1))$$

where $2 + 1/(M-1)$ is the expected singleton branch length: It is the sum of the leaf branch length of the tree (which is 2 in units of $2\,N_0$) and that part of the tree which gives rise to variants with derived allele frequency $M-1$ (which is $1/(M-1)$ in units of $2\,N_0$).

We then run this HMM via the forward- backward-algorithm to obtain local posterior probabilities. For the discretization intervals, we

Nature Genetics: doi:10.1038/ng.3015

We then use the maximum posterior path as best local estimate of T_s. The reason why this simple approach works so well despite all the approximations is the fact that the maximum likelihood path of T_s is very much dominated by the local emission rates, and not so much by the transition rates.

■ Testing the singleton branch length estimation with simulations

To test how the approximations in our singleton branch length estimation influence the demographic inference, we can use the true singleton branch length locally directly to use for T_s in the MSMC emission probability. As shown in *Supplementary Figure 3*, this comparison reveals that both the HMM estimation of T_s and the true value obtained from the actual trees from the simulation are remarkably similar, revealing almost no effect of either the approximation involved in using PSMC's transition probability, nor in using a constant effective population size for the estimation.

There is one important note about comparing with the true singleton branch length from simulations. It is quite tedious to compute from the Newick-trees (output from MaCS [6] and ms [5] coalescent simulators) the full *minor* allele singleton branch length. As defined above, T_s does not only include true singletons, but also mutations with derived allele count $M-1$. In these comparisons, we therefore change the definition slightly to $\tilde{T}_s$, which includes only singletons with *derived* allele count 1. This is very easy to extract from the Newick-tree format, since it is simply the sum of all leaf branches. Correspondingly, in the MSMC emission probability we have to use a slightly different term for category 2: $e(O \mid t, i, j) = \mu\left(\tilde{T}_s - 2t\right) / (M-2)$, which is straight forward to see, given the slightly simpler definition of $\tilde{T}_s$. We use the modified definition $\tilde{T}_s$ only for the MSMC runs using the true singleton branch lengths (dotted lines in *Supplementary Figures 3*).

7. Coalescent simulations

■ Zig-Zag simulation

We first generated a single population with a zig-zag type population history, implemented in *ms* [5] as a sequence of exponential growths and declines:

```
ms 4 1 -t 7156.0000000 -r 2000.0000 10000000 -eN 0 5 -eG 0.000582262 1318.18 -eG
0.00232905 -329.546 -eG 0.00931619 82.3865 -eG 0.0372648 -20.5966 -eG 0.149059 5.14916 -eN
0.596236 0.5 -T
```

Here, the first parameter (4) is the number of haplotypes, which we varied among 2, 4 and 8.

■ Split simulation

In the second scenario we simulated a single population with *MaCS* [6] that split into two constant size populations. The command line for this case is:

```
macs 4 30000000 -t 0.0007156 -r 0.0002 -I 2 2 2 -ej 0.116 2 1 -T
```

We again generated a dataset with 8 haplotypes for this scenario, replacing the first command line argument with "8", and the three parameters after "-I" with "2 4 4". Also, the parameter following "-ej" determines the scaled time of the split, where 0.116 corresponds a split 100kya using our generation time and mutation rate. We simulated various values for the split time between 10kya and 150kya, simply scaling the parameter after "-ej".

Since *MaCS* can simulate faster and more efficiently, we simulate 30Mb per simulation. With *ms*, we only simulate 10Mb. Note that *ms* is needed to simulate negative growths, which *MaCS* currently cannot do.

■ Simulations with sharp population size changes

We simulated two histories with sharp population size changes. One which mimics the true changes observed in CEU (see *Supplementary Figure 2a*):

```
macs 4 30000000 -t 0.0007156 -r 0.0002 -eN 0.0 10.8300726663 -eN 0.00116452394261
1.08300726663 -eN 0.0174678591392 0.216601453326 -eN 0.0465809577045 1.08300726663 -eN
0.0873392956959 3.24902179989 -eN 0.232904788522 1.08300726663 -T
```

and one which mimics the history observed in YRI (see *Supplementary Figure 2b*):

```
macs 4 30000000 -t 0.001 -r 0.0004 -eN 0.0 8.25 -eN 0.0025 0.825 -eN 0.0416666666667 2.475
-eN 0.166666666667 0.825 -T
```

Again, we replaced the first parameter "4" with "2" and "8" to simulate fewer or more haplotypes.

Nature Genetics: doi:10.1038/ng.3015

■ Population split with migration

This command line simulates a population split 100kya with subsequent migration of 0.0002 per generation until 50kya, as seen in *Supplementary Figure 2c*.

```
ms 4 10000000 -t 10000 -r 4000 -I 2 2 2 -ej 0.116 2 1 -eM 0.058 16 -T
```

■ Population split with population size changes

This simulates a population that split into two populations 120kya with varying population size changes resembling the CEU/YRI split without any migration, as seen in *Supplementary Figure 3d*.

```
macs 4 30000000 -t 0.001 -r 0.0004 -I 2 2 2 -eN 0 9.5  -en 0.000833333333333 1 0.95 -en
0.0025 2 0.95 -en 0.0125 1 0.19 -en 0.0333333333333 1 0.95 -en 0.0416666666667 2 2.85 -ej
0.05 2 1 -eN 0.166666666667 0.95 -T
```

■ Hotspot simulations

We used the software msHot [7] to simulate sequences under the true recombination map in Humans, obtained from the HapMap project. We simulated 100 chromosomes, each with 1Mb long sequences for 4 haplotypes. We chose the same command line as for the zig-zag simulation, with a scaled recombination rate of $4 N_0 r = 0.00055$, a scaled mutation rate of $4 N_0 \mu = 0.0007156$ and additional parameters

```
-v n <start_1> <end_1> <l_1> <start_2> <end_2> <l_2> ... <start_n> <end_n> <l_n>,
```

where each block of recombination rates is taken from a random chunk of the true human recombination map. The <l_i> values are given as multiples of the average recombination rate in humans of 1 cM/Mb.

8. Comparison with diCal

We applied diCal [8] to 10Mb of simulated sequence from 8 haplotypes. This sequence length was used in [8] to demonstrate the method, it is currently not practical to run it on longer data sets. We used the following command line to run diCal:

```
java -Xmx65G -d64 -jar diCal-v1.2/diCal.jar -F "chr1.fasta,chr2.fasta,...,chr10.fasta" -c
10 -I params.txt -n 8 -p "3 2 2 2 2 2 2 2 3" -t 1
```

where “chr1.fasta”, “chr2.fasta”,... denote 10 fasta files, each with 8 haplotypes of length 1Mb. We also tried “-t 0.5” and “-t 0.1” (see *Supplementary Figure 10*).

9. Processing Genomic Data

In contrast to PSMC [1], we do not need to bin all data into bins of 100bp but can simply go through every basepair in the genome. As described above, we optimized MSMC for efficient traversal of SNP-free homozygous segments, as well as missing data segments.

In practice, the MSMC implementation takes as input files simple tab-separated files, one for each chromosome, such as this:

```
1   58432   63  TCCC
1   58448   16  GAAA
1   68306   15  CTTT
1   68316   10  TCCC
1   69552   8   GCCC
1   69569   17  TCCC
1   801848  9730    CCCA
1   809876  1430    AAAG
1   825207  1971    CCCT,CCTC
1   833223  923 TCCC
```

Here, the columns are:

1. Chromosome
2. Position of the SNP in the chromosome
3. Number of called bases since the last SNP, *including* the site at this position.
4. The alleles at the site in question. Comma-separated variations can account for missing phasing information, as in seen in the line before the last line in the example above. In practice, we simply average the emission probability at that site over all variations.

Nature Genetics: doi:10.1038/ng.3015

We account for missing data in this format through the third column. This number is always larger than zero since it always includes the site at that position. However, if the number of called bases is smaller than the physical distance since the last SNP, the rest is assumed to be missing. This encoding of missing data neglects information about the exact positions of missing bases, but simply gives the total number of them since the last called site. In practice, we assume that all missing bases since the last called base form a contiguous block right after the last called base. This way we only need to apply the recursion in the forward- and backward vector twice: once for the block of missing data and once for the block of homozygous called bases. The error in this approximation should be negligible, since the forward- and backward variables will typically change on much larger genomic lengthscales than between individual SNPs, such that the exact distribution of missing bases between two SNPs should not affect the inference.

In our implementation on github, we provide scripts that facilitate the generation and processing of these input files from e.g. BAM files or Complete Genomics VCF files. We also provide additional documentation about the software as a README on github.

10. Relationship to Island-Migration models

In deriving the transition- and equilibrium probabilities for times of first coalescences we have parameterized our model using effective coalescence rates between all pairs of lineages, $\lambda^{i,j}(t)$. This parameterization is very general and allows us to model a) scenarios in which all haplotypes are sampled from one population, $\lambda^{i,j}(t) \equiv \lambda(t)$, with no differences between rates between different lineages; and b) scenarios with haplotypes sampled from different subpopulations. In the latter case, it is interesting to ask how our parameterization relates to a model with two separated populations and time-dependent migration between them.

Consider the case of two populations with time-dependent population sizes $N_1(t)$ and $N_2(t)$, and a time-dependent and symmetric migration rate $m(t)$ between them. We believe that such a parameterization is *equivalent* to a parameterization using coalescence rates within populations, $\lambda^{11}(t)$ and $\lambda^{22}(t)$, and a cross coalescence rate $\lambda^{12}(t)$, where the indices here mean populations, not lineages. We mean by "equivalent" that for every two-island model with symmetric migration rates, parameterized by $\{N_1(t), N_2(t), m(t)\}$, there exists a mapping

$$\{N_1(t), N_2(t), m(t)\} \rightarrow \left\{\lambda^{11}(t), \lambda^{12}(t), \lambda^{22}(t)\right\}$$

such that the distribution of first coalescence times, $q_0(t)$, is the same in both models.

We can indeed show this for the special case of two haplotypes drawn from different subpopulations, as shown below. We further hypothesize that the equivalence holds for more general island-migration models, too, but we emphasize that we cannot prove this.

It is important to highlight the fact that what we term "relative cross-coalescence rate", defined as $\gamma(t) = 2\,\lambda^{12}(t)/(\lambda^{11}(t) + \lambda^{22}(t))$ in the article, is *not* equivalent to the migration rate. The exact relationship is more subtle, even for the special case of two haplotypes (see below) and in principle would involve numerical solutions to differential equations. However, intuitively, it may help to imagine the migration rate $m(t)$ as being closely related to the *rate of change* of the relative cross-coalescence rate $\gamma(t)$: Consider a situation in which two populations have been separated since time T, but that there was a brief period of relatively strong migration between them around time $T_m < T$, where times are counted backwards in time. The expected relative cross coalescence rate $\gamma(t)$ would then be:

$$\gamma(t) = \begin{cases} 0 & \text{for } t < T_m \\ 0 < \gamma_m < 1 & \text{for } T_m < t < T \\ 1 & \text{for } t \geq T, \end{cases}$$

i.e. a step-like function, in which the brief pulse of migration would increase $\gamma(t)$ to some intermediate value γ_m, before it would reach 1 at the split time T. Clearly, if migration is more ongoing rather then short and strong, $\gamma(t)$ would increase more steadily, in a well defined way which is however difficult to derive analytically.

■ Constant migration rate, two haplotypes

Consider two haplotypes, each of which drawn from two separate populations. Let us first consider constant parameters $N_i(t) \equiv N_i$ and a constant migration rate $m(t) \equiv m$. As shown in [9] the distribution of first coalescence times $q_0(t)$ can in that case be computed from the matrix exponential of a Markov rate matrix. The five states of the Markov process are then: S_{11}, both genes are in population 1; S_{22}, both genes are in population 2; S_{12}, one gene is in population 1 and the other is in population 2; S_1, the genes have coalesced and the single gene is in population 1; and S_2, the genes have coalesced and the single gene is in population 2. It is straight forward to write down the 5×5 rate matrix, Q, for this process using the three parameters N_1, N_2 and m, as shown in the referenced article. The probability density of first coalescence times can then be expressed as the probability that starting with state S_{12} the system will eventually be in one of the absorbing states S_1 or S_2, which can be expressed by a matrix exponential:

$$P_{S_i}(t) = [\exp(Q\,t)]_{S_i,S_{12}}$$

where the indices denote the end and the start state. The probability density of coalescent times, $q_{0,\text{Hobolth}}(t)$ is then

$$q_{0,\text{Hobolth}}(t) = \frac{1}{2\,N_1} P_{S_1}(t) + \frac{1}{2\,N_2} P_{S_2}(t)$$

We can now very easily show that a single time-dependent parameter $\lambda^c(t)$, which denotes the effective rate of cross-coalescence between the two lineages, is sufficent to parameterize this probability density. As defined in chapter 1, the equilibrium probability of first coalescence times in this special case can be expressed as

$$q_0(t) = \lambda^c(t) \exp\left(- \int_0^t \lambda^c(t')\, dt' \right).$$

We can rewrite this as

$$q_0(t) = \frac{dL(t)}{dt} \exp(-(L(t) - L(0)))$$

with $L(t)$ being defined as the anti-derivative $dL(t)/dt = \lambda^c(t)$. It is then clear that the function $L(t)$ which solves the differential equation

$$\frac{dL(t)}{dt} \exp(-(L(t) - L(0))) = q_{0,\text{Hobolth}}(t)$$

leads directly to the desired solution for $\lambda^c(t)$. This shows that for this special case, both parameterizations are equivalent.

■ Time dependent migration rate

If population sizes and migration rates are time-dependent, the approach from [9] needs to be modified. In that case, the Markov transition matrix is itself time-dependent, expressed as $Q(t)$, and the formula for the equilibrium density can still be expressed as a matrix exponential, but with an additional integral:

$$P_{S_i}(t) = \left[\exp\left(\int_0^t Q(t')\, dt' \right) \right]_{S_i, S_{12}},$$

which still can in principle be parameterized via a single rate $\lambda^c(t)$ as shown above.

■ More general cases

In the more general case, if more than one haplotype is sampled from either or both of the two populations, if more than two populations are present, or if recombination comes into play, we still hypothesize that effective coalescence rates within and between populations are sufficient to model any island-migration model with symmetric migration rates. But as long as we lack a formal proof of this, we acknowledge that it is not entirely clear whether our parameterization is exactly mappable to an island model or whether it only approximates it in the most general case.

11. Normalization Proofs

■ Helpers

We define the antiderivative $\phi^{i,j}(t)$ with $\frac{d}{dt}\phi^{i,j}(t) = \lambda^{i,j}(t)$. Also we define $\Phi^i(t) = \sum_{j=1}^{M} \phi^{i,j}(t)$ such that $\frac{d}{dt}\Phi^i(t) = \Lambda^i(t)$, and similarly $\Phi(t) = \sum_{i<j} \phi^{i,j}(t)$ such that $\frac{d}{dt}\Phi(t) = \Lambda(t)$.

We then have

$$L^m(t_1; t_2) = \exp\left(- \int_{t_1}^{t_2} \Lambda^m(\nu)\, d\nu \right) = e^{\Phi^m(t_1) - \Phi^m(t_2)}$$

and

$$L(t_1; t_2) = \exp\left(- \int_{t_1}^{t_2} \Lambda(\nu)\, d\nu \right) = e^{\Phi(t_1) - \Phi(t_2)}.$$

We then have:

$$\int_s^t \Lambda(u)\, L(s; u)\, du = \int_s^t \Phi'(u)\, e^{\Phi(s) - \Phi(u)}\, du = e^{\Phi(s)} \int_{\Phi(s)}^{\Phi(t)} e^{-z}\, dz = e^{\Phi(s)} \left(e^{-\Phi(s)} - e^{-\Phi(t)} \right) = 1 - L(s; t) \tag{10}$$

and similarly

$$\int_s^t \Lambda^m(u)\, L^m(s; u)\, du = 1 - L^m(s; t) \tag{11}$$

■ Normalization of the equilibrium probability

The equilbrium probability is

$$q(i,\ j,\ t) = \lambda^{i,j}(t)\, L(0; t).$$

We check normalization by integrating

$$\sum_{i<j}^{M}\int_0^\infty q(i,\,j,\,t)\,dt = \int_0^\infty \sum_{i<j}^{M}\lambda^{i,j}(t)\,L(0;t)\,dt = \int_0^\infty \Lambda(t)\,L(0;t)\,dt = 1 - L(0;\infty) = 1$$

using equation 10.

■ Normalization of the conditional transition probability

In a bit more formal style, the total transition probability can be expressed using Kronecker-Deltas and the Heavyside-function Θ. We have

$$q(i,\,j,\,t \mid k,\,l,\,s,\,u,\,m) =$$

$$\delta(t-s)\,\delta_{i,k}\,\delta_{j,l}\left(\int_u^t \lambda^{m,m}(\kappa)\,L^m(u;\kappa)\,d\kappa + \left(1-\delta_{m,k}\right)\left(1-\delta_{m,l}\right)L^m(u;t)\right) + \begin{cases} \left(\delta_{m,i}+\delta_{m,j}\right)\lambda^{i,j}(t)\,L^m(u;t)\,\Theta(t-u) & \text{for } t \le s \\ \left(\delta_{m,k}+\delta_{m,l}\right)\lambda^{i,j}(t)\,L^m(u;s)\,L(s;t) & \text{for } t > s. \end{cases}$$

Now the normalization needs to hold for the total parameter space of the triple $(i,\,j,\,t)$:

$$\begin{aligned}
&\sum_{i<j}^{M}\int_0^\infty q(i,\,j,\,t \mid k,\,l,\,s,\,u,\,m)\,dt \\
&\quad= \sum_{i<j}^{M}\int_0^\infty \delta(t-s)\,\delta_{i,k}\,\delta_{j,l}\left(\int_u^t \lambda^{m,m}(\kappa)\,L^m(u;\kappa)\,d\kappa + \left(1-\delta_{m,k}\right)\left(1-\delta_{m,l}\right)L^m(u;t)\right)dt \\
&\qquad+ \sum_{i<j}^{M}\int_0^s \left(\delta_{m,i}+\delta_{m,j}\right)\lambda^{i,j}(t)\,L^m(u;t)\,\Theta(t-u)\,dt \\
&\qquad+ \sum_{i<j}^{M}\int_s^\infty \left(\delta_{m,k}+\delta_{m,l}\right)\lambda^{i,j}(t)\,L^m(u;s)\,L(s;t)\,dt =
\end{aligned} \tag{12}$$

We compute all three terms separately:

First Term:

$$\begin{aligned}
&\sum_{i<j}^{M}\int_0^\infty \delta(t-s)\,\delta_{i,k}\,\delta_{j,l}\left(\int_u^t \lambda^{m,m}(\kappa)\,L^m(u;\kappa)\,d\kappa + \left(1-\delta_{m,k}\right)\left(1-\delta_{m,l}\right)L^m(u;t)\right)dt \\
&\quad= \int_u^s \lambda^{m,m}(\kappa)\,L^m(u;\kappa)\,d\kappa + \left(1-\delta_{m,k}\right)\left(1-\delta_{m,l}\right)L^m(u;s)
\end{aligned}$$

Second Term:

$$\begin{aligned}
&\sum_{i<j}^{M}\int_0^s \left(\delta_{m,i}+\delta_{m,j}\right)\lambda^{i,j}(t)\,L^m(u;t)\,\Theta(t-u)\,dt \\
&\quad= \sum_{i<j}^{M}\int_u^s \left(\delta_{m,i}+\delta_{m,j}\right)\lambda^{i,j}(t)\,L^m(u;t)\,dt \\
&\quad= \sum_{i\neq m}\int_u^s \lambda^{m,i}(t)\,L^m(u;t)\,dt \\
&\quad= \int_u^s \left(\Lambda^m(t) - \lambda^{m,m}(t)\right)L^m(u;t)\,dt
\end{aligned}$$

Third Term:

$$\begin{aligned}
&\sum_{i<j}^{M}\int_s^\infty \left(\delta_{m,k}+\delta_{m,l}\right)\lambda^{i,j}(t)\,L^m(u;s)\,L(s;t)\,dt \\
&\quad= \left(\delta_{m,k}+\delta_{m,l}\right)L^m(u;s)\int_s^\infty \Lambda(t)\,L(s;t)\,dt
\end{aligned}$$

The sum of all terms:

$$\begin{aligned}
&\int_u^s \lambda^{m,m}(\kappa)\,L^m(u;\kappa)\,d\kappa + \left(1-\delta_{m,k}\right)\left(1-\delta_{m,l}\right)L^m(u;s) \\
&\qquad+ \int_u^s \left(\Lambda^m(t) - \lambda^{m,m}(t)\right)L^m(u;t)\,dt
\end{aligned}$$

Nature Genetics: doi:10.1038/ng.3015

$$
\begin{aligned}
&+\left(\delta_{m,k}+\delta_{m,l}\right) L^m(u;s) \int_s^\infty \Lambda(t)\, L(s;t)\, dt \\
&= \left(1-\delta_{m,k}\right)\left(1-\delta_{m,l}\right) L^m(u;s) + \int_u^s \Lambda^m(t)\, L^m(u;t)\, dt + \left(\delta_{m,k}+\delta_{m,l}\right) L^m(u;s) \int_s^\infty \Lambda(t)\, L(s;t)\, dt \\
&= \left(1-\delta_{m,k}\right)\left(1-\delta_{m,l}\right) L^m(u;s) + (1-L^m(u;s)) + \left(\delta_{m,k}+\delta_{m,l}\right) L^m(u;s)\,(1-L(s;\infty)) \\
&= L^m(u;s) + (1-L^m(u;s)) \\
&= 1
\end{aligned}
$$

□

References

1. Li, H. and R. Durbin, *Inference of human population history from individual whole-genome sequences*. Nature, 2011.
2. Marjoram, P. and J.D. Wall, *Fast "coalescent" simulation*. BMC genetics, 2006. **7**: p. 16.
3. McVean, G.A.T. and N.J. Cardin, *Approximating the coalescent with recombination*. Philosophical transactions of the Royal Society of London Series B, Biological sciences, 2005. **360**(1459): p. 1387-1393.
4. Press, W.H., *Numerical recipes : the art of scientific computing*. 3rd ed2007, Cambridge, UK ; New York: Cambridge University Press. xxi, 1235 p.
5. Hudson, R. R. *Generating samples under a Wright-Fisher neutral model of genetic variation*. Bioinformatics, 2002, 18: 337-338.
6. Chen, G.K., P. Marjoram, and J.D. Wall, *Fast and flexible simulation of DNA sequence data*. Genome Res, 2009. **19**(1): p. 136-142.
7. Hellenthal, G. and M. Stephens, *msHOT: modifying Hudson's ms simulator to incorporate crossover and gene conversion hotspots,* 2006, Bioinformatics.
8. Sheehan, S., Harris, K., & Song, Y. S. (2013). Estimating variable effective population sizes from multiple genomes: a sequentially markov conditional sampling distribution approach. Genetics, 194(3), 647–662. doi:10.1534/genetics.112.149096
9. Hobolth, A., L. Nørvang Andersen and Thomas Mailund, *On Computing the Coalescence Time Density in an Isolation-With-Migration Model With Few Samples,* Genetics, 2011. **187: p.** 1241-1243