psmc-rs: fast PSMC inference that reproduces the reference implementation

Yufan Han
School of Life Sciences, Fudan University, Shanghai, China
Correspondence: larryivanhan@gmail.com

Abstract

The pairwise sequentially Markovian coalescent (PSMC) infers the history of effective population size from a single diploid genome and is one of the most widely used methods of population genomics. Its reference implementation, the C program psmc, runs on one thread, which makes bootstraps and large cohorts slow. We present psmc-rs, a reimplementation of PSMC in Rust that fits the same model as C psmc. On 40 simulated and 55 real genomes, the two programs' estimates of population size differed by at most 0.15% over the time range each genome informs. On the simulated genomes, their histories were at least 1,500 times closer to each other than either was to the true history. On a simulated genome of 30 chromosomes of 100 Mb, psmc-rs was 14.1 times as fast as C psmc on one thread and 386 times as fast on 30 threads, and it used at most 148 MB of memory, against C psmc's 1,062 MB. Its results were byte-identical on any number of threads. Fitting all 2,504 unrelated high-coverage genomes of the 1000 Genomes Project took psmc-rs 257 core-hours, where C psmc would have needed about 3,700. psmc-rs also reads VCF and multihetsep files, accounts for partly callable bins, and runs PSMC's block bootstrap in parallel. It is open source under the MIT license, and the pipeline behind every result, with the inputs and fits of the 2,504 genomes, is openly available.

Introduction

The pairwise sequentially Markovian coalescent (PSMC; Li and Durbin, 2011) infers how the effective size (N_e) of a population changed over time from the two genome copies of a single diploid individual. Along the genome, the time to the most recent common ancestor (TMRCA) of the two copies changes at recombination events. PSMC treats this time as the hidden state of a hidden Markov model whose observations are short bins of sequence, each heterozygous or homozygous, and fits a piecewise-constant history of N_e by expectation–maximization (EM). Because it needs only one unphased genome, PSMC has been applied to modern and archaic humans (Li and Durbin, 2011; Prüfer et al., 2014; Mallick et al., 2016) and to a wide range of other species (Nadachowska-Brzyska et al., 2016; Hilgers et al., 2025), and it remains the baseline for newer coalescent hidden Markov models (Schiffels and Durbin, 2014; Terhorst et al., 2017; Cousins et al., 2025; Terhorst, 2025).

PSMC's reference implementation, the C program `psmc` (Li, 2025), runs on one thread. Twenty-five EM iterations on a 3 Gb genome take more than an hour, a bootstrap multiplies that time by its number of replicates, and population resources such as the high-coverage 1000 Genomes data (Byrska-Bishop et al., 2022) now hold thousands of genomes. Other programs port PSMC to other languages and toolkits (Paradis and Li, 2023; Lomakin, 2024; Korfmann, 2026) or extend its model (Schiffels and Wang, 2020; Liu et al., 2022; Cousins et al., 2024, 2025), and methods that combine many genomes or fit a richer model answer other questions. To replace C psmc, a faster program must keep its estimator and reproduce its results, so that new curves remain comparable with the many PSMC curves already published. The native PSMC of smckit has been checked

against C psmc on one chromosome (Korfmann, 2026); we are not aware of a program shown to reproduce C psmc's fits of whole genomes on the same inputs while running substantially faster than C psmc.

Here we present psmc-rs, a reimplementation of PSMC in Rust. We show that it reproduces C psmc's fits of simulated and real genomes. It is an order of magnitude faster than C psmc on one thread and, on genomes of many sequences, two orders of magnitude faster on many threads; it needs a fraction of the memory and gives results that do not depend on the number of threads. We then fit all 2,504 unrelated high-coverage genomes of the 1000 Genomes Project, and release their inputs and fits.

Results

psmc-rs fits the PSMC model as C psmc does

psmc-rs fits the PSMC model of Li and Durbin (2011) as C psmc 0.6.5 implements it: the same discretization of time, the same transition and emission probabilities, the same parameter patterns and starting point, and EM on the same objective (Methods). It reads C psmc's input format, `psmcfa`; VCF files, either all-sites consensus calls or variant calls with a mask of callable sites; and the multihetsep files of MSMC (Schiffels and Durbin, 2014). It writes the fitted model as JSON and as an HTML report and, on request, in C psmc's `.psmc` format, which C psmc's `psmc_plot.pl` draws. With the settings of C psmc's documentation, this command fits a human genome for 25 EM iterations on eight threads:

```
psmc-rs NA12878.psmcfa.gz NA12878.json 25 --pattern "4+25*2+4+6" --t-max 15 \
    --tr-ratio 5 --mu 1.25e-8 --gen-years 29 --threads 8 --psmc-out NA12878.psmc
```

psmc-rs differs from C psmc in four ways (Methods). First, it processes the sequences of a genome (its chromosomes or scaffolds) in parallel, each on one thread. Second, its maximization step (M-step) runs L-BFGS to convergence, within bounds on the parameters, where C psmc runs a Hooke–Jeeves search. Third, its expected emission counts include the last bin of each sequence, as both programs' log-likelihoods do; C psmc leaves that bin out of its counts. Fourth, when the input records how many of a bin's bases are callable, that is, could be genotyped, it weights the bin by that number.

psmc-rs reproduces C psmc

We fitted both programs, with the same inputs and settings, to 40 simulated genomes of 500 Mb, 10 under each of four demographic models, and to 55 real genomes: three turtles and 52 human genomes of the 1000 Genomes Project, two per population, which we call the comparison subset (experiments E1, E2 and E7; Methods). We compared the fits over each genome's informative window, the times between the 1% and 99% quantiles of the pairwise coalescence time; outside it, the data say little about N_e. As the difference between two fits we use Δ, the root-mean-square error (RMSE) of one fit's $\log_{10} N_e$ against the other's, interval by interval over this window (Methods). On every genome, Δ was at most 2.8×10^{-4} (Table 1, Fig. 1). Across all 4,620 intervals in the windows of these 95 genomes, the two programs' N_e differed by a median of 5.5×10^{-6} in $\log_{10} N_e$ and by at most 6.6×10^{-4}, that is, by at most 0.15%.

This difference is far smaller than the error of the method itself. On the simulated genomes, Δ ranged from 3.7×10^{-6} to 2.0×10^{-4} and was at least 1,500 times smaller than either program's error relative to the true history; the median Δ was 7,000 times smaller than psmc-rs's median error (Fig. 1e). The two programs' errors were equal to three decimals, from a mean RMSE of 0.048 for the constant-size model to 0.291 for the bottleneck model (Table S1). The difference is also about three times C psmc's own numerical noise. On the simulated genomes, the median Δ was 2.5×10^{-5}, against a median of 8.5×10^{-6} between C psmc's own fits on two

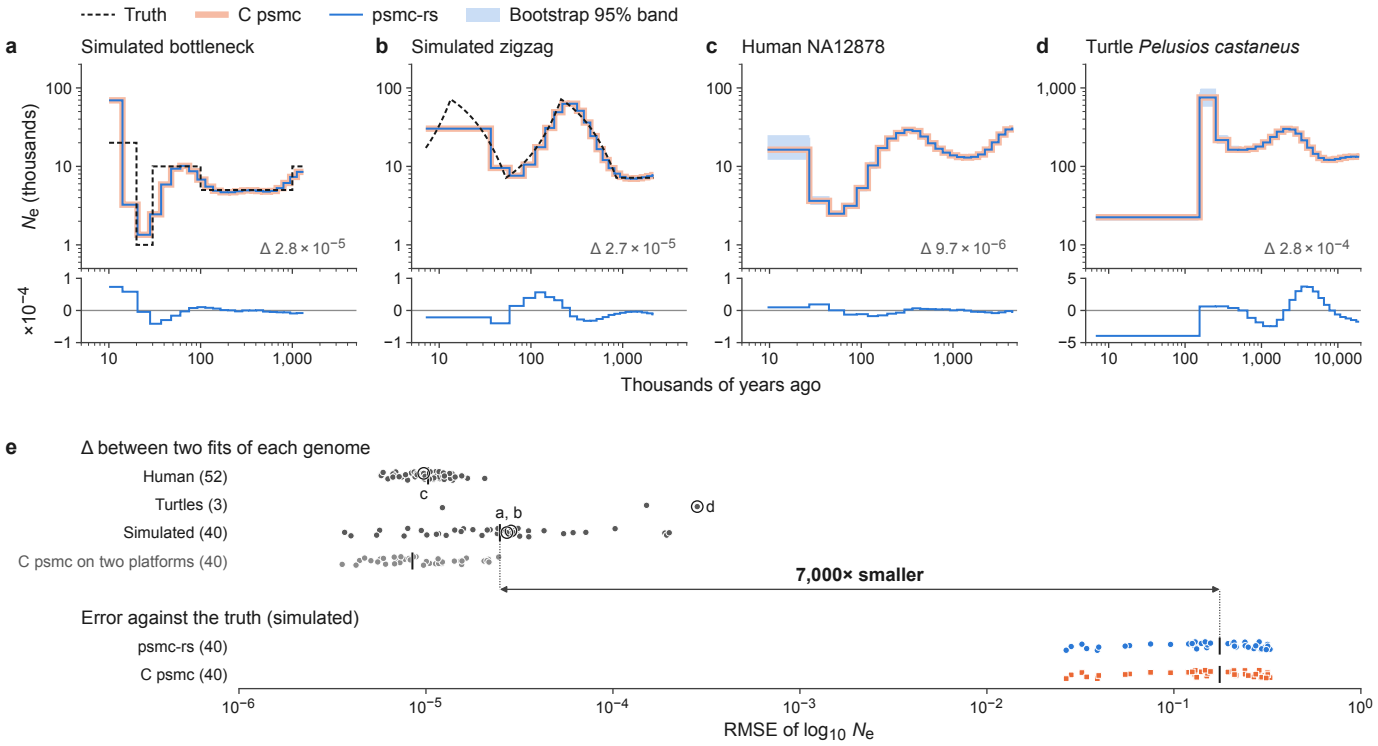


Figure 1. psmc-rs reproduces C psmc. (a–d) Histories fitted by both programs to the same genome, C psmc's (wide, pale orange) under psmc-rs's (thin, blue), over the genome's informative window (1% to 99% of its pairwise coalescence times). Below each, psmc-rs's $\log_{10} N_e$ minus C psmc's, interval by interval, in units of 10^{-4}; Δ, the RMSE of this difference over the window (Methods). (a, b) The first simulated genome of the bottleneck (a) and zigzag (b) models, with the true history (dashed); $\mu = 2.5 \times 10^{-8}$ per bp per generation, $g = 25$ years. (c, d) The human genome NA12878 and the turtle *Pelusios castaneus*, the genome of the largest Δ, with, in pale blue where it is wider than the curves, the pointwise 95% band of psmc-rs's bootstrap (100 replicates in blocks of 50 Mb, Methods; for the turtle, its 1,220 whole scaffolds). Both are scaled as in Methods; panel d and its strip have their own scales. The turtle's recent peak is an artefact of the pattern's single free λ for the four most recent intervals (Hilgers et al., 2025). (e) Δ for every genome both programs fitted: the 52 human genomes of the 1000 Genomes comparison subset (NA12878 among them), the three turtles and the 40 simulated genomes; rings, the genomes of a–d. The row "C psmc on two platforms" gives, for comparison, Δ between C psmc's own fits of the simulated genomes on two platforms (gcc without floating-point contraction on x86-64, Apple clang with fused multiply-adds on arm64; Methods). Below, each program's RMSE against the true history of the simulated genomes. Bars, medians of rows of at least ten genomes; the arrow joins the simulated genomes' median Δ to the median of psmc-rs's error on them: Δ is 7,000 times smaller.

Table 1. Agreement between psmc-rs and C psmc on every genome both programs fitted, with the same input and settings. NA12878 is in both the E2 and the E7 rows. Δ, the RMSE (range over genomes), and largest absolute difference $|d_k|$ of $\log_{10} N_e$, interval by interval over the informative window (Methods); largest relative differences of the scaled mutation rate θ, the scaled recombination rate ρ and the maximum coalescence time t_{max}, of the log-likelihood at the last iteration both programs report, and of the log-likelihood of the first expectation step (E-step), at the shared starting point, where the two agree at about the rounding of the six decimals C psmc prints.

		$\log_{10} N_e$ in the window		largest relative difference		
Genomes	n	Δ	largest $\lvert d_k \rvert$	θ, ρ, t_{max}	log-likelihood	first E-step
Simulated (E1)	40	3.7×10^{-6} to 2.0×10^{-4}	6.6×10^{-4}	5.5×10^{-4}	1.3×10^{-8}	7.0×10^{-13}
Turtles and NA12878 (E2)	4	9.7×10^{-6} to 2.8×10^{-4}	4.5×10^{-4}	1.0×10^{-3}	2.3×10^{-7}	6.9×10^{-14}
1000 Genomes (E7)	52	5.7×10^{-6} to 2.1×10^{-5}	5.8×10^{-5}	5.5×10^{-5}	2.8×10^{-9}	1.1×10^{-13}

platforms (gcc without floating-point contraction on x86-64, Apple clang with fused multiply-adds on arm64; Fig. 1e, row 'C psmc on two platforms'; Methods). Finally, it is far smaller than the sampling uncertainty of a fit: on the turtles and NA12878, the median width of psmc-rs's bootstrap 95% band was at least 260 times Δ (Fig. 1c, d).

The two programs compute the same likelihood; their small differences arise during the EM iterations. From the shared starting point, their first expectation steps (E-steps) gave log-likelihoods that agreed at about the rounding of the six decimals C psmc prints (a relative 7.0×10^{-13}). In both programs the log-likelihood never decreased from one iteration to the next, and the two programs' log-likelihoods differed by at most 0.065 units after any iteration (Fig. S1). A diagnostic on the simulated genomes (Methods) traced the differences mostly to the last bin of each sequence, not to the M-step's optimizer. Leaving that bin out of psmc-rs's expected emission counts, as C psmc does, lowered the median Δ from 2.5×10^{-5} to 6.2×10^{-6}, below C psmc's own noise. It brought the curves closer to C psmc's in 37 of the 40 genomes; the other three were already within that noise. Replacing L-BFGS with C psmc's Hooke–Jeeves search barely changed the median (2.6×10^{-5}). Each sequence has one last bin, so the difference should grow with the number of sequences: among the turtles and NA12878, Δ was smallest for NA12878, with 22 sequences, and largest for the turtle *Pelusios castaneus*, with 1,220 (Fig. 1c–e). It also depends on the history: the simulated genomes, of ten sequences each, reached up to 2.0×10^{-4}.

Both programs share PSMC's weakness at the most recent times, about which one genome says little. With the pattern used here, a single parameter, the first free λ (a size relative to N_0; Methods), sets N_e in the four most recent time intervals. In one of the 40 simulated genomes, this parameter ran away in both programs, outside the informative window: to psmc-rs's upper bound of 10^4 and, in C psmc, which has no bound, to 1.1×10^6. The turtles' curves show recent peaks of N_e (one shown in Fig. 1d), artefacts of this grouping that Hilgers et al. (2025) found in many published PSMC curves. Giving the four most recent intervals two or four free λs instead of one removed the peaks in all six fits and raised the log-likelihood by 49 to 586 units.

psmc-rs is fast and deterministic

To measure speed, we timed both programs on simulated constant-size genomes of 1, 3, 10 and 30 chromosomes of 100 Mb, each run alone on an exclusive node, with five repeats on four nodes (experiment E3; Fig. 2a, b, Table 2, Methods). On one thread, 25 EM iterations on the 3,000 Mb genome took psmc-rs 317 s (5.28 minutes) and C psmc 4,456 s (74.3 minutes): psmc-rs was 14.1 times as fast (Fig. 2a), and 13.9 to 14.2 times as fast at every size (Table 2). With the sequences in parallel, it was 101, 198 and 386 times as fast as C psmc on 8, 16 and 30 threads (Fig. 2a, Table 2), and fitted the 3,000 Mb genome in 11.5 s. Its parallel efficiency, its speed-up over its own one-thread time divided by the number of threads, was 0.88 to 0.97 on 2 to 30 threads (Fig. 2a, grey line).

Because each sequence runs on one thread, a genome's longest sequence bounds the speed-up. One 100 Mb chromosome took 10.46 to 10.97 s on 1, 8 and 30 threads alike (Fig. 2a, dotted line); the 30-thread speed-up holds because this genome has 30 chromosomes of equal length. A human genome's speed-up on many threads is bounded in the same way, by its chromosome 1 (249 Mb).

psmc-rs also needed far less memory than C psmc: 36 MB on one thread and 148 MB on 30, against 1,062 MB (Fig. 2b). C psmc keeps the forward and backward vectors of every bin of a sequence, and its E-step takes time quadratic in the number of states. psmc-rs keeps the forward vectors only at checkpoints and recomputes the rest, and for models of up to 256 states its E-step takes time linear in the number of states (Methods; Harris et al., 2014).

psmc-rs's results do not depend on the number of threads. For each genome size, all its fitted models, from

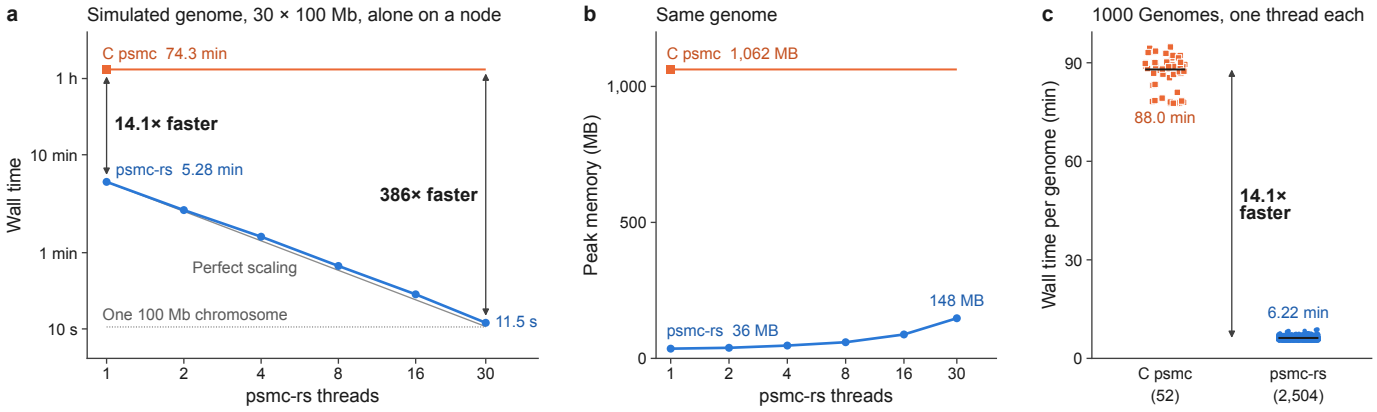


Figure 2. psmc-rs is faster than C psmc and needs less memory. (a, b) A simulated constant-size genome of 30 chromosomes of 100 Mb (E3), fitted for 25 EM iterations, each run alone on an exclusive AMD EPYC 7702 node; points are medians of five repeats on four nodes, whose wall times spread by at most 4.4% of their median. C psmc runs on one thread only: its square, at one thread, is drawn flat across the thread counts. (a) Wall time; arrows, the speed-up over C psmc on 1 and on 30 threads; grey line, perfect scaling of psmc-rs's one-thread time (the points' parallel efficiency is 0.88 to 0.97); dotted line, psmc-rs's time on one 100 Mb chromosome alone (10.46 to 10.97 s on 1, 8 and 30 threads). psmc-rs fits each sequence on one thread, so a genome's longest sequence bounds its run time, and the 30-thread speed-up needs at least 30 sequences of similar length (Table 2). (b) Peak memory (MB of 2^{20} bytes), the larger of the program's `ru_maxrss` and the sampled peak of its process tree. (c) The 1000 Genomes cohort (E7), each genome fitted on one thread of a node shared with other fits: wall time per genome of C psmc on the 52 genomes of the comparison subset (two per population) and of psmc-rs on all 2,504; bars, medians; arrow, the ratio of the medians (on the 52 genomes both programs fitted, the median per-genome ratio was 13.8). Both programs' fits ran in one job on one node, so their ratio is like for like, but these times are not comparable with those of a.

Table 2. Run time and memory of C psmc and psmc-rs (E3). Median wall time of 25 EM iterations over five repeats on exclusive AMD EPYC 7702 nodes, for simulated constant-size genomes of 100 Mb chromosomes; speed-up over C psmc (a ratio of unrounded medians) and peak memory (median, MB of 2^{20} bytes) on 3,000 Mb. – marks a setup not timed.

		Wall time, median (s)				3,000 Mb	
Tool	Threads	100 Mb	300 Mb	1,000 Mb	3,000 Mb	Speed-up	Memory (MB)
C psmc	1	152.5	449.4	1,478.3	4,455.9	1	1,062
psmc-rs	1	11.0	31.8	103.9	317.0	14.1	36
psmc-rs	2	–	–	–	163.3	27.3	39
psmc-rs	4	–	–	–	87.3	51.0	47
psmc-rs	8	10.5	11.0	21.5	43.9	101	59
psmc-rs	16	–	–	–	22.5	198	88
psmc-rs	30	10.5	11.0	11.5	11.5	386	148

every thread count, repeat and node, were byte-identical (75 fits in all), as were C psmc's (20 fits). The bootstrap runs of the turtles and NA12878, on 50 threads, refitted each genome's one-thread fit exactly. NA12878's bootstrap, run with the same seed on 50 and on 32 threads, wrote 103 of its 103 files byte for byte the same. Across platforms, results differ only slightly. We refitted the 40 simulated genomes with a build from the same source code on a laptop with another processor architecture, operating system and math library. No parameter differed from the cluster's fits by more than a relative 2.1×10^{-5}. The laptop's curves were within a Δ of 2.4×10^{-6} of the cluster's, a median of 124 times closer than C psmc's own fits on the same two platforms (Methods).

Partly callable bins, bootstrap and decoding

C psmc's input is made by `fq2psmcfa`, which marks a 100 bp bin missing only when more than 90% of its bases are uncallable. A bin with fewer uncallable bases is fitted as if all its bases had been observed, which biases PSMC's estimates (Lipson et al., 2015; Nadachowska-Brzyska et al., 2016). When the input records how

many of a bin's bases are callable, as a VCF with its callable mask does, psmc-rs weights the bin by that number (Methods), as recent coalescent hidden Markov models do (Cousins et al., 2024, 2025). Inputs that record no callable counts, such as `psmcfa` files, are fitted with C psmc's model unchanged.

We tested the weighting on simulated genomes with 10% or 30% of positions uncallable, either in short runs, which leave many bins partly callable, or in long runs, which empty whole bins (experiment E4). With short runs, taking every called (non-missing) bin as full, as `fq2psmcfa` does, underestimated the scaled mutation rate θ, and with it every size and time, by 11.9% or 34.9%. The weighted fit underestimated it by only 0.7% or 1.8% (Fig. S2). Multihetsep files record how many sites are callable between two segregating sites, but not which, so psmc-rs takes their called bins as full by default. With 30% of positions uncallable, their fits underestimated θ by 31.9% with short runs and 9.1% with long ones; a VCF read without its mask, by 35.5% and 56.8%.

psmc-rs runs PSMC's block bootstrap in parallel, each replicate drawing from its own random stream, so that a seed gives the same replicates on any number of threads. In a calibration study of one bottleneck model (experiment E5), the bootstrap's 95% bands held the mean of independent fits at, on average, 83% of the times in the informative window with blocks of 50 Mb and 92% with blocks of 5 Mb, but the true history at only 37% and 43% (Fig. S3). The bands measure the sampling variation of an estimate, not PSMC's bias.

Read from a VCF with no uncallable positions, a genome gives the same bins as its `psmcfa` and the same fit, bit for bit (experiment E6); with a callable mask, its partly callable bins are weighted, as above. From a multihetsep file, which ends at each sequence's last heterozygous site and places uncallable sites approximately, the fits differed from the `psmcfa` fits by a Δ of at most 6.7×10^{-3}. psmc-rs also decodes the posterior distribution of the TMRCA in every bin: in a simulated genome, the central 95% of this distribution held the true TMRCA in 95.6% of its bins (we did not compare this decoding with C psmc's). Before fitting, it checks every input (`psmc-rs qc`).

2,504 human genomes

We fitted psmc-rs to all 2,504 unrelated individuals of the high-coverage 1000 Genomes data (Byrska-Bishop et al., 2022): 26 populations of 61 to 113 individuals, sequenced to a median depth of 36.2× (experiment E7; Fig. 3). From the joint-called VCFs we made one `psmcfa` file per genome, with the strict accessibility mask (The 1000 Genomes Project Consortium, 2015a) and with every position whose record failed the callset's filters set missing (Methods). For NA12878, the heterozygous single-nucleotide variants from which this input was made matched Genome in a Bottle's benchmark (Zook et al., 2019; Wagner et al., 2022) with a precision of 0.9992 and a recall of 0.9966.

Fitting one genome per thread on a shared node, psmc-rs took a median of 6.22 minutes per genome (Fig. 2c) and 257 core-hours for the cohort. C psmc, run the same way in the same job on the 52 genomes of the comparison subset, took a median of 88.0 minutes per genome, which would make about 3,700 core-hours for the cohort. On these 52 genomes, Δ between the two programs was 5.7×10^{-6} to 2.1×10^{-5} (Table 1).

The histories have the features of published single-genome PSMC analyses of human genomes (Fig. 3a; Li and Durbin, 2011; Prüfer et al., 2014; Mallick et al., 2016). The median curves of the four superpopulations outside Africa fell to minima of 1,961 to 2,902 between 43 and 52 thousand years ago, whereas the African median fell only to 9,422. From 247 thousand years ago to the end of the window, 4.6 million years ago, the five superpopulations' medians were within 20% of one another.

Within each superpopulation, the population medians differed most over the span of the first free λ (drawn pale in Fig. 3), which one genome constrains poorly. There, the largest difference between two populations was 0.24 to 0.99 in $\log_{10} N_e$, depending on the superpopulation. When we split the first free λ into four, in one genome per population, the four values spanned a median of 1.1 in $\log_{10} N_e$: the single λ averages over sizes

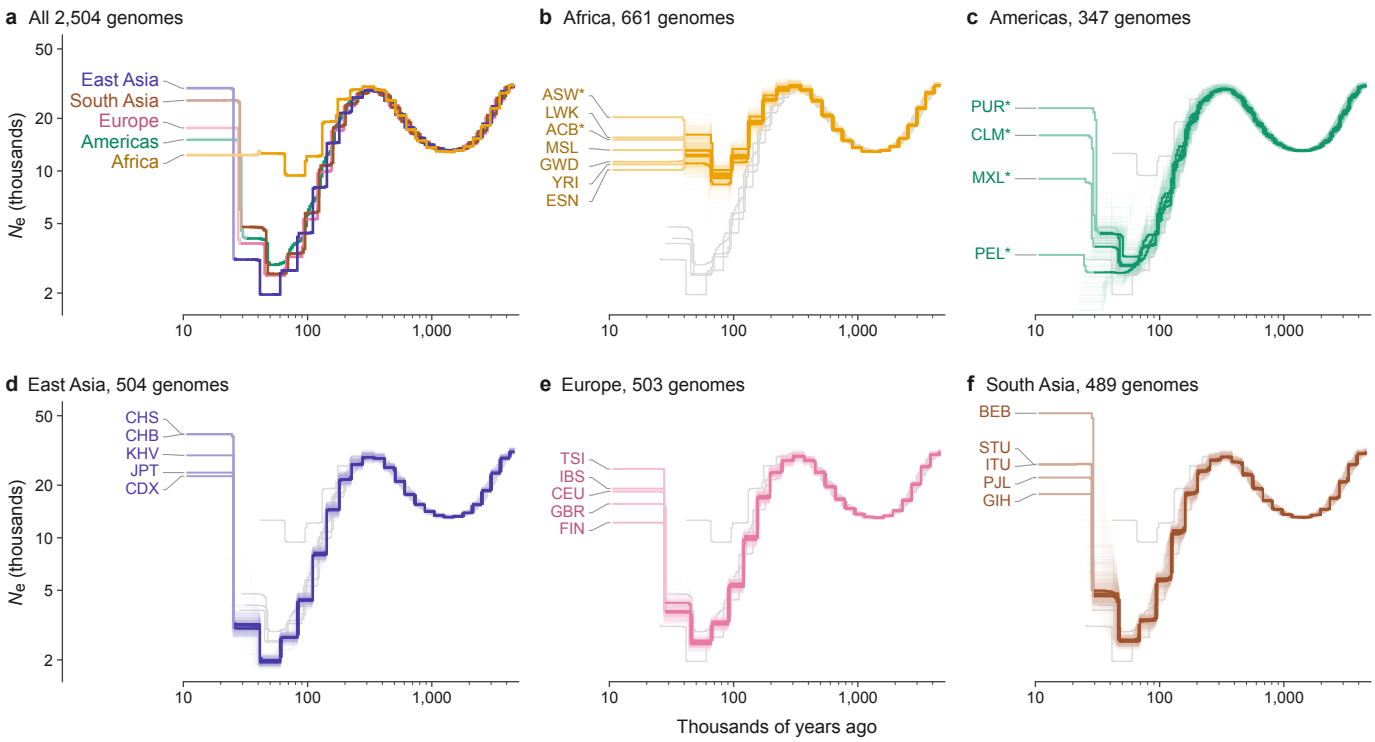


Figure 3. Population size histories of all 2,504 unrelated individuals in the high-coverage 1000 Genomes data, each genome fitted by psmc-rs (E7). (a) The median over each superpopulation's genomes. (b–f) One superpopulation each: its genomes (faint lines, which merge into a soft band where they agree), the median over each of its populations' genomes (61 to 113 genomes per population; 1000 Genomes population codes; *, admixed) and, in grey, the other four superpopulations' medians from a. Medians are pointwise, at each time of a common grid, and span the cohort's informative window, from 10.6 thousand to 4.6 million years ago: the medians over genomes of each genome's 1% and 99% quantiles of the pairwise coalescence time. Pale: PSMC's first free λ, a single parameter for the four most recent intervals, which one genome constrains poorly. Each median is pale until nine in ten of its genomes have passed the span of their first free λ; each genome starts where its own first free λ ends, and each grey median where it turns full in a. The fits are of the `psmcfa` inputs, which take every called bin as full; weighting bins by their callable bases raised the curves of the comparison subset almost uniformly, by a median of 0.044 in $\log_{10} N_e$ (Results). $\mu = 1.25 \times 10^{-8}$ per bp per generation, $g = 29$ years.

that, fitted separately, differ more than tenfold, though each of the four is itself poorly constrained. After the first free λ, the population medians stayed close (Fig. 3b–f). From its end to 150 thousand years ago, through the bottleneck and the rise after it, the highest and the lowest differed by a median over time of 0.015 to 0.084 in $\log_{10} N_e$, depending on the superpopulation, and by at most 0.19 to 0.24 at any time. At older times, this median was at most 0.015 (largest single value, 0.11). In East Asia, Europe and South Asia, the larger differences were brief: they fell where the populations' steps fell at slightly different times. Individual genomes were close too: after the first free λ, the middle 80% of a superpopulation's genomes spanned a median of 0.035 to 0.040 in $\log_{10} N_e$ outside the admixed Americas (Fig. 3b–f).

Fitting the 52 genomes of the comparison subset from their VCFs, with each bin weighted by its callable bases, raised the scaled mutation rate by a median of 8.5% over their `psmcfa` fits. This is close to the 8.2% expected if θ scales with the callable bases: a median of 7.6% of the bases in called bins were uncallable. The VCF fits lay almost uniformly above the `psmcfa` fits: interval by interval, by a median of 0.044 in $\log_{10} N_e$, with a median SD of 0.008 around that shift (Methods).

The Project's own PSMC release holds only the output of fits made from the low-coverage (mean 7.4×), imputed genotypes of the same individuals, and from high-coverage genomes of only 26 individuals (The 1000 Genomes Project Consortium, 2015a,b). Here, every genome was fitted from its high-coverage sequence, and the `psmcfa` inputs and the fits of all 2,504 genomes are released as a resource, with the provenance of every fit

(Data availability).

Discussion

psmc-rs reproduces C psmc's fits. On the same inputs, its histories differ from C psmc's by far less than the method's own error and than the sampling uncertainty of a fit. Most of the remaining difference comes from one bookkeeping choice, the last bin of each sequence: with that bin left out, as C psmc leaves it out, the two programs differ about as little as C psmc's own fits on two platforms. Analyses of inputs like these can therefore move to psmc-rs, and published PSMC analyses can be repeated or extended with it, without their conclusions changing because of the program. A VCF read with its callable mask is the exception: by default psmc-rs weights its partly callable bins, which C psmc's model does not, and `--bin-callable ignore` fits it as C psmc's input would be fitted. We tested one pattern of parameters and 25 EM iterations, which do not reach convergence (Fig. S1), on genomes of up to 1,220 sequences. In Tasmanian devils, PSMC curves changed little with assembly quality (Patton et al., 2019). Among the four genomes of E2, however, the difference between the two programs grew with the number of sequences, so fits of highly fragmented assemblies are worth comparing with C psmc's.

psmc-rs is an order of magnitude faster than C psmc on one thread and, on genomes of many sequences, two orders of magnitude faster on many threads. It needs a small fraction of C psmc's memory and gives the same results on any number of threads. For one genome, threads cut the wait. For a cohort or a bootstrap, which run many fits side by side, the speed per core and the smaller memory count: the 2,504 genomes took 257 core-hours, where C psmc would have needed about 3,700.

Because it keeps PSMC's model, psmc-rs also keeps PSMC's limits. One genome says little about the last few thousand generations, where the first free λ can run away; psmc-rs bounds it and warns when it does. Ancestral population structure can produce changes in a single-genome curve that are not changes in size (Mazet et al., 2016; Cousins et al., 2025), and methods that combine many genomes, such as MSMC2 (Schiffels and Wang, 2020), SMC++ (Terhorst et al., 2017) or PHLASH (Terhorst, 2025), resolve recent times better. The bootstrap measures sampling variation, not these biases (Fig. S3). Finally, because psmc-rs parallelizes across sequences, a genome assembled into a few long sequences gains less from many threads than one assembled into many.

For single genomes, psmc-rs makes the reference method fast without changing its results. Its machinery, a checkpointed E-step run in parallel across sequences with results that do not depend on the number of threads, could also serve MSMC2 (Schiffels and Wang, 2020), whose composite likelihood sums PSMC-like hidden Markov models over pairs of haplotypes within and between populations.

Methods

Model

psmc-rs fits the model of Li and Durbin (2011) with the conventions of C psmc 0.6.5-r74 (Li, 2025). Time is measured in units of $2N_0$ generations and divided into $n+1$ intervals $[t_k, t_{k+1})$, with $t_k = \alpha(e^{\beta k} - 1)$ for $k = 0, \ldots, n-1$, $\beta = \ln(1 + t_{\max}/\alpha)/n$, $\alpha = 0.1$, $t_n = t_{\max}$ and t_{n+1} a large value standing for infinity (1,000, as in C psmc). The population size in interval k is $\lambda_k N_0$. The free parameters are the scaled mutation rate θ and recombination rate ρ per bin, $t_{\max}$, and the free λs, each spanning consecutive intervals according to a pattern: `4+25*2+4+6` has 64 intervals and 28 free λs, the first spanning the four most recent intervals. The initial distribution, the transition probabilities and the expected coalescence time $\bar{t}_k$ in each interval are computed

exactly as C psmc computes them (psmc-rs's code is a line-by-line port of its `psmc_update_hmm`). A bin of s bases is missing (N) if more than 90% of its bases are uncallable, as in `fq2psmcfa`; otherwise it is heterozygous (K) if any of its callable sites is heterozygous, and homozygous (T) if none is. The emission probabilities are $e_T(k) = e^{-\theta \bar{t}_k}$, $e_K(k) = 1 - e_T(k)$ and $e_N(k) = 1$. Sizes and times are scaled with a mutation rate μ per bp per generation, the bin size s and a generation time g: $N_0 = \theta/(4\mu s)$, $N_e = \lambda_k N_0$, and a time t is $2N_0 tg$ years.

When the input records how many of a bin's s bases are callable, c, psmc-rs uses

$$P(T \mid k, c) = e_T(k)^{c/s} = e^{-\theta \bar{t}_k c/s}, \qquad P(K \mid k, c) = 1 - P(T \mid k, c), \tag{1}$$

the probability that c callable bases hold no heterozygous site under the Poisson bin likelihood of PSMC+ and cobraa (Cousins et al., 2024, 2025), restricted to PSMC's binary observations (`--bin-callable auto`, the default). Full bins, and inputs without counts, are fitted with C psmc's model. A VCF read with a callable mask gives c as s minus the bin's positions outside the mask, inside a negative mask, at records that failed the callset's filters, with a missing or partial genotype, or in the reference span of a heterozygous variant that is not a single-nucleotide variant. A multihetsep file gives only the number of callable sites between two of its rows; psmc-rs places the uncallable ones at the start of each such stretch, and `--bin-callable auto` takes the bins that result as full.

Inference

E-step. The forward–backward algorithm runs in double precision on each sequence, with the forward vector normalized at every bin; the log-likelihood is the sum of the logarithms of the normalizers. To bound memory, the forward vector is kept every 128 bins and the vectors between these checkpoints are recomputed, to the same bits, during the backward pass. For models of up to 256 states the forward and backward passes take time linear in the number of states, from the rank-one structure of PSMC's transition matrix (Harris et al., 2014). So do the expected transition counts, for models of any size: below and above the diagonal, a transition probability is the product of a factor of its row and a factor of its column, so C psmc's objective needs the counts only summed over the rows and columns of each triangle and over the diagonal, and the E-step adds these up from prefix and suffix sums at every bin. C psmc's `.psmc` output also reports the objective less its largest value over all models (its `QD` line), which needs every count; psmc-rs's leaves that line out. The expected emission counts include the last bin of every sequence, which C psmc leaves out of them. On x86-64 processors with AVX2 and FMA, psmc-rs chooses kernels compiled for them when it starts, and portable kernels otherwise; on aarch64 it uses NEON kernels. The JSON output and the HTML report record which kernels ran.

M-step. psmc-rs maximizes C psmc's objective, the expected complete-data log-likelihood of the transitions and emissions, over all free parameters with L-BFGS (Liu and Nocedal, 1989). It uses a memory of 20, an Armijo line search and gradients from a hand-written adjoint, on logits within the bounds $\theta, \rho \in [10^{-12}, 10]$ per bin, $t_{\max} \in [1, 200]$ and $\lambda \in [10^{-6}, 10^4]$. It stops when, in each of two consecutive iterations, the objective improved by at most a relative 10^{-13} and the next quasi-Newton step predicts no larger improvement, or when its line search finds no improvement; before stopping, it restarts from fresh estimates of the curvature, up to three times, if they predict a larger improvement. It also stops after 1,000 iterations. C psmc instead uses a Hooke–Jeeves direct search (Hooke and Jeeves, 1961) on unbounded parameters. EM starts, as C psmc's does, from $\theta_0 = -\ln(1 - K/L)$, with K the heterozygous and L the non-missing bins (with callable counts, the root of $\sum_i (1 - e^{-\theta_0 c_i/s}) = K$), $\rho_0 = \theta_0/r$ (`--tr-ratio`, 4 by default), $t_{\max}$ (`--t-max`, 15) and every $\lambda = 1$.

Parallelism and determinism. The E-step and the log-likelihood run in parallel over work items, contiguous groups of whole sequences whose grouping depends only on the number of sequences; partial sums are merged in the order of the items, so they do not depend on which thread finished first. The M-step is serial. The block bootstrap (Künsch, 1989) follows C psmc's: it cuts each sequence into blocks of 500,000 bins as `splitfa` does, a remainder shorter than 1.5 blocks staying in the last block, and each replicate draws blocks with replacement as C psmc's `psmc_resamp` does, starting afresh from its own θ_0. Replicates run in parallel, each drawing from its own ChaCha8 stream seeded from the bootstrap seed and its index. Posterior decoding (`--tmrca-out`) gives, for every bin, the posterior mean TMRCA, the most probable state and the states at which the posterior reaches 2.5% and 97.5%.

Experiments

All results come from one pipeline (`experiment/pipeline`) that simulates the genomes, runs both programs with matched settings, records how every output was made and draws every figure from its tables; `experiment/REPRODUCIBILITY.md` in the repository describes how to reproduce them. We name the experiments E1–E7, as their directories in the repository are named. C psmc ran the command of its documentation, `psmc -N25 -t15 -r5 -p "4+25*2+4+6"`: 25 EM iterations, with the pattern and maximum time that Li and Durbin (2011) used for human genomes. psmc-rs 0.2.0 ran the same settings with its own options. Both used 100 bp bins and one thread unless stated otherwise. Every seed was fixed in advance, from the SHA-256 of a string naming the experiment, model, replicate and stage.

Comparisons. The fits of the two programs to the same input have the same intervals, whose boundaries differ in years only through θ and $t_{\max}$, and are compared interval by interval. For each interval k of C psmc's fit that overlaps the window $[a, b]$, $d_k = \log_{10}(N_k^{\mathrm{rs}}/N_k^{\mathrm{C}})$, where N_k^{rs} and N_k^{C} are the two fits' N_{e} in interval k, weighted by $w_k = \ln(\min(e_k, b)/\max(s_k, a))$ for an interval from s_k to e_k; their RMSE, Δ, is $(\sum_k w_k d_k^2 / \sum_k w_k)^{1/2}$. The window spans the 1% to 99% quantiles of the pairwise coalescence time under the true model (simulations) or under the fitted one (C psmc's where the programs are compared, psmc-rs's otherwise). A cohort's window spans the medians, over its genomes, of these quantiles: for the 1000 Genomes cohort, 10.6 thousand to 4.6 million years ago. Fits of different inputs, whose θ, and so whose time scales, differ, are compared at the same times (E4), except the VCF and `psmcfa` fits of a 1000 Genomes genome. These are compared interval by interval, as above, with $d_k = \log_{10}(N_k^{\mathrm{VCF}}/N_k^{\mathrm{psmcfa}})$ on the `psmcfa` fit's intervals and window: the weighted mean $\bar{d} = \sum_k w_k d_k / \sum_k w_k$ is the shift between the two fits, and the weighted SD, $(\sum_k w_k (d_k - \bar{d})^2 / \sum_k w_k)^{1/2}$, the change of shape. A fit is compared with the truth by the RMSE of $\log_{10} N_{\mathrm{e}}$ over 200 log-spaced times in the window.

Simulated genomes (E1; Fig. 1, Table S1, Fig. S1). Four models, in diploid sizes and generations: constant size 10,000; a bottleneck (20,000 for 0–800 generations, 1,000 to 1,200, 10,000 to 4,000, 5,000 to 40,000, then 10,000); an expansion (10,000 for 0–400 generations, 1,000 to 2,400, 10,000 to 8,000, 5,000 to 40,000, 10,000 to 80,000, then 20,000); and the zigzag model of Schiffels and Durbin (2014) from stdpopsim 0.3.0 (Adrion et al., 2020). Each of 10 replicates per model was 10 unlinked chromosomes of 50 Mb, simulated for one diploid individual with msprime 1.4.4 (Kelleher et al., 2016; Baumdicker et al., 2022) under the coalescent with recombination, with $\mu = 2.5 \times 10^{-8}$ and $r = 10^{-8}$ per bp per generation, and scaled with $g = 25$ years.

The residual difference (E1). We traced the residual difference from C psmc on E1's 40 genomes with builds of psmc-rs modified for this purpose, which are not part of the release (`experiment/c_equivalence`). Each

change described in Results was made alone: the last bin of each sequence was left out of the expected emission counts, as C psmc leaves it out, or the M-step was replaced by a port of C psmc's Hooke–Jeeves search. psmc-rs's default fits were within a Δ of 2.5×10^{-6} (median 1.0×10^{-7}) of fits whose M-steps ran to the limit of precision. The port of C psmc's search, run with the last bin also left out, was traced on an Apple M1 Pro laptop against C psmc's own search in the first M-step on the repository's test input: it evaluated the same parameters, bit for bit, until the two searches ranked differently two points whose objectives differed by less than the two programs' rounding.

Real genomes (E2; Fig. 1c–e). NA12878's `psmcfa` file is the one made for E7 (below). The three turtles are side-necked turtles (Pleurodira): *Emydura subglobosa*, *Hydromedusa tectifera* and *Pelusios castaneus*. The turtle `psmcfa` files are those of Hilgers et al. (2025). The turtles were scaled with $\mu = 4.6 \times 10^{-9}$, the mean of nine trios of the painted turtle (Bergeron et al., 2023), and $g = 15$ years, close to the generation times that Sozzoni et al. (2025) used for two of the species; NA12878 with $\mu = 1.25 \times 10^{-8}$ (Scally and Durbin, 2012) and $g = 29$ years (Fenner, 2005). Each genome was checked with `psmc-rs qc` and bootstrapped with 100 replicates on 50 threads, in blocks of 50 Mb as `splitfa` cuts them (a sequence's last block keeps a remainder shorter than 1.5 blocks), which made every block of *P. castaneus* a whole scaffold. The turtles were also refitted with the pattern's first free λ split into two (`2+2+25*2+4+6`) or four (`1+1+1+1+25*2+4+6`).

Performance (E3; Fig. 2, Table 2). Genomes of 1, 3, 10 and 30 chromosomes of 100 Mb were simulated under constant size. C psmc ran on one thread and psmc-rs on 1, 8 and 30 threads, and also on 2, 4 and 16 threads on 3,000 Mb: 19 setups of five repeats, each repeat on an exclusive AMD EPYC 7702 node (four nodes in all). Before each run the node had to be quiet, with at most one core and 10% of cores busy over 2 s; during the run, other processes could use at most 0.5 core and 5% of cores, or the run was repeated later. Peak memory, in MB of 2^{20} bytes, is the larger of the program's `ru_maxrss` and its process tree's resident memory sampled every 0.5 s. Outputs (C psmc's `.psmc`, psmc-rs's JSON) were compared byte for byte with the same program's output on the fewest threads in the first repeat.

Partly callable bins, bootstrap and decoding (E4–E6; Figs S2 and S3). E4 masked five bottleneck genomes of E1's size with uncallable runs of mean 20 kb (clustered) or 20 bp (dispersed) covering 10% or 30% of positions. It compared psmc-rs on the VCF with its mask, with and without weighting; a consensus sequence converted by `fq2psmcfa`, which gave the unweighted fit exactly; multihetsep converted from the VCF and its mask; and the VCF without its mask, each against the fit of the same genome without missing data. E5 simulated 50 bottleneck genomes of 6×50 Mb, fitted each, and bootstrapped 10 of them with 50 replicates, with blocks of 50 and 5 Mb. E6 read E1's first replicate of each model as `psmcfa`, as VCF and as multihetsep, and decoded E1's first bottleneck replicate with its fitted model (`--tmrca-out` on 4 threads) against its true TMRCA, re-simulated from the recorded seeds.

1000 Genomes (E7; Fig. 3). The inputs were the joint-called VCFs of the 3,202 high-coverage samples (Byrska-Bishop et al., 2022) for chromosomes 1–22 (123,574,606 records), of which we used the 2,504 unrelated individuals; the GRCh38 strict accessibility mask (The 1000 Genomes Project Consortium, 2015a); and, as a negative mask shared by all samples, the position of every record whose FILTER was neither PASS nor '.' (11,256,876 records). Each chromosome was downloaded and checked against its published size and MD5 checksum, read in chunks of 15 Mb with `bcftools` 1.24 (Danecek et al., 2021) and split into one VCF per sample of its heterozygous or missing genotypes; `psmc-rs convert --vcf-mode mask` then wrote each sample's

psmcfa. For NA18507, NA19239 and NA12878, chromosome 21 converted directly from the joint VCF gave files byte-identical to those made through the per-sample VCFs. Every genome was fitted from its psmcfa. The comparison subset, NA18507, NA19239, NA12878 and others drawn by seed to make two genomes per population (52 in all), was also fitted by C psmc, and from its VCFs with the masks and `--bin-callable auto`. In each population, the first genome of the subset (NA18507 in YRI, NA12878 in CEU, elsewhere the first drawn) was also fitted with four free λs in the four most recent intervals (`1+1+1+1+25*2+4+6`), and NA18507, NA19239 and NA12878 were bootstrapped (100 replicates on 32 threads). Core-hours are the summed wall times of fits that ran on one thread each; C psmc's for the cohort are its median wall time per genome times 2,504. Population and superpopulation curves are the medians of N_e across individuals, at each time of a common grid over the cohort's window. At each time, the populations' spread in a superpopulation is $\log_{10}$ of the highest over the lowest of its population medians, and its genomes' spread is $\log_{10}$ of the 90% over the 10% quantile of its genomes. We give the largest value of the populations' spread until the first free λ has ended in each of the superpopulation's populations (at the median over the population's genomes). After that, we give medians over the grid times: of the populations' spread up to 150 thousand years ago and at older times, and of the genomes' spread at all times. For the comparison with Genome in a Bottle, NA12878's PASS records with a heterozygous genotype of single-base alleles were matched by position and alleles with the heterozygous single-nucleotide variants of HG001's GRCh38 v4.2.1 benchmark (Wagner et al., 2022) inside both the strict mask and the benchmark's regions (2,109 Mb).

Software and computing. The simulations and analyses used Python 3.13.13 with msprime 1.4.4, tskit 1.0.3, stdpopsim 0.3.0, NumPy 2.5.3, SciPy 1.18.1, pandas 3.0.6 and Matplotlib 3.11.2, and htslib and bcftools 1.24. psmc-rs 0.2.0 was built with rustc 1.99.0 and the release profile, without build flags; every psmc-rs fit on the cluster used the AVX2 and FMA kernels. To compare platforms, both programs refitted E1's genomes from byte-identical inputs on an Apple M1 Pro laptop (arm64, macOS): psmc-rs built from the same source code by rustc 1.97.1, with its NEON kernels, and C psmc built by Apple clang with `-O1`, which fuses multiply-adds (`experiment/crossplatform`). C psmc was lh3/psmc 0.6.5-r74, from sources identical to its commit 82fa2ad, with `fq2psmcfa` from the same sources, built with gcc 11.4.1 and `-O3 -march=x86-64-v3 -ffp-contract=off`, whose output was byte-identical to that of the Makefile's `-O2` build; `generate_multihetsep.py` came from msmc-tools (Schiffels, 2023). The fits ran on a computing cluster, on AMD EPYC 7702 nodes of 128 cores, and the figures were drawn from the tables on an Apple M1 Pro laptop. Every output has a provenance record of its command, settings, seeds, the checksums of its inputs and outputs, the identities of the tools and how the job ran.

Data availability

The 1000 Genomes high-coverage VCFs are available from the International Genome Sample Resource (`https://www.internationalgenome.org`), the strict accessibility mask from the 1000 Genomes Project, and Genome in a Bottle's HG001 v4.2.1 benchmark from NIST. The inputs of the 1000 Genomes analysis, the psmcfa files of the 2,504 genomes, the comparison subset's VCFs and the masks, are available on Zenodo (`https://doi.org/10.5281/zenodo.23764975`); the manifest of their sources, commands and checksums, every fit, bootstrap and table, and the turtle and NA12878 psmcfa files of E2 are in the repository (`experiment/runs`, `experiment/real_data`). The turtle psmcfa files come from Hilgers et al. (2025).

Code availability

psmc-rs is available at `https://github.com/larryivan/psmc-rs` under the MIT license; the release used here, psmc-rs 0.2.0, is archived on Zenodo (`https://doi.org/10.5281/zenodo.23762893`). The experiment pipeline, its configuration, the committed results of every experiment and the code that draws every figure are in the same repository (`experiment/`); `experiment/REPRODUCIBILITY.md` describes how to reproduce them. The numbers quoted in this article are generated from the committed results by `manuscript/make_numbers.py`.

Acknowledgements

We thank Dr Ke Wang and the members of her laboratory at Fudan University for their support. We thank the 1000 Genomes Project, the New York Genome Center, Genome in a Bottle and Hilgers et al. (2025) for making their data available.

Funding

This work received no specific funding.

Conflict of interest

The author declares no conflict of interest.

Use of AI tools

The text of this manuscript was polished with the assistance of AI. The author takes full responsibility for the results and the content of the article.

References

Jeffrey R. Adrion, Christopher B. Cole, Noah Dukler, Jared G. Galloway, Ariella L. Gladstein, Graham Gower, Christopher C. Kyriazis, Aaron P. Ragsdale, Georgia Tsambos, Franz Baumdicker, et al. A community-maintained standard library of population genetic models. *eLife*, 9:e54967, 2020. doi: 10.7554/elife.54967.

Franz Baumdicker, Gertjan Bisschop, Daniel Goldstein, Graham Gower, Aaron P. Ragsdale, Georgia Tsambos, Sha Zhu, Bjarki Eldon, E. Castedo Ellerman, Jared G. Galloway, et al. Efficient ancestry and mutation simulation with msprime 1.0. *Genetics*, 220(3):iyab229, 2022. doi: 10.1093/genetics/iyab229.

Lucie A. Bergeron, Søren Besenbacher, Jiao Zheng, Panyi Li, Mads Frost Bertelsen, Benoit Quintard, Joseph I. Hoffman, Zhipeng Li, Judy St. Leger, Changwei Shao, et al. Evolution of the germline mutation rate across vertebrates. *Nature*, 615(7951):285–291, 2023. doi: 10.1038/s41586-023-05752-y.

Marta Byrska-Bishop, Uday S. Evani, Xuefang Zhao, Anna O. Basile, Haley J. Abel, Allison A. Regier, André Corvelo, Wayne E. Clarke, Rajeeva Musunuri, Kshithija Nagulapalli, et al. High-coverage whole-genome sequencing of the expanded 1000 Genomes Project cohort including 602 trios. *Cell*, 185(18): 3426–3440.e19, 2022. doi: 10.1016/j.cell.2022.08.004.

Trevor Cousins, Daniel Tabin, Nick Patterson, David Reich, and Arun Durvasula. Accurate inference of population history in the presence of background selection. *bioRxiv*, 2024. doi: 10.1101/2024.01.18.576291. Preprint.

Trevor Cousins, Aylwyn Scally, and Richard Durbin. A structured coalescent model reveals deep ancestral structure shared by all modern humans. *Nature Genetics*, 57(4):856–864, 2025. doi: 10.1038/s41588-025-02117-1.

Petr Danecek, James K. Bonfield, Jennifer Liddle, John Marshall, Valeriu Ohan, Martin O. Pollard, Andrew Whitwham, Thomas Keane, Shane A. McCarthy, Robert M. Davies, et al. Twelve years of SAMtools and BCFtools. *GigaScience*, 10(2):giab008, 2021. doi: 10.1093/gigascience/giab008.

Jack N. Fenner. Cross-cultural estimation of the human generation interval for use in genetics-based population divergence studies. *American Journal of Physical Anthropology*, 128(2):415–423, 2005. doi: 10.1002/ajpa.20188.

Kelley Harris, Sara Sheehan, John A. Kamm, and Yun S. Song. Decoding coalescent hidden Markov models in linear time. In Roded Sharan, editor, *Research in Computational Molecular Biology (RECOMB 2014)*, volume 8394 of *Lecture Notes in Computer Science*, pages 100–114. Springer International Publishing, 2014. doi: 10.1007/978-3-319-05269-4_8.

Leon Hilgers, Shenglin Liu, Axel Jensen, Thomas Brown, Trevor Cousins, Regev Schweiger, Katerina Guschanski, and Michael Hiller. Avoidable false PSMC population size peaks occur across numerous studies. *Current Biology*, 35(4):927–930.e3, 2025. doi: 10.1016/j.cub.2024.09.028.

Robert Hooke and T. A. Jeeves. "Direct search" solution of numerical and statistical problems. *Journal of the ACM*, 8(2):212–229, 1961. doi: 10.1145/321062.321069.

Jerome Kelleher, Alison M. Etheridge, and Gilean McVean. Efficient coalescent simulation and genealogical analysis for large sample sizes. *PLOS Computational Biology*, 12(5):e1004842, 2016. doi: 10.1371/journal.pcbi.1004842.

Kevin Korfmann. smckit: a preservation-first interface for SMC methods, version 0.0.1b2. `https://github.com/kevinkorfmann/smckit`, commit b5f0a20, 2026.

Hans R. Künsch. The jackknife and the bootstrap for general stationary observations. *The Annals of Statistics*, 17(3):1217–1241, 1989. doi: 10.1214/aos/1176347265.

Heng Li. psmc: implementation of the pairwise sequentially Markovian coalescent (PSMC) model, version 0.6.5-r74. `https://github.com/lh3/psmc`, commit 82fa2ad, 2025.

Heng Li and Richard Durbin. Inference of human population history from individual whole-genome sequences. *Nature*, 475(7357):493–496, 2011. doi: 10.1038/nature10231.

Mark Lipson, Po-Ru Loh, Sriram Sankararaman, Nick Patterson, Bonnie Berger, and David Reich. Calibrating the human mutation rate via ancestral recombination density in diploid genomes. *PLOS Genetics*, 11(11): e1005550, 2015. doi: 10.1371/journal.pgen.1005550.

Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1–3):503–528, 1989. doi: 10.1007/bf01589116.

Junfeng Liu, Xianchao Ji, and Hua Chen. Beta-PSMC: uncovering more detailed population history using beta distribution. *BMC Genomics*, 23(1):785, 2022. doi: 10.1186/s12864-022-09021-6.

Artem Lomakin. psmc-python: a reimplementation of the PSMC method in Python. `https://github.com/tulerpetontidae/psmc-python`, commit 5a6484f, 2024.

Swapan Mallick, Heng Li, Mark Lipson, Iain Mathieson, Melissa Gymrek, Fernando Racimo, Mengyao Zhao, Niru Chennagiri, Susanne Nordenfelt, Arti Tandon, et al. The Simons Genome Diversity Project: 300 genomes from 142 diverse populations. *Nature*, 538(7624):201–206, 2016. doi: 10.1038/nature18964.

O. Mazet, W. Rodríguez, S. Grusea, S. Boitard, and L. Chikhi. On the importance of being structured: instantaneous coalescence rates and human evolution—lessons for ancestral population size inference? *Heredity*, 116(4):362–371, 2016. doi: 10.1038/hdy.2015.104.

Krystyna Nadachowska-Brzyska, Reto Burri, Linnéa Smeds, and Hans Ellegren. PSMC analysis of effective population sizes in molecular ecology and its application to black-and-white *Ficedula* flycatchers. *Molecular Ecology*, 25(5):1058–1072, 2016. doi: 10.1111/mec.13540.

Emmanuel Paradis and Heng Li. psmcr: R port of psmc, version 0.1-4. `https://github.com/emmanuelparadis/psmcr`, commit a784ee7, 2023.

Austin H. Patton, Mark J. Margres, Amanda R. Stahlke, Sarah Hendricks, Kevin Lewallen, Rodrigo K. Hamede, Manuel Ruiz-Aravena, Oliver Ryder, Hamish I. McCallum, Menna E. Jones, et al. Contemporary

demographic reconstruction methods are robust to genome assembly quality: a case study in Tasmanian devils. *Molecular Biology and Evolution*, 36(12):2906–2921, 2019. doi: 10.1093/molbev/msz191.

Kay Prüfer, Fernando Racimo, Nick Patterson, Flora Jay, Sriram Sankararaman, Susanna Sawyer, Anja Heinze, Gabriel Renaud, Peter H. Sudmant, Cesare de Filippo, et al. The complete genome sequence of a Neanderthal from the Altai Mountains. *Nature*, 505(7481):43–49, 2014. doi: 10.1038/nature12886.

Aylwyn Scally and Richard Durbin. Revising the human mutation rate: implications for understanding human evolution. *Nature Reviews Genetics*, 13(10):745–753, 2012. doi: 10.1038/nrg3295.

Stephan Schiffels. msmc-tools: tools and utilities for MSMC and MSMC2. `https://github.com/stschiff/msmc-tools`, commit 0d1d169, 2023.

Stephan Schiffels and Richard Durbin. Inferring human population size and separation history from multiple genome sequences. *Nature Genetics*, 46(8):919–925, 2014. doi: 10.1038/ng.3015.

Stephan Schiffels and Ke Wang. MSMC and MSMC2: the multiple sequentially Markovian coalescent. In Julien Y. Dutheil, editor, *Statistical Population Genomics*, volume 2090 of *Methods in Molecular Biology*, pages 147–166. Springer US, New York, NY, 2020. doi: 10.1007/978-1-0716-0199-0_7.

Marcella Sozzoni, Jennifer Balacco, Massimo Bellavita, Anna Brüniche-Olsen, Giulio Formenti, Nivesh Jain, Bonhwang Koo, Jacquelyn Mountcastle, Marc Palmada-Flores, Vladimir Trifonov, et al. Quaternary habitat fluctuations and demographic dynamics in turtles inferred from environmental niche modelling and whole genome data. *Molecular Ecology Resources*, 25(8):e70040, 2025. doi: 10.1111/1755-0998.70040.

Jonathan Terhorst. Accelerated Bayesian inference of population size history from recombining sequence data. *Nature Genetics*, 57(10):2570–2577, 2025. doi: 10.1038/s41588-025-02323-x.

Jonathan Terhorst, John A. Kamm, and Yun S. Song. Robust and scalable inference of population history from hundreds of unphased whole genomes. *Nature Genetics*, 49(2):303–309, 2017. doi: 10.1038/ng.3748.

The 1000 Genomes Project Consortium. A global reference for human genetic variation. *Nature*, 526(7571): 68–74, 2015a. doi: 10.1038/nature15393.

The 1000 Genomes Project Consortium. PSMC output for Phase 3 individuals. `https://ftp.1000genomes.ebi.ac.uk/vol1/ftp/release/20130502/supporting/psmc/`, 2015b. Tarballs `psmc_low_coverage_20141021.tar.gz` and `psmc_26_wgs_pcr_free_20141103.tar.gz`, and their README (A. Auton, 21 July 2015).

Justin Wagner, Nathan D. Olson, Lindsay Harris, Ziad Khan, Jesse Farek, Medhat Mahmoud, Ana Stankovic, Vladimir Kovacevic, Byunggil Yoo, Neil Miller, et al. Benchmarking challenging small variants with linked and long reads. *Cell Genomics*, 2(5):100128, 2022. doi: 10.1016/j.xgen.2022.100128.

Justin M. Zook, Jennifer McDaniel, Nathan D. Olson, Justin Wagner, Hemang Parikh, Haynes Heaton, Sean A. Irvine, Len Trigg, Rebecca Truty, Cory Y. McLean, et al. An open resource for accurately benchmarking small variant and reference calls. *Nature Biotechnology*, 37(5):561–566, 2019. doi: 10.1038/s41587-019-0074-6.

Supplementary information

Figure S1 and Table S1 (experiment E1), Figure S2 (E4) and Figure S3 (E5). As in the main text, every result in the legends is generated from the committed results.

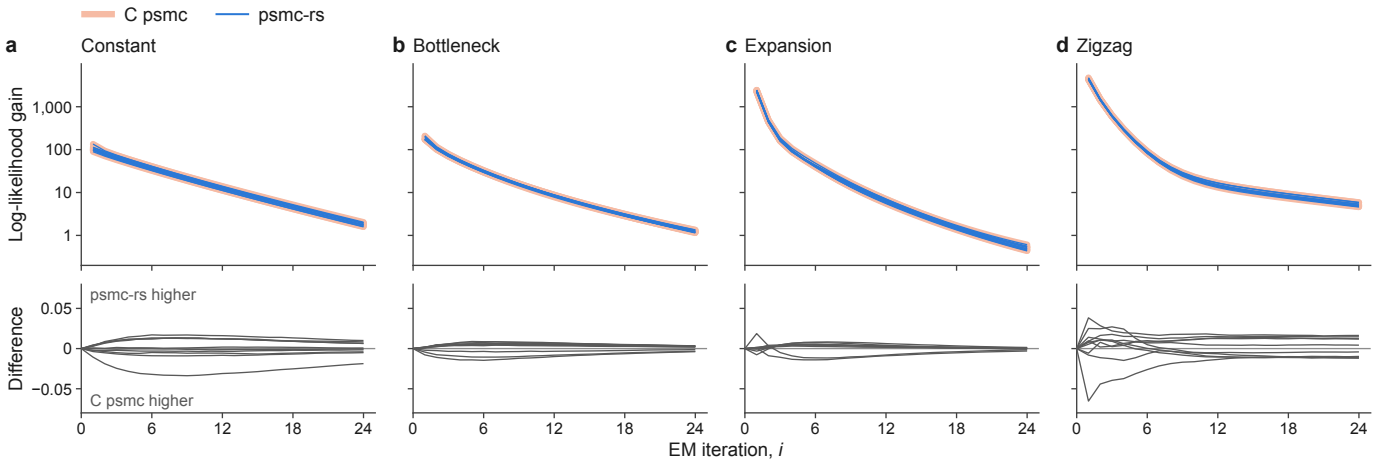


Figure S1. C psmc and psmc-rs climb the same likelihood at every EM iteration (E1). One column per simulated model, 10 genomes each. Above: the log-likelihood gained in iteration i, $\mathrm{LL}(i) - \mathrm{LL}(i-1)$, one line per genome and program (C psmc wide and pale underneath, psmc-rs thin and blue on top). All 1,920 gains were positive: the log-likelihood never decreased in either program, and in iteration 24 it still rose by 0.43 to 5.9 units. Below: psmc-rs's log-likelihood after iteration i minus C psmc's, one line per genome. From the same starting model ($i = 0$) the two agreed within the rounding of the six decimals C psmc prints (at most 4.9×10^{-7}). After that they differed by at most 0.065; in each panel and iteration, the largest difference was at least 84 times smaller than the smallest gain. After iteration 24, psmc-rs's was the higher in 23 of the 40 genomes. Over all iterations, θ and ρ of the two programs differed by at most a relative 3.9×10^{-4} and 3.6×10^{-4}. C psmc prints no log-likelihood after its last iteration, so the panels end at $i = 24$.

Table S1. Accuracy of both programs, and their agreement, on simulated genomes (E1). For each model, 10 genomes of 500 Mb. Window: the informative window of the true model, 1% to 99% of its pairwise coalescence times. RMSE vs truth: each program's RMSE of $\log_{10} N_e$ against the true history over the window (mean ± SD over the genomes). psmc-rs vs C psmc: Δ, the RMSE of $\log_{10} N_e$ between the two programs' fits, interval by interval over the window (median over the genomes), and the largest absolute difference $|d_k|$ in any interval of any genome (Methods).

	Window (years)		RMSE vs truth		psmc-rs vs C psmc	
Model	from	to	C psmc	psmc-rs	Δ (median)	max $\lvert d_k \rvert$
Constant	5,025	2,302,585	0.048 ± 0.023	0.048 ± 0.023	7.8×10^{-6}	9.2×10^{-5}
Bottleneck	10,050	1,322,585	0.291 ± 0.021	0.291 ± 0.021	3.9×10^{-5}	6.6×10^{-4}
Expansion	5,025	1,052,585	0.240 ± 0.043	0.240 ± 0.043	3.0×10^{-5}	4.8×10^{-4}
Zigzag	7,141	2,167,766	0.138 ± 0.013	0.138 ± 0.013	1.9×10^{-5}	6.6×10^{-5}

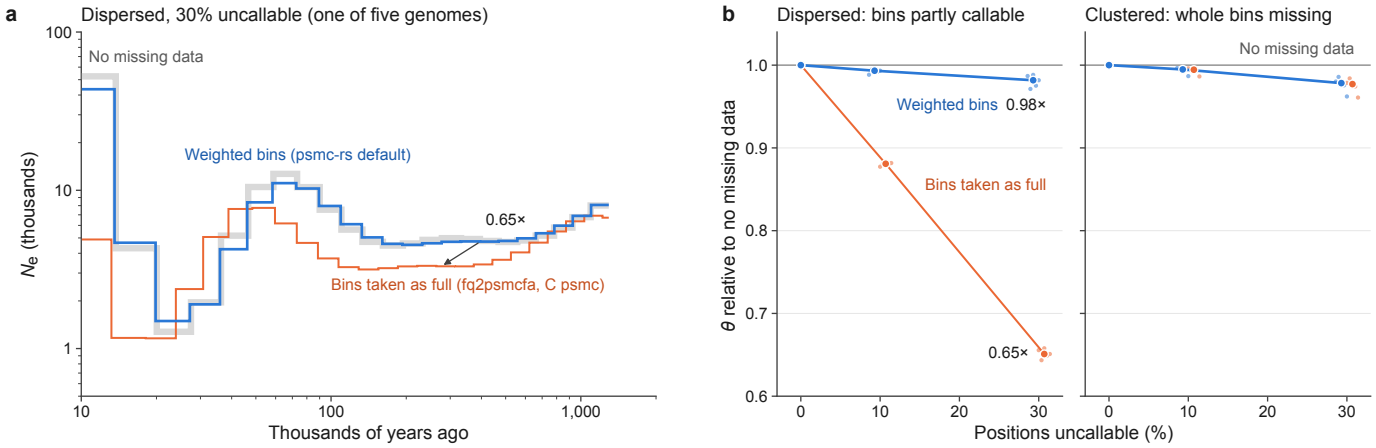


Figure S2. Taking partly callable bins as full biases PSMC; weighting each bin by its callable bases removes most of the bias (E4). We read five simulated bottleneck genomes of 500 Mb without missing data, or from a VCF whose callable mask leaves 10% or 30% of positions uncallable in dispersed (mean run 20 bp) or clustered (mean run 20 kb) runs. Blue: psmc-rs's default, each called bin weighted by its callable bases (`--bin-callable auto`). Orange: C psmc's input, the bins `fq2psmcfa` makes from the consensus sequence, every called bin taken as full (fitted by psmc-rs with C psmc's model; Fig. 1). (a) One genome with the 30% dispersed mask, the one whose weighted fit has the median RMSE against its fit without missing data, and that fit (wide grey), over the model's informative window; the arrow is the factor by which taking bins as full scales θ, and with it every size and time. (b) θ relative to the fit of the same genome without missing data; large circles, medians, joined by lines and offset sideways so that both routes show where they agree (in the clustered facet the orange line, which would lie under the blue, is left out); small dots, replicates; numbers, the dispersed medians with 30% missing. Dispersed runs leave many called bins with only part of their bases: about half (48%) with 10% missing and nearly all (92%) with 30%. Taken as full, they made θ 11.9% and 34.9% low, against 0.7% and 1.8% weighted. Clustered runs empty whole bins, which both models drop. Not drawn: multihetsep input, whose bins are taken as full, made θ 31.9% low with 30% dispersed and 9.1% with 30% clustered missing data; the VCF read without its mask, 35.5% and 56.8%. Simulations: $\mu = 2.5 \times 10^{-8}$ per bp per generation, $g = 25$ years.

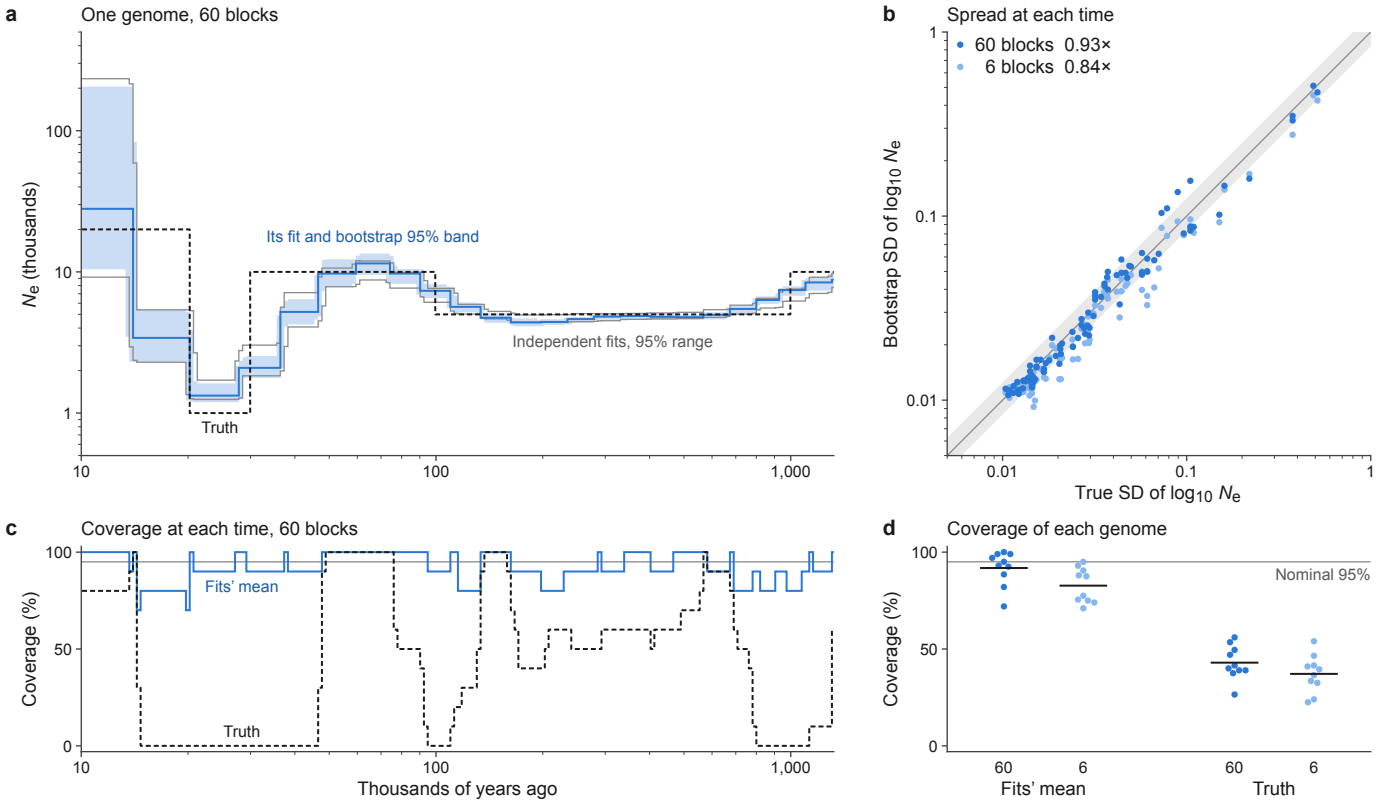


Figure S3. The bootstrap measures the sampling variation of a PSMC fit, not PSMC's bias (E5). The fits of 50 independent bottleneck genomes of 6×50 Mb give the true sampling distribution of the estimate; 10 of them were bootstrapped (50 replicates) in 60 blocks of 5 Mb (blue) or 6 of 50 Mb (light blue); 60 blocks is about what the default 50 Mb blocks give a 3 Gb genome. (a) One bootstrapped genome (60 blocks): its fit and bootstrap 95% band, the truth (dashed) and, in grey, the 95% range of the independent fits, over the model's informative window. (b) Bootstrap SD of $\log_{10} N_e$ (median over the bootstrapped genomes) against the true SD, the SD over the independent fits, one dot per distinct pair of values over the window's times; grey line, identity; grey band, where a bootstrap SD equal to the true SD would fall, given the χ^2 95% interval of an SD estimated from 50 fits; numbers, each block count's median ratio of the two SDs. (c) At each time, the share of the 10 bootstrapped genomes whose band (60 blocks) holds the fits' mean, their geometric mean (blue), or the truth (dashed). (d) The share of the window's times at which each genome's band holds the fits' mean or the truth; dots, genomes; bars, means. Grey lines in c and d, the nominal 95%. The bands are nearly as wide as the fits' spread (bootstrap SD 0.93 and 0.84 times the true one, median over the window) but centred on a biased fit: PSMC smooths sudden changes in size, and the fits' median absolute bias, 0.061 in $\log_{10} N_e$, was about twice their median SD, 0.028, so about as large as the half-width of a 95% band. One demographic model was studied. Simulations: $\mu = 2.5 \times 10^{-8}$ per bp per generation, $g = 25$ years.