Bobay *BMC Bioinformatics* (2020) 21:264
https://doi.org/10.1186/s12859-020-03619-x

BMC Bioinformatics

SOFTWARE Open Access

# *CoreSimul*: a forward-in-time simulator of genome evolution for prokaryotes modeling homologous recombination

Louis-Marie Bobay

Correspondence: ljbobay@uncg.edu
Department of Biology, University of North Carolina Greensboro, 321 McIver Street, PO Box 26170, Greensboro, NC 27402, USA

## Abstract

**Background:** Prokaryotes are asexual, but these organisms frequently engage in homologous recombination, a process that differs from meiotic recombination in sexual organisms. Most tools developed to simulate genome evolution either assume sexual reproduction or the complete absence of DNA flux in the population. As a result, very few simulators are adapted to model prokaryotic genome evolution while accounting for recombination. Moreover, many simulators are based on the coalescent, which assumes a neutral model of genomic evolution, and those are best suited for organisms evolving under weak selective pressures, such as animals and plants. In contrast, prokaryotes are thought to be evolving under much stronger selective pressures, suggesting that forward-in-time simulators are better suited for these organisms.

**Results:** Here, I present *CoreSimul*, a forward-in-time simulator of core genome evolution for prokaryotes modeling homologous recombination. Simulations are guided by a phylogenetic tree and incorporate different substitution models, including models of codon selection.

**Conclusions:** *CoreSimul* is a flexible forward-in-time simulator that constitutes a significant addition to the limited list of available simulators applicable to prokaryote genome evolution.

**Keywords:** Simulator, Genome evolution, Homologous recombination, Prokaryotes

## Background

Many bioinformatic tools rely on genome simulators to infer parameters or to validate new methodologies [1–4]. Although a large diversity of genome simulators have been released, such tools are usually designed for specific tasks and are not adapted to all types of analyses [5–13]. Specifically, relatively few simulators have been implemented to simulate the evolution of prokaryote genomes [1, 4, 14], whose biology differs substantially from eukaryotic organisms [15]. One key difference is the inability of prokaryotes to engage in meiotic recombination. Instead, prokaryotes engage in homologous recombination through gene conversion, which consists in the non-

reciprocal transfer and replacement of a sequence by an homologous one, typically leading to the exchange of short sequences [16–18].

Recent analyses are suggesting a prevalent role of homologous recombination in prokaryotes and the vast majority of bacterial species appears to be impacted by this process, indicating that simulation frameworks incorporating homologous recombination are needed [16, 19–21]. Indeed, the presence of homologous recombination prevents the application of purely clonal frameworks to simulate prokaryote evolution, as well as methods assuming true sexual reproduction [22]. Moreover, the strong selective constraints typically acting on bacterial genome evolution [23, 24] must be taken into account in order to simulate realistic genome datasets.

Multiple coalescent-based simulators have been implemented and several can be applied to simulate prokaryote evolution with homologous recombination [4, 25, 26]. Although coalescent-based simulators offer interesting properties to simulate the evolution of genomic sequences, forward-in-time simulators present alternative qualities that can be better suited for certain tasks. In particular, coalescent-based simulators are designed to simulate sequences under a neutral model of evolution [27], an assumption that is likely violated in prokaryotes, where adaptive evolution could be predominant [28, 29]. Although multiple forward-in-time simulators have been implemented [1, 27], they are rarely adapted to simulate the evolution of prokaryote genomes.

Here I present *CoreSimul*, an efficient forward-in-time simulator to model prokaryotic genome evolution with homologous recombination and selection along phylogenetic trees (https://github.com/lbobay/CoreSimul).

## Implementation

*CoreSimul* generates a set of prokaryote genomes based on a phylogenetic tree. *CoreSimul* allows to simulate both the core genome—the set of genes conserved across all genomes—and the accessory genome—the set of genes shared by a subset of genomes—of a population or a species. The *CoreSimul* process starts by generating a random core genome sequence of length $L$ and with a GC-content $GC$ specified by the user. Alternatively, an input genome can be directly provided by the user. The sequence is assumed to represent a nucleotide concatenate of protein coding genes without intergenic DNA. This sequence is then evolved in silico following a branching process respecting the topology of the input tree. The rate of substitutions $m$ is based on the branch length of the input tree, and this rate can be modified by the user with a rescaling coefficient (e.g. a rescaling coefficient of 0.5 will reduce the length of all branches by half). In order to mimic the effect of purifying selection acting on coding sequences, the sequence can be evolved with different substitution rates across codon positions: the relative rates of the three codon positions can be specified by the user (see user manual for recommendations). When specified, the relative rate of substitution across codon positions does not change the overall rate of substitution, which is imposed by the branch length of the input tree and the rescaling coefficient. In addition, several substitution models can be specified: Juke and Cantor (JC69), Kimura 2-parameters (K2P), Kimura 3-parameters (K3P) or General Time Reversible (GTR), in which case, the substitutions transition/transversion ratio $\kappa$ or other parameters must be specified. Finally, the genomes are evolved with a recombination rate $\rho$, which is defined relative to the

substitution rate $m$. Homologous recombination events are internal to the simulated dataset and no imports from external sources are modelled (unless gene gains are allowed). Note that *CoreSimul* makes a clear distinction between homologous recombination, which consists in the replacement of a sequence by an homologous one present in the simulated dataset, from horizontal gene transfer, which consists in the gain of a new sequence external to the simulated genomes (see below). In effect, recombination leads to the exchange of single nucleotide polymorphisms while gene gains lead to the acquisition of new sequences.

In order to mimic more realistic conditions, the different sequences present at any given time are evolved simultaneously and only sequences overlapping in time are allowed to recombine with one another (i.e. recombination with ancestral sequences is not allowed). Concretely, the phylogenetic tree is divided in multiple "time segments" of overlapping branches (Fig. 1a). For each time segment $t$, each sequence receives a number of mutations $M_t$ and a number of recombination events $R_t$ defined by a Poisson process of mean $m.l$ and $\rho.l$, respectively, with $l$ the length of the branch in the time segment, $m$ the mutation rate and $\rho$ the recombination rate. The mutation and recombination events are then introduced at random in different sequences of the time segment: a random sequence of the time segment is pulled and a mutation event or a recombination event is introduced randomly (this step is repeated until all the mutation and recombination events specific to each sequence have been introduced). The donor sequence of each recombination event is pulled randomly from the set of sequences in the time segment. The position of each recombination event is chosen at random along the sequence and its size is defined by a geometric distribution of mean $\delta$ specified by the user (genomes are assumed to be linear).

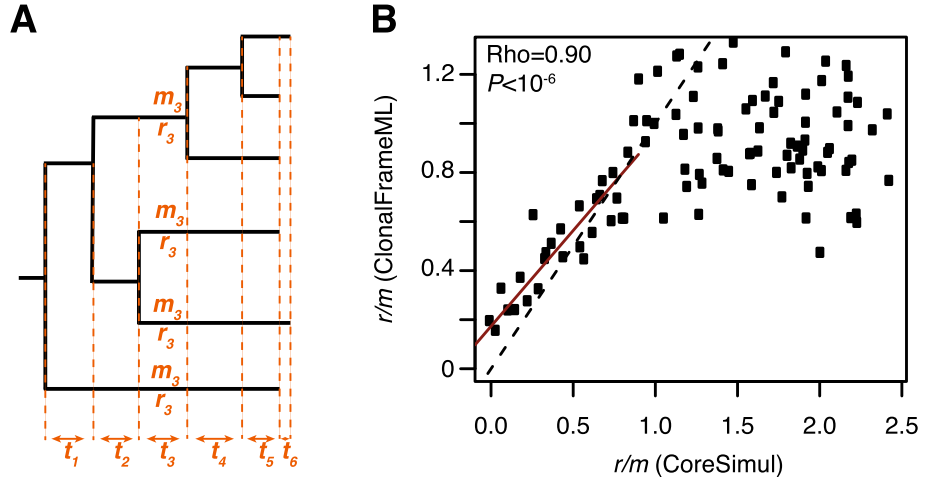


**Fig. 1 a**. Schematic representation of the *CoreSimul* process. The tree is divided in multiple time segments with the same number of overlapping branches. Only branches in the same time segment can recombine with one another. For each branch in the time segment, an average number $m_t$ of mutations is introduced following a Poisson distribution and an average number $r_t$ of recombination events is introduced—also following a Poisson distribution—between the branches with $r_t = \rho.m_t$ with $\rho$ the recombination rate. Mutation events and recombination events are introduced in a random order in the different branches of the time segment, simultaneously. **b**. Comparison between *CoreSimul* simulations and recombination rate predictions of *ClonalFrameML*. The black dashed line represents the theoretical expectation between *CLonalFrameML* predictions and the recombination rates in the simulated dataset. The red line represents the observed linear regression between the simulated *r/m* values and the *r/m* values predicted by *ClonalFrameML* (note that only the data points for $r/m \leq 1$ were used for the regression)

During the simulation the number of nucleotide polymorphisms (SNPs) exchanged by each recombination event is recorded to generate the statistic $\eta$, which represents the average number of polymorphic alleles exchanged by recombination. Using this statistic, the effective recombination rate $r/m$ is defined from the relationship $r/m = \rho.\eta.\delta$ as in [2] and is returned to the user at the end of the simulation. The effective recombination rate $r/m$ is frequently used to measure recombination rates and represents the number of polymorphisms exchanged by recombination relative to the number of polymorphisms introduced by mutation.

*CoreSimul* further offers the possibility to simulate uneven rates of homologous recombination along the genome as a function of sequence divergence. Multiple works have experimentally determined that the frequency of homologous recombination decreases with sequence divergence following a log-linear relationship [30–36]. As a result, it is predicted that homologous sequences with higher sequence identity will be much more likely to recombine than more divergent sequences. When specified by the user, *CoreSimul* introduces a biased probability of homologous recombination using the relationship $p = 10^{-\pi\Phi}$, with $p$ the probability to recombine, $\pi$ the sequence divergence and $\Phi$ the slope of the relationship between sequence divergence and the frequency of homologous recombination [33]. The coefficient $\Phi$ is species-specific and has only been determined for several species [33]. Thus, by default, *CoreSimul* uses a parameter of $\Phi = 18.1$, as experimentally inferred for *Streptococcus pneumoniae*, which is intermediate relative to other species with known $\Phi$ values such as *Bacillus subtilis* and *Escherichia coli* [33].

In addition, rates of gene gains and losses can be specified to simulate the evolution of accessory genes. In the *CoreSimul* framework, gene gains are modeled as external horizontal gene transfers, independent from homologous recombination events, which are only modeled as internal events. The rates of gene gains and losses are specified as a function of the substitution rate following a Poisson distribution (i.e. proportional to branch length). Assuming purifying selection, entire genes can be gained or lost but not fragments thereof. Simulating genomes with gene gains and/or losses results in a genome composed of core genes and accessory genes: Core genes are those genes present in all the genomes of the dataset, while accessory genes are those genes only found in a fraction of the genomes. During the simulation, homologous accessory genes are free to engage in homologous recombination with one another if the sequence is present in both the donor and the recipient genomes. In the case where a recombining fragment overlaps an accessory and a core gene, the accessory portion of the gene will not be transferred from the donor to the recipient genome if this sequence is absent in the recipient genome (i.e. only the fragment of the core gene will be transferred). Consequently, homologous recombination will not introduce partial indels from one genome to another but will transfer entire genes or lead to the deletion of entire genes if both extremities of the recombining fragment are located within homologous sequences shared by both the donor and the recipient genomes. Note that if no gene gains or losses are specified, *CoreSimul* will only simulate the evolution of the core genome.

*CoreSimul* is implemented in Python 3.7 but can also run on Python 2.7 without modification. It requires the Python library NumPy. No other dependencies are required. The parameters of the simulation must be provided in a control file. *CoreSimul*

is compatible with Mac and Linux operating systems. *CoreSimul* can be freely downloaded at https://github.com/lbobay/CoreSimul. The program also includes a user manual with detailed information, recommendations and examples.

## Results and discussion

To test *CoreSimul*, we simulated the evolution of a genome following the topology and branch lengths of a previously published tree [24]; the tree was originally built using the core genome of 34 genomes of *Acinetobacter pittii* using RAxML v8 [37]. The genomes were simulated with a length of 100,000 bp, a GC-content of 50%, $\kappa = 1$, and no selection (identical substitution rates across codon positions). The tree was rescaled with parameter 0.05 to reduce the number of polymorphisms in the simulated alignments. One hundred simulations were conducted with recombination rates $\rho$ varying from 0 to 5. The average recombination tract length was set at $\delta = 100$ bp across all simulations. The 50 simulated datasets were then analyzed with *ClonalFrameML* [3] with the same input tree and with default parameters. We observed nearly identical values between the recombination rates generated with *CoreSimul* and the rates predicted by *ClonalFrameML* when recombination rate was low (Fig. 1b; Spearman's Rho = 0.90,$P < 10^{-6}$ for $r/m \leq 1$). A very similar relationship (Supplementary Figure 1; Spearman's Rho = 0.89, $P < 10^{-6}$ for $r/m \leq 1$) was observed when the sequences were evolved with more realistic parameters that closely match this species, i.e. GC = 45%, $\kappa = 1.6$ and different substitution rates across codon positions (0.15, 0.07 and 0.78 for codon positions 1, 2 and 3, respectively). Note that for higher recombination rates ($r/m > 1$ in this case), *ClonalFrameML* tends to substantially underestimate recombination rates (Fig. 1b) as reported by the authors of the program [3]. In addition, we found stronger discrepancies between our simulations and the $r/m$ ratios inferred by *ClonalFrameML* for datasets simulated with higher substitution rates.

In addition, we conducted multiple verifications to ensure that the entered parameters were simulated as expected: The GC-content, the alignment length and the transition/transversion ratio $\kappa$ (or the rates for other substitution models) were systematically found to match the simulation parameters. We also observed that the polymorphisms at the different codon positions matched the relative substitution frequency specified across first, second and third codon positions. Finally, we could retrieve the same tree topology as the input tree when building the phylogenetic tree from the simulated sequences with RAxML v8 [37]. However, tree topologies were observed to differ from the input tree when simulated with higher recombination rates, which is expected due to the increase of homoplasies in the sequences when simulated with higher recombination rates. Overall, these different tests confirmed that the sequences were simulated as expected.

## Conclusions

*CoreSimul* is a forward-in-time simulator, specifically built to simulate prokaryote evolution with homologous recombination and negative selection across codon positions. Such a tool can be used to infer parameters or to test other bioinformatic tools. In addition, because it is based on a phylogenetic framework, *CoreSimul* incorporates population structure information, which can be applied to population models and phylogenetic analyses. Although many genome simulators have been implemented,

*CoreSimul* presents several key differences relative to existing tools: i) it does not rely on the coalescent but uses a tree-guided simulation framework, ii) it is specifically designed to simulate the evolution of prokaryotic genomes with a prokaryote-specific model of homologous recombination, iii) it can be run with customized parameters and models of substitution, iv) it offers the possibility to model gene gains and gene losses, thereby simulating the evolution of core and accessory genes and v) it can model varying rates homologous recombination as a function of sequence divergence. Overall, *CoreSimul* constitutes a substantial addition to the current list of genome simulators. This addition is particularly valuable considering the limited number of forward-in-time simulators currently available to model prokaryote genome evolution.

## Availability and requirements

**Project name**: CoreSimul.

**Project home page:** https://github.com/lbobay/CoreSimul

**Operating system(s):** MacOS, Linux.

**Programming language:** Python.

**Other requirements:** Numpy library.

**License:** MIT License.

**Any restrictions to use by non-academics:** None.

## Supplementary information

**Supplementary information** accompanies this paper at https://doi.org/10.1186/s12859-020-03619-x.

**Additional file 1: Figure S1.** Comparison between *CoreSimul* simulations with selection and recombination rate predictions of *ClonalFrameML*. Simulations were run with parameters that closely match the sequence parameters of *A. pittii*: GC = 45%, transition/transversion ratio $\kappa = 1.6$, relative substitution rates of codon positions: 0.15, 0.07 and 0.78 for codon positions 1, 2 and 3, respectively. The black dashed line represents the theoretical expectation between *ClonalFrameML* predictions and the recombination rates in the simulated dataset. The red line represents the observed linear regression between the simulated *r/m* values and the *r/m* values predicted by *ClonalFrameML* (note that only the data points for $r/m \leq 1$ were used for the regression).

### Acknowledgments

We thank Kasie Raymann and Caroline Stott for helpful comments on *CoreSimul* and the manuscript.

### Author's contributions

LMB wrote the program, analyzed the data and wrote the manuscript. All authors read and approved the final manuscript.

### Funding

This work was supported by the National Science Foundation under Grant No. DEB-1831730 and by the National Institute Of General Medical Sciences of the National Institutes of Health under Award Number R01GM132137. The funding body had no role in the design of the study, collection, analysis and interpretation of data and in writing of the manuscript.

### Availability of data and materials

The scripts and datasets used for this analysis are freely accessible on https://github.com/lbobay/CoreSimul.

### Ethics approval and consent to participate

Not applicable.

### Consent for publication

Not applicable.

### Competing interests

The author declares that he has no competing interests.

Received: 2 March 2020 Accepted: 19 June 2020
Published online: 24 June 2020

**References**

1. Dalquen DA, Anisimova M, Gonnet GH, Dessimoz C. ALF--a simulation framework for genome evolution. Mol Biol Evol. 2012;29(4):1115–23.
2. Didelot X, Falush D. Inference of bacterial microevolution using multilocus sequence data. Genetics. 2007;175(3):1251–66.
3. Didelot X, Wilson DJ. ClonalFrameML: efficient inference of recombination in whole bacterial genomes. PLoS Comput Biol. 2015;11(2):e1004041.
4. Brown T, Didelot X, Wilson DJ, De Maio N. SimBac: simulation of whole bacterial genomes with homologous recombination. Microb Genom. 2016;2(1):e000044.
5. Rambaut A, Grassly NC. Seq-gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. Comput Appl Biosci. 1997;13(3):235–8.
6. Gonnet GH, Hallett MT, Korostensky C, Bernardin L. Darwin v. 2.0: an interpreted computer language for the biosciences. Bioinformatics. 2000;16(2):101–3.
7. Tuffery P. CS-PSeq-gen: simulating the evolution of protein sequence under constraints. Bioinformatics. 2002;18(7):1015–6.
8. Yang Z. PAML 4: phylogenetic analysis by maximum likelihood. Mol Biol Evol. 2007;24(8):1586–91.
9. Chadeau-Hyam M, Hoggart CJ, O'Reilly PF, Whittaker JC, De Iorio M, Balding DJ. Fregene: simulation of realistic sequence-level data in populations and ascertained samples. BMC Bioinformatics. 2008;9:364.
10. Arenas M, Posada D. Simulation of genome-wide evolution under heterogeneous substitution models and complex multispecies coalescent histories. Mol Biol Evol. 2014;31(5):1295–301.
11. Spielman SJ, Wilke CO. Pyvolve: a flexible Python module for simulating sequences along phylogenies. PLoS One. 2015; 10(9):e0139047.
12. Mallo D, De Oliveira ML, Posada D. SimPhy: Phylogenomic simulation of gene, locus, and species trees. Syst Biol. 2016; 65(2):334–44.
13. Saber MM, Shapiro BJ. Benchmarking bacterial genome-wide association study methods using simulated genomes and phenotypes. Microb Genom. 2020;6(3).
14. Beiko RG, Charlebois RL. A simulation test bed for hypotheses of genome evolution. Bioinformatics. 2007;23(7):825–31.
15. Bobay LM. The prokaryotic species concept and challenges. In: The Pangenome. Edited by Tettelin H. Cham: Springer; 2020.
16. Bobay LM, Traverse CC, Ochman H. Impermanence of bacterial clones. Proc Natl Acad Sci U S A. 2015;112(29):8893–900.
17. Touchon M, Hoede C, Tenaillon O, Barbe V, Baeriswyl S, Bidet P, Bingen E, Bonacorsi S, Bouchier C, Bouvet O, et al. Organised genome dynamics in the Escherichia coli species results in highly diverse adaptive paths. PLoS Genet. 2009;5: e1000344.
18. Kuzminov A. Recombinational repair of DNA damage in Escherichia coli and bacteriophage lambda. Microbiol Mol Biol Rev. 1999;63:751–813.
19. Davies JL, Simancik F, Lyngso R, Mailund T, Hein J. On recombination-induced multiple and simultaneous coalescent events. Genetics. 2007;177(4):2151–60.
20. Vos M, Didelot X. A comparison of homologous recombination rates in bacteria and archaea. ISME J. 2009;3(2):199–208.
21. Bobay LM, Ochman H. Biological species are universal across Life's domains. Genome Biol Evol. 2017;9(3):491–501.
22. Smith JM, Smith NH, O'Rourke M, Spratt BG. How clonal are bacteria? Proc Natl Acad Sci U S A. 1993;90(10):4384–8.
23. Price MN, Arkin AP. Weakly deleterious mutations and low rates of recombination limit the impact of natural selection on bacterial genomes. MBio. 2015;6(6):e01302–15.
24. Bobay LM, Ochman H. Factors driving effective population size and pan-genome evolution in bacteria. BMC Evol Biol. 2018;18(1):153.
25. De Maio N, Wilson DJ. The bacterial sequential Markov coalescent. Genetics. 2017;206(1):333–43.
26. Sipola A, Marttinen P, Corander J. Bacmeta: simulator for genomic evolution in bacterial metapopulations. Bioinformatics. 2018;34(13):2308–10.
27. Kessner D. Novembre J: forqs: forward-in-time simulation of recombination, quantitative traits and selection. Bioinformatics. 2014;30(4):576–7.
28. Rocha EPC. Neutral theory, microbial practice: challenges in bacterial population genetics. Mol Biol Evol. 2018;35(6): 1338–47.
29. Charlesworth J, Eyre-Walker A. The rate of adaptive evolution in enteric bacteria. Mol Biol Evol. 2006;23(7):1348–56.
30. Roberts MS, Cohan FM. The effect of DNA sequence divergence on sexual isolation in Bacillus. Genetics. 1993;134(2):401–8.
31. Zawadzki P, Roberts MS, Cohan FM. The log-linear relationship between sexual isolation and sequence divergence in Bacillus transformation is robust. Genetics. 1995;140(3):917–32.
32. Vulic M, Dionisio F, Taddei F, Radman M. Molecular keys to speciation: DNA polymorphism and the control of genetic exchange in enterobacteria. Proc Natl Acad Sci U S A. 1997;94(18):9763–7.
33. Majewski J, Zawadzki P, Pickerill P, Cohan FM, Dowson CG. Barriers to genetic exchange between bacterial species: Streptococcus pneumoniae transformation. J Bacteriol. 2000;182(4):1016–23.
34. Majewski J. Sexual isolation in bacteria. FEMS Microbiol Lett. 2001;199(2):161–9.
35. Kung SH, Retchless AC, Kwan JY, Almeida RP. Effects of DNA size on transformation and recombination efficiencies in Xylella fastidiosa. Appl Environ Microbiol. 2013;79(5):1712–7.
36. Dixit PD, Pang TY, Maslov S. Recombination-driven genome evolution and stability of bacterial species. Genetics. 2017; 207(1):281–95.
37. Stamatakis A. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. Bioinformatics. 2014;30(9):1312–3.

## Publisher's Note