# Review of *Simulation of bacterial populations with SLiM* by Cury *et al.*

October 28, 2020

Simulating genetic data under various scenarios is standard practice in evolutionary biology. This is usually done using coalescent simulations, which work well for neutral models. The stated aim of the paper under review is to "go beyond the limitations of the coalescent" (p. 2).

The authors pursue this by showing how the forward simulator SLiM, apparently first published in 2013, can be adapted to bacterial populations; its original target are eukaryotes. The results are similar to those obtained with classical tools such as ms. And while ms is much faster than SLiM under most scenarios, it is overtaken under high recombination.

The paper is essentially an addition to the SLiM manual, it contains no new biology or algorithm. However, SLiM appears to be a good forward simulator and there are bound to be scenarios not covered by current coalescent simulators, though fast gene conversion isn't one of them.

The authors state that ms gets slow with high recombination and msprime so far lacks gene conversion. However, macs is a practical simulator with fast gene conversion published in 2008, which isn't mentioned. Here are a few additional detailed comments:

1. p. 2: The authors criticize that gene conversion in ms is based on a linear chromosome, whereas SLiM implements the circular chromosomes found in bacteria. Where does this make a difference?

2. p. 3: The authors recommend a burn-in of $5N_{\mathrm{e}}$ generations, but caution that this also does not guarantee equilibrium. What is the probability of reaching equilibrium as a function of burn-in length? Or is that not known?

3. p. 7: All entries in Figure 1 should be based on the same number of replicates, even if times need to be extrapolated from smaller runs.

4. p. 7: What is the memory requirement of ms and FastSimBac in the lower panel of Figure 1?

5. p. 10: Recombination makes ms slow, but a more appropriate comparison might be to macs.

6. p. 10: As in Figure 1, the run times should be based on the same number of replicates.