Simulation of pangenome evolution

Sodapop pangenome evolution module explicitly simulates Sela, Wolf and Koonin's Prokaryotic genome size evolution model[1,2] with few changes. In this model, the selective advantage of gene gain, i.e. the advantage of having x+1 genes instead of x genes, depends of the genome size, which is measured by the number of genes in the genome (x). The gene loss selection coefficient has the opposite sign of gene gain and more precisely, gene gain is slightly beneficial while gene loss is slightly deleterious[2]. The selection coefficient of gene gain and gene loss can thus be described by the following formula:

$$s_{gain}(\mathrm{x}) = \mathrm{a} + b \cdot \mathrm{x} = -\mathrm{s}_{\mathrm{loss}}(\mathrm{x})$$

where s_{gain} is the selection coefficient of gene gain through horizontal gene transfer (HGT), " a " is a constant input parameter of the simulation, " b " is a constant input parameter that represents the benefit or cost associated with the gain of a single gene, x represents genome size (the number of genes in a species' genome) and s_{loss} is the selection coefficient of gene loss. We modified that formula to simulate a model where each gene has its own constant selective advantage which is exponentially distributed and independent of genome size (x), i.e. b = 0. This change allows simulations to reproduce the shape of gene mobility in a real dataset (**Figure 1**); Note that the expected value of an exponential distribution is 1/ λ. In this case:

$$s_{gain} = s_{gene} = -\mathrm{s}_{\mathrm{loss}}$$

where s_{gene} ~ Exp(λ), λ is an input parameter of the simulation and 1/λ represents the expected value of the distribution of HGT selection coefficient.

Moreover, in the model, genome size (x) influences the gene gain and gene loss rates. Indeed, as genome size increases, the gene gain rate decreases, and the gene loss rate increases to reach an equilibrium around a certain genome size x0 [2]. Therefore, when genome size (x) is smaller than genome size at equilibrium (x0) – i.e. x0 represents the genome size at which gain rate equals loss rate – the cell should have a higher probability to gain genes than to lose genes. As for when genome size (x) is higher than genome size at equilibrium (x0), the cell should have a higher probability to lose genes than to gain new ones. To consider the stochastic component of evolution, the cells and genes that are involved in each gain or loss events are randomly selected. Also, the number of gain or loss events are drawn from a Poisson distribution with the gain and loss rate representing the rate parameter of the distribution:

$$G_{rate} \sim \text{Poisson}(\lambda = \text{s}' \cdot x^{\lambda^{+}})$$

$$L_{rate} \sim \text{Poisson}(\lambda = \text{r}' \cdot x^{\lambda^{-}})$$

where G_{rate} is the gain rate, i.e. the number of gene gain events per cell per generation, L_{rate} is the loss rate, i.e. the number of gene loss events per cell per generation, x represents genome size (number of genes in species genome) and r', s', λ^{+} and λ^{-} are simulation input parameters. It is also worth noting that HGT is not restricted to cells from 2 different species but could also happen, on rarer occasions, within species.

Furthermore, we chose to implement this model in the SodaPop software as it allows to simulate mutations and a Wright-Fischer process for asexual populations [1]. In SodaPop, the current mutation model is equivalent to the Jukes-Cantor model in which all single nucleotide changes occur at the same constant rate [3]. We also implemented a distribution of non-synonymous mutation fitness effects where 30% of mutations are lethal, as previously reported in literature [4], and 70% are drawn from N(μ=-0.02, σ=0.01). Synonymous mutations are considered neutral unless the user provides data on species codon usage and the related fitness effects (See Sodapop manual). SodaPop also offers flexibility as the user can build the initial setup of the simulation by himself [1]. We created scripts to facilitate the initial simulation setup (https://github.com/arnaud00013/SodaPop/tree/Sodapop-pev/tools). The scripts allow to define each species' abundance, gene content and even define the genes that are mobile within species' genomes (https://github.com/arnaud00013/SodaPop/blob/Sodapop-pev/tools/Setup_SodaPop_with_PEV. py). At the start of the simulation, the pool of genes of the microbial community contains housekeeping genes that are present in all species and accessory genes that are randomly distributed across species. Within this set of accessory genes, some are randomly tagged as "mobile" and the quantity of mobile genes in the pool of accessory genes is defined by the user. Mobile genes can be transferred and lost while other genes can only be lost. For each set of simulations sharing the same input parameters, we ran 10 replicates to show that results were reproducible. Each simulation included 5000 cells, 10 species, 500 genes per cell at equilibrium and a simulation time of 10^5 generations with a timestep of 10^4 generations to save simulation data. Population sizes in simulation are smaller relative to natural because of hardware memory limitations [2,5]. To make sure this limitation does not cause undesirable effects, like the accumulation of deleterious mutations leading to extinction, also known as Muller's Ratchet [6], we maintained species abundance constant. To avoid a lack of genetic diversity in the simulated population due to small population sizes, we also increased prokaryotic mutation rate up to the order of 10^{-7} mutations per site per generation. Although these parameters are not typical for prokaryotes, the simulation results are still valuable because they reproduce the trends observed in the real Fiji dataset **(Figures 3).** Moreover, genome size equilibrium has been reached for every simulation. Thus, the simulation the results do not depend on the initial conditions (**Figures 2**). The software is available on GitHub (https://github.com/arnaud00013/SodaPop).

Pangenome evolution simulations step by step

*The version of SodaPop that includes the pangenome module, i.e. Sodapop-pev, can be downloaded at https://github.com/arnaud00013/SodaPop (See https://louisgt.github.io/SodaPop/2017/09/05/Setup-and-installation.html for installation).

1) Create the initial setup for the simulation with the script https://github.com/arnaud00013/SodaPop/blob/Sodapop-pev/tools/Setup_SodaPop_with_PEV.py. This script creates the input files necessary for running Sodapop simulations with the pangenome evolution module. All header files should be in files/headers/ directory of the Sodapop workspace (See https://github.com/arnaud00013/SodaPop/tree/Sodapop-pev/files/header). The script is interactive and asks the user for the genome size of each species.

   -Arguments:

   (i) The absolute path of the SodaPop workspace

   (ii) The name of the ".gene" header file

   (iii) The name of the ".cell" header file

   (iv) The name of the population data header file

   (v) The number of species simulated

   (vi) The size of the whole community pool of accessory genes

(vii) The number of core genes at the start of the simulation

(viii) The number of accessory genes that are mobile.

-Dependencies: Python3 (possible to specify the version in the header of the file; python3.6 by default)

-Example of command used for our paper on short timescales pangenome evolution: "python3.6 Setup_SodaPop_with_PEV.py ~/Sodapop-pev/ header.gene header.cell header_pop.dat 10 5000 100 2000"

2) (OPTIONAL) Add the codon usage data of a species in its .cell file with the script "https://github.com/arnaud00013/SodaPop/blob/Sodapop-pev/tools/add_Codon_Usage_data_into_Sodapop_cell_file_from_Kasuza_db_like_file.sh." This script takes as input the path of a file in Kazusa database codon usage-like format (1st input argument) and adds the data in the input .cell file (2nd input argument). You need to run the script for each species simulated. You can modify it

   -Arguments:

   (i) The absolute path of a file containing the codon usage data of the species (in Kazusa database codon usage-like format [7]; See https://www.kazusa.or.jp/codon/)

   (ii) The name of the ".cell" file

   -Example :

   "./add_Codon_Usage_data_into_Sodapop_cell_file_from_Kasuza_db_like_file.sh ~/Sodapop-pev/E_coli_codon_usage.txt ~/Sodapop-pev/0.cell"

3) Run sodasumm to create the initial population snapshot file [1](See https://louisgt.github.io/SodaPop/2017/09/05/Running-a-basic-simulation.html)

   -Example:

   "sodasumm ~/Sodapop-pev/files/start/population.dat 0"

4) Run the simulation with the pangenome evolution module

   -Arguments (See https://louisgt.github.io/SodaPop/2017/09/05/Command-line-flags.html):

   "--sim-type s" allows to use selection coefficients distributions

"--f 9" allows to select the fitness function 9, which needs to be selected for the pangenome evolution module

"--normal --alpha -0.02 --beta 0.01" defines the distribution of mutation selection coefficient as N(μ=-0.02, σ=0.01)

"-p ~/SodaPop_pev/files/start/population.snap" defines the absolute path to the initial population snapshot

"-g ~/SodaPop_pev/files/genes/gene_list.dat " defines the absolute path to the gene list file

"-o simulation_test1" defines the name of the output workspace (created if it does not already exist and overwritten if it already exists)

"-t 20000" defines the timestep, i.e. the number of generations between 2 snapshots

"-n 5000 " defines the number of cells simulated

"-m 100001" defines the simulation time, i.e. the number of generations simulated accounting for generation 0.

"-s 2" allows to save snapshots in the long format with DNA sequence

"-V" activates the pangenome evolution module and forces the use of the fitness function 9, which accounts for mutation, gain and loss fitness effects

"--exp_rate_s_hgt 1E4 " defines the rate parameter λ of the exponential distribution of HGT fitness effect (where 1/λ is the expected value of the distribution)

"--bForSx 0" defines b = 0 in the formula of gene gain selection coefficient

"--rPrime 7.2E-15" defines r' in the formula of gene loss rate

"--sPrime 56250" defines s' in the formula of gene gain rate

"--lambdaPlus -2" defines λ^{+} in the formula of gene gain rate

"--lambdaMinus 5" defines λ^{-} in the formula of gene loss rate

"-e" allows to keep a log of all arising mutations during the simulations

"-T" allows to track pangenomes evolution events (Gain and loss of genes), and the evolution of genome size, loss rate and gain rate

"--execVA" allows to run a population genetics analysis at the mobile gene level at the end of the simulation (execVA stands for "execute mobile genes variants analysis")

"-u 6" defines the number of CPUs to use for the population genetics analysis

"--simulCUB" actives the simulation of codon usage bias

"--stdCubDfe" defines the standard deviation (σ) of the distribution of synonymous mutation fitness effect, where the selection coefficient of a synonymous mutation follows the distribution N(μ=0, σ). This distribution allows to define synonymous mutations as neutral on average while allowing non-neutral synonymous mutations. Synonymous mutations selection coefficients are drawn from the positive part of the distribution (s>=0) if they increase gene CAI and from the negative part of the distribution (s<=0) otherwise.

-Example:

```
"sodapop --sim-type s -f 9 --normal --alpha -0.02 --beta 0.01 -p ~/SodaPop_pev/files/start/population.snap -g ~/SodaPop_pev/files/genes/gene_list.dat -o simulation_test1 -t 20000 -n 5000 -m 100001 -s 2 -V --exp_rate_s_hgt 1E4 --bForSx 0 --rPrime 7.2E-15 --sPrime 56250 --lambdaPlus -2 --lambdaMinus 5 -e -T --execVA -u 6 "
```

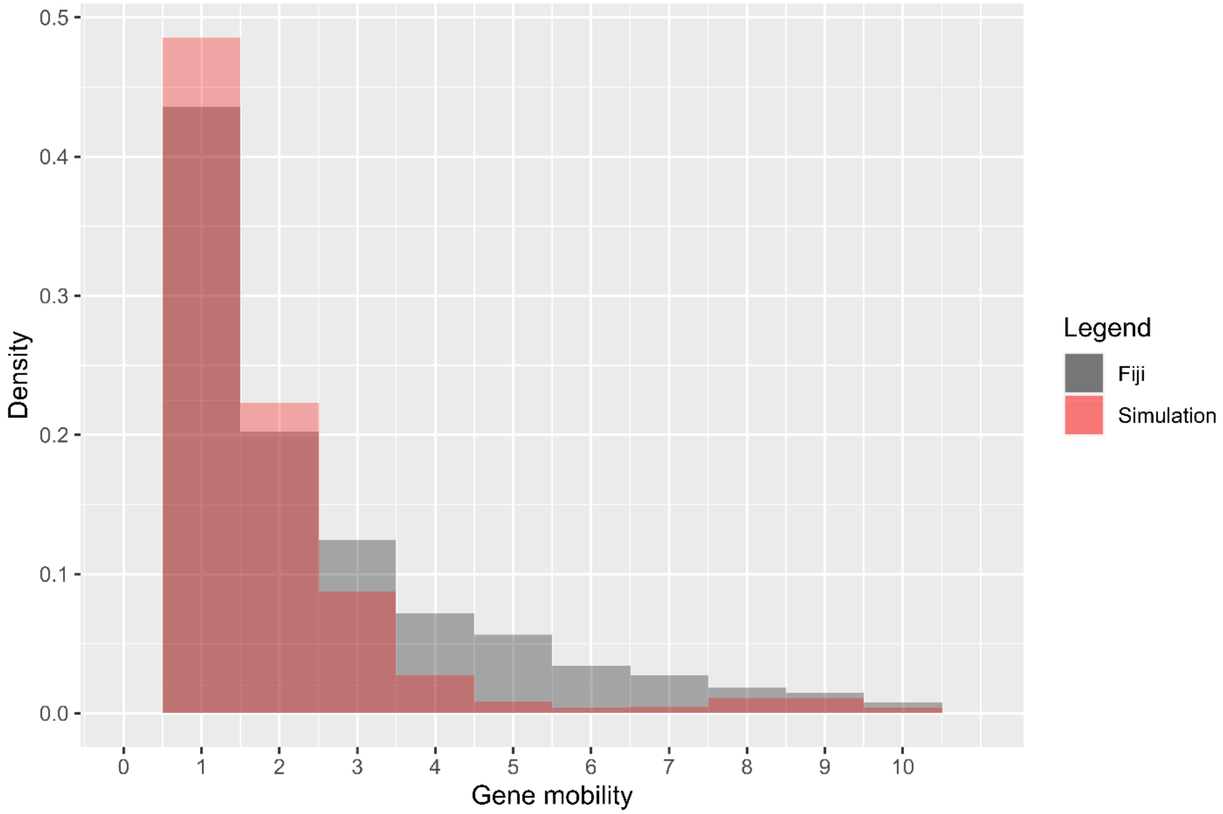
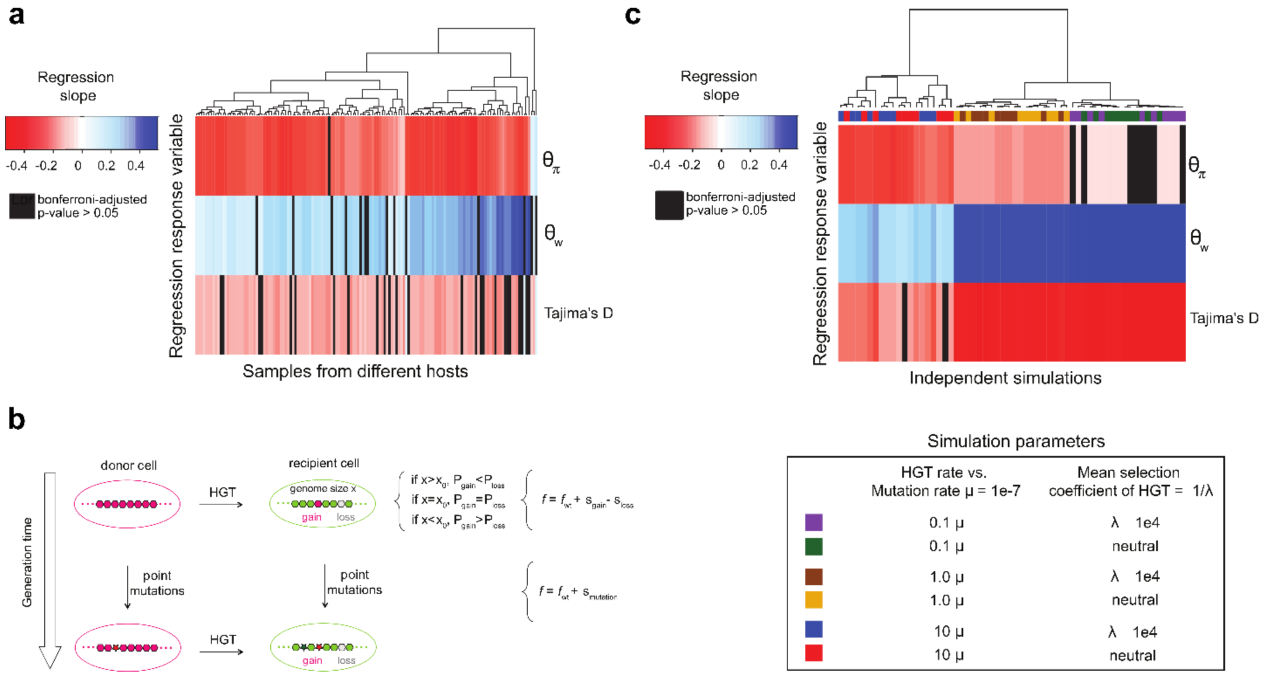


Figure 1. Gene mobility distribution in simulation vs in Fiji dataset.

Simulations in which HGT is slightly adaptive and the HGT selection coefficient is exponentially distributed produce a gene mobility distribution with a similar shape than the one observed in the Fiji dataset. However, their quantitative similarity is not significant (Kolmogorov-Smirnov test p-value < 0.05) and the range of mobility in simulation is [1,10] while it is [1,16] in the Fiji dataset so the distribution is truncated here to a maximum of 10 species. The simulation presented (red) included 5000 cells, 10 species, 500 genes per cells at equilibrium, a simulation time of 10^5 generations, HGT rate = 10µ and HGT selection coefficient parameter λ = 1E5 (HGT is slightly adaptive; *s* = 1E-5 in average).

Figures 2 Genome size equilibrium across simulations (https://github.com/arnaud00013/SodaPop/blob/Sodapop-pev/docs/Genome_Size_Equilibrium_All_Simulations.pdf)

This set of figures represents the time series of genome size during the simulations mentioned in this manuscript: a) HGT rate = 0.01μ and HGT is neutral, b) HGT rate = 0.01μ and λ = 1E6, c) HGT rate = 0.01μ and λ = 1E5, d) HGT rate = 0.01μ and λ = 1E4, e) HGT rate = 0.1μ and HGT is neutral, f) HGT rate = 0.1μ and λ = 1E6, g) HGT rate = 0.1μ and λ = 1E5, h) HGT rate = 0.1μ and λ = 1E4, i) HGT rate = 1μ and HGT is neutral, j) HGT rate = 1μ and λ = 1E6, k) HGT rate = 1μ and λ = 1E5, l) HGT rate = 0.01μ and λ = 1E4, m) HGT rate = 10μ and HGT is neutral, n) HGT rate = 10μ and λ = 1E6, o) HGT rate = 10μ and λ = 1E5 and p) HGT rate = 10μ and λ = 1E4, where the expected selection coefficient of HGT (s) = 1/λ. It is important to show that these time series are on dynamic equilibrium, i.e. genome size fluctuates around a certain value, because it supports the fact that our results are not dependent on the initial conditions of the simulations.



Simulation parameters

	HGT rate vs. Mutation rate μ = 1e-7	Mean selection coefficient of HGT = 1/λ
🟪	0.1 μ	λ 1e4
🟩	0.1 μ	neutral
🟫	1.0 μ	λ 1e4
🟨	1.0 μ	neutral
🟦	10 μ	λ 1e4
🟥	10 μ	neutral

Figure 3. Gene mobility is negatively correlated with Tajima's *D* in real and simulated microbiomes. A) Real data from Fiji. The heatmap shows the slope of a regression model in which either θ_π, θ_W or *Tajima's D* is the response variable and gene mobility is the explanatory variable (across samples). Regression p-values were obtained through a *t*-test. The heatmap contains non-significant regressions results after Bonferroni p-value filter (black), negative significant correlations (red) and positive significant correlations (blue). Data standardization was performed before each regression to respect the *t*-test's assumption of normality. Heatmap rows and columns were clustered with Euclidean distance and complete linkage clustering.

B) Representation of simulation events over two generations. In the first generation, a gene gain event occurs through HGT. Gene gain is represented by the transfer of gene from a donor cell to a recipient cell and increases the genome size of this recipient cell. The probability of future gene gain or gene loss events (P_{gain} and P_{loss} respectively) is determined by the difference between the current genome size of the cell (x) and the equilibrium genome size (x_0). At equilibrium, the probability of gene gain and gene loss is the same by definition ($P_{gain}=P_{loss}$). An increase of genome size until it exceeds the equilibrium point ($x > x_0$) leads to gene loss being more likely than gene gain ($P_{gain}<P_{loss}$). Gene gain also increases the fitness ($f > f_{WT}$) of the recipient cell based on the selection coefficient of the transferred gene (s_{gain}). In the model, each gene has its own selective coefficient which is drawn from an exponential distribution *exp(λ)* with an expected value of *1/λ*. Gene gain is either slightly beneficial or neutral in this model and has the opposite fitness effect of gene loss, which is slightly deleterious or neutral ($-s_{gain} = s_{loss}$ where $s_{gain} >= 0$). Gene loss decreases the genome size of the target cell and in case this decrease leads to a smaller genome size than equilibrium, the probability of gene gain becomes higher than the probability of gene loss ($P_{gain} > P_{loss}$). Gene loss also decreases the fitness of the target cell ($f < f_{WT}$) based on the selection coefficient of the lost gene (s_{loss}). Finally, as represented in the second generation, mutations can also occur and change the fitness of the cell based on a selective coefficient ($s_{mutation}$) which is drawn from a distribution (Methods).

C) Simulated data. The heatmap shows the slope of a regression model in which either θ_π, θ_w or *Tajima's D* is the response variable and gene mobility is the explanatory variable (across simulation replicates). Simulations with different parameter for HGT rate and or distributions of selective coefficients ($s \sim exp(\lambda)$) are color-coded (n=10 replicates per simulation).

References

1 Gauthier, L., Di Franco, R. & Serohijos, A. W. R. SodaPop: a forward simulation suite for the evolutionary dynamics of asexual populations on protein fitness landscapes. *Bioinformatics* **35**, 4053-4062, doi:10.1093/bioinformatics/btz175 (2019).

2 Sela, I., Wolf, Y. I. & Koonin, E. V. Theory of prokaryotic genome evolution. *Proc. Natl. Acad. Sci. U. S. A.* **113**, 11399-11407, doi:10.1073/pnas.1614083113 (2016).

3 Jukes, T. H. & Cantor, C. R. in *Mammalian protein metabolism* Vol. III (ed Elsevier) Ch. 24, 21-132 (Academic Press, 1969).

4 Eyre-Walker, A. & Keightley, P. D. The distribution of fitness effects of new mutations. *Nat. Rev. Genet.* **8**, 610-618, doi:10.1038/nrg2146 (2007).

5 Bobay, L. M. & Ochman, H. Factors driving effective population size and pan-genome evolution in bacteria. *BMC Evol. Biol.* **18**, 153, doi:10.1186/s12862-018-1272-4 (2018).

6 Bachtrog, D. & Gordo, I. Adaptive evolution of asexual populations under Muller's ratchet. *Evolution* **58**, 1403-1413, doi:10.1111/j.0014-3820.2004.tb01722.x (2004).

7 Nakamura, Y., Gojobori, T. & Ikemura, T. Codon usage tabulated from international DNA sequence databases: status for the year 2000. *Nucleic Acids Res.* **28**, 292, doi:10.1093/nar/28.1.292 (2000).