Introduction to AlphaSimR

Chris Gaynor

2017-12-15

The AlphaSimR package provides a framework for simulating plant and animal breeding programs in R. It is based on the AlphaSim program (Faux et al. 2016). This document will introduce the basic steps for running a simulation in AlphaSimR.

The basic steps of an AlphaSimR simulation are:

1. Create founder haplotypes using `runMacs`
2. Set simulation parameters using `createSimulation`
3. Add traits using `addTraitA` or any other addTrait function
4. Add SNP chips using `addSnpChip` (optional)
5. Create one or more initial populations using `newPop`
6. Perform a mix of crossing and selection to simulate a breeding program

Creating Founder Haplotypes

The first step in an AlphaSimR simulation is to create a set of founder haplotypes. To do this, run the function `runMacs`. The `runMacs` function uses a built in version of the MaCS program (Chen, Marjoram, and Wall 2009). The `runMacs` function allows for simulation of haplotypes using several predefined species histories or can be used to create a custom population using MaCS command line arguments.

Setting Simulation Parameters

Parameters for a simulation are stored in an object of `SimParam-class`. The function `createSimulation` creates this object. The `SimParam-class` is a container for global parameters that will be used by many functions in AlphaSimR. To avoid specifying a `SimParam-class` in all functions that use one, it is recommended that you give your `SimParam-class` the name `SIMPARAM`. All functions that take a `SimParam-class` as an arguement will search your global environment for an object called `SIMPARAM` if the argument is `NULL`.

Adding Traits

Traits are added to the simulation using one of the fullowing functions: `addTraitA`, `addTraitAD`, `addTraitAG`, or `addTraitADG`. These functions can be called multiple times to run simulations with multiple traits. These functions can also be used to simulate correlated pleiotropic traits. Simulations without any traits are possible as long as you don't use a selection function.

Adding SNP Chips

SNP chips are added to the simulation using the `addSnpChip` or `addStructuredSnpChips` functions. As with traits, these functions can be called multiple times to assign multiple SNP chips. SNP chips are not necessary

for a simulation, so this step can be skipped.

Creating a Population

The main objects in an AlphaSimR simulation are `Pop-class` objects. These objects represent a population that contains one or more individuals. Many AlphaSimR functions use a `Pop-class` object as an argument and return a `Pop-class` object as a result.

To create your first `Pop-class` object use the function `newPop`. This function should only be called after you have finished all of the above steps.

Working with Populations

Listed below are some functions for carrying out operations on populations.

- Viewing genotypes: `pullSnpGeno`, `pullSnpHaplo`, `pullQtlGeno`, `pullQtlHaplo`, `pullSegSitHaplo`
- Viewing summary data: `meanG`, `meanP`, `varG`, `varP`, `varAD`
- Setting phenotypes: `setPheno`, `calcPheno`, `setPhenoGCA`
- Making selections: `selectInd`, `selectFam`, `selectWithinFam`, `selectMale`, `selectFemale`
- Crossing: `makeCross`, `randCross`, `makeCross2`, `randCross2`, `self`, `makeDH`
- Genomic selection: `writeRecords`, `setEBV`

References

Chen, Gary K., Paul Marjoram, and Jeffery D. Wall. 2009. "Fast and Flexible Simulation of Dna Sequence Data." *Genome Research* 19:136–42. http://genome.cshlp.org/content/19/1/136.

Faux, Anne-Michelle, Gregor Gorjanc, R. Chris Gaynor, Mara Battagin, Stefan M. Edwards, David L. Wilson, Sarah J. Hearne, Serap Gonen, and John M. Hickey. 2016. "AlphaSim: Software for Breeding Program Simulation." *The Plant Genome* 9 (3). https://doi.org/10.3835/plantgenome2016.02.0013.