Breeding Scheme Language

Shiori Yabe, Hiroyoshi Iwata, and Jean-Luc Jannink

June 8, 2016

INDEX

Description

```
Package: BreedingSchemeLanguage
Type: Package
Title: Describe and simulate breeding schemes
Version: 1.0
Date: 2016-03-08
Author: Shiori Yabe, Hiroyoshi Iwata, and Jean-Luc Jannink
Maintainer: Shiori Yabe <syabe@affrc.go.jp>
Description: Users can simulate their planned breeding schemes by using functions that
imitate events in breeding.
License: GPL-3
Depends: R (>= 3.0.0), Rcpp (>= 0.11.0), snowfall
Imports:
    ggplot2,
    rrBLUP
LinkingTo: Rcpp
LazyLoad: yes
NeedsCompilation: yes
RoxygenNote: 5.0.1
```

Introduction

This documentation describes how to start and use the R package "BreedingSchemeLanguage". BreedingSchemeLanguage (BSL) was developed for breeders to conduct breeding simulations in a simple and flexible system. Users can use BSL under the R environment.

The BSL utilizes the coalescent-based whole genome simulator "GENOME" (Liang et al., 2006) to simulate a founder population for simulation.

If you use the program, the appropriate citation is:
__________________________________.

Reference for GENOME:
Liang L., Zöllner S., Abecasis G.R. (2006) GENOME: a rapid coalescent-based whole genome simulator

Required environment

Downloading and installing R is necessary to use the package. Users can download R on the web page:
The Comprehensive R Archive Network (CRAN) https://cran.r-project.org

The BSL package needs Rtools or Xcode for windows and Mac PC, respectively. These tools are used to install the BSL package from GitHub(https://github.com) and to compile C++ code on your PC. Compiling C++ code is automatically done when you install the BSL package.
Rtools: https://cran.r-project.org/bin/windows/Rtools/
Xcode: Mac App Store (You need your "Apple ID".)

Install BreedingSchemeLanguage

On your R screen, you can write the code as below:

```
install.packages(“devtools”)                # only at the first time
library(devtools)
install_github("syabe/BreedingSchemeLanguage")                  # only at the first time
library(BreedingSchemeLanguage)
```

Once you install the package, the first and second sentences can be removed.
If you asked the mirror cite when you conduct the code, please chose one place you prefer.

Now, you can use “Breeding Scheme Language” on your PC!

Describe breeding scheme

Help

You can call help page by two types of code:

?FUNCTION

help(FUNCTION)

FUNCTION should be replaced with the name of function you hope to ask.

Functions

- defineSpecies(loadData = NULL, saveDataFileName = "previousData", nSim = 1, nCore = 1, nChr = 7, lengthChr = 150, effPopSize = 100, nMarkers = 1000, nQTL = 50, propDomi = 0, nEpiLoci = 0)
- initializePopulation(nPop = 100, gVariance = 1)
- phenotype(errorVar = 1, popID = NULL)
- genotype()
- predictBreedVal(popID = NULL, trainingPopID = NULL)
- select(nSelect = 40, random = F, popID = NULL)
- cross(nProgeny = 100, equalContribution = F, popID = NULL)
- selfFertilize(nProgenyPerInd = 1, popID = NULL)
- doubledHaploid(nProgeny = 100, popID = NULL)
- plotData(ymax = NULL, add = F, addDataFileName = “plotData”)
- outputResults(summarize = T, directory = NULL, saveDataFileName = “BSLoutput”)

We are preparing this section now.

Population ID

Initial population ID = 0

Population ID will be incremented by these functions:

1. select

2. cross

3. selfFertilize
4. doubledHaploid

Be careful that:
select() creates a new population that is a subset of the candidate population

Example 1
“Phenotypic selection & genomic selection”

Code:

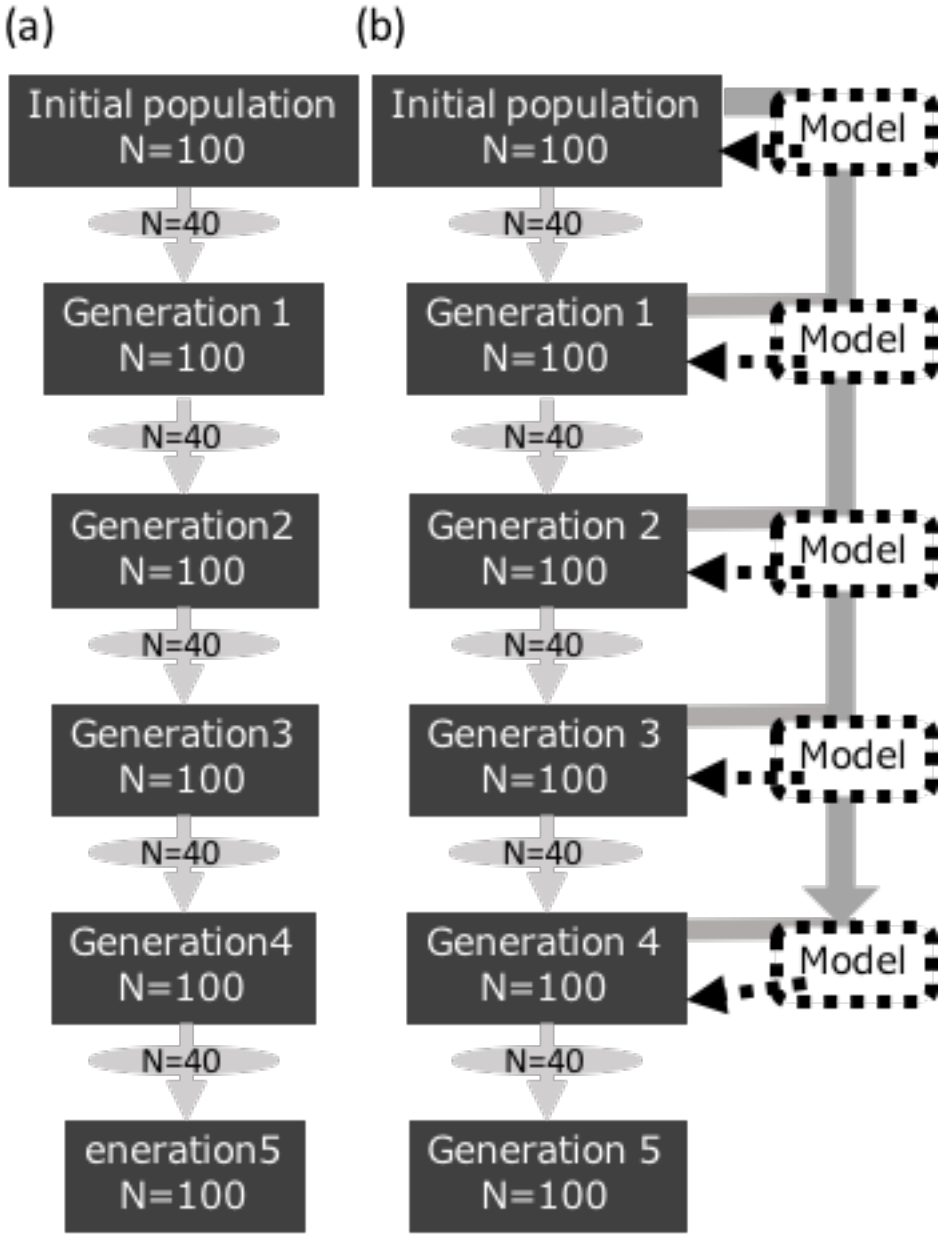
1. phenotypic selection (Fig. a)

```
defineSpecies(nSim = 5)
initializePopulation()
phenotype()
select()
cross()
phenotype()
select()
cross()
phenotype()
select()
cross()
plotData()
```

2. genomic selection (Fig. b)

```
defineSpecies(loadData="previousData")
initializePopulation()
phenotype()
genotype()
predictBreedVal()
select()
cross()
phenotype()
genotype()
predictBreedVal()
select()
cross()
phenotype()
genotype()
```

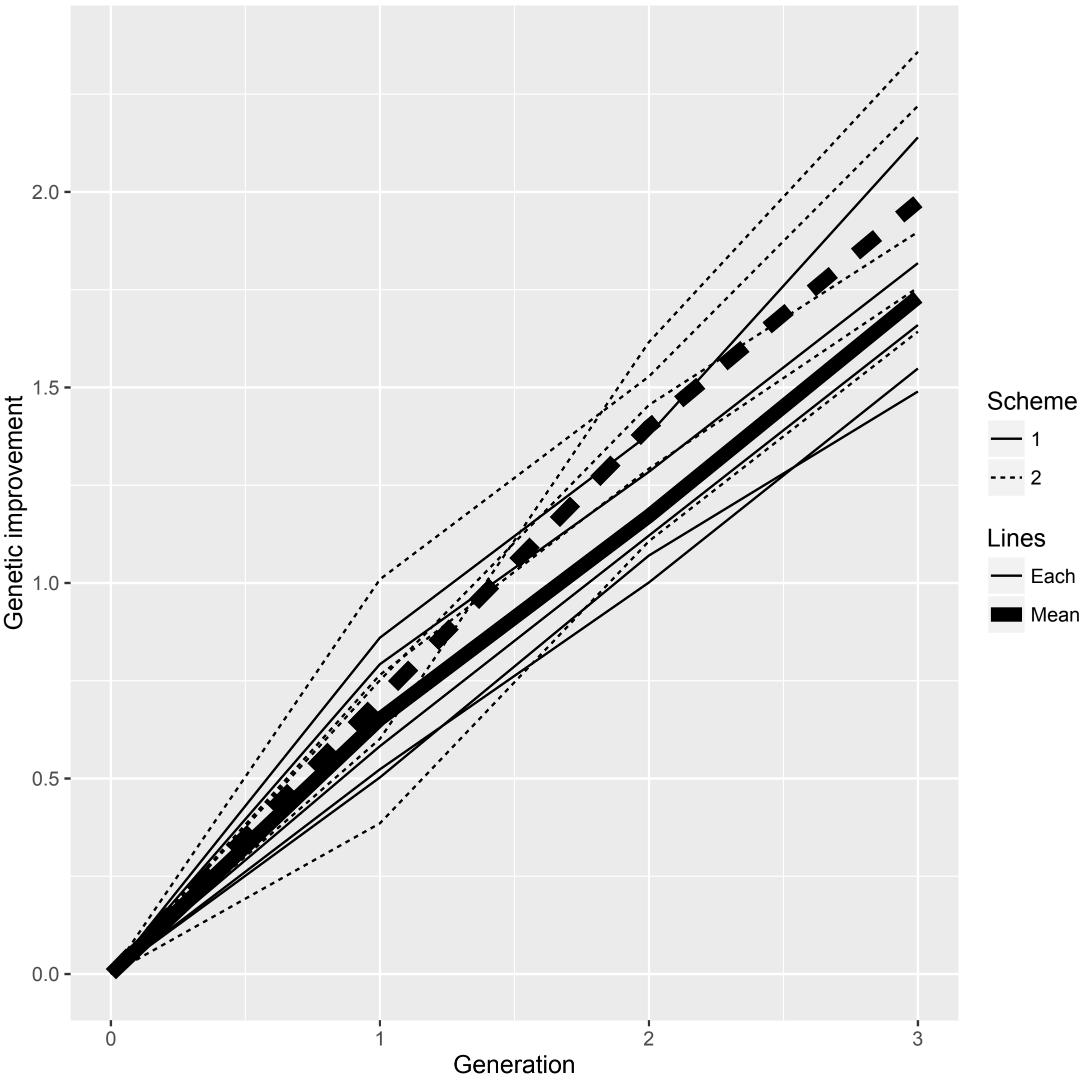
```
predictBreedVal()
select()
cross()
plotData(add=T)
```



Simulation result:

Scheme 1 is phenotypic selection, and Scheme 2 is genomic relection.

Bold lines represent the mean values of simulation trials (in this case, 5 times of simulations), and each thin line is the result of a simulation trial.


Genetic improvement
2.0
1.5
1.0
0.5
0.0
0
1
2
3
Generation
Scheme
1
2
Lines
Each
Mean

Now, we are preparing this manual…