Simulating the GORP

N.R Okereke

2025-01-09

***Load Packages***

library(AlphaSimR)

## Warning: package 'AlphaSimR' was built under R version 4.3.3

## Loading required package: R6

library(tidyverse)

## Warning: package 'lubridate' was built under R version 4.3.3

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.4 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.2

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ dplyr::mutate() masks AlphaSimR::mutate()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

set.seed(7)

***Simulate male sterile lines***

# Donor line  
ms\_lines <- runMacs2(nInd = 3, nChr = 10, segSites = 100, genLen = 2, inbred = TRUE)

***Simulate Diverse Founder Lines***

# Diverse founder lines (35 founders for the NAM population)  
diverseFounders <- runMacs2(nInd = 35, nChr = 10, segSites = 100, genLen = 2, inbred = TRUE)

***Simulate parameters for both donor and diverse lines***

# SimParam for reference and diverse founders  
SP <- SimParam$new(ms\_lines)  
SP$addTraitA(nQtlPerChr = 5)  
#SP$addSnpChip(nSnpPerChr = 500)  
  
SP2 <-SimParam$new(diverseFounders)  
SP2$addTraitA(nQtlPerChr = 5)  
#SP2$addSnpChip(nSnpPerChr = 500)

***Create new populations***

#create population  
pop <- newPop(ms\_lines, simParam = SP)  
pop2 <- newPop(diverseFounders, simParam = SP2)

***Perform random crosses of ms\_lines with founder lines***

#set.seed(7)  
F1 <- hybridCross(pop, pop2, crossPlan = "testcross")

***increase recombination rate***

# Adjust recombination ratio  
#SP$setRecombRatio(2)

***Self F1 generation and generate 6 more generations by SSD*** ***This should represent first GORP CYCLE development***

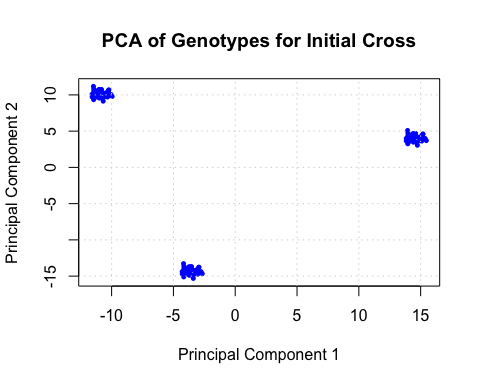
# Generate F2 population from each F1 population  
for (gen in 1:7) {  
 # Perform selfing for each generation of SSD  
 ssdPop <- self(F1, nProgeny = 1) # one progeny per individual (single seed descent)  
   
 # Optionally, track the progress of SSD (e.g., genetic diversity loss)  
 # print(paste("Generation", gen, "completed"))  
}

***Pull segregating sites and visualize***

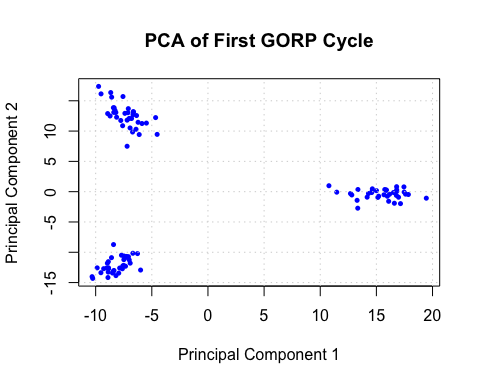
genodata <- pullSegSiteGeno(F1)  
genodata2 <- pullSegSiteGeno(ssdPop)

***Visualize Data First Cycle***

pca <- prcomp(genodata)  
  
pc1 <- pca$x[, 1] # Principal Component 1  
pc2 <- pca$x[, 2] # Principal Component 2  
  
# Step 6: Plot PCA of the SSD population (Base R)  
plot(  
 pc1, pc2,  
 xlab = "Principal Component 1",  
 ylab = "Principal Component 2",  
 main = "PCA of Genotypes for Initial Cross",  
 col = "blue", # Point color  
 pch = 16, # Point shape  
 cex = 0.7 # Point size  
)  
grid()



pca <- prcomp(genodata2)  
  
pc1 <- pca$x[, 1] # Principal Component 1  
pc2 <- pca$x[, 2] # Principal Component 2  
  
# Step 6: Plot PCA of the SSD population (Base R)  
plot(  
 pc1, pc2,  
 xlab = "Principal Component 1",  
 ylab = "Principal Component 2",  
 main = "PCA of First GORP Cycle",  
 col = "blue", # Point color  
 pch = 16, # Point shape  
 cex = 0.7 # Point size  
)  
grid()



***Perform random crosses among first population***

F1\_2 <- randCross(ssdPop, nCrosses = 105, nProgeny = 2)

***Self F1 of the second GORP cycle initial cross and generate 6 more generations by SSD*** ***This should represent second GORP CYCLE development***

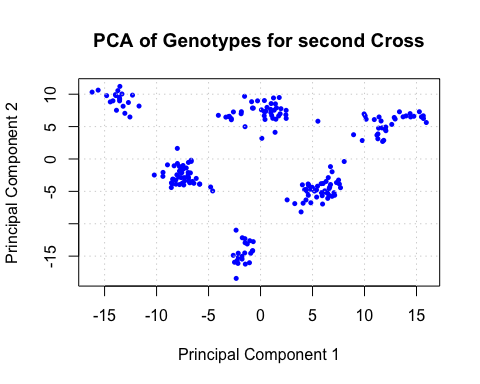
# Generate F2 population from each F1 population  
for (gen in 1:7) {  
 # Perform selfing for each generation of SSD  
 ssdPop2 <- self(F1\_2, nProgeny = 1) # one progeny per individual (single seed descent)  
   
 # Optionally, track the progress of SSD (e.g., genetic diversity loss)  
 # print(paste("Generation", gen, "completed"))  
}

***Pull segregating sites and visualize AGAIN***

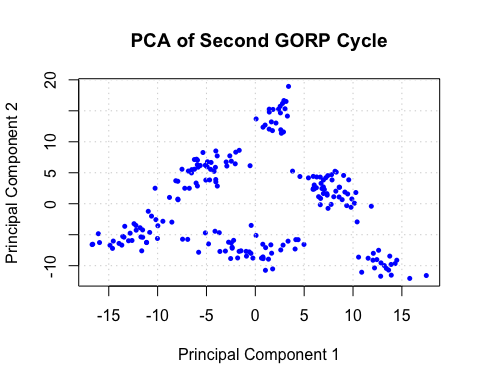
genodata3 <- pullSegSiteGeno(F1\_2)  
genodata4 <- pullSegSiteGeno(ssdPop2)

***Visualize Data second Cycle***

pca <- prcomp(genodata3)  
  
pc1 <- pca$x[, 1] # Principal Component 1  
pc2 <- pca$x[, 2] # Principal Component 2  
  
# Step 6: Plot PCA of the SSD population (Base R)  
plot(  
 pc1, pc2,  
 xlab = "Principal Component 1",  
 ylab = "Principal Component 2",  
 main = "PCA of Genotypes for second Cross",  
 col = "blue", # Point color  
 pch = 16, # Point shape  
 cex = 0.7 # Point size  
)  
grid()



pca <- prcomp(genodata4)  
  
pc1 <- pca$x[, 1] # Principal Component 1  
pc2 <- pca$x[, 2] # Principal Component 2  
  
# Step 6: Plot PCA of the SSD population (Base R)  
plot(  
 pc1, pc2,  
 xlab = "Principal Component 1",  
 ylab = "Principal Component 2",  
 main = "PCA of Second GORP Cycle",  
 col = "blue", # Point color  
 pch = 16, # Point shape  
 cex = 0.7 # Point size  
)  
grid()



***Perform random crosses among second population***

F1\_3 <- randCross(ssdPop2, nCrosses = 210, nProgeny = 2)

***Self F1 of the second GORP cycle initial cross and generate 6 more generations by SSD*** ***This should represent third GORP CYCLE development***

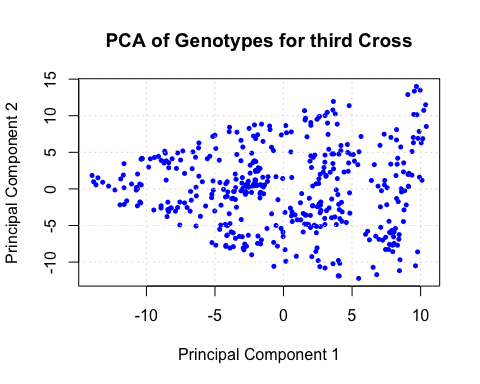
# Generate F2 population from each F1 population  
for (gen in 1:7) {  
 # Perform selfing for each generation of SSD  
 ssdPop3 <- self(F1\_3, nProgeny = 1) # one progeny per individual (single seed descent)  
}

***Pull segregating sites and visualize AGAIN x3***

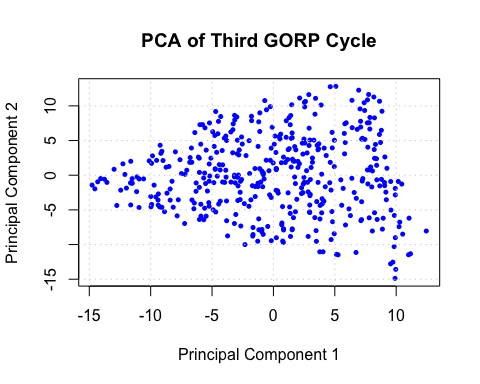
genodata5 <- pullSegSiteGeno(F1\_3)  
genodata6 <- pullSegSiteGeno(ssdPop3)

***Visualize Data second Cycle***

pca <- prcomp(genodata5)  
  
pc1 <- pca$x[, 1] # Principal Component 1  
pc2 <- pca$x[, 2] # Principal Component 2  
  
# Step 6: Plot PCA of the SSD population (Base R)  
plot(  
 pc1, pc2,  
 xlab = "Principal Component 1",  
 ylab = "Principal Component 2",  
 main = "PCA of Genotypes for third Cross",  
 col = "blue", # Point color  
 pch = 16, # Point shape  
 cex = 0.7 # Point size  
)  
grid()



pca <- prcomp(genodata6)  
  
pc1 <- pca$x[, 1] # Principal Component 1  
pc2 <- pca$x[, 2] # Principal Component 2  
  
# Step 6: Plot PCA of the SSD population (Base R)  
plot(  
 pc1, pc2,  
 xlab = "Principal Component 1",  
 ylab = "Principal Component 2",  
 main = "PCA of Third GORP Cycle",  
 col = "blue", # Point color  
 pch = 16, # Point shape  
 cex = 0.7 # Point size  
)  
grid()



***Perform random crosses among third population***

F1\_4 <- randCross(ssdPop3, nCrosses = 420, nProgeny = 2)

***Self F1 of the second GORP cycle initial cross and generate 6 more generations by SSD*** ***This should represent third GORP CYCLE development***

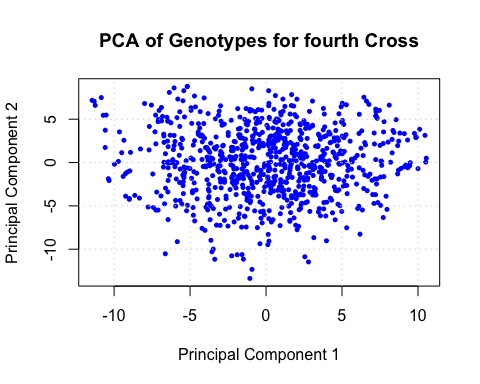
# Generate F2 population from each F1 population  
for (gen in 1:7) {  
 # Perform selfing for each generation of SSD  
 ssdPop4 <- self(F1\_4, nProgeny = 1) # one progeny per individual (single seed descent)  
}

***Pull segregating sites and visualize AGAIN x3***

genodata7 <- pullSegSiteGeno(F1\_4)  
genodata8 <- pullSegSiteGeno(ssdPop4)

***Visualize Data second Cycle***

pca <- prcomp(genodata7)  
  
pc1 <- pca$x[, 1] # Principal Component 1  
pc2 <- pca$x[, 2] # Principal Component 2  
  
# Step 6: Plot PCA of the SSD population (Base R)  
plot(  
 pc1, pc2,  
 xlab = "Principal Component 1",  
 ylab = "Principal Component 2",  
 main = "PCA of Genotypes for fourth Cross",  
 col = "blue", # Point color  
 pch = 16, # Point shape  
 cex = 0.7 # Point size  
)  
grid()



pca <- prcomp(genodata8)  
  
pc1 <- pca$x[, 1] # Principal Component 1  
pc2 <- pca$x[, 2] # Principal Component 2  
  
# Step 6: Plot PCA of the SSD population (Base R)  
plot(  
 pc1, pc2,  
 xlab = "Principal Component 1",  
 ylab = "Principal Component 2",  
 main = "PCA of Fourth GORP Cycle",  
 col = "blue", # Point color  
 pch = 16, # Point shape  
 cex = 0.7 # Point size  
)  
grid()

