

# Package: Ghat (via r-universe)

September 11, 2024

**Title** Quantifying Evolution and Selection on Complex Traits

**Version** 0.1.0

**Description** Functions are provided for quantifying evolution and selection on complex traits. The package implements effective handling and analysis algorithms scaled for genome-wide data and calculates a composite statistic, denoted Ghat, which is used to test for selection on a trait. The package provides a number of simple examples for handling and analysing the genome data and visualising the output and results. Beissinger et al., (2018) <[doi:10.1534/genetics.118.300857](https://doi.org/10.1534/genetics.118.300857)>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**Depends** R (>= 3.0.0)

**URL** <https://www.genetics.org/content/209/1/321>

**BugReports** <https://github.com/Medhat86/Ghat/issues>

**Suggests** knitr, rmarkdown

**Imports** rrBLUP

**NeedsCompilation** no

**Author** Medhat Mahmoud [aut, cre], Ngoc-Thuy Ha [aut], Timothy Beissinger [aut]

**Maintainer** Medhat Mahmoud <[medhat.mahmoud@gwdg.de](mailto:medhat.mahmoud@gwdg.de)>

**Date/Publication** 2019-08-02 10:00:05 UTC

**Repository** <https://medhat86.r-universe.dev>

**RemoteUrl** <https://github.com/medhat86/ghat>

**RemoteRef** HEAD

**RemoteSha** 495bf5c312e854e1b24dcb5708ddb16385878fa5

## Contents

Ghat . . . . .	2
ld_decay . . . . .	4
Maize_wqs . . . . .	5
<b>Index</b>	<b>6</b>

---

Ghat *Quantifying evolution and selection on complex traits*

---

### Description

G-hat: R function to estimate G-hat from allele frequency and effect size data.

### Usage

```
Ghat(effects = effects, change = change, method = "scale",
     perms = 1000, plot = "Both", blockSize = 1000, num_eff = NULL)
```

### Arguments

effects	Vector of allele effects.
change	Vector of changes in allele frequency (could be positive, negative or zero).
method	"vanilla" (assumes complete linkage equilibrium between markers), "trim" (excludes markers to approximate linkage equilibrium some of the extreme values) or "scale" (scales results to reflect underlying levels of linkage LD)
perms	Number of permutations to run.
plot	"Ghat", "Cor", or "Both", Should a plot of the Ghat or correlation test be returned?
blockSize	How large should blocks for trimming be? Only required if method = "trim".
num_eff	The effective number of independent markers, to be used only in conjunction with the "scale" method, above (see "ld_decay" function or use help (?ld_decay).

### Value

Ghat Ghat-value

Cor Correlation between alleles frequencies and their effects

p.val two-sided P-value of Evidence of selection

plot relationship between estimated allelic effects at individual SNPs and the change in allele frequency over generations

## Examples

```
#Example-1 Both SNP effects and change in allele frequency are known
maize <- Maize_wqs[[1]]
result.adf <- Ghat(effects =maize[,1], change=maize[,2], method="scale",
                  perms=1000, plot="Ghat", num_eff=54.74819)
mtext(paste("WQS ADF test for selection, pval = ", round(result.adf$p.val,4)))
message (c(result.adf$Ghat , result.adf$Cor , result.adf$p.va))
```

```
#Example-2 Both SNP effects and change in allele frequency are known
#####
## step 1: #run rrBLUP and estimating allels effects      ##
#####

library(Ghat)
library(parallel)
library(rrBLUP)
phe <- Maize_wqs[[2]]
map <- Maize_wqs[[3]]
gen <- Maize_wqs[[4]]
phe <-phe[which(is.na(phe[,2])==FALSE),]
gen <-gen[which(is.na(phe[,2])==FALSE),]
result <- mixed.solve(phe[,2],
                      Z= as.matrix(gen[,2:ncol(gen)]),
                      X= model.matrix(phe[,2]~phe[,3]),
                      K=NULL, SE=FALSE, return.Hinv=FALSE,
                      method="ML")

#####
## step 2: is to calculate the allele frequency at Cycle 1 and 3##
#####
CycleIndicator <- as.numeric(unlist(strsplit(gen$X,
split="_C")) [seq(2,2*nrow(gen),2)])
Cycle1 <- gen[which(CycleIndicator == 1),]
Cycle3 <- gen[which(CycleIndicator == 3),]
CycleList <- list(Cycle1,Cycle3)
frequencies <- matrix(nrow=ncol(gen)-1,ncol=2)
for(i in 1:2){
  frequencies[,i] <- colMeans(CycleList[[i]][,-1],na.rm=TRUE)/2
}
frequencies <- as.data.frame(frequencies)
names(frequencies) <- c("Cycle1","Cycle3")
change<-frequencies$Cycle3-frequencies$Cycle1

#####
## step 3: Calculate LD Decay      ##
#####
ld <- ld_decay (gen=gen, map=map,
               max_win_snp=2000, max.chr=10,
               cores=1, max_r2=0.03)
```

```
#####
## step 4: Calculate Ghat                                     ##
#####
Ghat.adf <- Ghat(effects=result$u, change=change, method = "scale",
                perms=1000,plot="Ghat", num_eff = 54.74819)

message (paste("Ghat=" , Ghat.adf$Ghat,
              "Cor=" , Ghat.adf$Cor ,
              "P-val=", Ghat.adf$p.va, sep = " "))
```

ld\_decay

*Evaluation of Linkage Disequilibrium Decay***Description**

ld\_decay: R function for calculating the effective number of independent markers

**Usage**

```
ld_decay(gen = gen, map = map, max_win_snp = 2000,
         max.chr = max.chr, cores = 1, max_r2 = max_r2)
```

**Arguments**

gen	Matrix of genotype data. Individuals in rows, genotypes (0, 1, 2) in columns.
map	Dataframe including the name for each marker with a corresponding chromosome number and physical position.
max_win_snp	The maximum number of markers in each window. Sets the maximum number of markers allowed per window within a chromosome before estimating the LD. Default is 2000.
max.chr	Chromosomes above this number will be excluded from the analysis.
cores	Numer of cores for using parallelized calculation, Default is 1 for windows machine.
max_r2	the threshold of $r^2$ to calculate the effective number of independent markers.

**Value**

cor: Correlation matrix

ch\_eff\_nmark: The Number of independent marker per chromosome

eff\_nmark: The effective number of independent markers

### Examples

```
library("parallel")
gen      <- Maize_wqs[[4]]
map      <- Maize_wqs[[3]]
Res_ld <- ld_decay (gen=gen, map=map, max_win_snp=2000,
                  max.chr=10, cores=1, max_r2=0.03)
```

---

Maize_wqs	<i>The Wisconsin Quality Synthetic (WQS) maize population datasets.</i>
-----------	-------------------------------------------------------------------------

---

### Description

The Wisconsin Quality Synthetic (WQS) maize population datasets.

### Usage

```
Maize_wqs
```

### Format

A list of 4 data sets:

**Dataset-1** Data frame including all SNP effects and changes in allele frequencies between cycle 2 and 5.

**Dataset-2** BLUP breeding values for Acid Detergent Fiber (ADF) involving 5 generations of selection.

**Dataset-3** Map file: Each line of the Map file describes a single marker and must contain at least three columns. 1: chromosome number; 2: SNP (snp id); 3: SNP position (in base-pairs (bp)).

**Dataset-4** Maize Genotype (Illumina MaizeSNP50 BeadChip); an Infinium HD assay (Illumina, Inc. San Diego, CA). 10,017 SNP markers (0,1 and 2) after filtration, distributed across the maize genome (Ganal et al.2011).

### Source

Lorenz, A. J., Beissinger, T.M., Rodrigues, R., de Leon, N. 2015. Selection for silage yield and composition did not affect genomic diversity within the Wisconsin Quality Synthetic maize population. *Genes Genomes Genetics*. DOI: 10.1534/g3.114.015263.

### Examples

```
maize <- Maize_wqs[[1]]
phe   <- Maize_wqs[[2]]
map   <- Maize_wqs[[3]]
gen   <- Maize_wqs[[4]]
```

# Index

\* **datasets**

Maize\_wqs, [5](#)

Ghat, [2](#)

ld\_decay, [4](#)

Maize\_wqs, [5](#)