

# Package ‘fsthet’

May 8, 2026

**Version** 1.0.1

**Title** Fst-Heterozygosity Smoothed Quantiles

**Description** A program to generate smoothed quantiles for the Fst-heterozygosity distribution. Designed for use with large numbers of loci (e.g., genome-wide SNPs). The best case for analyzing the Fst-heterozygosity distribution is when many populations (>10) have been sampled. See Flanagan & Jones (2017) <[doi:10.1093/jhered/esx048](https://doi.org/10.1093/jhered/esx048)>.

**Author** Sarah P. Flanagan and Adam G. Jones

**Maintainer** Sarah P. Flanagan <[spflanagan.phd@gmail.com](mailto:spflanagan.phd@gmail.com)>

**License** GPL-2

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Depends** R (>= 2.10)

**Repository** CRAN

**Date/Publication** 2018-03-20 18:27:53 UTC

## Contents

allele.counts . . . . .	2
bins . . . . .	3
calc.actual.fst . . . . .	3
calc.allele.freq . . . . .	4
calc.betahat . . . . .	4
calc.exp.het . . . . .	5
calc.fst . . . . .	6
calc.theta . . . . .	7
ci.means . . . . .	7
cis . . . . .	8
cis.list . . . . .	9
fhetboot . . . . .	9
find.outliers . . . . .	10
find.quantiles . . . . .	11

fst.boot . . . . .	12
fst.boot.means . . . . .	13
fst.boot.onecol . . . . .	14
fst.options.print . . . . .	15
fsthet . . . . .	15
fsts . . . . .	16
fsts.beta . . . . .	16
fsts.betahat . . . . .	17
fsts.theta . . . . .	18
gpop . . . . .	18
make.bins . . . . .	19
my.read.genepop . . . . .	20
p.boot . . . . .	21
plotting.cis . . . . .	21
quant.out . . . . .	23
remove.spaces . . . . .	23
var.fst . . . . .	24
<b>Index</b>	<b>25</b>

---

allele.counts	<i>This counts the number of alleles at a locus.</i>
---------------	--

---

## Description

This counts the number of times each allele occurs at a locus from a list of genotypes (the sum of all the counts is 2\*number of individuals).

## Usage

```
allele.counts(genotypes)
```

## Arguments

genotypes      A list of genotypes.

## Value

AlleleCounts    The number of times each allele is recorded at the locus.

## Examples

```
#create a random sample of genotypes
genotypes<-sample(c("0101","0102","0202"),50,replace=TRUE)
counts<-allele.counts(genotypes)
```

---

bins	<i>Example heterozygosity bins from fsthet.</i>
------	---

---

**Description**

This is a list with a data.frame of bins (the lower and upper bounds for each heterozygosity bin) and a list of fst values that fall into each bin, with the name of each set of Fst values being the upper heterozygosity bound from the data.frame of bins.

**Usage**

```
bins
```

**Format**

```
list
```

**Source**

```
bins<-make.bins(fsts)
```

**References**

See Flanagan & Jones

---

calc.actual.fst	<i>This calculates global Fsts from a genepop dataframe.</i>
-----------------	--

---

**Description**

This calculates global Fsts from a genepop dataframe. This does not include bootstrapping.

**Usage**

```
calc.actual.fst(df, fst.choice="fst")
```

**Arguments**

df	Provide the genepop dataframe (from my.read.genepop).
fst.choice	Specify which type of fst calculation should be used. See fst.options.print for the choices.

**Value**

fsts	This returns a dataframe with Locus, Ht, and Fst characters.
------	--

**Examples**

```

gpop<-data.frame(popinfo=c(rep("POP 1", 20),rep("POP 2", 20)),ind.names=c(1:20,1:20),
  loc0=sample(c("0101","0102","0202"),40,replace=TRUE))
fst<-calc.actual.fst(gpop)
## Not run:
gfile<-system.file("extdata", "example.genepop.txt",package = 'fsthet')
gpop<-my.read.genepop(gfile)
fst<-calc.actual.fst(gpop)

## End(Not run)

```

---

calc.allele.freq      *This calculates allele frequencies.*

---

**Description**

This calculates allele frequencies from a list of genotypes.

**Usage**

```
calc.allele.freq(genotypes)
```

**Arguments**

genotypes      A list of genotypes.

**Value**

obs.af      A list of observed allele frequencies in the genotypes list.

**Examples**

```

#create a random sample of genotypes
genotypes<-sample(c("0101","0102","0202"),50,replace=TRUE)
af<-calc.allele.freq(genotypes)

```

---

calc.betahat      *This calculates beta-hat, the Fst value used in Lositan.*

---

**Description**

This calculates Weir & Cockerham (1993)'s beta-hat. Beaumont & Nichols (1996) used this formulation in FDIST2 (and is implemented in Lositan) See the vignette for details on the calculation of beta.

**Usage**

```
calc.betahat(df, i)
```

**Arguments**

**df** A dataframe containing the genepop information, where the first column is the population ID.

**i** Column number containing genotype information.

**Value**

**ht** HB (or 1-F1). This is a single numerical value.

**fst** The calculated betahat value  $((F0-F1)/(1-F1))$  for this locus.

**Examples**

```
gpop<-data.frame(popinfo=c(rep("POP 1", 20),rep("POP 2", 20)),ind.names=c(1:20,1:20))
for(i in 1:40){
  gpop[1:20,(i+2)]<-sample(c("0101","0102","0202"),20,replace=TRUE)
  gpop[21:40,(i+2)]<-sample(c("0101","0102","0202"),20,replace=TRUE)
}
bh<-calc.betahat(gpop, 3) #calculate betahat for the SNP

gfile<-system.file("extdata", "example.genepop.txt",package = 'fsthet')
gpop<-my.read.genepop(gfile)
beta1<-calc.betahat(gpop,3) #calculate betahat for the first SNP
```

---

calc.exp.het

*This calculates expected heterozygosities.*

---

**Description**

This calculates expected heterozygosities from a list of allele frequencies.

**Usage**

```
calc.exp.het(af)
```

**Arguments**

**af** is a list of allele frequencies.

**Value**

**ht** The expected heterozygosity under Hardy-Weinberg expectations. This is a single numerical value.

**Examples**

```
#create a random sample of genotypes
genotypes<-sample(c("0101","0102","0202"),50,replace=TRUE)
af<-calc.allele.freq(genotypes)
hs<-calc.exp.het(af)
```

---

calc.fst

*This calculates Fst.*

---

**Description**

This calculates Fst. The calculation is done as  $(Ht-Hs)/Ht$ , where Ht is the expected heterozygosity for all populations and Hs is the expected heterozygosity for each population. This calculation is used in bootstrapping functions.

**Usage**

```
calc.fst(df, i)
```

**Arguments**

df	A dataframe containing the genepop information, where the first column is the population ID.
i	Column number containing genotype information.

**Value**

ht	The expected heterozygosity under Hardy-Weinberg expectations. This is a single numerical value.
fst	The calculated Fst value for this locus.

**Examples**

```
gpop<-data.frame(popinfo=c(rep("POP 1", 20),rep("POP 2", 20)),ind.names=c(1:20,1:20),
  loc0=sample(c("0101","0102","0202"),40,replace=TRUE))
fst1<-calc.fst(gpop,3)
## Not run:
gfile<-system.file("extdata", "example.genepop.txt",package = 'fsthet')
gpop<-my.read.gpop(gfile)
fst1<-calc.fst(gpop,3) #calculate fst for the first SNP

## End(Not run)
```

---

calc.theta	<i>This calculates theta.</i>
------------	-------------------------------

---

**Description**

This calculates Weir (1990)'s theta. See the vignette for details on the calculation of beta.

**Usage**

```
calc.theta(df, i)
```

**Arguments**

df	A dataframe containing the genepop information, where the first column is the population ID.
i	Column number containing genotype information.

**Value**

ht	T2. This is a single numerical value.
fst	The calculated theta value (T1/T2) for this locus.

**Examples**

```
gpop<-data.frame(popinfo=c(rep("POP 1", 20),rep("POP 2", 20)),ind.names=c(1:20,1:20),
  loc0=sample(c("0101","0102","0202"),40,replace=TRUE))
theta1<-calc.theta(gpop, 3)
## Not run:
gfile<-system.file("extdata", "example.genepop.txt",package = 'fsthet')
gpop<-my.read.gpop(gfile)
theta1<-calc.theta(gpop,3) #calculate theta for the first SNP

## End(Not run)
```

---

ci.means	<i>This calculates the average confidence intervals from multiple bootstrap outputs.</i>
----------	--

---

**Description**

This calculates the mean upper and lower confidence intervals from a list of bootstrap CI matrices.

**Usage**

```
ci.means(boot.out.list)
```

**Arguments**

`boot.out.list` A list of matrices. Each matrix is the CIs from `fst.boot` (`boot.out[[3]]`).

**Value**

`avg.cil` A list of the average lower CI values

`avg.ciu` A list of the average upper CI values

**Examples**

```
## Not run:
gfile<-system.file("extdata", "example.genepop.txt",package = 'fsthet')
gpop<-my.read.genepop(gfile)
quant.out<-fst.boot(gpop, bootstrap = FALSE)
quant.list<-ci.means(quant.out[[3]])
```

```
## End(Not run)
```

---

`cis`

*Example dataframe of smoothed quantiles from fsthet*

---

**Description**

Example list of data.frames with smoothed quantiles from `fsthet` output from numerical simulations. The data were generated using a numerical analysis with  $N_m = 10$ , 75 demes, and 5 population samples taken. No selection was imposed. This is a list with a single data.frame containing values from 95 percent smoothed quantiles.

**Usage**

```
cis
```

**Format**

```
list
```

**Source**

Ninety-five percent smoothed quantiles, using the dataframe `gpop`.

**References**

See Flanagan & Jones

---

cis.list	<i>Example list of CI matrices from bootstrap output from numerical simulations</i>
----------	---

---

**Description**

Example list of CI data.frames from fsthet output from numerical simulations. The data were generated using a numerical analysis with  $N_m = 10$ , 75 demes, and 5 population samples taken. No selection was imposed. This is a list of data.frames containing values from 99 percent and 95 percent smoothed quantiles.

**Usage**

```
cis.list
```

**Format**

```
list
```

**Source**

From multiple smoothed quantile alpha thresholds, using the dataframe gpop.

**References**

See Flanagan & Jones

---

fhetboot	<i>This is a wrapper to run the bootstrapping and plot the confidence intervals and significant loci.</i>
----------	---

---

**Description**

This calculates global Fsts from a genepop dataframe and then does: p-value calculations plots the Heterozygosity-Fst relationship with smoothed CIs outputs the loci lying outside the confidence intervals. Returns a data frame containing Locus ID, Ht, Fst, P-value, a Benjamini-Hochberg-corrected P-value, and a true/false value of whether it's an outlier.

**Usage**

```
fhetboot(gpop, fst.choice="fst", alpha=0.05, nreps=10)
```

**Arguments**

gpop	Provide the genepop dataframe (from my.read.genepop).
fst.choice	Specify which type of fst calculation should be used. See fst.options.print for the choices.
alpha	The alpha value for the confidence intervals and the p-value adjustment calculations (default is 0.05).
nreps	The number of bootstrap replicates to use. The default is 10.

**Value**

fsts	This returns a dataframe with Locus, Ht, Fst, P-value, correcte P-value, and True/False of whether it's an outlier.
------	---

**Examples**

```
## Not run:
gfile<-system.file("extdata", "example.genepop.txt",package = 'fsthet')
gpop<-my.read.genepop(gfile)
out.dat<-fhetboot(gpop, fst.choice="fst", alpha=0.05,nreps=10)

## End(Not run)
```

---

find.outliers	<i>This identifies all of the SNPs outside of the smoothed quantiles in the dataset.</i>
---------------	--

---

**Description**

This identifies all of the SNPs outside of the smoothed quantiles in the dataset.

**Usage**

```
find.outliers(df, boot.out, ci.df = NULL, file.name = NULL)
```

**Arguments**

df	Provide the dataframe with Ht and Fst values.
boot.out	Bootstrap output. You must provide this.
ci.df	List of confidence intervals. You may provide this in addition to bootstrap output to save a small amount of time.
file.name	You may provide a file name to output the outliers to a csv file. Otherwise, the function will only return the outliers.

**Value**

out	A list of the outlier loci
-----	----------------------------

**Examples**

```
## Not run:
gfile<-system.file("extdata", "example.genepop.txt",package = 'fsthet')
gpop<-my.read.gpop(gfile)
fst<-calc.actual.fst(gpop)
boot.out<-as.data.frame(t(replicate(10, fst.boot(gpop))))
outliers<-find.outliers(fst,boot.out)

## End(Not run)
```

---

find.quantiles	<i>Generates quantiles from binned Fst values</i>
----------------	---

---

**Description**

This takes the output from make.bins and calculates the smoothed quantiles.

**Usage**

```
find.quantiles(bins,bin.fst,ci=0.05)
```

**Arguments**

bins	A dataframe containing with upper and lower het and Fst values for each bin (output from make.bins).
bin.fst	A list with the Fst values for each bin (output from make.bins).
ci	A value for the confidence intervals alpha (default is 0.05).

**Value**

fst.CI	A list of data.frames, one for each ci value with the upper and lower Fst quantiles for each Heterozygosity bin.
--------	--

**Examples**

```
gpop<-data.frame(popinfo=c(rep("POP 1", 20),rep("POP 2", 20)),ind.names=c(1:20,1:20))
for(i in 1:40){
  gpop[1:20,(i+2)]<-sample(c("0101","0102","0202"),20,replace=TRUE)
  gpop[21:40,(i+2)]<-sample(c("0101","0102","0202"),20,replace=TRUE)
}
fst<-calc.actual.fst(gpop)
nloci<-(ncol(gpop)-2)
boot.out<-as.data.frame(t(replicate(nloci, fst.boot.onecol(gpop,"fst"))))
bins<-make.bins(boot.out,25,Ht.name="V1",Fst.name="V2")
fst.CI<-find.quantiles(bins$bins,bins$bin.fst)
```

---

fst.boot	<i>This is the major bootstrapping function to calculate confidence intervals.</i>
----------	--

---

### Description

This randomly samples all of the loci, with replacement (so if you have 200 loci, it will choose 200 loci to calculate Fst for, but some may be sampled multiply) It makes use of `fst.boot.onerow`. To calculate the confidence intervals, this function bins the Fst values based on heterozygosity values. The bins are overlapping and each bin is the width of `smooth.rate`. The Fst value which separates the top  $100*(ci/2)$  and bottom  $100*(ci/2)$  percent in each bin are the upper and lower CIs. This function can be slow. We recommend running it 10 times to generate confidence intervals for analysis.

### Usage

```
fst.boot(df, fst.choice="fst", ci=0.05, num.breaks=25, bootstrap = TRUE, min.per.bin=20)
```

### Arguments

<code>df</code>	A dataframe containing the genepop information, where the first column is the population ID.
<code>fst.choice</code>	A character defining which fst calculation is to be used. See <code>fst.options.print()</code> for the choices.
<code>ci</code>	A value for the confidence intervals alpha (default is 0.05).
<code>num.breaks</code>	The number of breaks used to create bins (default is 25)
<code>bootstrap</code>	A TRUE/FALSE statement telling the program whether to bootstrap and then determine the bins or to calculate bins and confidence intervals from the empirical dataset without bootstrapping. The default is TRUE, which means bootstrapping occurs.
<code>min.per.bin</code>	The minimum number of loci that are required for a bin to be retained. Default is 20.

### Value

<code>Fsts</code>	The bootstrapped Fst and Ht values
<code>Bins</code>	A dataframe containing the bins start and stop Ht values.
<code>fst.CI</code>	A list of dataframes containing the lower and upper confidence intervals' Ht values.

### Examples

```
gpop<-data.frame(popinfo=c(rep("POP 1", 20),rep("POP 2", 20)),ind.names=c(1:20,1:20))
for(i in 1:40){
  gpop[1:20,(i+2)]<-sample(c("0101","0102","0202"),20,replace=TRUE)
  gpop[21:40,(i+2)]<-sample(c("0101","0102","0202"),20,replace=TRUE)
}
```

```

fsts<-calc.actual.fst(gpop)
quant.out<-as.data.frame(t(replicate(1, fst.boot(gpop,bootstrap=FALSE))))
## Not run:
  gfile<-system.file("extdata", "example.genepop.txt",package = 'fsthet')
  gpop<-my.read.genepop(gfile)
  fsts<-calc.actual.fst(gpop)
  quant.out<-as.data.frame(t(replicate(1, fst.boot(gpop,bootstrap=FALSE))))

## End(Not run)

```

---

fst.boot.means	<i>Calculates mean values within the bins.</i>
----------------	--

---

## Description

This calculates mean heterozygosity and Fst values for each bin used in bootstrapping.

## Usage

```
fst.boot.means(boot.out)
```

## Arguments

boot.out      The first item in the output lists from fst.boot (aka boot.out[[1]]).

## Value

bmu            A dataframe containing four columns: heterozygosity Fst the number of loci in the bin the lower Ht value for the bin and the upper Ht value for the bin.

## Examples

```

gpob<-data.frame(popinfo=c(rep("POP 1", 20),rep("POP 2", 20)),ind.names=c(1:20,1:20))
for(i in 1:40){
  gpob[1:20,(i+2)]<-sample(c("0101", "0102", "0202"),20,replace=TRUE)
  gpob[21:40,(i+2)]<-sample(c("0101", "0102", "0202"),20,replace=TRUE)
}
fsts<-calc.actual.fst(gpob)
boot.out<-as.data.frame(t(replicate(1, fst.boot(gpob))))
outliers<-find.outliers(fsts,boot.out)
## Not run:
  gfile<-system.file("extdata", "example.genepop.txt",package = 'fsthet')
  gpob<-my.read.genepop(gfile)
  fsts<-calc.actual.fst(gpob)
  boot.out<-as.data.frame(t(replicate(10, fst.boot(gpob))))
  outliers<-find.outliers(fsts,boot.out)

## End(Not run)

```

---

fst.boot.onecol	<i>This bootstraps across all individuals to calculate a bootstrapped Fst for a randomly-sampled locus.</i>
-----------------	---

---

### Description

This calculates Fst using calc.fst. It randomly selects a column containing genotype information for all individuals. It then calculates Fst and Ht for that locus.

### Usage

```
fst.boot.onecol(df, fst.choice)
```

### Arguments

df	A dataframe containing the genepop information, where the first column is the population ID.
fst.choice	A character defining which fst calculation is to be used. The three options are: Wright's Fst (Wright, wright, WRIGHT, W, w) Weir and Cockerham 1993's beta (WeirCockerham, weircockerham, wc, WC) Corrected Weir and Cockerham 1993's beta from Beaumont and Nichols 1996 (WeirCockerhamCorrected, weircockerhamcorrected, corrected, wcc, WCC)

### Value

ht.fst	A vector containin Ht and Fst
--------	-------------------------------

### Examples

```
gpop<-data.frame(popinfo=c(rep("POP 1", 20),rep("POP 2", 20)),ind.names=c(1:20,1:20))
for(i in 1:40){
  gpop[1:20,(i+2)]<-sample(c("0101","0102","0202"),20,replace=TRUE)
  gpop[21:40,(i+2)]<-sample(c("0101","0102","0202"),20,replace=TRUE)
}
fsts<-calc.actual.fst(gpop)
nloci<-(ncol(gpop)-2)
boot.out<-as.data.frame(t(replicate(nloci, fst.boot.onecol(gpop,"fst"))))
## Not run:
gfile<-system.file("extdata", "example.genepop.txt",package = 'fsthet')
gpop<-my.read.genepop(gfile)
fsts<-calc.actual.fst(gpop)
nloci<-(ncol(gpop)-2)
boot.out<-as.data.frame(t(replicate(nloci, fst.boot.onecol(gpop,"fst"))))
outliers<-find.outliers(fsts,boot.out)

## End(Not run)
```

---

fst.options.print	<i>This prints the options for choosing an Fst calculation.</i>
-------------------	---

---

**Description**

This prints the options for choosing an Fst calculation.

**Usage**

```
fst.options.print()
```

**Examples**

```
fst.options.print()
```

---

fsthet	<i>This is a wrapper to generate and plot the smoothed quantiles and identify outliers.</i>
--------	---

---

**Description**

This calculates global Fsts from a genepop dataframe and then does: calculates smoothed quantiles plots the Heterozygosity-Fst relationship with smoothed quantiles outputs the loci lying outside the quantiles. Returns a data frame containing Locus ID, Ht, Fst, and a true/false value of whether it's an outlier.

**Usage**

```
fsthet(gpop, fst.choice="fst", alpha=0.05)
```

**Arguments**

gpop	Provide the genepop dataframe (from my.read.genepop).
fst.choice	Specify which type of fst calculation should be used. See fst.options.print for the choices.
alpha	The alpha value for the quantiles (default is 0.05 to generate 95 percent quantiles).

**Value**

fstst	This returns a dataframe with Locus, Ht, Fst, and True/False of whether it's an outlier.
-------	--

## Examples

```
## Not run:
gfile<-system.file("extdata", "example.genepop.txt",package = 'fsthet')
gpop<-my.read.gpop(gfile)
out.dat<-fsthet(gpop)

## End(Not run)
```

---

fstst

*Example fst calculations from a genepop file.*

---

## Description

Example fst calculations from a genepop file. The original data were generated by using a numerical analysis with  $N_m = 10$ , 75 demes, and 5 population samples taken. No selection was imposed. The fstst were calculated using `calc.actual.fst(gpop)` This file contains a dataframe with 2000 columns and 3 rows. The first column is the Locus ID, the second column is the  $H_t$  for that locus, and the third column is the  $F_{st}$  for that locus.

## Usage

fstst

## Format

data.frame

## Source

Generated by numerical analysis

## References

See Flanagan & Jones

---

fstst.beta

*Example fst calculations from a genepop file.*

---

## Description

Example fst calculations using beta (`fst.choice="var"`) from a genepop file. The original data were generated by using a numerical analysis with  $N_m = 10$ , 75 demes, and 5 population samples taken. No selection was imposed. The fstst were calculated using `calc.actual.fst(gpop,fst.choice="var")` This file contains a dataframe with 2000 columns and 3 rows. The first column is the Locus ID, the second column is the  $H_t$  for that locus, and the third column is the  $F_{st}$  for that locus.

**Usage**

```
fsts.beta
```

**Format**

```
data.frame
```

**Source**

Generated by numerical analysis

**References**

See Flanagan & Jones

---

```
fsts.betahat
```

*Example fst calculations from a genepop file.*

---

**Description**

Example fst calculations using betahat (fst.choice="betahat") from a genepop file. The original data were generated by using a numerical analysis with  $N_m = 10$ , 75 demes, and 5 population samples taken. No selection was imposed. The fsts were calculated using `calc.actual.fst(gpop,fst.choice="betahat")`. This file contains a dataframe with 2000 columns and 3 rows. The first column is the Locus ID, the second column is the Ht for that locus, and the third column is the Fst for that locus.

**Usage**

```
fsts.betahat
```

**Format**

```
data.frame
```

**Source**

Generated by numerical analysis

**References**

See Flanagan & Jones

---

`fsts.theta`*Example fst calculations from a genepop file.*

---

**Description**

Example fst calculations using theta (`fst.choice="theta"`) from a genepop file. The original data were generated by using a numerical analysis with  $N_m = 10$ , 75 demes, and 5 population samples taken. No selection was imposed. The fsts were calculated using `calc.actual.fst(gpop,fst.choice="theta")`. This file contains a dataframe with 2000 columns and 3 rows. The first column is the Locus ID, the second column is the  $H_t$  for that locus, and the third column is the  $F_{st}$  for that locus.

**Usage**`fsts.theta`**Format**`data.frame`**Source**

Generated by numerical analysis

**References**

See Flanagan &amp; Jones

---

`gpop`*Example genepop file from numerical simulations*

---

**Description**

Example genepop file from numerical simulations. It was generated by using a numerical analysis with  $N_m = 10$ , 75 demes, and 5 population samples taken. No selection was imposed. This file contains a dataframe with 2002 columns and 250 rows. The first two columns are the population name and the individual name. The remaining columns are genotypes for each locus (one column per locus). Each row is an individual.

**Usage**`gpop`**Format**`data.frame`

**Source**

Generated by numerical analysis

**References**

See Flanagan & Jones

---

make.bins	<i>This sorts Fst values into a designated number of overlapping heterozygosity bins.</i>
-----------	---

---

**Description**

This breaks up Fst values into a designated number of overlapping heterozygosity bins. It returns a list containing a data.frame called bins a list called bin.fst with the Fst values for each of the Het categories.

**Usage**

```
make.bins(fsts,num.breaks=25, Ht.name="Ht", Fst.name="Fst",min.per.bin=20)
```

**Arguments**

fsts	A dataframe containing at least the columns with heterozygosity and Fst values.
num.breaks	The number of breaks used to create bins (default is 25)
Ht.name	Provide the name of the column with the heterozygosity values, unless the column is named "Ht".
Fst.name	Provide the name of the column with the Fst values, unless the column is named "Fst".
min.per.bin	If you have a smaller dataset, you can change the minimum number of loci required to be in each bin. Default is 20.

**Value**

```
list(bins, bin.fst)
```

A list with a data.frame called bins with the upper and lower Fst and Ht values and a list called bin.fst with the Fst values for each of the Het categories.

**Examples**

```
gpop<-data.frame(popinfo=c(rep("POP 1", 20),rep("POP 2", 20)),ind.names=c(1:20,1:20))
for(i in 1:40){
  gpop[1:20,(i+2)]<-sample(c("0101","0102","0202"),20,replace=TRUE)
  gpop[21:40,(i+2)]<-sample(c("0101","0102","0202"),20,replace=TRUE)
}
fsts<-calc.actual.fst(gpop)
nloci<-(ncol(gpop)-2)
```

```

boot.out<-as.data.frame(t(replicate(nloci, fst.boot.onecol(gpop,"fst"))))
bins<-make.bins(boot.out,25,Ht.name="V1",Fst.name="V2")
## Not run:
  gfile<-system.file("extdata", "example.genepop.txt",package = 'fsthet')
  gpop<-my.read.genepop(gfile)
  fsts<-calc.actual.fst(gpop)
  nloci<-(ncol(gpop)-2)
  boot.out<-as.data.frame(t(replicate(nloci, fst.boot.onecol(gpop))))
  make.bins(boot.out,25)

## End(Not run)

```

---

my.read.genepop

*This reads a genepop file into R*


---

## Description

This reads a genepop file into R. It was adapted from a similar function in adegenet.

## Usage

```
my.read.genepop(file, ncode = 2L, quiet = FALSE)
```

## Arguments

file	is the filename of the genepop file.
quiet	If quiet = FALSE updates will be printed. If quiet = T status updates will not be printed.
ncode	Do not change this argument.

## Value

res	A dataframe with the Population ID in column 1, the Individual ID in column 2, and the genotypes in columns following that. There is one row per individual.
-----	--

## References

<http://adegenet.r-forge.r-project.org/>

## Examples

```

gfile<-system.file("extdata", "example.genepop.txt",package = 'fsthet')
gpob<-my.read.genepop(gfile)

```

---

p.boot *Calculates mean values within the bins.*

---

### Description

This calculates mean heterozygosity and Fst values for each bin used in bootstrapping.

### Usage

```
p.boot(actual.fsts, boot.out,boot.means=NULL)
```

### Arguments

actual.fsts      The first item in the output lists from fst.boot.  
 boot.out         The output from a bootstrapping run. Either supply this or boot.means.  
 boot.means      The output from fst.boot.means. Either supply this or bootstrapping output.

### Value

pvals            A numeric containing uncorrected p-values for each locus. The names attribute are the locus names.

### Examples

```
## Not run:
gfile<-system.file("extdata", "example.genepop.txt",package = 'fsthet')
gpop<-my.read.genepop(gfile)
fsts<-calc.actual.fst(gpop)
boot.out<-as.data.frame(t(replicate(10, fst.boot(gpop))))
boot.pvals<-p.boot(fsts,boot.out=boot.out)

## End(Not run)
```

---

plotting.cis *This plots a dataframe of fsts with bootstrapped confidence intervals.*

---

### Description

This plots a dataframe of fsts with bootstrapped confidence intervals.

### Usage

```
plotting.cis(df,boot.out,ci.df=NULL,sig.list=NULL,Ht.name="Ht",Fst.name="Fst",
ci.col="red", pt.pch=1,file.name=NULL,sig.col=ci.col,make.file=TRUE)
```

**Arguments**

<code>df</code>	A dataframe of Fst and Ht values. It must have at least two columns, one named "Ht" and one named "Fst". Or you must pass the column names to the function
<code>boot.out</code>	Bootstrap output. You must either provide this or a list of confidence interval values.
<code>ci.df</code>	Data frame of confidence intervals. You must either provide this or bootstrap output.
<code>sig.list</code>	List of significant locus names (this acts as a way to highlight particular loci). This is optional and colors some of the points using the same shape as <code>pt.pch</code> and the color of <code>sig.col</code> (default <code>sig.color</code> is same as <code>ci.col</code> ).
<code>Ht.name</code>	Provide the name of the column with the heterozygosity values, unless the column is named "Ht".
<code>Fst.name</code>	Provide the name of the column with the Fst values, unless the column is named "Fst".
<code>ci.col</code>	You can input the colors of the confidence intervals to be plotted. First is the 95 percent CI, second is the 99 percent CI. Defaults are "red" and "gold".
<code>pt.pch</code>	You can change the point shape here. Default is 1 (open circles)
<code>sig.col</code>	The color of the significant loci, if that option is taken. The default is the same color as the confidence interval.
<code>file.name</code>	You can provide the filename. If not provided, default is "OutlierLoci" in the current directory.
<code>make.file</code>	A boolean value (TRUE or FALSE). If TRUE, a file will be created with the plot. If FALSE, the plot will be made in R only (and can be further annotated).

**Examples**

```

gpop<-data.frame(popinfo=c(rep("POP 1", 20),rep("POP 2", 20)),ind.names=c(1:20,1:20),
                 loc0=sample(c("0101","0102","0202"),40,replace=TRUE),
                 loc1=sample(c("0101","0102","0202"),40,replace=TRUE))
fst<-calc.actual.fst(gpop)
bins<-make.bins(fst)
cis<-find.quantiles(bins = bins$bins,bin.fst = bins$bin.fst)
quant.list<-cis$CI0.95
plotting.cis(df=fst,ci.df=quant.list,make.file=FALSE)
## Not run:
load(fst)
bins<-make.bins(fst)
cis<-find.quantiles(bins = bins$bins,bin.fst = bins$bin.fst)
quant.list<-cis$CI0.95
plotting.cis(df=fst,ci.df=quant.list,make.file=FALSE)

## End(Not run)

```

---

 quant.out

*Example fsthet output based on numerical simulations*


---

**Description**

Example fsthet output based on numerical simulations Allelic information was generated by using a numerical analysis with  $N_m = 10$ , 75 demes, and 5 population samples taken. No selection was imposed. This is a list of three structures. The first is a data.frame containing the Ht and Fst values. The second is a data.frame of the bins with the lower heterozygosity values and the upper heterozygosity values for each bin. The third is a list of data.frames with the lower (Low) and upper (Upp) Fst values for each bin (the bins are in "LowHet" and "UppHet" columns.)

**Usage**

```
quant.out
```

**Format**

```
list
```

**Source**

Smoothed quantiles generated from the dataframe gpop.

**References**

See Flanagan & Jones

---

 remove.spaces

*This removes spaces from a character vector*


---

**Description**

This removes spaces from a before and after words in a character vector. It was adapted from a similar function in adegenet.

**Usage**

```
remove.spaces(charvec)
```

**Arguments**

charvec is a vector of characters containing spaces to be removed.

**Value**

charvec A vector of characters without spaces

## References

<http://adegenet.r-forge.r-project.org/>

## Examples

```
charvec<-c("this ", " is", " a", " test")
remove.spaces(charvec)
```

---

var.fst

*This calculates Cockerham & Weir's Beta.*

---

## Description

This calculates Weir & Cockerham (1993)'s Fst. The calculation is based on variance in allele frequencies. See the vignette for details on the calculation of beta.

## Usage

```
var.fst(df, i)
```

## Arguments

df	A dataframe containing the genepop information, where the first column is the population ID.
i	Column number containing genotype information.

## Value

ht	$2p\bar{p}(1-p\bar{p})$ . This is a single numerical value.
fst	The calculated beta value for this locus.

## Examples

```
gpop<-data.frame(popinfo=c(rep("POP 1", 20),rep("POP 2", 20)),ind.names=c(1:20,1:20),
  loc0=sample(c("0101","0102","0202"),40,replace=TRUE),
  loc1=sample(c("0101","0102","0202"),40,replace=TRUE))
var1<-var.fst(gpop,3)
## Not run:
gfile<-system.file("extdata", "example.genepop.txt",package = 'fsthet')
gpop<-my.read.gpop(gfile)
var1<-var.fst(gpop,3) #calculate variance-based for the first SNP

## End(Not run)
```

# Index

`allele.counts`, 2

`bins`, 3

`calc.actual.fst`, 3

`calc.allele.freq`, 4

`calc.betahat`, 4

`calc.exp.het`, 5

`calc.fst`, 6

`calc.theta`, 7

`ci.means`, 7

`cis`, 8

`cis.list`, 9

`fhetboot`, 9

`find.outliers`, 10

`find.quantiles`, 11

`fst.boot`, 12

`fst.boot.means`, 13

`fst.boot.onecol`, 14

`fst.options.print`, 15

`fsthet`, 15

`fsts`, 16

`fsts.beta`, 16

`fsts.betahat`, 17

`fsts.theta`, 18

`gpop`, 18

`make.bins`, 19

`my.read.genepop`, 20

`p.boot`, 21

`plotting.cis`, 21

`quant.out`, 23

`remove.spaces`, 23

`var.fst`, 24