Package ‘qtl2fst’

July 22, 2025

Version 0.30

Date 2024-11-23

Title Database Storage of Genotype Probabilities for QTL Mapping

Description Uses the 'fst' package to store genotype probabilities on disk for the 'qtl2' package. These genotype probabilities are a central data object for mapping quantitative trait loci (QTL), but they can be quite large. The facilities in this package enable the genotype probabilities to be stored on disk, leading to reduced memory usage with only a modest increase in computation time.

Author Karl W Broman [aut, cre] (ORCID:
<`https://orcid.org/0000-0002-4914-6671`>),
Brian S Yandell [aut] (ORCID: <`https://orcid.org/0000-0002-8774-9377`>),
Petr Simecek [aut] (ORCID: <`https://orcid.org/0000-0002-2922-7183`>)

Maintainer Karl W Broman <`broman@wisc.edu`>

Depends R (>= 3.1.0)

Imports fst, qtl2 (>= 0.24)

Suggests testthat, knitr, rmarkdown

VignetteBuilder knitr

License GPL-3

URL `https://github.com/rqtl/qtl2fst`

ByteCompile true

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Repository CRAN

Date/Publication 2024-11-23 21:00:02 UTC

Contents

`calc_genoprob_fst` *Calculate conditional genotype probabilities and write to fst database*

Description

Uses a hidden Markov model to calculate the probabilities of the true underlying genotypes given the observed multipoint marker data, with possible allowance for genotyping errors.

Usage

```
calc_genoprob_fst(
  cross,
  fbase,
  fdir = ".",
  map = NULL,
  error_prob = 0.0001,
  map_function = c("haldane", "kosambi", "c-f", "morgan"),
  lowmem = FALSE,
  quiet = TRUE,
  cores = 1,
  compress = 0,
  overwrite = FALSE
)
```

Arguments

`cross`	Object of class `"cross2"`. For details, see the R/qtl2 developer guide.
`fbase`	Base of filename for fst database.
`fdir`	Directory for fst database.
`map`	Genetic map of markers. May include pseudomarker locations (that is, locations that are not within the marker genotype data). If NULL, the genetic map in `cross` is used.
`error_prob`	Assumed genotyping error probability

`map_function`	Character string indicating the map function to use to convert genetic distances to recombination fractions.
`lowmem`	If `FALSE`, split individuals into groups with common sex and crossinfo and then precalculate the transition matrices for a chromosome; potentially a lot faster but using more memory.
`quiet`	If `FALSE`, print progress messages.
`cores`	Number of CPU cores to use, for parallel calculations. (If `0`, use `parallel::detectCores()`.) Alternatively, this can be links to a set of cluster sockets, as produced by `parallel::makeCluster()`.
`compress`	Amount of compression to use (value in the range 0-100; lower values mean larger file sizes)
`overwrite`	If FALSE (the default), refuse to overwrite any files that already exist.

Details

This is like calling `qtl2::calc_genoprob()` and then `fst_genoprob()`, but in a way that hopefully saves memory by doing it one chromosome at a time.

Value

A list containing the attributes of `genoprob` and the address for the created fst database. Components are:

- `dim` - List of all dimensions of 3-D arrays.
- `dimnames` - List of all dimension names of 3-D arrays.
- `is_x_chr` - Vector of all is_x_chr attributes.
- `chr` - Vector of (subset of) chromosome names for this object.
- `ind` - Vector of (subset of) individual names for this object.
- `mar` - Vector of (subset of) marker names for this object.
- `fst` - Path and base of file names for the fst database.

See Also

`qtl2::calc_genoprob()`, `fst_genoprob()`

Examples

```
library(qtl2)
grav2 <- read_cross2(system.file("extdata", "grav2.zip", package="qtl2"))
gmap_w_pmar <- insert_pseudomarkers(grav2$gmap, step=1)
fst_dir <- file.path(tempdir(), "grav2_genoprob")
dir.create(fst_dir)
probs_fst <- calc_genoprob_fst(grav2, "grav2", fst_dir, gmap_w_pmar, error_prob=0.002)

# clean up: remove all the files we created
unlink(fst_files(probs_fst))
```

`cbind.fst_genoprob` *Join genotype probabilities for different chromosomes*

Description

Join multiple genotype probability objects, as produced by `fst_genoprob()` for different individuals.

Usage

```
## S3 method for class 'fst_genoprob'
cbind(..., fbase = NULL, fdir = NULL, overwrite = FALSE, quiet = FALSE)
```

Arguments

`...`	Genotype probability objects as produced by `fst_genoprob()`. Must have the same set of individuals.
`fbase`	Base of fileame for fst database. Needed if objects have different fst databases.
`fdir`	Directory for fst database.
`overwrite`	If FALSE (the default), refuse to overwrite existing `.fst` files.
`quiet`	If TRUE, don't show any messages. Passed to `fst_genoprob()`.

Value

A single genotype probability object.

See Also

`rbind.fst_genoprob()`

Examples

```
library(qtl2)
grav2 <- read_cross2(system.file("extdata", "grav2.zip", package="qtl2"))
map <- insert_pseudomarkers(grav2$gmap, step=1)
probsA <- calc_genoprob(grav2[1:5,1:2], map, error_prob=0.002)
probsB <- calc_genoprob(grav2[1:5,3:4], map, error_prob=0.002)
dir <- tempdir()
fprobsA <- fst_genoprob(probsA, "exampleAc", dir, overwrite=TRUE)
fprobsB <- fst_genoprob(probsB, "exampleBc", dir, overwrite=TRUE)

# use cbind to combine probabilities for same individuals but different chromosomes
fprobs <- cbind(fprobsA, fprobsB, fbase = "exampleABc", overwrite=TRUE)

# clean up: remove all the files we created
unlink(fst_files(fprobsA))
unlink(fst_files(fprobsB))
unlink(fst_files(fprobs))
```

`fst_extract` *Extract genotype probabilities from fst database*

Description

Extract genotype probabilities from fst database as an ordinary calc_genoprob object.

Usage

```
fst_extract(object)

fst2calc_genoprob(object)
```

Arguments

`object`	Object of class `"fst_genoprob"`, linking to an fst database of genotype probabilities.

Details

The genotype probabilities are extracted from the fst database. Each chromosome is extracted in turn.

Value

An object of class `"calc_genoprob"` (a list of 3-dimensional arrays).

Functions

- `fst2calc_genoprob`: Deprecated version (to be deleted)

See Also

`fst_genoprob()`

Examples

```
library(qtl2)
grav2 <- read_cross2(system.file("extdata", "grav2.zip", package="qtl2"))
map <- insert_pseudomarkers(grav2$gmap, step=1)
probs <- calc_genoprob(grav2, map, error_prob=0.002)
dir <- tempdir()
fprobs <- fst_genoprob(probs, "grav2", dir, overwrite=TRUE)
nprobs <- fst_extract(fprobs)

# clean up: remove all the files we created
unlink(fst_files(fprobs))
```

`fst_files` *List files used in fst_genoprob object*

Description

List all of the files used in an fst_genoprob object.

Usage

```
fst_files(object)
```

Arguments

`object`	An object of class `"fst_genoprob"` as created by `fst_genoprob()`.

Value

Vector of character strings with the full paths for all of the files used for the input `object`.

See Also

`fst_path()`

Examples

```
library(qtl2)
grav2 <- read_cross2(system.file("extdata", "grav2.zip", package="qtl2"))
probs <- calc_genoprob(grav2, error_prob=0.002)
dir <- tempdir()
fprobs <- fst_genoprob(probs, "grav2", dir, overwrite=TRUE)

fst_path(fprobs)
fst_files(fprobs)

# clean up: remove all the files we created
unlink(fst_files(fprobs))
```

`fst_genoprob` *Store genotype probabilities in fst database*

Description

Save an R/qtl2 genotype probabilities object to a set of fst files for fast access with reduced memory usage.

Usage

```
fst_genoprob(
  genoprob,
  fbase,
  fdir = ".",
  compress = 0,
  verbose = TRUE,
  overwrite = FALSE,
  quiet = !verbose
)
```

Arguments

`genoprob`	Object of class `"calc_genoprob"`. For details, see the R/qtl2 developer guide and `qtl2::calc_genoprob()`.
`fbase`	Base of filename for fst database.
`fdir`	Directory for fst database.
`compress`	Amount of compression to use (value in the range 0-100; lower values mean larger file sizes)
`verbose`	Opposite of `quiet`; deprecated argument (to be removed).
`overwrite`	If FALSE (the default), refuse to overwrite any files that already exist.
`quiet`	If FALSE (the default), show messages about fst database creation.

Details

The genotype probabilities are stored in separate databases for each chromosome as tables of (indivduals*genotypes) x (positions) in directory `fst`. The `dim`, `dimnames` and `is_x_chr` elements of the object have information about the entire fst database. If a `fst_genoprob` object is a subset of another such object, the `chr`, `ind`, and `mar` contain information about what is in the subset. However, the `fst` databases are not altered in a subset, and can be restored by `fst_restore()`. The actual elements of an `"fst_genoprob"` object are only accessible to the user after a call to `unclass()`; instead the usual access to elements of the object invoke `subset.fst_genoprob()`.

Value

A list containing the attributes of `genoprob` and the address for the created fst database. Components are:

- `dim` - List of all dimensions of 3-D arrays.
- `dimnames` - List of all dimension names of 3-D arrays.
- `is_x_chr` - Vector of all is_x_chr attributes.
- `chr` - Vector of (subset of) chromosome names for this object.
- `ind` - Vector of (subset of) individual names for this object.
- `mar` - Vector of (subset of) marker names for this object.
- `fst` - Path and base of file names for the fst database.

Functions

- `fst_genoprob()`: Deprecated version (to be deleted)

See Also

`fst_path()`, `fst_extract()`, `fst_files()`, `replace_path()`, `fst_restore()`

Examples

```
library(qtl2)
grav2 <- read_cross2(system.file("extdata", "grav2.zip", package="qtl2"))
map <- insert_pseudomarkers(grav2$gmap, step=1)
probs <- calc_genoprob(grav2, map, error_prob=0.002)
dir <- tempdir()
fprobs <- fst_genoprob(probs, "grav2", dir, overwrite=TRUE)

# clean up: remove all the files we created
unlink(fst_files(fprobs))
```

`fst_path` *Path used in fst_genoprob object*

Description

Get the path used in an fst_genoprob object.

Usage

```
fst_path(object)
```

Arguments

`object`	An object of class `"fst_genoprob"` as created by `fst_genoprob()`.

Value

Character string with path (and initial file stem) for files used in the input `object`.

See Also

`fst_files()`, `replace_path()`

Examples

```
library(qtl2)
grav2 <- read_cross2(system.file("extdata", "grav2.zip", package="qtl2"))
probs <- calc_genoprob(grav2, error_prob=0.002)
dir <- tempdir()
fprobs <- fst_genoprob(probs, "grav2", dir, overwrite=TRUE)

fst_path(fprobs)
fst_files(fprobs)

# clean up: remove all the files we created
unlink(fst_files(fprobs))
```

`fst_restore` *Restore fst_genoprob object to original dimensions.*

Description

Any `"fst_genoprob"` object has embedded its original data and dimensions. This resets elements `ind`, `chr` and `mar` to the full set.

Usage

```
fst_restore(object)

fst_genoprob_restore(object)
```

Arguments

`object`	Object of class `"fst_genoprob"` as produced by `fst_genoprob()`.

Details

Object is unclassed and elements `ind`, `chr` and `mar` are changed before reseting attributes as `"fst_genoprob"` object. See `fst_genoprob()` for details on the object.

Value

Input `object` with dimensions restored.

Functions

- `fst_genoprob_restore()`: Deprecated version (to be removed).

See Also

`fst_genoprob()`, `fst_extract()`

Examples

```
library(qtl2)
grav2 <- read_cross2(system.file("extdata", "grav2.zip", package="qtl2"))
map <- insert_pseudomarkers(grav2$gmap, step=1)
probs <- calc_genoprob(grav2, map, error_prob=0.002)
dir <- tempdir()
fprobs <- fst_genoprob(probs, "grav2", dir, overwrite=TRUE)

# subset probabilities
fprobs2 <- subset(fprobs, chr=1:2)

# use object to get the full probabilities back
fprobs5 <- fst_restore(fprobs2)

# clean up: remove all the files we created
unlink(fst_files(fprobs))
```

---

genoprob_to_alleleprob_fst

Convert genotype probabilities to allele probabilities and write to fst database

---

Description

Reduce genotype probabilities (as calculated by `qtl2::calc_genoprob()`) to allele probabilities, writing them to an fst database.

Usage

```
genoprob_to_alleleprob_fst(
  probs,
  fbase,
  fdir = ".",
  quiet = TRUE,
  cores = 1,
  compress = 0,
  overwrite = FALSE
)
```

Arguments

`probs`	Genotype probabilities, as calculated from `qtl2::calc_genoprob()`.
`fbase`	Base of filename for fst database.
`fdir`	Directory for fst database.
`quiet`	IF `FALSE`, print progress messages.

`cores`	Number of CPU cores to use, for parallel calculations. (If `0`, use `parallel::detectCores()`.) Alternatively, this can be links to a set of cluster sockets, as produced by `parallel::makeCluster()`.
`compress`	Amount of compression to use (value in the range 0-100; lower values mean larger file sizes)
`overwrite`	If FALSE (the default), refuse to overwrite any files that already exist.

Details

This is like calling `qtl2::genoprob_to_alleleprob()` and then `fst_genoprob()`, but in a way that hopefully saves memory by doing it one chromosome at a time.

Value

Link to fst database for the `probs` input with probabilities collapsed to alleles rather than genotypes.

See Also

`qtl2::genoprob_to_alleleprob()`, `fst_genoprob()`

Examples

```
library(qtl2)
iron <- read_cross2(system.file("extdata", "iron.zip", package="qtl2"))
gmap_w_pmar <- insert_pseudomarkers(iron$gmap, step=1)

# genotype probabilities
fst_dir <- file.path(tempdir(), "iron_genoprob")
dir.create(fst_dir)
probs_fst <- calc_genoprob_fst(iron, "iron", fst_dir, gmap_w_pmar, error_prob=0.002)

# allele probabilities
fst_dir_apr <- file.path(tempdir(), "iron_alleleprob")
dir.create(fst_dir_apr)
aprobs_fst <- genoprob_to_alleleprob_fst(probs_fst, "iron", fst_dir_apr)

# clean up: remove all the files we created
unlink(fst_files(probs_fst))
unlink(fst_files(aprobs_fst))
```

`rbind.fst_genoprob` *Join genotype probabilities for different individuals*

Description

Join multiple genotype probability objects, as produced by `fst_genoprob()` for different individuals.

Usage

```
## S3 method for class 'fst_genoprob'
rbind(..., fbase = NULL, fdir = NULL, overwrite = FALSE, quiet = FALSE)
```

Arguments

`...`	Genotype probability objects as produced by `fst_genoprob()`. Must have the same set of markers and genotypes.
`fbase`	Base of fileame for fst database. Needed if objects have different fst databases.
`fdir`	Directory for fst database.
`overwrite`	If FALSE (the default), refuse to overwrite existing `.fst` files
`quiet`	If TRUE, don't show any messages. Passed to `fst_genoprob()`.

Value

A single genotype probability object.

See Also

`cbind.fst_genoprob()`

Examples

```
library(qtl2)
grav2 <- read_cross2(system.file("extdata", "grav2.zip", package="qtl2"))
map <- insert_pseudomarkers(grav2$gmap, step=1)
probsA <- calc_genoprob(grav2[1:5,], map, error_prob=0.002)
probsB <- calc_genoprob(grav2[6:12,], map, error_prob=0.002)
dir <- tempdir()
fprobsA <- fst_genoprob(probsA, "exampleAr", dir, overwrite=TRUE)
fprobsB <- fst_genoprob(probsB, "exampleBr", dir, overwrite=TRUE)

# use rbind to combine probabilities for same chromosomes but different individuals
fprobs <- rbind(fprobsA, fprobsB, fbase = "exampleABr")

# clean up: remove all the files we created
unlink(fst_files(fprobsA))
unlink(fst_files(fprobsB))
unlink(fst_files(fprobs))
```

`replace_path` *Replace the path used in fst_genoprob object*

Description

Replace the path used in an fst_genoprob object.

Usage

```
replace_path(object, path)
```

Arguments

`object`	An object of class `"fst_genoprob"` as created by `fst_genoprob()`.
`path`	New path (directory + file stem as a single character string) to be used in the object.

Value

The input `object` with the path replaced. If any of the expected files don't exist with the new path, warnings are issued.

See Also

`fst_path()`, `fst_files()`

Examples

```
library(qtl2)
grav2 <- read_cross2(system.file("extdata", "grav2.zip", package="qtl2"))
probs <- calc_genoprob(grav2, error_prob=0.002)
dir <- tempdir()
fprobs <- fst_genoprob(probs, "grav2", dir, overwrite=TRUE)

# move the probabilities into a different directory
new_dir <- file.path(tempdir(), "subdir")
if(!dir.exists(new_dir)) dir.create(new_dir)
for(file in fst_files(fprobs)) {
   file.rename(file, file.path(new_dir, basename(file)))
}

# revise the path in fprobs
new_path <- sub(dir, new_dir, fst_path(fprobs), fixed=TRUE)
fprobs <- replace_path(fprobs, new_path)
```

`subset_fst_genoprob` *Subsetting genotype probabilities*

Description

Pull out a specified set of individuals and/or chromosomes from the results of `fst_genoprob()`.

Usage

```
subset_fst_genoprob(x, ind = NULL, chr = NULL, mar = NULL, ...)

## S3 method for class 'fst_genoprob'
subset(x, ind = NULL, chr = NULL, mar = NULL, ...)
```

Arguments

`x`	Genotype probabilities as output from `fst_genoprob()`.
`ind`	A vector of individuals: numeric indices, logical values, or character string IDs
`chr`	A vector of chromosomes: logical values, or character string IDs. Numbers are interpreted as character string IDs.
`mar`	A vector of marker names as character string IDs.
`...`	Ignored.

Value

The input genotype probabilities, with the selected individuals and/or chromsomes.

Examples

```
library(qtl2)
grav2 <- read_cross2(system.file("extdata", "grav2.zip", package="qtl2"))

pr <- calc_genoprob(grav2)
dir <- tempdir()
fpr <- fst_genoprob(pr, "grav2", dir)

# keep just individuals 1:5, chromosome 2
prsub <- fpr[1:5,2]
# keep just chromosome 2
prsub2 <- fpr[,2]

# clean up: remove all the files we created
unlink(fst_files(fpr))
```

`summary.fst_genoprob` *Summary of an fst_genoprob object*

Description

Summarize an fst_genoprob object

Usage

```
## S3 method for class 'fst_genoprob'
summary(object, ...)
```

Arguments

`object`	An object of class "`fst_genoprob`", as output by `fst_genoprob()`.
`...`	Ignored.

Examples

```
library(qtl2)
grav2 <- read_cross2(system.file("extdata", "grav2.zip", package="qtl2"))

pr <- calc_genoprob(grav2)
dir <- tempdir()
fpr <- fst_genoprob(pr, "grav2", dir)

# summary of fst_genoprob object
summary(fpr)

# clean up: remove all the files we created
unlink(fst_files(fpr))
```

Index