

Package ‘parati’

August 27, 2026

Type Package

Title Parental Allele Transmission Inference for Trio Genotype Data

Version 1.0.0

Description Infers maternal and paternal transmitted and non-transmitted alleles from phased trio genotype data. The package supports SNP-level analyses of genetic nurture and transgenerational effects. It interoperates with Bioconductor VCF infrastructure through support for VariantAnnotation::VCF objects and returns R objects for downstream analysis.

URL <https://github.com/newche/parati>

BugReports <https://github.com/newche/parati/issues>

License GPL-3 + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Imports data.table, methods, openxlsx, R.utils, vcfR,
VariantAnnotation, SummarizedExperiment, BiocGenerics,
GenomeInfoDb

Suggests BiocStyle, knitr, optparse, rmarkdown, testthat (>= 3.0.0),
waldo

VignetteBuilder knitr

biocViews Genetics, SNP, Sequencing, VariantAnnotation, Software

git_url <https://git.bioconductor.org/packages/parati>

git_branch RELEASE_3_23

git_last_commit 4abcf9

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-08-26

Author Jinyi Che [aut, cre]

Maintainer Jinyi Che <chejinyi777@gmail.com>

Contents

haplotype_infer	2
parati_run	3
read_vcf_by_chr	4
vcf_dt_to_vcfR	4
write_vcf_dt	5
write_vcf_obj	6
Index	7

haplotype_infer	<i>Infer parental transmitted and non-transmitted alleles</i>
-----------------	---

Description

Given trio genotype data for a single family, infer maternal and paternal transmitted and non-transmitted alleles.

Usage

```
haplotype_infer(vcf_dt, hap_length = 5e+05)
```

Arguments

vcf_dt	A 'data.table' containing fixed VCF columns and genotype columns named 'M', 'P', and 'B'.
hap_length	Integer, haplotype window length.

Details

This function preserves the original PARATI inference semantics: - deterministic Mendelian inference for non-triple-heterozygote patterns - local haplotype matching for triple-heterozygote sites - low-similarity / ambiguous sites are left as missing (".l.")

Value

- A named list containing:
- vcf_trans** A 'data.table' with transmitted alleles.
 - vcf_nontrans** A 'data.table' with non-transmitted alleles.
 - sim_perc_summary** A 'data.table' summarizing inference statistics.

parati_run	<i>Run PARATI inference workflow</i>
------------	--------------------------------------

Description

Main function to infer parental transmitted and non-transmitted alleles from phased trio genotype data. The function accepts either a VCF file path or a 'VCF' object from 'VariantAnnotation', and returns R objects without writing files by default.

Usage

```
parati_run(vcf, fam, chr = NULL, hap_length = 5e+05)
```

Arguments

vcf	Either a character path to a phased VCF/VCF.GZ file, or a 'VariantAnnotation::VCF' object.
fam	Either a character path to a family index table ('.xlsx') or a 'data.frame' / 'data.table' with columns 'FamilyIndex', 'IndividualID', and 'Role'.
chr	Optional chromosome identifier to subset variants.
hap_length	Integer, haplotype window length.

Value

A named list containing:

vcf_trans A 'data.table' with transmitted alleles.

vcf_nontrans A 'data.table' with non-transmitted alleles.

sim_perc_summary A 'data.table' summarizing inference statistics.

Examples

```
vcf_file <- system.file("extdata", "Toy_TrioGenotype.vcf.gz", package = "parati")
fam_file <- system.file("extdata", "Toy_FamilyIndexTable.xlsx", package = "parati")
res <- parati_run(vcf = vcf_file, fam = fam_file, chr = 1)
names(res)
```

read_vcf_by_chr	<i>Read VCF by chromosome</i>
-----------------	-------------------------------

Description

Reads a VCF file and subsets variants to a given chromosome.

Usage

```
read_vcf_by_chr(vcf_file, chr)
```

Arguments

vcf_file	Character scalar, path to a VCF/VCF.GZ file.
chr	Character or integer chromosome identifier.

Value

A 'data.table' containing VCF rows for the selected chromosome.

Examples

```
vcf_file <- system.file("extdata", "Toy_TrioGenotype.vcf.gz", package = "parati")
vcf_chr <- read_vcf_by_chr(vcf_file, chr = 1)
dim(vcf_chr)
```

vcf_dt_to_vcfR	<i>Convert data.table to vcfR object</i>
----------------	--

Description

Converts a 'data.table' representing VCF rows into a 'vcfR' object.

Usage

```
vcf_dt_to_vcfR(df, meta = character())
```

Arguments

df	A 'data.table' containing VCF rows.
meta	Character vector of VCF meta lines.

Value

A 'vcfR' object.

Examples

```
vcf_file <- system.file("extdata", "Toy_TrioGenotype.vcf.gz", package = "parati")
vcf_dt <- read_vcf_by_chr(vcf_file, chr = 1)
vcf_obj <- vcf_dt_to_vcfR(vcf_dt)
class(vcf_obj)
stopifnot(inherits(vcf_obj, "vcfR"))
```

write_vcf_dt	<i>Write VCF data.table to file</i>
--------------	-------------------------------------

Description

Writes a 'data.table' representing VCF rows to a VCF file.

Usage

```
write_vcf_dt(df, file)
```

Arguments

df	A 'data.table' containing VCF rows.
file	Character scalar, output file path.

Value

'NULL', invisibly. The VCF file is written to 'file'.

Examples

```
vcf_file <- system.file("extdata", "Toy_TrioGenotype.vcf.gz", package = "parati")
vcf_dt <- read_vcf_by_chr(vcf_file, chr = 1)
outfile <- tempfile(fileext = ".vcf")
write_vcf_dt(vcf_dt, outfile)
file.exists(outfile)
```

write_vcf_obj	<i>Write vcfR object to file</i>
---------------	----------------------------------

Description

Writes a 'vcfR' object to a VCF file.

Usage

```
write_vcf_obj(vcf_obj, file)
```

Arguments

vcf_obj	A 'vcfR' object.
file	Character scalar, output file path.

Value

'NULL', invisibly. The VCF file is written to 'file'.

Examples

```
vcf_file <- system.file("extdata", "Toy_TrioGenotype.vcf.gz", package = "parati")
vcf_dt <- read_vcf_by_chr(vcf_file, chr = 1)
vcf_obj <- vcf_dt_to_vcfR(vcf_dt)
stopifnot(inherits(vcf_obj, "vcfR"))
outfile <- tempfile(fileext = ".vcf")
write_vcf_obj(vcf_obj, outfile)
stopifnot(file.exists(outfile))
```

Index

haplotype_infer, [2](#)

parati_run, [3](#)

read_vcf_by_chr, [4](#)

vcf_dt_to_vcfR, [4](#)

write_vcf_dt, [5](#)

write_vcf_obj, [6](#)