Package ‘BIGr’

June 11, 2024

Title (B)reeding (I)nsight (G)enomics Functions for Polypoid and Diploid Species

Version 0.1.1

Description This package contains the functions developed to analyze polyploid and diploid species within Breeding Insight.

License `use_apache_license()`

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Depends R (>= 4.0.0)

Imports doParallel,
dplyr,
foreach,
Rdpack (>= 0.7),
tidyr (>= 1.3.1)

RdMacros Rdpack

R topics documented:

`calculate_Het` *Calculate Observed Heterozygosity from a Genotype Matrix*

Description

This function calculates the observed heterozygosity from a genotype matrix. It assumes that the samples are the columns, and the genomic markers are in rows. Missing data should be set as NA, which will then be ignored for the calculations. All samples must have the same ploidy.

Usage

```
calculate_Het(geno, ploidy)
```

Arguments

`geno`	Genotype matrix or data.frame
`ploidy`	The ploidy of the species being analyzed

Value

A dataframe of observed heterozygosity values for each sample

---

`calculate_MAF` *Calculate Minor Allele Frequency from a Genotype Matrix*

Description

This function calculates the allele frequency and minor allele frequency from a genotype matrix. It assumes that the Samples are the columns, and the genomic markers are in rows. Missing data should be set as NA, which will then be ignored for the calculations. All samples must have the same ploidy.

Usage

```
calculate_MAF(df, ploidy)
```

Arguments

`df`	Genotype matrix or data.frame
`ploidy`	The ploidy of the species being analyzed

Value

A dataframe of AF and MAF values for each marker

`capture_diversity.Gmat` *Estimate Minimum Number of Individuals to Sample to Capture Population Genomic Diversity (Genotype Matrix)*

Description

This function can be used to estimate the number of individuals to sample from a population in order to capture a desired percentage of the genomic diversity. It assumes that the samples are the columns, and the genomic markers are in rows. Missing data should be set as NA, which will then be ignored for the calculations. All samples must have the same ploidy. This function was adapted from a previously developed Python method (Sandercock et al., 2023) (https://github.com/alex-sandercock/Capturing_genomic_diversity/)

Usage

```
capture_diversity.Gmat(
  df,
  ploidy,
  r2_threshold = 0.9,
  iterations = 10,
  sample_list = NULL,
  parallel = FALSE,
  batch = 1,
  save.result = TRUE
)
```

Arguments

`df` Genotype matrix or data.frame with the count of alternate alleles (0=homozygous reference, 1 = heterozygous, 2 = homozygous alternate)

`ploidy` The ploidy of the species being analyzed

`r2_threshold` The ratio of diversity to capture (default = 0.9)

`iterations` The number of iterations to perform to estimate the average result (default = 10)

`sample_list` The list of samples to subset from the dataset (optional)

`parallel` Run the analysis in parallel (True/False) (default = FALSE)

`save.result` Save the results to a .txt file? (default = TRUE)

Value

A data.frame with minimum number of samples required to match or exceed the input ratio

References

Sandercock, A. M., Westbrook, J. W., Zhang, Q., & Holliday, J. A. (2023). The road to restoration: Identifying and conserving the adaptive legacy of American chestnut. bioRxiv, 2023-05.

`check_ped` *Evaluate Pedigree File for Accuracy*

Description

Check a pedigree file for accuracy and output suspected errors

Usage

```
check_ped(ped.file)
```

Arguments

`ped.file` path to pedigree text file. The pedigree file is a 3-column pedigree tab separated file with columns labeled as id sire dam in any order

Details

check_ped takes a 3-column pedigree tab separated file with columns labeled as id sire dam in any order and checks for:

- Ids that appear more than once in the id column
- Ids that appear in both sire and dam columns
- Direct (e.g. parent is a offspring of his own daughter) and indirect (e.g. a great grandparent is son of its granchild) dependencies within the pedigree.
- Individuals included in the pedigree as sire or dam but not on the id column and reports them back with unknown parents (0).

When using check_ped, do a first run to check for repeated ids and parents that appear as sire and dam. Once these errors are cleaned run the function again to check for dependencies as this will provide the most accurate report.

Note: This function does not change the input file but prints any errors found in the console.

Value

A list of dataframes of error types, and the output printed to the console

Examples

```
##Get list with a dataframe for each error type
ped_errors <- check_ped(ped.file = "example_ped.txt")

##Access the "messy parents" dataframe result
ped_errors$messy_parents

##Get list of sample IDs with messy parents error
messy_parent_ids <- ped_errors$messy_parents$id
print(messy_parent_ids)
```

`dosage_ratios` *Calculate the Percentage of Each Dosage Value*

Description

This function calculates the percentage of each dosage value within a genotype matrix. It assumes that the samples are the columns, and the genomic markers are in rows. Missing data should be set as NA, which will then be ignored for the calculations. All samples must have the same ploidy.

Usage

```
dosage_ratios(matrix_data, ploidy)
```

Arguments

`ploidy`	The ploidy of the species being analyzed
`df`	Genotype matrix or data.frame

Value

A data.frame with percentages of dosage values in the genotype matrix

flip_dosage *Switch Dosage Values from a Genotype Matrix*

Description

This function converts the dosage count values to the opposite value. This is primarily used when converting dosage values from reference based (0 = homozygous reference) to alternate count based (0 = homozygous alternate). It assumes that the Samples are the columns, and the genomic markers are in rows. Missing data should be set as NA, which will then be ignored for the calculations. All samples must have the same ploidy.

Usage

```
flip_dosage(df, ploidy, is_reference = TRUE)
```

Arguments

`df`	Genotype matrix or data.frame
`ploidy`	The ploidy of the species being analyzed

Value

A genotype matrix

get_countsMADC *Obtain Read Counts from MADC File*

Description

This function takes the MADC file as input and retrieves the ref and alt counts for each sample, and converts them to ref, alt, and size(total count) matrices for dosage calling tools. At the moment, only the read counts for the Ref and Alt target loci are obtained while the additional loci are ignored.

Usage

```
get_countsMADC(madc_file)
```

Arguments

`madc_file`	Path to MADC file

Value

A list of read count matrices for reference, alternate, and total read count values

`updog2vcf` *Export Updog Results as VCF*

Description

This function will convert an Updog output to a VCF file

Usage

```
updog2vcf(multidog.object, ploidy, output.file)
```

Arguments

`multidog.object`	updog output object with class "multidog" from dosage calling
`ploidy`	The ploidy of the species being analyzed
`output.file`	output file name and path

Details

When performing dosage calling for multiple SNPs using Updog, the output file contains information for all loci and all samples. This function will convert the updog output file to a VCF file, while retaining the information for the values that are commonly used to filter low quality and low confident dosage calls.

Value

A vcf file

References

Updog R package

Index