Package ‘FnR’

April 26, 2024

Title Inbreeding and Numerator Relationship Coefficients

Version 1.0.0

Description Compute inbreeding coefficients using the method of Meuwissen and Luo (1992) <doi:10.1186/1297-9686-24-4-305>, and numerator relationship coefficients between individuals using the method of Van Vleck (2007) <https://pubmed.ncbi.nlm.nih.gov/18050089/>.

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

URL https://github.com/nilforooshan/FnR

BugReports https://github.com/nilforooshan/FnR/issues

NeedsCompilation no

Author Mohammad Ali Nilforooshan [aut, cre]
(<https://orcid.org/0000-0003-0339-5442>)

Maintainer Mohammad Ali Nilforooshan <m.a.nilforooshan@gmail.com>

R topics documented:

`calcR` *Compute numerator relationship coefficients between two distinct groups of individuals*

Description

Compute numerator relationship coefficients between two distinct groups of individuals

Usage

```
calcR(ped, set1, set2, type = "notdam-notsire")
```

Arguments

ped	: A data frame with integer columns corresponding to ID, SIRE, and DAM. IDs should be sequential, starting from 1. Missing parents (SIRE and DAM) are denoted as 0.
set1	: A set of individual IDs.
set2	: A set of individual IDs, distinct from set1.
type	: "notdam-notsire" (default), "sire-sire", or "dam-dam" relationships "notdam-notsire" requires set1 and set2 individuals not to be members of ped$DAM and ped$SIRE, respectively. "sire-sire" requires set1 and set2 individuals to be members of ped$SIRE. "dam-dam" requires set1 and set2 individuals to be members of ped$DAM.

Value

: Numerator relationship coefficients between set1 and set2 individuals in the form of a matrix (a partition of the numerator relationship matrix **A**).

Examples

```
# A sample pedigree data frame:
ped <- data.frame(
    ID = 1:12,
    SIRE = c(0, 0, 0, 2, 2, 0, 4, 6, 0, 6, 10, 10),
    DAM = c(0, 0, 0, 1, 1, 0, 3, 5, 7, 8, 9, 0)
)

# Example 1: Calculate relationship coefficients between two groups of animals,
# one's members not among dams, and the members of the other not among sires.
calcR(ped, set1 = c(12, 6), set2 = c(11, 8), type = "notdam-notsire")
# Since `"notdam-notsire"` is the default type, `type = "notdam-notsire"` might be omitted.

# Example 2: Calculate relationship coefficients between dam 7 and dams 8 and 9.
calcR(ped, set1 = 7, set2 = 8:9, type = "dam-dam")

# Example 3: Calculate relationship coefficients between sires 2 & 6 and sires 4 & 10.
calcR(ped, set1 = c(2, 6), set2 = c(4, 10), type = "sire-sire")
```

resume_inbreed *Calculate inbreeding coefficients from scratch or resume for new individuals in the pedigree*

Description

Calculate inbreeding coefficients from scratch or resume for new individuals in the pedigree

Usage

```
resume_inbreed(ped, f = c(), d = c(), export_d = FALSE)
```

Arguments

`ped`	: A data frame with integer columns corresponding to ID, SIRE, and DAM. IDs should be sequential, starting from 1. Missing parents (SIRE and DAM) are denoted as 0.
`f`	: (Optional) If available, the vector of inbreeding coefficients from the previous calculation of inbreeding coefficients with less number of animals in the pedigree.
`d`	: (Optional) If available, the vector of the diagonal elements of the diagonal matrix $\mathbf{D}$ in $\mathbf{A} = \mathbf{TDT}'$ from the previous calculation of inbreeding coefficients with less number of animals in the pedigree, where $\mathbf{A}$ is the numerator relationship matrix.
`export_d`	: `FALSE` (default) or `TRUE`. If `TRUE`, vector `d` is retuned for future use.

Value

: Vector of inbreeding coefficients if `export_d == FALSE`, or a list containing the vector of inbreeding coefficients and the vector of `d` coefficients if `export_d == TRUE`.

Examples

```
# A sample pedigree data frame:
ped <- data.frame(
    ID = 1:12,
    SIRE = c(0, 0, 0, 2, 2, 0, 4, 6, 0, 6, 10, 10),
    DAM = c(0, 0, 0, 1, 1, 0, 3, 5, 7, 8, 9, 0)
)

oldped <- ped[1:9, ]
(oldrun <- resume_inbreed(oldped, export_d = TRUE))
resume_inbreed(ped)
resume_inbreed(ped, f = oldrun$f)
resume_inbreed(ped, f = oldrun$f, d = oldrun$d)
```

Index