Package ‘hypergeo2’

September 18, 2024

Title Generalized Hypergeometric Function with Tunable High Precision

Version 0.1.0

Description Computation of generalized hypergeometric function with tunable high precision in a vectorized manner, with the floating-point datatypes from 'mpfr' or 'gmp' library. The computation is limited to real numbers.

License MIT + file LICENSE

Suggests ggplot2, hypergeo, microbenchmark, testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.2

LinkingTo BH, Rcpp

Imports Rcpp

URL https://github.com/zhuxr11/hypergeo2

BugReports https://github.com/zhuxr11/hypergeo2/issues

SystemRequirements gmp, mpfr

NeedsCompilation yes

Author Xiurui Zhu [aut, cre]

Maintainer Xiurui Zhu <zxr6@163.com>

Repository CRAN

Date/Publication 2024-09-18 11:50:02 UTC

Contents

genhypergeo *Generalized hypergeometric function*

Description

genhypergeo computes generalized hypergeometric function with vectorized input. This function is available in Rcpp as hypergeo2::genhypergeo_vec(). Its non-vectorized version is available in Rcpp as hypergeo2::genhypergeo_vec(). To use them, please use [[Rcpp::depends(hypergeo2)]] and #include "hypergeo2.h".

Usage

```
genhypergeo(
  U,
  L,
  z,
  prec = NULL,
  check_mode = TRUE,
  log = FALSE,
  backend = c("mpfr", "gmp")
)
```

Arguments

U, L List of numeric vectors for upper and lower values.

z Numeric vector as common ratios.

prec List of NULL or (unsigned) integers as precision level during computation, a.k.a the number of precise digits of floating-point datatypes. This argument is vectorized: you may use different precision settings for different input elements. If NULL, double precision (default) is used.

check_mode Logical vector indicating whether the mode of x should be checked for obvious convergence failures. This argument is vectorized: you may use different check modes for different input elements.

log Logical (1L) indicating whether result is given as log(result). This argument is **NOT** vectorized: only its first element is used.

backend One of the following: 'mpfr' (default) or 'gmp', for the realization of floating-point datatype of tunable precision. This argument is **NOT** vectorized: you may only input character (1L).

Details

Sometimes, computing generalized hypergeometric function in double precision is not sufficient, even though we only need 6-8 accurate digits in the results (see example). Here, two floating-point datatypes are provided: mpfr_float ('mpfr') and gmp_float ('gmp'). By comparison, the 'mpfr' backend is safer, since it defines Inf while the 'gmp' backend throws overflow exception (see references). But the 'gmp' backend results in more accurate results at the same precision, since it usually uses higher precision than set (see reference and validate it on yourself with the examples).

Value

Numeric vector as the results of computation (at double precision). Warnings are issued if failing to converge.

Note

Change log:

- 0.1.0 Xiurui Zhu - Initiate the function.

Author(s)

Xiurui Zhu

References

For the floating-point datatypes of tunable precision:

- Documentation about mpfr_float
- Documentation about gmp_float
- Documentation about higher precision of gmp_float datatype

Examples

```
U <- c(-28.2, 11.8, 15.8)
L <- c(12.8, 17.8)
z <- 1
# hypergeo results
if (length(find.package("hypergeo", quiet = TRUE)) > 0L) {
  hypergeo::genhypergeo(U = U, L = L, z = z)
}
# Default (double) precision: this may result in cancellation error on some platforms
tryCatch(
  genhypergeo(U = U, L = L, z = z),
  error = function(err) {
    if (grepl("Cancellation is so severe that no bits in the result are correct",
              conditionMessage(err)) == TRUE) {
      message("! Cancellation error on your platform: ",
              "you may need a higher [prec] than double ([prec = NULL]): ",
              conditionMessage(err))
    } else {
      stop(err)
    }
  }
)
# Precision of 20 digits, default ('mpfr') backend
genhypergeo(U = U, L = L, z = z, prec = 20L)
# Precision of 20 digits, 'gmp' backend
genhypergeo(U = U, L = L, z = z, prec = 20L, backend = "gmp")
# Precision of 25 digits, default ('mpfr') backend
genhypergeo(U = U, L = L, z = z, prec = 25L)
```

```
# Precision of 25 digits, 'gmp' backend
genhypergeo(U = U, L = L, z = z, prec = 25L, backend = "gmp")
```

Index