

Package: GFisher (via r-universe)

October 10, 2026

Type Package

Title Generalized Fisher's Combination Tests Under Dependence

Version 0.3.1

Date 2026-09-02

Description This R package provides accurate and computationally efficient methods for computing p-values for a general family of Fisher-type statistics (GFisher), including Fisher's combination, Good's statistic, and Lancaster's statistic. It supports flexible weighting schemes and an omnibus procedure that adapts the weights and degrees of freedom to the data. The package includes p-value calculation methods based on moment-ratio matching and joint distribution approximations. It offers both pure R implementations and optimized C++ code via RcppArmadillo for improved performance on large-scale data. For fast calculations under independence, the optional 'coga' package is required for functions with the '_ind' suffix.

URL <https://zwulab.github.io/GFisher/>,
<https://github.com/ZWuLab/GFisher>,
<https://doi.org/10.1111/biom.13634>

BugReports <https://github.com/ZWuLab/GFisher/issues>

License GPL-3

Encoding UTF-8

LazyData true

Depends R (>= 4.3.0)

Imports Rcpp, stats, mvtnorm, Matrix

LinkingTo Rcpp, RcppArmadillo

Suggests testthat (>= 3.0.0), knitr, rmarkdown, microbenchmark, coga

VignetteBuilder knitr

RoxygenNote 7.3.1

Repository <https://zwulab.r-universe.dev>

Date/Publication 2026-09-02 21:30:17 UTC
RemoteUrl https://github.com/zwulab/gfisher
RemoteRef HEAD
RemoteSha 1d88b2002d847690b0a0211c543f6c8cadd0df4f

Contents

check_coga	2
clear_GFisher_cache	3
get_GFisher_cache_size	4
get_GFisher_cache_stats	4
getGFishercoef_cached_cpp	5
p.GFisher	5
p.GFisher_ind	8
p.GFisher_ind_wl	10
pval.oGFisher	11
pval.oGFisher_ind	13
stat.GFisher	15
stat.oGFisher	17
stat.oGFisher_ind	19
Index	22

check_coga	<i>Check and Install coga Package for Independent Functions</i>
------------	---

Description

Helper function to check if the coga package is installed and optionally install it. The coga package is required for functions with `_ind` suffix (e.g., `p.GFisher_ind`, `stat.oGFisher_ind`).

Usage

```
check_coga(install = FALSE)
```

Arguments

install	Logical. If TRUE and coga is not installed, attempt to install it from CRAN. Default is FALSE.
---------	--

Details

The GFisher package includes optimized functions for calculating p-values when input p-values are independent. These functions (with `_ind` suffix) provide faster and more accurate calculations by leveraging the `coga` package for weighted sums of chi-square distributions.

Functions requiring `coga`:

- `p.GFisher_ind` - Calculate p-value under independence
- `p.GFisher_ind_w1` - Calculate p-value with equal weights (does not require `coga`)
- `stat.oGFisher_ind` - Calculate oGFisher statistics under independence
- `pval.oGFisher_ind` - Calculate oGFisher p-value under independence

If `coga` is not available, you can still use the general functions (e.g., `p.GFisher`) with `M = diag(n)` to indicate independence.

Value

Logical (invisibly). TRUE if `coga` is installed (or successfully installed), FALSE otherwise.

See Also

`p.GFisher_ind`, `stat.oGFisher_ind`

Examples

```
# Check if coga is available
check_coga()

# Install coga if not available
## Not run:
check_coga(install = TRUE)

## End(Not run)
```

<code>clear_GFisher_cache</code>	<i>Clear GFisher Coefficient Cache</i>
----------------------------------	--

Description

Clears the internal cache used by `getGFishercoef_cached_cpp`. This can be useful to free memory or force recomputation.

Usage

```
clear_GFisher_cache()
```

Value

The number of entries that were cleared from the cache.

`get_GFisher_cache_size`*Get GFisher Coefficient Cache Size*

Description

Returns the current number of entries in the coefficient cache.

Usage

```
get_GFisher_cache_size()
```

Value

The number of cached coefficient values.

`get_GFisher_cache_stats`*Get GFisher Cache Statistics*

Description

Returns detailed statistics about the coefficient cache.

Usage

```
get_GFisher_cache_stats()
```

Value

A list containing cache statistics:

- `size`: Current number of entries
- `max_size`: Maximum cache size before automatic clearing
- `memory_kb`: Approximate memory usage in KB
- `fill_ratio`: Percentage of max capacity used

getGFishercoef_cached_cpp

Cached Compute Hermite Polynomial Coefficients for GFisher

Description

Fast C++ implementation with caching for repeated df values. This can significantly speed up analyses where the same df values are used repeatedly (e.g., df=1,2,3 across thousands of genes).

Usage

```
getGFishercoef_cached_cpp(D, M, p_type)
```

Arguments

D	Vector of degrees of freedom for a GFisher statistic.
M	Correlation matrix of the input Z-scores.
p_type	String: "two" for two-sided, "one" for one-sided input p-values.

Details

This function provides cached computation of coefficients using the same integration routines as getGFishercoef_cpp but with an efficient caching layer at the individual coefficient level. When the same df value is requested multiple times, the cached result is returned immediately without recomputing expensive integrals. The cache has a size limit to prevent unbounded memory growth.

Cache efficiency: For genome-wide analyses where df values are reused across many tests, this caching can provide significant speedups by avoiding redundant numerical integration.

Value

A list of 4 vectors of integrals:

- For two-sided: list(coeff2, coeff4, coeff6, coeff8)
- For one-sided: list(coeff1, coeff2, coeff3, coeff4)

p.GFisher

GFisher Test P-Value Calculation

Description

Calculate the p-value of a GFisher test statistic under dependence, using moment matching methods to account for correlation structure among input statistics.

Usage

```
p.GFisher(
  q,
  df,
  w,
  M,
  p.type = "two",
  method = "HYB",
  nsim = NULL,
  seed = NULL,
  use.cpp = TRUE
)
```

Arguments

q	Numeric value of the observed GFisher statistic (typically from <code>stat.GFisher</code>).
df	Vector of degrees of freedom for inverse chi-square transformation for each p-value. If all dfs are equal, it can be defined by a single constant.
w	Vector of non-negative weights. If a single value, equal weights are used. Note: Weights are automatically normalized to sum to 1.
M	Correlation matrix of the Z-scores from which the input p-values were obtained. Must be a square matrix with dimension equal to the length of df.
p.type	Character string specifying p-value type: • "two": Two-sided input p-values (default) • "one": One-sided input p-values
method	Character string specifying the calculation method: • "MR": Simulation-assisted moment ratio matching (recommended for tail probabilities) • "HYB": Moment ratio matching by quadratic approximation (default, faster but only for two-sided p-values) • "GB": Brown's method with calculated variance
nsim	Number of simulations used in the "MR" method. Default is 5e4. Larger values provide more accurate estimates but take longer to compute.
seed	Optional seed for random number generation in simulation-based methods. Default is NULL (no seed set).
use.cpp	Logical indicating whether to use C++ implementation for method="MR". Default is TRUE. Set to FALSE to use pure R implementation.

Details

Compute the P-Value of a GFisher Test

This function calculates the p-value for a GFisher test statistic under dependence by approximating the null distribution using moment matching.

Method Selection:

- **HYB (Hybrid)**: Uses quadratic approximation based on the first four moments. Faster but only works for two-sided p-values. Best for moderate test statistics.
- **MR (Moment Ratio)**: Uses simulation to estimate moments, then applies moment matching. More accurate for extreme tail probabilities and works for both one-sided and two-sided p-values.
- **GB (Generalized Brown)**: Uses Brown's method with calculated variance. Simplest approximation but may be less accurate.

The function automatically handles near-singular correlation matrices by finding the nearest positive definite matrix when necessary.

Weights are automatically normalized to sum to 1.

Value

A numeric value representing the p-value of the GFisher test.

Author(s)

Hong Zhang, Zheyang Wu

References

Zhang, H., & Wu, Z. (2023). The generalized Fisher's combination and accurate p-value calculation under dependence. *Biometrics*, 79(2), 1159-1172.

See Also

[stat.GFisher](#), [stat.oGFisher](#), [pval.oGFisher](#)

Examples

```
# Example 1: Simple case with equal correlation
set.seed(123)
n <- 10
M <- matrix(0.3, n, n) + diag(0.7, n, n) # Correlation matrix
zscore <- matrix(rnorm(n), nrow = 1) %*% chol(M)
pval <- 2 * (1 - pnorm(abs(zscore))) # Two-sided p-values

# Calculate statistic and p-value with df=2
gf1 <- stat.GFisher(pval, df = 2, w = 1)
p.GFisher(gf1, df = 2, w = 1, M = M, method = "HYB")
p.GFisher(gf1, df = 2, w = 1, M = M, method = "MR", nsim = 5e4)

# Example 2: Varying df and weights
gf2 <- stat.GFisher(pval, df = 1:n, w = 1:n)
p.GFisher(gf2, df = 1:n, w = 1:n, M = M, method = "HYB")
p.GFisher(gf2, df = 1:n, w = 1:n, M = M, method = "MR", nsim = 5e4)
```

p.GFisher_ind *GFisher P-Value for Independent Inputs*

Description

Calculate the p-value of a GFisher test statistic when input p-values are independent. This is substantially faster than using `p.GFisher` with `M` as the identity matrix.

Usage

```
p.GFisher_ind(q, df, w, isExact = TRUE)
```

Arguments

<code>q</code>	Numeric value of the observed GFisher statistic.
<code>df</code>	Vector of degrees of freedom for each p-value transformation. Cannot be single value if the number of independent chi-square variables is greater than 1.
<code>w</code>	Vector of non-negative weights. If single value, it is expanded to a vector of equal weights. Note: Weights are automatically normalized to sum to 1, consistent with all functions in this package.
<code>isExact</code>	Logical. If TRUE (default), uses exact calculation via the <code>coga</code> package. If FALSE, uses an approximation which may be less accurate but faster.

Details

Calculate P-Value of GFisher Under Independence

Under independence, the GFisher statistic follows a linear combination of independent chi-square distributions:

$$S = \sum_{i=1}^n w_i \chi_{d_i}^2$$

This function computes $P(S > q)$ using:

- **Exact method** (`isExact = TRUE`): Uses the `coga` package to compute the distribution of a weighted sum of gamma random variables (chi-square is a special case of gamma). This method is accurate even for very small p-values.
- **Approximation** (`isExact = FALSE`): Uses a faster approximation from the `coga` package, which may be less accurate for extreme tail probabilities.

Weights are automatically scaled so that $\sum w_i = 1$.

Note on dependencies:

This function requires the `coga` package, which is listed in `Suggests`. If `coga` is not installed, the function will fail with an error message.

Value

A numeric p-value.

Author(s)

Hong Zhang, Zheyang Wu

References

Zhang, H., & Wu, Z. (2023). The generalized Fisher's combination and accurate p-value calculation under dependence. *Biometrics*, 79(2), 1159-1172.

Witkovsky, V. (2016). Numerical inversion of a characteristic function: An alternative tool to form the probability distribution of output quantity in linear measurement models. *Acta IMEKO*, 5(3), 32-44. (Describes the characteristic function inversion methodology underlying the *coga* package)

See Also

[p.GFisher](#), [p.GFisher_ind_w1](#), [stat.oGFisher_ind](#)

Examples

```
## Not run:
# Requires coga package
if (requireNamespace("coga", quietly = TRUE)) {
  set.seed(123)
  n <- 10
  df <- runif(n, 0.5, 5)
  w <- abs(rnorm(n))
  q <- 40

  # Exact calculation
  p_exact <- p.GFisher_ind(q = q, df = df, w = w, isExact = TRUE)
  print(p_exact)

  # Approximation (faster but less accurate)
  p_approx <- p.GFisher_ind(q = q, df = df, w = w, isExact = FALSE)
  print(p_approx)

  # Compare with general method (should be similar but slower)
  p_general <- p.GFisher(q = q, df = df, w = w, M = diag(n),
                        p.type = "two", method = "HYB")
  print(p_general)
}

## End(Not run)
```

`p.GFisher_ind_w1`

GFisher P-Value for Independent Inputs with Equal Weights

Description

Fast calculation of GFisher p-value when input p-values are independent and have equal weights (or weight = 1). Uses the fact that the sum of independent chi-square random variables is also chi-square distributed.

Usage

```
p.GFisher_ind_w1(q, df)
```

Arguments

`q` Numeric value of the observed GFisher statistic.
`df` Vector of degrees of freedom for each p-value transformation.

Details

Calculate P-Value of GFisher Under Independence with Equal Weights

When weights are equal (or all equal to $1/n$), the GFisher statistic under independence is:

$$S = \frac{1}{n} \sum_{i=1}^n \chi_{d_i}^2 \sim \frac{1}{n} \chi_{\sum d_i}^2$$

Value

A numeric p-value.

Author(s)

Hong Zhang, Zheyang Wu

References

Zhang, H., & Wu, Z. (2023). The generalized Fisher's combination and accurate p-value calculation under dependence. *Biometrics*, 79(2), 1159-1172.

See Also

[p.GFisher_ind](#), [p.GFisher](#)

Examples

```
# Example: Fast calculation with equal weights
set.seed(123)
n <- 10
df <- rep(2, n) # Equal df
pval <- runif(n)
q <- stat.GFisher(pval, df = df, w = 1)

# Fast exact p-value
p_fast <- p.GFisher_ind_w1(q, df)
print(p_fast)

# Compare with standard chi-square test (df = 2n)
p_chisq <- pchisq(q, df = sum(df), lower.tail = FALSE)
print(p_chisq)
```

pval.oGFisher	<i>Omnibus GFisher (oGFisher) Test P-Value</i>
---------------	--

Description

Calculate the p-value for an omnibus GFisher test, which combines multiple GFisher statistics using either Cauchy combination or multivariate normal distribution.

Usage

```
pval.oGFisher(
  p,
  DF,
  W,
  M,
  p.type = "two",
  method = "HYB",
  combine = "cct",
  nsim = NULL,
  seed = NULL
)
```

Arguments

p	A numeric vector of input p-values for the oGFisher test.
DF	A matrix of degrees of freedom for inverse chi-square transformation for each p-value. Each row represents a GFisher test.
W	A matrix of non-negative weights. Each row represents a GFisher test.
M	Correlation matrix of the input Z-scores from which the input p-values were obtained.

p.type	Character string: "two" for two-sided (default), "one" for one-sided input p-values.
method	Character string specifying calculation method: • "MR": Simulation-assisted moment ratio matching • "HYB": Moment ratio matching by quadratic approximation (default) • "GB": Brown's method with calculated variance
combine	Character string specifying combination method: • "cct": oGFisher using the Cauchy combination test (default) • "mvn": oGFisher using multivariate normal distribution (minimum p-value approach)
nsim	Number of simulations used in the "MR" method. Default is 5e4.
seed	Optional seed for random number generation. Default is NULL.

Details

Compute the oGFisher Test P-Value

This function performs the omnibus GFisher test by:

1. Computing individual GFisher statistics and p-values for each test configuration
2. Combining these p-values using either:
 - **CCT (Cauchy Combination Test)**: Robust to dependence, analytically tractable
 - **MVN (Multivariate Normal)**: Uses correlation structure to compute minimum p-value distribution

Cauchy Combination Method (combine = "cct"):

The CCT approach is preferred when the individual tests may be highly dependent. The p-value is calculated as $P(\text{Cauchy} > \text{cct})$ where cct is the Cauchy combination statistic from [stat.oGFisher](#).

Minimum P-Value Method (combine = "mvn"):

This approach accounts for the correlation among GFisher tests using their theoretical correlation matrix (computed via [getGFisherCOR](#)). The p-value is:

$$P = 1 - P(\text{all normalized p-values} > \Phi^{-1}(1 - \text{minp}))$$

where the probability is computed using the multivariate normal distribution.

Value

A list with the following components:

stat	The test statistic: CCT statistic if combine = "cct", or minimum p-value if combine = "mvn"
pval	The p-value of the oGFisher test
pval_indi	Vector of individual p-values for each GFisher test
stat_indi	Vector of individual GFisher statistics

Author(s)

Hong Zhang, Zheyang Wu

References

Zhang, H., & Wu, Z. (2023). The generalized Fisher's combination and accurate p-value calculation under dependence. *Biometrics*, 79(2), 1159-1172.

Liu, Y., & Xie, J. (2020). Cauchy combination test: a powerful test with analytic p-value calculation under arbitrary dependency structures. *Journal of the American Statistical Association*, 115(529), 393-402.

See Also

[stat.GFisher](#), [p.GFisher](#), [stat.oGFisher](#), [getGFisherCOR](#)

Examples

```
# Example: oGFisher test with Cauchy combination
set.seed(123)
n <- 10
M <- matrix(0.3, n, n) + diag(0.7, n, n)
zscore <- matrix(rnorm(n), nrow = 1) %%% chol(M)
pval <- 2 * (1 - pnorm(abs(zscore)))

# Define multiple GFisher tests
DF <- rbind(rep(1, n), rep(2, n))
W <- rbind(rep(1, n), 1:10)

# CCT combination
result_cct <- pval.oGFisher(pval, DF, W, M, p.type = "two",
                           method = "HYB", combine = "cct")
print(result_cct)

# MVN combination (minimum p-value)
result_mvn <- pval.oGFisher(pval, DF, W, M, p.type = "two",
                           method = "HYB", combine = "mvn")
print(result_mvn)

# Alternative: single df per test
DF_short <- rbind(1, 2)
result <- pval.oGFisher(pval, DF = DF_short, W, M, p.type = "two",
                       method = "HYB", combine = "cct")
```

Description

Calculate the p-value for an omnibus GFisher test when input p-values are independent. This is the gold standard for independent inputs.

Usage

```
pval.oGFisher_ind(p, DF, W, combine = "cct")
```

Arguments

p	A numeric vector of input p-values.
DF	A matrix of degrees of freedom. Each row is the df vector for a GFisher test.
W	A matrix of weights. Each row is the weight vector for a GFisher test.
combine	Character string: "cct" for Cauchy combination (default), or "minp" for minimum p-value with multivariate normal distribution.

Details

Calculate oGFisher P-Value Under Independence

This function is the gold standard for oGFisher tests under independence.

Cauchy Combination (combine = "cct"):

Uses the Cauchy combination test statistic. The p-value is:

$$P = P(\text{Cauchy} > \text{cct})$$

For extremely large CCT values ($> 1e15$), uses the approximation $P \approx 1/(\text{cct} \cdot \pi)$.

Minimum P-Value (combine = "minp"):

Uses the correlation structure among GFisher tests (computed via [getGFisherCOR](#) with M = I) to calculate the p-value of the minimum p-value statistic via multivariate normal distribution.

Value

A list with the following components:

pval	The p-value of the oGFisher test
pval_indi	Vector of individual p-values for each GFisher test

Author(s)

Hong Zhang, Zheyang Wu

References

Zhang, H., & Wu, Z. (2023). The generalized Fisher's combination and accurate p-value calculation under dependence. *Biometrics*, 79(2), 1159-1172.

Liu, Y., & Xie, J. (2020). Cauchy combination test: a powerful test with analytic p-value calculation under arbitrary dependency structures. *Journal of the American Statistical Association*, 115(529), 393-402.

See Also

[pval.oGFisher](#), [stat.oGFisher_ind](#), [p.GFisher_ind](#)

Examples

```
## Not run:
# Requires coga package
if (requireNamespace("coga", quietly = TRUE)) {
  set.seed(122)
  n <- 10
  nGF <- 20

  # Create test configurations
  DF <- matrix(runif(n * nGF, 0.5, 5), ncol = n) / 10
  W <- abs(matrix(rnorm(n * nGF), ncol = n))

  # Test data
  p <- runif(n)
  p[1] <- 0.00001

  # CCT combination (default)
  result_cct <- pval.oGFisher_ind(p = p, DF = DF, W = W, combine = "cct")
  print(result_cct)

  # Minimum p-value combination
  result_minp <- pval.oGFisher_ind(p = p, DF = DF, W = W, combine = "minp")
  print(result_minp)

  # Compare with general method
  result_general_cct <- pval.oGFisher(p = p, DF = DF, W = W, M = diag(n),
                                     method = "MR", combine = "cct")
  print(result_general_cct)
}

## End(Not run)
```

stat.GFisher

GFisher Test Statistic

Description

Compute the GFisher test statistics $S = \sum_i w_i F_{d_i}^{-1}(1 - P_i)$, based on a vector of p-values P_i 's, degrees of freedom d_i 's, and weights w_i 's. $F_{d_i}^{-1}$ is the inverse CDF of the chi-square distribution with d_i degrees of freedom.

Usage

```
stat.GFisher(p, df = 2, w = 1)
```

Arguments

p	A numeric vector of input p-values for the GFisher test. Must be between 0 and 1.
df	Degrees of freedom for inverse chi-square transformation for each p-value. It can be a vector of the same length as p, indicating each transformation function might have a different df, or a single number, indicating the same degrees of freedom for all. Default is 2 (standard Fisher's method).
w	A numeric vector of non-negative weights for each p-value. Default is 1 (equal weights). If a single value is provided, equal weights are used for all p-values. Note: Weights are automatically normalized to sum to 1 (i.e., $w \leftarrow w / \sum(w)$). This normalization improves numerical stability and is statistically equivalent to other normalizations such as $\sum w_i = n$.

Details

Compute the GFisher Test Statistic

The statistic is calculated based on normalized weights, i.e., $w / \sum(w)$, ensuring that the weights always sum to 1.

When all degrees of freedom equal 2, the function uses a faster implementation equivalent to Fisher's combination method: $-2 \log(p)$.

The GFisher test generalizes Fisher's combination method by allowing:

- Different degrees of freedom for each p-value transformation
- Flexible weighting schemes
- Handling of dependent test statistics through correlation structure (see [p.GFisher](#))

Value

A numeric value representing the GFisher test statistic.

Author(s)

Hong Zhang, Zheyang Wu

References

Zhang, H., & Wu, Z. (2023). The generalized Fisher's combination and accurate p-value calculation under dependence. *Biometrics*, 79(2), 1159-1172.

See Also

[p.GFisher](#), [stat.oGFisher](#), [pval.oGFisher](#)

Examples

```
# Example 1: Equal weights with df=2 (standard Fisher's method)
set.seed(123)
n <- 10
pval <- runif(n)
stat.GFisher(pval, df = 2, w = 1)

# Example 2: Equal weights with explicit vectors
stat.GFisher(pval, df = rep(2, n), w = rep(1, n))

# Example 3: Varying degrees of freedom and weights
stat.GFisher(pval, df = 1:n, w = 1:n)

# Example 4: Small p-values (signal detection)
pval_signal <- c(0.001, 0.002, 0.05, runif(7))
stat.GFisher(pval_signal, df = 2, w = 1)
```

stat.oGFisher

Omnibus GFisher (oGFisher) Test Statistics

Description

Calculate statistics for multiple GFisher tests with different degrees of freedom and weights, along with their combination via Cauchy combination test (CCT) or minimum p-value.

Usage

```
stat.oGFisher(
  p,
  DF,
  W,
  M,
  p.type = "two",
  method = "HYB",
  nsim = NULL,
  seed = NULL
)
```

Arguments

p	A numeric vector of input p-values for the oGFisher test.
DF	A matrix of degrees of freedom for inverse chi-square transformation for each p-value. Each row represents a GFisher test. It can be a matrix with one column, indicating the same degrees of freedom for all p-values's chi-square transformations across different tests.
W	A matrix of non-negative weights. Each row represents a GFisher test. Must have the same dimensions as DF.

M	Correlation matrix of the input Z-scores from which the input p-values were obtained.
p.type	Character string: "two" for two-sided (default), "one" for one-sided input p-values.
method	Character string specifying calculation method: • "MR": Simulation-assisted moment ratio matching • "HYB": Moment ratio matching by quadratic approximation (default) • "GB": Brown's method with calculated variance
nsim	Number of simulations used in the "MR" method. Default is 5e4.
seed	Optional seed for random number generation. Default is NULL.

Details

Compute the oGFisher Test Statistics

The oGFisher (omnibus GFisher) test evaluates multiple GFisher statistics simultaneously, each with potentially different degrees of freedom and weighting schemes. This allows for a flexible and powerful approach to combining p-values.

The function computes:

1. Individual GFisher statistics for each row of DF and W
2. Corresponding p-values for each statistic
3. Minimum p-value across all tests
4. Cauchy combination statistic (CCT) for robust aggregation

Cauchy Combination:

The CCT statistic is computed as $\bar{T} = \frac{1}{m} \sum_{i=1}^m \tan[\pi(0.5 - P_i)]$, where P_i are the individual GFisher p-values. This approach is robust to dependence and handles very small p-values through a special transformation.

P-values larger than 0.9 are capped at 0.9 to improve stability. Very small p-values ($< 1e-15$) are handled using a special approximation: $1/(P_i \cdot \pi)$.

Value

A list with the following components:

STAT	Vector of GFisher test statistics, one for each row of DF
PVAL	Vector of individual p-values for each GFisher test
minp	Minimum p-value among all GFisher tests
cct	Cauchy combination test statistic for combining the p-values

Author(s)

Hong Zhang, Zheyang Wu

References

Zhang, H., & Wu, Z. (2023). The generalized Fisher's combination and accurate p-value calculation under dependence. *Biometrics*, 79(2), 1159-1172.

Liu, Y., & Xie, J. (2020). Cauchy combination test: a powerful test with analytic p-value calculation under arbitrary dependency structures. *Journal of the American Statistical Association*, 115(529), 393-402.

See Also

[stat.GFisher](#), [p.GFisher](#), [pval.oGFisher](#)

Examples

```
# Example: Multiple GFisher tests with different df and weights
set.seed(123)
n <- 10
M <- matrix(0.3, n, n) + diag(0.7, n, n)
zscore <- matrix(rnorm(n), nrow = 1) %*% chol(M)
pval <- 2 * (1 - pnorm(abs(zscore)))

# Define two GFisher tests
DF <- rbind(rep(1, n), rep(2, n))
W <- rbind(rep(1, n), 1:10)

# Calculate oGFisher statistics
result <- stat.oGFisher(pval, DF, W, M, p.type = "two", method = "HYB")
print(result)

# Alternative: single df per test (expanded internally)
DF_short <- rbind(1, 2)
result2 <- stat.oGFisher(pval, DF = DF_short, W, M, p.type = "two", method = "HYB")
```

stat.oGFisher_ind	<i>oGFisher Statistics for Independent Inputs</i>
-------------------	---

Description

Calculate oGFisher statistics when input p-values are independent. This is the gold standard for independent inputs and is substantially faster than using [stat.oGFisher](#) with $M = \text{diag}(n)$.

Usage

```
stat.oGFisher_ind(p, DF, W)
```

Arguments

p	A numeric vector of input p-values.
DF	A matrix of degrees of freedom. Each row is the df vector for a GFisher test.
W	A matrix of weights. Each row is the weight vector for a GFisher test.

Details

Calculate oGFisher Statistics Under Independence

This function is the gold standard for computing oGFisher statistics when inputs are independent.

It computes:

1. Individual GFisher statistics using [stat.GFisher](#)
2. P-values using [p.GFisher_ind](#) (fast independent method)
3. Minimum p-value and Cauchy combination statistic

The Cauchy combination statistic handles very small p-values ($< 1e-15$) using the approximation $1/(P_i \cdot \pi)$.

Value

A list with the following components:

STAT	Vector of GFisher test statistics, one for each row of DF
PVAL	Vector of individual p-values for each GFisher test
minp	Minimum p-value among all GFisher tests
cct	Cauchy combination test statistic for combining the p-values

Author(s)

Hong Zhang, Zheyang Wu

References

Zhang, H., & Wu, Z. (2023). The generalized Fisher's combination and accurate p-value calculation under dependence. *Biometrics*, 79(2), 1159-1172.

See Also

[stat.oGFisher](#), [p.GFisher_ind](#), [pval.oGFisher_ind](#)

Examples

```
## Not run:
# Requires coga package
if (requireNamespace("coga", quietly = TRUE)) {
  set.seed(123)
  n <- 10
  nGF <- 3
```

```
# Create test configurations
DF <- matrix(runif(n * nGF, 0.5, 5), ncol = n) / 10
W <- abs(matrix(rnorm(n * nGF), ncol = n))

# Under H0
p <- runif(n)
result_null <- stat.oGFisher_ind(p = p, DF = DF, W = W)
print(result_null)

# With potential signal
p[1] <- 0.00001
result_signal <- stat.oGFisher_ind(p = p, DF = DF, W = W)
print(result_signal)

# Compare with general method (should be similar but slower)
result_general <- stat.oGFisher(p = p, DF = DF, W = W, M = diag(n), method = "HYB")
print(result_general)
}

## End(Not run)
```

Index

check_coga, [2](#)
clear_GFisher_cache, [3](#)

get_GFisher_cache_size, [4](#)
get_GFisher_cache_stats, [4](#)
getGFishercoef_cached_cpp, [5](#)
getGFisherCOR, [12–14](#)

p.GFisher, [3, 5, 8–10, 13, 16, 19](#)
p.GFisher_ind, [2, 3, 8, 10, 15, 20](#)
p.GFisher_ind_w1, [3, 9, 10](#)
pval.oGFisher, [7, 11, 15, 16, 19](#)
pval.oGFisher_ind, [3, 13, 20](#)

stat.GFisher, [6, 7, 13, 15, 19, 20](#)
stat.oGFisher, [7, 12, 13, 16, 17, 19, 20](#)
stat.oGFisher_ind, [2, 3, 9, 15, 19](#)