Package: symmoments (via r-universe)

May 28, 2026

Title Symbolic Central and Noncentral Moments of the Multivariate Normal Distribution

Version 1.2.2.9000

Maintainer Florian Schuberth <f.schuberth@utwente.nl>

Description Symbolic central and non-central moments of the multivariate normal distribution. Computes a standard representation, LateX code, and values at specified mean and covariance matrices.

Imports Rdpack, mvtnorm, cubature, combinat, multipol, mpoly

RdMacros Rdpack

Roxygen list(markdown = TRUE)

RoxygenNote 8.0.0

Suggests ape, knitr, rmarkdown

License GPL

Encoding UTF-8

LazyLoad yes

NeedsCompilation no

Config/roxygen2/version 8.0.0

VignetteBuilder knitr

URL https://github.com/FloSchuberth/symmoments

BugReports https://github.com/FloSchuberth/symmoments/issues

Config/pak/sysreqs libgmp3-dev make libicu-dev

Repository https://floschuberth.r-universe.dev

Date/Publication 2026-05-28 06:33:18 UTC

RemoteUrl https://github.com/floschuberth/symmoments

RemoteRef HEAD

RemoteSha 067a3785eae503341db72bf54e8700827f41eaff

Contents

`callmultmoments` *Compute Multivariate Moment Symbolically*

Description

Computes a multivariate normal moment by initializing variables, calling `multmoments`, and constructing output.

Usage

```
callmultmoments(moment)
```

Arguments

`moment`	A vector `c(k1, ..., kn)` specifying the moment $X_1^{k_1} \cdots X_n^{k_n}$.

Details

Each row of the representation gives the exponents for a single product of covariance terms. For example, `(1, 2, 0)` represents $S_{11}^1 S_{12}^2 S_{22}^0$, where the S_{ij} are the covariances.

The full moment is the sum of these terms multiplied by their respective coefficients. If the sum of the exponents is odd, the moment is 0.

Value

An object of class `'moment'`, which is a list with three components:

- `moment`: The input moment vector.
- `representation`: A matrix containing the representation in terms of upper-triangular matrices.
- `coefficients`: The coefficients corresponding to the rows of the representation.

If the sum of the exponents is odd, returns `-1` and prints "Sum of powers is odd. Moment is 0."

If any exponent is negative, returns `-2` and prints "All components of the moment must be non-negative."

If any exponent is not an integer, returns `-3` and prints "All components of the moment must be integers."

Author(s)

Kem Phillips <kemphillips@comcast.net>

References

Phillips K (2010). "R Functions to Symbolically Compute the Central Moments of the Multivariate Normal Distribution." *Journal of Statistical Software*, **33**(1), 1–14. doi:10.18637/jss.v033.c01.

See Also

`multmoments`, and the methods `toLatex`, `evaluate`, and `simulate` in the `symmoments` package.

Examples

```
# Compute the moment for the 4-dimensional moment c(1,2,3,4):
m.1234 <- callmultmoments(c(1, 2, 3, 4))
```

---

convert.mpoly *Convert Between mpoly and List Representations of Multivariate Polynomials*

Description

Converts an `mpoly` object to a simple list representation, or converts a simple list representation back to an `mpoly` object.

Usage

```
convert.mpoly(poly)
```

Arguments

poly	An mpoly object, or a list containing powers and coefficients that define a multivariate polynomial.

Details

The list representation consists of 2 components:

- powers: A matrix where each row represents the exponents/powers of X for a single term in the multivariate polynomial.
- coeff: A numeric vector where each element is the coefficient for the corresponding row/term in powers.

Value

If poly is of class 'mpoly', returns a list with two components (powers and coeff). If poly is a list of this form, returns the corresponding mpoly object.

Author(s)

Kem Phillips <kemphillips@comcast.net>

References

Phillips K (2010). "R Functions to Symbolically Compute the Central Moments of the Multivariate Normal Distribution." *Journal of Statistical Software*, **33**(1), 1–14. doi:10.18637/jss.v033.c01.

See Also

convert.multipol, evaluate_expected.polynomial, integrate.polynomial

Examples

```
## Not run:
library(mpoly)

# Create an mpoly object
t0 <- mpoly::mpoly(list(
  c(coef = 3, x1 = 2),
  c(coef = 2, x1 = 1, x2 = 3),
  c(coef = -4, z = 2),
  c(coef = 1, x1 = 1, x2 = 2, z = 1)
))

# Convert from mpoly to list representation
t1 <- convert.mpoly(t0)

# Convert from list representation back to an mpoly object
t2 <- convert.mpoly(t1)

## End(Not run)
```

`convert.multipol` *Convert Between multipol and List Representations of Multivariate Polynomials*

Description

Converts a `multipol` object to a simple list representation, or converts a simple list representation back to a `multipol` object.

Usage

```
convert.multipol(poly)
```

Arguments

`poly`	A `multipol` object, or a list containing powers and coefficients that define a multivariate polynomial.

Details

The list representation consists of 2 components:

- `powers`: A matrix where each row represents the exponents/powers of X for a single term in the multivariate polynomial.
- `coeff`: A numeric vector where each element is the coefficient for the corresponding row/term in `powers`.

Value

If `poly` is of class `'multipol'`, returns a list with two components (`powers` and `coeff`). If `poly` is a list of this form, returns the corresponding `multipol` object.

Author(s)

Kem Phillips <kemphillips@comcast.net>

References

Phillips K (2010). "R Functions to Symbolically Compute the Central Moments of the Multivariate Normal Distribution." *Journal of Statistical Software*, **33**(1), 1–14. doi:10.18637/jss.v033.c01.

See Also

`convert.mpoly`, `evaluate_expected.polynomial`, `integrate.polynomial`

Examples

```
## Not run:
library(mpoly)
library(multipol)

# Create an mpoly object to work with
t0 <- mpoly::mpoly(list(
  c(coef = 3, x1 = 2),
  c(coef = 2, x1 = 1, x2 = 3),
  c(coef = -4, z = 2),
  c(coef = 1, x1 = 1, x2 = 2, z = 1)
))

# Convert from mpoly to list representation
t1 <- convert.mpoly(t0)

# Convert from list representation to a multipol object
t2 <- convert.multipol(t1)

# Convert back to a list representation
t3 <- convert.multipol(t2)

## End(Not run)
```

`evaluate` *Evaluate a Moment or Polynomial*

Description

Evaluate a Moment or Polynomial

Usage

```
evaluate(object, sigma)
```

Arguments

`object`	An object of class `symmoment` or a multi-dimensional polynomial.
`sigma`	A numeric matrix representing the covariance matrix.

Value

The evaluated result.

`evaluate.moment` *Evaluate a Multivariate Moment*

Description

Generic method for class `moment` to compute the numerical value of a moment at a specified covariance matrix from the output of `callmultmoments`.

Usage

```
## S3 method for class 'moment'
evaluate(object, sigma)
```

Arguments

`object`	An object of class `'moment'`.
`sigma`	An upper-triangular matrix of covariance terms expressed as a vector at which the moment is to be evaluated.

Details

`object` is normally the output of a call to `callmultmoments`. This is a list with the first component being the moment itself, the second component being the set of upper-triangular matrices representing the moment, and the third component containing their corresponding coefficients. This is an object of class `'moment'`.

Value

The numeric value of the moment evaluated at the specified covariance matrix.

Author(s)

Kem Phillips <kemphillips@comcast.net>

References

Phillips K (2010). "R Functions to Symbolically Compute the Central Moments of the Multivariate Normal Distribution." *Journal of Statistical Software*, **33**(1), 1–14. doi:10.18637/jss.v033.c01.

See Also

`callmultmoments`, and the methods `simulate` and `toLatex` from the `symmoments` package.

Examples

```
# Evaluates the moment at c(1,2,3,4) at the following covariance matrix:
#   4 2 1 1
#   2 3 1 1
#   1 1 2 1
#   1 1 1 2
evaluate(callmultmoments(c(1, 2, 3, 4)), c(4, 2, 1, 1, 3, 1, 1, 2, 1, 2))
```

`evaluate_expected.polynomial`
Evaluate the Expected Value of a Multivariate Polynomial

Description

Evaluates the expected value of a multivariate polynomial assuming a specified non-central multivariate normal distribution.

Usage

```
evaluate_expected.polynomial(poly, mu, sigma, envir = "symmoments")
```

Arguments

`poly`	An object of class `'mpoly'` or `'multipol'`, or a simple list containing coefficients and powers defining the polynomial.
`mu`	A vector of real numbers representing the mean vector μ of the multivariate normal distribution.
`sigma`	A vector giving an upper-triangular matrix, stacked by row, representing the covariance matrix of the multivariate distribution.
`envir`	A character string specifying the environment containing the central moments needed for the calculation. Defaults to `'symmoments'`.

Details

This function searches the environment specified in the `envir` argument for the central moments required to complete the expected value expansion. The default is the `symmoments` environment. The computation will stop with an error message if any required central moment is missing from `envir`.

Value

The expected value of the multivariate polynomial evaluated at the specified multivariate normal mean and covariance matrix.

Author(s)

Kem Phillips <kemphillips@comcast.net>

References

Phillips K (2010). "R Functions to Symbolically Compute the Central Moments of the Multivariate Normal Distribution." *Journal of Statistical Software*, **33**(1), 1–14. doi:10.18637/jss.v033.c01.

See Also

evaluate_noncentral, make.all.moments

Examples

```
## Not run:
library(mpoly)

# Define an mpoly object for a multivariate polynomial and determine
# its expected value at a specified mean and covariance matrix:
# Note: All moments up to c(2,3,2) must exist in the symmoments environment.
# Run make.all.moments(c(2,3,2)) beforehand if necessary.

t0 <- mpoly(list(
  c(coef = 3, x1 = 2),
  c(coef = 2, x1 = 1, x2 = 3),
  c(coef = -4, z = 2),
  c(coef = 1, x1 = 1, x2 = 2, z = 1)
))

evaluate_expected.polynomial(t0, c(1, 2, 3), c(1, 0, 0, 1, 0, 1))

## End(Not run)
```

---

evaluate_noncentral *Evaluate a Non-Central Multivariate Moment*

Description

Computes the numerical value of a non-central moment at a specified mean and specified covariance matrix.

Usage

```
evaluate_noncentral(moment, mu, sigma, envir = "symmoments")
```

Arguments

moment	A vector of non-negative integers representing the non-central moment to be evaluated: $X_1^{k_1} \cdots X_n^{k_n}$.
mu	A vector of real numbers representing the mean vector μ of the multivariate normal distribution.

sigma	An upper-triangular matrix of covariance terms for the multivariate normal distribution, expressed as a vector stacked by row, at which the moment is to be evaluated.
envir	A character string specifying the environment containing the central moments needed for the calculation. Defaults to 'symmoments'.

Details

This function searches the environment specified in the envir argument for the central moments required to complete the non-central expansion. The default is the symmoments environment. All even central moments less than or equal to the requested moment vector must be present. The computation will stop with an error message if any required central moment is missing from envir.

Value

The numeric value of the non-central moment evaluated at the specified mean and covariance matrix.

Author(s)

Kem Phillips <kemphillips@comcast.net>

References

Phillips K (2010). "R Functions to Symbolically Compute the Central Moments of the Multivariate Normal Distribution." *Journal of Statistical Software*, **33**(1), 1–14. doi:10.18637/jss.v033.c01.

See Also

evaluate.moment, make.all.moments

Examples

```
## Not run:
# Evaluates the expected value of X1^3 X2 X3^2 at mean c(3,4,1)
# and at the following covariance matrix:
#    4 2 1
#    2 3 1
#    1 1 2
# Note: requires all central moments up to c(3,1,2) to exist in 'symmoments'.
# If needed, run: make.all.moments(c(3,1,2))
evaluate_noncentral(c(3, 1, 2), c(3, 4, 1), c(4, 2, 1, 3, 1, 2))

# Using central moments stored instead in the global environment:
evaluate_noncentral(c(3, 1, 2), c(3, 4, 1), c(4, 2, 1, 3, 1, 2), '.GlobalEnv')

## End(Not run)
```

`integrate.polynomial` *Numerically Integrate a Multivariate Polynomial*

Description

Integrates a multivariate polynomial against a specified non-central multivariate normal distribution using ordinary numerical integration via the `adaptIntegrate` function from the `cubature` package.

Usage

```
integrate.polynomial(poly, mu, sigma, lower = NULL, upper = NULL)
```

Arguments

`poly`	An object of class `'mpoly'` or `'multipol'`, or a simple list containing two components (`coeff` and `powers`) defining the polynomial.
`mu`	A numeric vector giving the mean vector μ of the multivariate normal distribution.
`sigma`	A square matrix specifying the covariance matrix of the multivariate normal distribution.
`lower`	A numeric vector of the lower limits of integration, containing one element for each dimension. If `NULL` (the default), it defaults to -6 times the standard deviations from the mean.
`upper`	A numeric vector of the upper limits of integration, containing one element for each dimension. If `NULL` (the default), it defaults to $+6$ times the standard deviations from the mean.

Details

Defaults for `lower` and `upper` boundaries are set to ± 6 times the standard deviations (the square roots of the diagonal elements of the covariance matrix `sigma`).

If the polynomial is defined by a simple list, it must contain two components:

- `powers`: A matrix where each row represents the exponents/powers for a single term in the polynomial.
- `coeff`: A numeric vector where each element is the coefficient of the corresponding row in `powers`.

For example, the list structure equivalent to the polynomial in the examples section is:
`list(coeff = c(3, 2, -4, 1), powers = matrix(c(2,0,0, 1,3,0, 0,0,2, 1,2,1), ncol = 3, byrow = TRUE))`

Value

The expected value of the polynomial numerically integrated against the specified multivariate normal distribution.

Author(s)

Kem Phillips <kemphillips@comcast.net>

References

Phillips K (2010). "R Functions to Symbolically Compute the Central Moments of the Multivariate Normal Distribution." *Journal of Statistical Software*, **33**(1), 1–14. doi:10.18637/jss.v033.c01.

See Also

evaluate_expected.polynomial, multmoments, evaluate, simulate

Examples

```
## Not run:
library(mpoly)

# Define an mpoly object for a multivariate polynomial
t0 <- mpoly(list(
  c(coef = 3, x1 = 2),
  c(coef = 2, x1 = 1, x2 = 3),
  c(coef = -4, z = 2),
  c(coef = 1, x1 = 1, x2 = 2, z = 1)
))

# Numerically integrate against a specified mean and covariance identity matrix
integrate.polynomial(t0, c(1, 2, 3), matrix(c(1,0,0, 0,1,0, 0,0,1), nrow = 3, byrow = TRUE))

## End(Not run)
```

---

make.all.moments *Create All Moments Up to Specified Size in Environment symmoments*

Description

Create all central moment objects of a specified or smaller size in the symmoments environment.

Usage

```
make.all.moments(moment, verbose = TRUE)
```

Arguments

moment	A vector c(k1, ..., kn) specifying the highest moment to compute.
verbose	If TRUE (the default), the names of the moments are shown as the algorithm progresses; if FALSE, progress is hidden.

Details

Unsorted moments (those whose exponents are not in numeric order) are created in the `symmoments` environment using the `tounsorted` function to transform them from their sorted counterpart. If the `symmoments` environment does not exist, the user is prompted to create it using `symmoments <- new.env()`.

If a sorted moment does not exist, it is automatically created. Moments of lower dimension are not created; for example, if `c(2, 4)` is input, `m20` is created, but `m2` is not.

Naming Conventions:

- Moments are named using the structure `mij..l`, e.g., `m136`.
- If any exponent is greater than 9, lowercase letters, and then uppercase letters are used. For example, `m3bA` represents the moment `c(3, 11, 36)`.
- The largest single exponent allowed by this alphanumeric encoding scheme is $9 + 26 + 26 = 61$.

If an object with a name of this form already exists in the target environment but is not an object of class `"moment"`, it will be silently overwritten by the new moment object.

Value

All objects of class `'moment'` up to the value given in `moment` are created in the environment `symmoments`.

Author(s)

Kem Phillips <kemphillips@comcast.net>

References

Phillips K (2010). "R Functions to Symbolically Compute the Central Moments of the Multivariate Normal Distribution." *Journal of Statistical Software*, **33**(1), 1–14. doi:10.18637/jss.v033.c01.

See Also

`callmultmoments`, `tounsorted`

Examples

```
## Not run:
# Create the symmoments environment if it does not exist
symmoments <- new.env()

# Compute all moments up to c(3,3)
make.all.moments(c(3, 3))

## End(Not run)
```

print.moment *Print the Representation of a Multivariate Moment*

Description

Prints an object of class 'moment'.

Usage

```
## S3 method for class 'moment'
print(x, ...)
```

Arguments

x	An object of class 'moment', usually the output of callmultmoments.
...	Included only for consistency with the generic function.

Details

Prints the moment as E[X1**k1 X2**k2 ...]: followed by the lines of the representation matrix with the corresponding coefficient attached to each row.

Author(s)

Kem Phillips <kemphillips@comcast.net>

References

Phillips K (2010). "R Functions to Symbolically Compute the Central Moments of the Multivariate Normal Distribution." *Journal of Statistical Software*, **33**(1), 1–14. doi:10.18637/jss.v033.c01.

See Also

callmultmoments

Examples

```
print(callmultmoments(c(1, 2, 3)))
```

`simulate.moment` *Compute a Multivariate Moment Using Monte Carlo Integration*

Description

Computes a multivariate normal moment by Monte Carlo integration.

Usage

```
## S3 method for class 'moment'
simulate(object, nsim, seed = NULL, Mean, Sigma, ...)
```

Arguments

`object`	An object of class `'moment'` representing $E[X_1^{k_1} \cdots X_n^{k_n}]$.
`nsim`	The number of samples to generate in computing the integral.
`seed`	An integer for the random number generator (`set.seed`).
`Mean`	The mean vector of $(X_1, \ldots, X_n)$.
`Sigma`	Covariance matrix of $(X_1, \ldots, X_n)$, dimension $n \times n$, expressed as a vector stacked by row.
`...`	Included only for consistency with the generic function.

Value

An approximate numerical value of the specified moment.

Note

Non-central moments can be approximated by specifying `Mean`. For central moments, set `Mean` to a vector of zeros.

The `mvtnorm` package must be installed for this function to utilize `rmvnorm`.

Author(s)

Kem Phillips <kemphillips@comcast.net>

References

Rizzo ML (2008). *Statistical Computing with R*, 1st edition. Chapman & Hall/CRC, Boca Raton.

See Also

`callmultmoments`, and the methods `toLatex` and `evaluate`.

Examples

```
# Using 10000 samples, estimate the central moment for the moment c(2,4)
# at the specified covariance matrix and mean (0,0):
library(mvtnorm)
simulate(callmultmoments(c(2, 4)), 10000, NULL, c(0, 0), c(2, 1, 1, 4))
```

`toLatex.moment` *LaTeX a Multivariate Moment*

Description

Computes a LaTeX representation sorted lexicographically of an object of class `'moment'`.

Usage

```
## S3 method for class 'moment'
toLatex(object, ...)
```

Arguments

`object`	An object of class `'moment'`, usually the output of `callmultmoments`.
`...`	Included only for consistency with the generic function.

Details

The first element of the result is the moment expressed as an expected value (`E[...] =`). The remaining lines are the LaTeX representation broken at appropriate intervals for printing. (Individual terms for high dimensions will still overrun a printed line.)

Double backslashes (`\\`) are inserted where LaTeX requires a backslash. These can be reset to single backslashes by writing the output to a file using the standard R function `writeLines`.

Value

A character vector giving the LaTeX code for the symbolic moment.

Author(s)

Kem Phillips <kemphillips@comcast.net>

References

Phillips K (2010). "R Functions to Symbolically Compute the Central Moments of the Multivariate Normal Distribution." *Journal of Statistical Software*, **33**(1), 1–14. doi:10.18637/jss.v033.c01.

See Also

`callmultmoments`, and the `evaluate` method.

Examples

```
toLatex(callmultmoments(c(1, 2, 3)))
```

`toLatex_noncentral` *Compute a LaTeX Expression for a Non-Central Moment*

Description

Computes a LaTeX expression for a non-central multivariate normal moment.

Usage

```
toLatex_noncentral(moment, envir = "symmoments")
```

Arguments

`moment`	A vector `c(k1, ..., kn)` specifying the moment $X_1^{k_1} \cdots X_n^{k_n}$.
`envir`	A character string specifying the environment that contains the required central moments. Defaults to `'symmoments'`.

Details

All required central moment objects must already exist in the specified environment (the default is `'symmoments'`). However, if only the sorted version of an unsorted moment exists in that environment, the `tounsorted` function will automatically be called to transform and obtain it.

Value

A character string giving the LaTeX representation of the non-central moment where X follows a multivariate normal distribution.

Author(s)

Kem Phillips <kemphillips@comcast.net>

References

Phillips K (2010). "R Functions to Symbolically Compute the Central Moments of the Multivariate Normal Distribution." *Journal of Statistical Software*, **33**(1), 1–14. doi:10.18637/jss.v033.c01.

See Also

`make.all.moments`, `tounsorted`, `callmultmoments`, and the method `toLatex`.

Examples

```
## Not run:
# Compute the LaTeX representation of the 2-dimensional non-central moment c(1,3).
# Note: This requires that all central moments up to c(1,3) have already been
# generated in the symmoments environment using make.all.moments.
toLatex_noncentral(c(1, 3))

## End(Not run)
```

`toMatching` *Convert a Phylogenetic Tree from a Moment L-Matrix to Matching Form*

Description

Converts a tree structure represented in a moment format into an ape matching format structure.

Usage

```
toMatching(L, type = NULL, tip.label = NULL)
```

Arguments

`L`	The input tree structure. This can be an `L-matrix` object, a square L matrix, or an L matrix in reduced upper-triangular (vector) form.
`type`	A character string, either `'square'` or `'ut'`. This must be specified if `L` is a raw matrix or vector rather than a formal `L-matrix` object. Defaults to `NULL`.
`tip.label`	A character vector containing custom labels for the tips. If `NULL` (the default), labels fallback to `"a"` through `"z"` if there are at most 26 tips; otherwise, 3-letter combinations of the form `"aaa"`, `"aab"`, etc., are generated.

Details

An `L-matrix` object is a list containing the following 5 components:

- `L`: The L-matrix in full square form.
- `L.ut`: The L-matrix in reduced upper-triangular form.
- `Newick`: The Newick string representation of the tree structure.
- `tip.label`: A character vector of the tip labels.
- `tip.label.n`: An integer specifying the total number of tips.

Value

A matching representation of the phylogenetic tree corresponding to the input. The output list is assigned the class `'L-matching'`, which contains 5 components including the tree in matching format.

Author(s)

Kem Phillips <kemphillips@comcast.net>

References

Phillips K (2010). "R Functions to Symbolically Compute the Central Moments of the Multivariate Normal Distribution." *Journal of Statistical Software*, **33**(1), 1–14. doi:10.18637/jss.v033.c01.

Diaconis PW, Holmes SP (1998). "Matchings and phylogenetic trees." *Proceedings of the National Academy of Sciences*, **95**(25), 14600–14602. doi:10.1073/pnas.95.25.14600.

See Also

toMoment, toNewick

Examples

```
# Create a Newick character string
exam.Newick <- "(((a,b),c),d);"

# Convert to a moment L-matrix
exam.moment <- toMoment(exam.Newick)

# Convert to matching format
exam.matching <- toMatching(exam.moment)
```

---

toMoment *Convert a Tree from Newick or Matching to Moment Format*

Description

Converts a phylogenetic tree from a Newick character string or an ape matching matrix into a moment L-matrix object.

Usage

```
toMoment(inputobject, tip.label = NULL)
```

Arguments

inputobject	A tree structure represented as a Newick format character string, or a matching object as defined in the **ape** package.
tip.label	A character vector specifying rearranged labels for the tips. If provided, these must be the original tip labels. Defaults to NULL.

Details

The returned L-matrix class object consists of 5 internal components:

- `L`: The L-matrix represented in full square form.
- `L.ut`: The L-matrix represented in upper-triangular form.
- `Newick`: The Newick string representation of the tree structure.
- `tip.label`: A character vector holding the labels of the tips.
- `tip.label.n`: An integer specifying the total number of tips.

Value

A moment L-matrix object corresponding to the input phylogenetic tree object.

Author(s)

Kem Phillips <kemphillips@comcast.net>

References

Phillips K (2010). "R Functions to Symbolically Compute the Central Moments of the Multivariate Normal Distribution." *Journal of Statistical Software*, **33**(1), 1–14. doi:10.18637/jss.v033.c01.

Felsenstein, J. (1990). The Newick tree format. http://evolution.genetics.washington.edu/phylip/newicktree.html

Diaconis PW, Holmes SP (1998). "Matchings and phylogenetic trees." *Proceedings of the National Academy of Sciences*, **95**(25), 14600–14602. doi:10.1073/pnas.95.25.14600.

See Also

toNewick, toMatching

Examples

```
# Create a Newick character string
exam.Newick <- "(((a,b),c),d);"

# Convert to a moment L-matrix
exam.moment <- toMoment(exam.Newick)

# Convert to a matching object
exam.matching <- toMatching(exam.moment)

# Convert back to a moment object
backto.moment <- toMoment(exam.matching)
```

toNewick *Convert a Phylogenetic Tree from a Moment L-Matrix to Newick Form*

Description

Converts a tree structure represented in a moment format back into a Newick format string.

Usage

```
toNewick(L, type = NULL, tip.label = NULL)
```

Arguments

L	The input tree structure. This can be an L-matrix object, a square L matrix, or an L matrix in reduced upper-triangular (vector) form.
type	A character string, either 'square' or 'ut'. This must be specified if L is a raw matrix or vector rather than a formal L-matrix object. Defaults to NULL.
tip.label	A character vector containing custom labels for the tips. If NULL (the default), labels fallback to "a" through "z" if there are at most 26 tips; otherwise, 3-letter combinations of the form "aaa", "aab", etc., are generated.

Details

An L-matrix object is a list containing the following 5 components:

- L: The L-matrix in full square form.
- L.ut: The L-matrix in reduced upper-triangular form.
- Newick: The Newick string representation of the tree structure.
- tip.label: A character vector of the tip labels.
- tip.label.n: An integer specifying the total number of tips.

Value

A character string representing the Newick format of the phylogenetic tree corresponding to the input. The output list is assigned the class 'L-Newick', which contains 5 components including the tree string.

Author(s)

Kem Phillips <kemphillips@comcast.net>

References

Phillips K (2010). "R Functions to Symbolically Compute the Central Moments of the Multivariate Normal Distribution." *Journal of Statistical Software*, **33**(1), 1–14. doi:10.18637/jss.v033.c01.

Felsenstein, J. (1990). The Newick tree format. http://evolution.genetics.washington.edu/phylip/newicktree.html

Diaconis PW, Holmes SP (1998). "Matchings and phylogenetic trees." *Proceedings of the National Academy of Sciences*, **95**(25), 14600–14602. doi:10.1073/pnas.95.25.14600.

See Also

toMoment, toMatching

Examples

```
# Create a Newick character string
exam.Newick <- "(((a,b),c),d);"

# Convert to a moment L-matrix
exam.moment <- toMoment(exam.Newick)

# Convert back to Newick format
backto.Newick <- toNewick(exam.moment)
```

---

tounsorted *Compute an Unsorted Central Moment Object from a Sorted Object*

Description

Produces an unsorted central moment object from a sorted object of class `'moment'`.

Usage

```
tounsorted(moment, sorted.moment)
```

Arguments

`moment`	The unsorted target moment to obtain, specified in vector form (e.g., `c(3, 1, 2)`).
`sorted.moment`	A sorted object of class `'moment'` to use as the base for creating the unsorted moment.

Details

Unsorted moments are those whose exponents are not in sorted numerical order (e.g., `m312` vs `m123`). The unsorted moment's representation is calculated by rearranging the rows and columns of the sorted moment's matrices successively.

Value

An object of class `'moment'`, which is a list containing the following three components:

`moment`	The input unsorted moment vector.
`representation`	A matrix containing the representation in terms of upper-triangular matrices, rearranged to match the target unsorted order.
`coefficients`	A numeric vector of the coefficients corresponding to the rows of the representation matrix.

Author(s)

Kem Phillips <kemphillips@comcast.net>

References

#' Phillips K (2010). "R Functions to Symbolically Compute the Central Moments of the Multivariate Normal Distribution." *Journal of Statistical Software*, **33**(1), 1–14. doi:10.18637/jss.v033.c01.

See Also

`multmoments`, `callmultmoments`

Examples

```
# Obtain unsorted moment m312 from sorted base m123
tounsorted(c(3, 1, 2), callmultmoments(c(1, 2, 3)))
```

Index