# Package 'CircMLE'

October 12, 2022

**Title** Maximum Likelihood Analysis of Circular Data

**Version** 0.3.0

**Description** A series of wrapper functions to implement the 10 maximum likelihood models of animal orientation described by Schnute and Groot (1992) <DOI:10.1016/S0003-3472(05)80068-5>. The functions also include the ability to use different optimizer methods and calculate various model selection metrics (i.e., AIC, AICc, BIC). The ability to perform variants of the Hermans-Rasson test and Pycke test is also included as described in Landler et al. (2019) <DOI:10.1186/s12898-019-0246-8>. The latest version also includes a new method to calculate circular-circular and circular-linear distance correlations.

**Maintainer** Robert Fitak <rfitak9@gmail.com>

**Author** Robert Fitak [aut, cre], Sönke Johnsen [aut]

**Depends** R (>= 3.2.2)

**License** GPL (>= 2)

**Encoding** UTF-8

**LazyData** true

**URL** https://www.r-project.org

**NeedsCompilation** no

**Date** 2020-8-31

**Repository** CRAN

**Imports** circular (>= 0.4-7), stats, energy (>= 1.7-7)

**RoxygenNote** 7.1.1

**Date/Publication** 2020-09-01 21:30:03 UTC

## R topics documented:

---

`check_data` *Data Checking Function*

---

## Description

Make sure data is in the right format. Datasets are coerced into class 'circular' of type = angles, units = radians, and modulo = 2pi. It is recommended to set these atributes ahead of time.

## Usage

```
check_data(data)
```

## Arguments

| | |
|---|---|
| `data` | A vector, class 'circular' is recommended but not required |

## Examples

```
testdata = circular::rvonmises(100, mu = circular::circular(pi), kappa = 3,
   control.circular = list(units = "degrees"))
check_data(testdata)
```

circ_mle *circ_mle*

## Description

Run all 10 maximum likelihood models of circular orientation

## Usage

```
circ_mle(
  data,
  criterion = "AIC",
  nchains = 5,
  BadStart = 10^9,
  niter = 5000,
  method = "BFGS",
  lambda.min = 0.25,
  q.diff,
  exclude = NULL
)
```

## Arguments

| | |
|---|---|
| data | A vector of class 'circular' |
| criterion | chose from either "AIC", "AICc", or "BIC" for the model comparison information criterion. (default = "AIC") |
| nchains | A positive integer indicating the number of chains to run. Only the chain with the lowest log likelihood is returned (default = 5) |
| BadStart | An integer to replace the log likelihood when starting parameters of the optimizer fall outside the preset bounds. This is usually set to a large number. Can also be set to Inf, but will result in an error if a method other than "Nelder-Mead" is chosen. (default = 10^9) |
| niter | The maximum number of iterations for the optimizing algorithm. Equivalent to the 'maxit' control parameter of the optim() function. See ?optim for more details. (default = 5000) |
| method | A character string indicating the optimizing algorithm to use. Either "BFGS" or "Nelder-Mead" are recommended. See ?optim for more details. (default = "BFGS") |
| lambda.min | The minimum proportional size of the first distribution. Must be between 0 and 1. (default = 0.25) |
| q.diff | The minimum difference (in radians) in preferred direction for bimodal models. Must be set between 0 and pi. (default = pi/4) |
| exclude | A character vector of the models to be excluded from the calculations. (The default is to include all 10 models). For example, exclude = c("M1", "M3A", "M5B"). |

## Value

A list with 4 elements:

$results: A data frame consisting of a row for each model (rownames) with the columns: 1 = number of free parameters, 2 = mu1, 3 = kappa1, 4 = lamda, 5 = mu2, 6 = kappa2, 7 = negative log likelihood, 8 = Counts.function, 9 = Counts.Gradient, 10 = convergence, 11 = message, 12 = AIC, 13 = AICc, 14 = BIC, 15 = delta AIC, 16 = delta AICc, 17 = delta BIC, 18 = relative likelihoods of criterion chosen, 19 = model weights (probabilities) for criterion chosen, 20 = evidence ratios for the best model selected by the criterion. See ?optim for details on columns 8-11.

$bestmodel: The best model according to the criterion chosen

$rt: A two-element vector giving the test statistic and p-value for the Rayleigh Test

$hessians: A list with each element containing the hessian matrix for each model. Used for calculating connfidence intervals of parameters.

## Examples

```
testdata = circular::rvonmises(100, mu = circular::circular(pi), kappa = 3)
circ_mle(testdata)
```

---

`ci_circmle` *ci_circmle*

---

## Description

Calculate the 95% confidence interval for estimated model parameters

## Usage

```
ci_circmle(circmle, model)
```

## Arguments

| | |
|---|---|
| `circmle` | A list consisting of the output from function 'circ_mle' |
| `model` | character string indicating the model to be used to estimate parameter uncertainty. Must be one of c("M2A", "M2B", "M2C", "M3A", "M3B", "M4A", "M4B", "M5A", "M5B"). Default = the $bestmodel from the circmle object. |

## Value

A data frame with a column for the parameter name, the maximum likelihood estimate (MLE), standard error (SE), and 95% confidence interval (CI) for each estimated model parameter.

## Examples

```
testdata = circular::rvonmises(100, mu = circular::circular(pi), kappa = 3)
fit <- circ_mle(testdata)
ci_circmle(fit)
```

`dcor.circular` *circular distance correlation function*

## Description

Perform a distance correlation between circular datasets or between circular and linear datasets.

## Usage

```
dcor.circular(x, y, method = "chord", type = "c-c", ...)
```

## Arguments

| | |
|---|---|
| `x` | A vector of class 'circular', or numeric vector of angles measured in radians |
| `y` | A vector of class 'circular', numeric vector of angles measured in radians, or numeric vector |
| `method` | the distance measure to be used. This must be one of the following functions: '"angularseparation"', '"chord"', '"geodesic"', or '"circ.range"' (default = "chord"). see ?dist.circular for additional details. |
| `type` | if 'type == "c-c"' then perform a circular-circular distance corellation, else if 'type == "c-l"' then perform a circular-linear distance corellation (default = "c-c"). |
| `...` | additional parameters passed to the dcor.test function |

## Value

Same as from the `dcor.test` function: a list with class 'htest'containing

method: description of test

statistic: observed value of the test statistic

estimate: dCov(x,y) or dCor(x,y)

estimates: a vector: [dCov(x,y), dCor(x,y), dVar(x), dVar(y)]

replicates: replicates of the test statistic

p.value: approximate p-value of the test

n: sample size

data.name: description of data

## See Also

`dcor` `dcov` `DCOR` `dcor.test` `dist.circular`

## Examples

```
# Circular-circular distance corellation
x <- circular::rvonmises(n = 50, mu = circular::circular(0), kappa = 3)
y <- x + circular::rvonmises(n = 50, mu = circular::circular(pi), kappa = 10)
dcor.circular(x, y)

# Run permutation test with 9999 iterations
dcor.circular(x, y, R = 9999)

# Circular-linear distance corellation
x <- circular::rvonmises(n = 50, mu = circular::circular(0), kappa = 3)
y <- as.numeric(x) + rnorm(50, mean = 5, sd = 2)
dcor.circular(x, y, type = "c-l", R = 9999)
```

---

# HR_test *Hermans-Rasson test function*

---

## Description

Perform variants of the Hermans-Rasson test.

## Usage

```
HR_test(data, original = F, iter = 9999)
```

## Arguments

| | |
|---|---|
| data | A vector of class 'circular', or numeric vector of angles measured in radians |
| original | A logical of whether or not to run the original version of the Hermans-Rasson test or the newer version described in Landler et al. (2019) doi: 10.1186/s12898-019-0246-8 (default = F) |
| iter | The number of bootstrap replicates to perform in order to estimate the p-value of the test. (default = 9999) |

## Value

A numeric vector of the test statistic (T) and associated p-value

## Examples

```
testdata = circular::rvonmises(20, mu = circular::circular(pi), kappa = 3)
HR_test(testdata, iter = 999)
```

`lr_test` *Likelihood ratio test for nested models*

## Description

Statistically test nested models for the rejection of the null model in favor of the alternative model.

## Usage

```
lr_test(data, null_model, alt_model)
```

## Arguments

| | |
|---|---|
| `data` | A vector of class 'circular' |
| `null_model` | character string indicating the null model to be used. Must be one of c("M1", "M2A", "M2B", "M2C", "M3A", "M3B", "M4A", "M4B", "M5A", "M5B"). |
| `alt_model` | character string indicating the alternative model to be used. Must be one of c("M1", "M2A", "M2B", "M2C", "M3A", "M3B", "M4A", "M4B", "M5A", "M5B"). |

## Value

A list with elements:

$null_model: Vector with the model name and number of free parameters

$alt_model: Vector with the model name and number of free parameters

$lr: likelihood ratio test statistic 2 * (-log(L1) - -log(L2))

$df: degrees of freedom (difference in number of parameters between models)

$p_val: probability of rejecting null model in favor of alternative model due to chance (e.g, if <0.05, data favors alternative model).

## Examples

```
testdata = circular::rvonmises(100, mu = circular::circular(pi), kappa = 3)
lr_test(testdata, "M1", "M2A")
```

## M1 *Model M1 MLE function*

### Description

Run Maximum likelihood estimation for model M1.

### Usage

```
M1(data)
```

### Arguments

| | |
|---|---|
| data | A vector of class 'circular' |

### Value

A list with the elements:

$lik: The negative log likelihood of data for model M1

### Examples

```
testdata = circular::rvonmises(100, mu = circular::circular(pi), kappa = 3)
M1(testdata)
```

## M2A *Model M2A MLE function*

### Description

Run Maximum likelihood estimation for model M2A.

### Usage

```
M2A(data, BadStart, nchains, method, niter)
```

### Arguments

| | |
|---|---|
| data | A vector of class 'circular' |
| BadStart | An integer to replace the log likelihood when starting parameters of the optimizer fall outside the preset bounds. This is usually set to a large number. Can also be set to Inf, but will result in an error if a method other than "Nelder-Mead" is chosen. (default = 10^9) |
| nchains | A positive integer indicating the number of chains to run. Only the chain with the lowest log likelihood is returned (default = 5) |

| | |
|---|---|
| method | A character string indicating the optimizing algorithm to use. Either "BFGS" or "Nelder-Mead" are recommended. See ?optim for more details. (default = "BFGS") |
| niter | The maximum number of iterations for the optimizing algorithm. Equivalent to the 'maxit' control parameter of the optim() function. See ?optim for more details. (default = 5000) |

## Value

A list with elements (same as for function optim()):

$par: Vector with the optimized mean angle (mu1) and concentration parameter (kappa1)

$lik: The negative log likelihood

$counts: A two-element integer vector giving the number of calls to 'fn' and 'gr' respectively. See ?optim() for details.

$convergence: An integer code. '0' indicates successful completion (which is always the case for '"SANN"' and '"Brent"'). Possible error codes are:

'1' indicates that the iteration limit 'maxit' had been reached.

'10' indicates degeneracy of the Nelder-Mead simplex.

'51' indicates a warning from the '"L-BFGS-B"' method; see component 'message' for further details.

'52' indicates an error from the '"L-BFGS-B"' method; see component 'message' for further details.

$message: A character string giving any additional information returned by the optimizer, or 'NULL'.

## Examples

```
testdata = circular::rvonmises(100, mu = circular::circular(pi), kappa = 3)
M2A(testdata)
```

---

# M2B *Model M2B MLE function*

---

## Description

Run Maximum likelihood estimation for model M2B.

## Usage

```
M2B(data, BadStart, nchains, method, niter)
```

## Arguments

| | |
|---|---|
| data | A vector of class 'circular' |
| BadStart | An integer to replace the log likelihood when starting parameters of the optimizer fall outside the preset bounds. This is usually set to a large number. Can also be set to Inf, but will result in an error if a method other than "Nelder-Mead" is chosen. (default = 10^9) |
| nchains | A positive integer indicating the number of chains to run. Only the chain with the lowest log likelihood is returned (default = 5) |
| method | A character string indicating the optimizing algorithm to use. Either "BFGS" or "Nelder-Mead" are recommended. See ?optim for more details. (default = "BFGS") |
| niter | The maximum number of iterations for the optimizing algorithm. Equivalent to the 'maxit' control parameter of the optim() function. See ?optim for more details. (default = 5000) |

## Value

A list with elements (same as for function optim()):

$par: Vector with the optimized mean angle (mu1) and concentration parameter (kappa1)

$lik: The negative log likelihood

$counts: A two-element integer vector giving the number of calls to 'fn' and 'gr' respectively. See ?optim() for details.

$convergence: An integer code. '0' indicates successful completion (which is always the case for '"SANN"' and '"Brent"'). Possible error codes are:

'1' indicates that the iteration limit 'maxit' had been reached.

'10' indicates degeneracy of the Nelder-Mead simplex.

'51' indicates a warning from the '"L-BFGS-B"' method; see component 'message' for further details.

'52' indicates an error from the '"L-BFGS-B"' method; see component 'message' for further details.

$message: A character string giving any additional information returned by the optimizer, or 'NULL'.

## Examples

```
testdata = circular::rvonmises(100, mu = circular::circular(pi), kappa = 3)
M2B(testdata)
```

M2C *Model M2C MLE function*

## Description

Run Maximum likelihood estimation for model M2C.

## Usage

```
M2C(data, BadStart, nchains, method, niter, lambda.min)
```

## Arguments

| | |
|---|---|
| data | A vector of class 'circular' |
| BadStart | An integer to replace the log likelihood when starting parameters of the optimizer fall outside the preset bounds. This is usually set to a large number. Can also be set to Inf, but will result in an error if a method other than "Nelder-Mead" is chosen. (default = 10^9) |
| nchains | A positive integer indicating the number of chains to run. Only the chain with the lowest log likelihood is returned (default = 5) |
| method | A character string indicating the optimizing algorithm to use. Either "BFGS" or "Nelder-Mead" are recommended. See ?optim for more details. (default = "BFGS") |
| niter | The maximum number of iterations for the optimizing algorithm. Equivalent to the 'maxit' control parameter of the optim() function. See ?optim for more details. (default = 5000) |
| lambda.min | The minimum proportional size of the first distribution. Must be between 0 and 1. (default = 0.25) |

## Value

A list with elements (same as for function optim()):

$par: Vector with the optimized mean angle (mu1), concentration parameter (kappa1) and proportional size (lambda) of the first distribution (mu1 and kappa1).

$lik: The negative log likelihood

$counts: A two-element integer vector giving the number of calls to 'fn' and 'gr' respectively. See ?optim() for details.

$convergence: An integer code. '0' indicates successful completion (which is always the case for '"SANN"' and '"Brent"'). Possible error codes are:

'1' indicates that the iteration limit 'maxit' had been reached.

'10' indicates degeneracy of the Nelder-Mead simplex.

'51' indicates a warning from the '"L-BFGS-B"' method; see component 'message' for further details.

'52' indicates an error from the '"L-BFGS-B"' method; see component 'message' for further details.

$message: A character string giving any additional information returned by the optimizer, or 'NULL'.

## Examples

```
testdata = circular::rvonmises(100, mu = circular::circular(pi), kappa = 3)
M2C(testdata)
```

---

M3A *Model M3A MLE function*

---

## Description

Run Maximum likelihood estimation for model M3A.

## Usage

```
M3A(data, BadStart, nchains, method, niter)
```

## Arguments

| | |
|---|---|
| data | A vector of class 'circular' |
| BadStart | An integer to replace the log likelihood when starting parameters of the optimizer fall outside the preset bounds. This is usually set to a large number. Can also be set to Inf, but will result in an error if a method other than "Nelder-Mead" is chosen. (default = 10^9) |
| nchains | A positive integer indicating the number of chains to run. Only the chain with the lowest log likelihood is returned (default = 5) |
| method | A character string indicating the optimizing algorithm to use. Either "BFGS" or "Nelder-Mead" are recommended. See ?optim for more details. (default = "BFGS") |
| niter | The maximum number of iterations for the optimizing algorithm. Equivalent to the 'maxit' control parameter of the optim() function. See ?optim for more details. (default = 5000) |

## Value

A list with elements (same as for function optim()):

$par: Vector with the optimized mean angle (mu1) and concentration parameter (kappa1).

$lik: The negative log likelihood

$counts: A two-element integer vector giving the number of calls to 'fn' and 'gr' respectively. See ?optim() for details.

$convergence: An integer code. '0' indicates successful completion (which is always the case for '"SANN"' and '"Brent"'). Possible error codes are:

'1' indicates that the iteration limit 'maxit' had been reached.

'10' indicates degeneracy of the Nelder-Mead simplex.

'51' indicates a warning from the '"L-BFGS-B"' method; see component 'message' for further details.

'52' indicates an error from the '"L-BFGS-B"' method; see component 'message' for further details.

$message: A character string giving any additional information returned by the optimizer, or 'NULL'.

## Examples

```
testdata = circular::rvonmises(100, mu = circular::circular(pi), kappa = 3)
M3A(testdata)
```

---

M3B *Model M3B MLE function*

---

## Description

Run Maximum likelihood estimation for model M3B.

## Usage

```
M3B(data, BadStart, nchains, method, niter)
```

## Arguments

| | |
|---|---|
| data | A vector of class 'circular' |
| BadStart | An integer to replace the log likelihood when starting parameters of the optimizer fall outside the preset bounds. This is usually set to a large number. Can also be set to Inf, but will result in an error if a method other than "Nelder-Mead" is chosen. (default = 10^9) |
| nchains | A positive integer indicating the number of chains to run. Only the chain with the lowest log likelihood is returned (default = 5) |
| method | A character string indicating the optimizing algorithm to use. Either "BFGS" or "Nelder-Mead" are recommended. See ?optim for more details. (default = "BFGS") |
| niter | The maximum number of iterations for the optimizing algorithm. Equivalent to the 'maxit' control parameter of the optim() function. See ?optim for more details. (default = 5000) |

## Value

A list with elements (same as for function optim()):

$par: Vector with the optimized mean angle (mu1), concentration parameter (kappa1), and concentration parameter (kappa2).

$lik: The negative log likelihood

$counts: A two-element integer vector giving the number of calls to 'fn' and 'gr' respectively. See ?optim() for details.

$convergence: An integer code. '0' indicates successful completion (which is always the case for '"SANN"' and '"Brent"'). Possible error codes are:

'1' indicates that the iteration limit 'maxit' had been reached.

'10' indicates degeneracy of the Nelder-Mead simplex.

'51' indicates a warning from the '"L-BFGS-B"' method; see component 'message' for further details.

'52' indicates an error from the '"L-BFGS-B"' method; see component 'message' for further details.

$message: A character string giving any additional information returned by the optimizer, or 'NULL'.

## Examples

```
testdata = circular::rvonmises(100, mu = circular::circular(pi), kappa = 3)
M3B(testdata)
```

---

`M4A` *Model M4A MLE function*

---

## Description

Run Maximum likelihood estimation for model M4A.

## Usage

```
M4A(data, BadStart, nchains, method, niter, lambda.min)
```

## Arguments

| | |
|---|---|
| `data` | A vector of class 'circular' |
| `BadStart` | An integer to replace the log likelihood when starting parameters of the optimizer fall outside the preset bounds. This is usually set to a large number. Can also be set to Inf, but will result in an error if a method other than "Nelder-Mead" is chosen. (default = 10^9) |
| `nchains` | A positive integer indicating the number of chains to run. Only the chain with the lowest log likelihood is returned (default = 5) |
| `method` | A character string indicating the optimizing algorithm to use. Either "BFGS" or "Nelder-Mead" are recommended. See ?optim for more details. (default = "BFGS") |
| `niter` | The maximum number of iterations for the optimizing algorithm. Equivalent to the 'maxit' control parameter of the optim() function. See ?optim for more details. (default = 5000) |
| `lambda.min` | The minimum proportional size of the first distribution. Must be between 0 and 1. (default = 0.25) |

## Value

A list with elements (same as for function optim()):

$par: Vector with the optimized mean angle (mu1), concentration parameter (kappa1), and proportional size (lambda) of the first distribution.

$lik: The negative log likelihood

$counts: A two-element integer vector giving the number of calls to 'fn' and 'gr' respectively. See ?optim() for details.

$convergence: An integer code. '0' indicates successful completion (which is always the case for '"SANN"' and '"Brent"'). Possible error codes are:

'1' indicates that the iteration limit 'maxit' had been reached.

'10' indicates degeneracy of the Nelder-Mead simplex.

'51' indicates a warning from the '"L-BFGS-B"' method; see component 'message' for further details.

'52' indicates an error from the '"L-BFGS-B"' method; see component 'message' for further details.

$message: A character string giving any additional information returned by the optimizer, or 'NULL'.

## Examples

```
testdata = circular::rvonmises(100, mu = circular::circular(pi), kappa = 3)
M4A(testdata)
```

---

# M4B *Model M4B MLE function*

---

## Description

Run Maximum likelihood estimation for model M4B.

## Usage

```
M4B(data, BadStart, nchains, method, niter, lambda.min)
```

## Arguments

| | |
|---|---|
| data | A vector of class 'circular' |
| BadStart | An integer to replace the log likelihood when starting parameters of the optimizer fall outside the preset bounds. This is usually set to a large number. Can also be set to Inf, but will result in an error if a method other than "Nelder-Mead" is chosen. (default = 10^9) |
| nchains | A positive integer indicating the number of chains to run. Only the chain with the lowest log likelihood is returned (default = 5) |
| method | A character string indicating the optimizing algorithm to use. Either "BFGS" or "Nelder-Mead" are recommended. See ?optim for more details. (default = "BFGS") |

| | |
|---|---|
| niter | The maximum number of iterations for the optimizing algorithm. Equivalent to the 'maxit' control parameter of the optim() function. See ?optim for more details. (default = 5000) |
| lambda.min | The minimum proportional size of the first distribution. Must be between 0 and 1. (default = 0.25) |

## Value

A list with elements (same as for function optim()):

$par: Vector with the optimized mean angle (mu1), concentration parameter (kappa1), concentration parameter (kappa2), and proportional size (lambda) of the first distribution.

$lik: The negative log likelihood

$counts: A two-element integer vector giving the number of calls to 'fn' and 'gr' respectively. See ?optim() for details.

$convergence: An integer code. '0' indicates successful completion (which is always the case for '"SANN"' and '"Brent"'). Possible error codes are:

'1' indicates that the iteration limit 'maxit' had been reached.

'10' indicates degeneracy of the Nelder-Mead simplex.

'51' indicates a warning from the '"L-BFGS-B"' method; see component 'message' for further details.

'52' indicates an error from the '"L-BFGS-B"' method; see component 'message' for further details.

$message: A character string giving any additional information returned by the optimizer, or 'NULL'.

## Examples

```
testdata = circular::rvonmises(100, mu = circular::circular(pi), kappa = 3)
M4B(testdata)
```

---

M5A *Model M5A MLE function*

---

## Description

Run Maximum likelihood estimation for model M5A.

## Usage

```
M5A(data, BadStart, nchains, method, niter, lambda.min, q.diff)
```

## Arguments

| | |
|---|---|
| data | A vector of class 'circular' |
| BadStart | An integer to replace the log likelihood when starting parameters of the optimizer fall outside the preset bounds. This is usually set to a large number. Can also be set to Inf, but will result in an error if a method other than "Nelder-Mead" is chosen. (default = 10^9) |
| nchains | A positive integer indicating the number of chains to run. Only the chain with the lowest log likelihood is returned (default = 5) |
| method | A character string indicating the optimizing algorithm to use. Either "BFGS" or "Nelder-Mead" are recommended. See ?optim for more details. (default = "BFGS") |
| niter | The maximum number of iterations for the optimizing algorithm. Equivalent to the 'maxit' control parameter of the optim() function. See ?optim for more details. (default = 5000) |
| lambda.min | The minimum proportional size of the first distribution. Must be between 0 and 1. (default = 0.25) |
| q.diff | The minimum difference (in radians) in preferred direction for bimodal models. Must be set between 0 and pi. (default = pi/4) |

## Value

A list with elements (same as for function optim()):

$par: Vector with the optimized mean angle (mu1), concentration parameter (kappa1), mean angle (mu2), and proportional size (lambda) of the first distribution.

$lik: The negative log likelihood

$counts: A two-element integer vector giving the number of calls to 'fn' and 'gr' respectively. See ?optim() for details.

$convergence: An integer code. '0' indicates successful completion (which is always the case for '"SANN"' and '"Brent"'). Possible error codes are:

'1' indicates that the iteration limit 'maxit' had been reached.

'10' indicates degeneracy of the Nelder-Mead simplex.

'51' indicates a warning from the '"L-BFGS-B"' method; see component 'message' for further details.

'52' indicates an error from the '"L-BFGS-B"' method; see component 'message' for further details.

$message: A character string giving any additional information returned by the optimizer, or 'NULL'.

## Examples

```
testdata = circular::rvonmises(100, mu = circular::circular(pi), kappa = 3)
M5A(testdata)
```

M5B *Model M5B MLE function*

## Description

Run Maximum likelihood estimation for model M5B.

## Usage

```
M5B(data, BadStart, nchains, method, niter, lambda.min, q.diff)
```

## Arguments

| | |
|---|---|
| data | A vector of class 'circular' |
| BadStart | An integer to replace the log likelihood when starting parameters of the optimizer fall outside the preset bounds. This is usually set to a large number. Can also be set to Inf, but will result in an error if a method other than "Nelder-Mead" is chosen. (default = 10^9) |
| nchains | A positive integer indicating the number of chains to run. Only the chain with the lowest log likelihood is returned (default = 5) |
| method | A character string indicating the optimizing algorithm to use. Either "BFGS" or "Nelder-Mead" are recommended. See ?optim for more details. (default = "BFGS") |
| niter | The maximum number of iterations for the optimizing algorithm. Equivalent to the 'maxit' control parameter of the optim() function. See ?optim for more details. (default = 5000) |
| lambda.min | The minimum proportional size of the first distribution. Must be between 0 and 1. (default = 0.25) |
| q.diff | The minimum difference (in radians) in preferred direction for bimodal models. Must be set between 0 and pi. (default = pi/4) |

## Value

A list with elements (same as for function optim()):

$par: Vector with the optimized mean angle (mu1), concentration parameter (kappa1), mean angle (mu2), concentration parameter (kappa2), and proportional size (lambda) of the first distribution.

$lik: The negative log likelihood

$counts: A two-element integer vector giving the number of calls to 'fn' and 'gr' respectively. See ?optim() for details.

$convergence: An integer code. '0' indicates successful completion (which is always the case for '"SANN"' and '"Brent"'). Possible error codes are:

'1' indicates that the iteration limit 'maxit' had been reached.

'10' indicates degeneracy of the Nelder-Mead simplex.

‘51’ indicates a warning from the ‘"L-BFGS-B"’ method; see component ‘message’ for further details.

‘52’ indicates an error from the ‘"L-BFGS-B"’ method; see component ‘message’ for further details.

$message: A character string giving any additional information returned by the optimizer, or ‘NULL’.

## Examples

```
testdata = circular::rvonmises(100, mu = circular::circular(pi), kappa = 3)
M5B(testdata)
```

---

`plot_circMLE` *plot_circMLE*

---

## Description

Plot the observed data and specific model from circ_mle output

## Usage

```
plot_circMLE(data, table, model, bins, shrink, col, lwd, lty)
```

## Arguments

| | |
|---|---|
| `data` | A vector of class ’circular’ |
| `table` | A list containing the output from the circ_mle function |
| `model` | The name of the model to be plotted. (default = first model of "table") |
| `bins` | The number of bins for the circular histogram. (defualt = 18) |
| `shrink` | The value by which to shrink the size of the plotted circle. Larger numbers shrink the circle, smaller numbers expand the circle. (default = 1.5) |
| `col` | Vector of colors used for plotting, up to four colors can be specified. The order is: histogram color, mean vector color, model density color, predicted mean direction(s) color(s). (default = c("grey", "red", "black", "black")) |
| `lwd` | Vector of line weights used for plotting, up to 3 weights can be specified. The order is: mean vector weight, model density weight, predicted mean direction(s) weight(s). (default = c(2, 2, 2)) |
| `lty` | Vector of line weights used for plotting, up to 3 weights can be specified. The order is: mean vector weight, model density weight, predicted mean direction(s) weight(s). (default = c("solid", "dashed", "dashed")) |

## Value

A plot is returned to the default image device.

### Note

In the resulting plot, the observed vector length is equal to rho (vector strength). However, the predicted direction(s) from the model are drawn with vector length of 1. Also, if the model "M1" is selected, by definition no predicted direction is plotted.

### Examples

```
testdata = circular::rvonmises(100, mu = circular::circular(pi), kappa = 3)
out = circ_mle(testdata)
plot_circMLE(testdata, out)
plot_circMLE(testdata, out, model = "M4A")
```

---

## pycke_test *Pycke test function*

---

### Description

Perform the Pycke test.

### Usage

```
pycke_test(data, iter = 9999)
```

### Arguments

| | |
|---|---|
| data | A vector of class 'circular', or numeric vector of angles measured in radians |
| iter | The number of bootstrap replicates to perform in order to estimate the p-value of the test. (default = 9999) |

### Value

A numeric vector of the test statistic (T) and associated p-value

### Examples

```
testdata = circular::rvonmises(20, mu = circular::circular(pi), kappa = 3)
pycke_test(testdata, iter = 999)
```

# Index