Package 'midi'

May 8, 2026

Title Microstructure Information from Diffusion Imaging

Version 0.1.0

Description An implementation of a taxonomy of models of restricted diffusion in biological tissues parametrized by the tissue geometry (axis, diameter, density, etc.). This is primarily used in the context of diffusion magnetic resonance (MR) imaging to model the MR signal attenuation in the presence of diffusion gradients. The goal is to provide tools to simulate the MR signal attenuation predicted by these models under different experimental conditions. The package feeds a companion 'shiny' app available at <https://midi-pastrami.apps.math.cnrs.fr> that serves as a graphical interface to the models and tools provided by the package. Models currently available are the ones in Neuman (1974) <doi:10.1063/1.1680931>, Van Gelderen et al. (1994) <doi:10.1006/jmrb.1994.1038>, Stanisz et al. (1997) <doi:10.1002/mrm.1910370115>, Soderman & Jonsson (1995) <doi:10.1006/jmra.1995.0014> and Callaghan (1995) <doi:10.1006/jmra.1995.1055>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.1

Depends R (>= 3.5)

URL https://github.com/lmjl-alea/midi, https://lmjl-alea.github.io/midi/

BugReports https://github.com/lmjl-alea/midi/issues

Imports cli, ggplot2, plotly, purrr, R6, rlang, withr

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Aymeric Stamm [aut, cre] (ORCID: <https://orcid.org/0000-0002-8725-3654>)

Maintainer Aymeric Stamm <aymeric.stamm@cnrs.fr>

Repository CRAN

Date/Publication 2024-04-03 19:43:03 UTC

Contents

`autoplot.bundle` *Plots a cross section of a cylinder bundle using* **ggplot2**

Description

Plots a cross section of a cylinder bundle from an object of class `bundle` as generated by `simulate_bundle()` using **ggplot2**.

Usage

```
## S3 method for class 'bundle'
autoplot(object, grid_size = 100L, ...)
```

Arguments

`object`	An object of class `bundle` as generated by `simulate_bundle()`.
`grid_size`	An integer value specifying the number of points on which the unit circle should be discretized to draw the spheres. Defaults to `100L`.
`...`	Additional arguments to be passed to the `ggplot2::autoplot()` method.

Value

A `ggplot2::ggplot()` object.

Examples

```
density <- 0.5
voxel_size <- 5 # micrometers
withr::with_seed(1234, {
  out <- simulate_bundle(density, voxel_size)
})
ggplot2::autoplot(out)
```

`CallaghanCompartment` *Callaghan's model for restricted diffusion in a cylinder*

Description

A class to model restricted diffusion in a cylinder using the Callaghan's model.

Super class

`midi::CylinderRadialCompartment` -> `CallaghanCompartment`

Methods

Public methods:

- `CallaghanCompartment$clone()`

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
CallaghanCompartment$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

References

Callaghan, P. T. (1995). Pulsed-gradient spin-echo NMR for planar, cylindrical, and spherical pores under conditions of wall relaxation. Journal of magnetic resonance, Series A, 113(1), 53-59.

CylinderBundleCompartment

Cylinder bundle compartment class

Description

A class to model restricted diffusion in a bundle of cylinders.

Methods

Public methods:

- `CylinderBundleCompartment$new()`
- `CylinderBundleCompartment$get_signal()`
- `CylinderBundleCompartment$clone()`

Method `new()`**:** Instantiates a new cylinder bundle compartment.

Usage:

```
CylinderBundleCompartment$new(
  axis,
  radius,
  diffusivity,
  cylinder_density,
  axial_diffusivity = NULL,
  radial_diffusivity = NULL,
  n_cylinders = 1L,
  axis_concentration = Inf,
  radius_sd = 0,
 radial_model = c("soderman", "callaghan", "stanisz", "neuman", "vangelderen"),
  seed = 1234
)
```

Arguments:

`axis` A numeric vector of length 3 and unit norm specifying the mean axis of the cylinder population.

`radius` A positive numeric value specifying the mean radius of the cylinder population in meters.

`diffusivity` A positive numeric value specifying the diffusivity within the cylinders in $m^2.s^{-1}$.

`cylinder_density` A numeric value specifying the density of the cylinders in the voxel. Must be between 0 and 1.

`axial_diffusivity` A numeric value specifying the axial diffusivity in the space outside the cylinders in $m^2.s^{-1}$. If not provided, defaults to a tortuosity model reducing the intrinsic diffusivity depending on orientation dispersion. Defaults to `NULL`.

`radial_diffusivity` A numeric value specifying the radial diffusivity in the space outside the cylinders in $m^2.s^{-1}$. If not provided, defaults to a tortuosity model reducing the axial diffusivity depending on radius heterogeneity. Defaults to `NULL`.

`n_cylinders` An integer value specifying the number of cylinders in the bundle. Defaults to `1L`.

`axis_concentration` A numeric value specifying the concentration of cylinders along the mean axis. Defaults to `Inf`.

`radius_sd` A numeric value specifying the standard deviation of the radius of the cylinder population in meters. Defaults to `0`.

`radial_model` A character string specifying the radial model to use. Choices are `"soderman"`, `"callaghan"`, `"stanisz"`, `"neuman"`, and `"vangelderen"`. Defaults to `"soderman"`.

`seed` An integer value specifying the seed for the random number generator. Defaults to `1234`.

Returns: An instance of the `CylinderBundleCompartment` class.

Method `get_signal()`**:** Computes the signal attenuation predicted by the model.

Usage:

```
CylinderBundleCompartment$get_signal(
  small_delta,
  big_delta,
  G,
  direction,
  echo_time = NULL,
  n_max = 20L,
  m_max = 50L
)
```

Arguments:

`small_delta` A numeric value specifying the duration of the gradient pulse in seconds.

`big_delta` A numeric value specifying the duration between the gradient pulses in seconds.

`G` A numeric value specifying the strength of the gradient in $T.m^{-1}$.

`direction` A length-3 numeric vector specifying the direction of the gradient.

`echo_time` A numeric value specifying the echo time in seconds.

`n_max` An integer value specifying the maximum order of the Bessel function. Defaults to `20L`.

`m_max` An integer value specifying the maximum number of extrema for the Bessel function. Defaults to `50L`.

Returns: A numeric value storing the predicted signal attenuation.

Examples:

```
cylinderBundleComp <- CylinderBundleCompartment$new(
  axis = c(0, 0, 1),
  radius = 1e-5,
  diffusivity = 2.0e-9,
  cylinder_density = 0.5,
  radial_model = "soderman"
)
cylinderBundleComp$get_signal(
  small_delta = 0.03,
  big_delta = 0.03,
  G = 0.040,
  direction = c(0, 0, 1)
)
```

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
CylinderBundleCompartment$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
## ------------------------------------------------
## Method `CylinderBundleCompartment$get_signal`
## ------------------------------------------------

cylinderBundleComp <- CylinderBundleCompartment$new(
  axis = c(0, 0, 1),
  radius = 1e-5,
  diffusivity = 2.0e-9,
  cylinder_density = 0.5,
  radial_model = "soderman"
)
cylinderBundleComp$get_signal(
  small_delta = 0.03,
  big_delta = 0.03,
  G = 0.040,
  direction = c(0, 0, 1)
)
```

`CylinderCompartment` *Cylinder compartment class*

Description

A class to model restricted diffusion in a cylinder.

Methods

Public methods:

- `CylinderCompartment$new()`
- `CylinderCompartment$get_signal()`
- `CylinderCompartment$clone()`

Method `new()`**:** Instantiates a new cylinder compartment.

Usage:

```
CylinderCompartment$new(
  axis,
  radius,
  diffusivity,
  radial_model = c("soderman", "callaghan", "stanisz", "neuman", "vangelderen")
)
```

Arguments:

`axis` A length-3 numeric vector specifying the axis of the cylinder.

`radius` A numeric value specifying the radius of the cylinder in meters.

`diffusivity` A numeric value specifying the diffusivity within the cylinder in $m^2.s^{-1}$.

`radial_model` A character string specifying the radial model to use. Choices are `"soderman"`, `"callaghan"`, `"stanisz"`, `"neuman"`, and `"vangelderen"`. Defaults to `"soderman"`.

Returns: An instance of the `CylinderCompartment` class.

Method `get_signal()`**:** Computes the signal attenuation predicted by the model.

Usage:

```
CylinderCompartment$get_signal(
  small_delta,
  big_delta,
  G,
  direction,
  echo_time = NULL,
  n_max = 20L,
  m_max = 50L
)
```

Arguments:

`small_delta` A numeric value specifying the duration of the gradient pulse in seconds.

`big_delta` A numeric value specifying the duration between the gradient pulses in seconds.

`G` A numeric value specifying the strength of the gradient in $T.m^{-1}$.

`direction` A length-3 numeric vector specifying the direction of the gradient.

`echo_time` A numeric value specifying the echo time in seconds.

`n_max` An integer value specifying the maximum order of the Bessel function. Defaults to `20L`.

`m_max` An integer value specifying the maximum number of extrema for the Bessel function. Defaults to `50L`.

Returns: A numeric value storing the predicted signal attenuation.

Examples:

```
cylinderComp <- CylinderCompartment$new(
  axis = c(0, 0, 1),
  radius = 1e-6,
  diffusivity = 2.0e-9,
  radial_model = "soderman"
)
cylinderComp$get_signal(
  small_delta = 0.03,
  big_delta = 0.03,
  G = 0.040,
  direction = c(0, 0, 1)
)
```

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
CylinderCompartment$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
## ------------------------------------------------
## Method `CylinderCompartment$get_signal`
## ------------------------------------------------

cylinderComp <- CylinderCompartment$new(
  axis = c(0, 0, 1),
  radius = 1e-6,
  diffusivity = 2.0e-9,
  radial_model = "soderman"
)
cylinderComp$get_signal(
  small_delta = 0.03,
  big_delta = 0.03,
  G = 0.040,
  direction = c(0, 0, 1)
)
```

`CylinderRadialCompartment`

Cylinder radial compartment class

Description

A class to model restricted diffusion in a cylinder in the plane perpendicular to the cylinder axis.

Methods

Public methods:

- `CylinderRadialCompartment$new()`
- `CylinderRadialCompartment$get_signal()`
- `CylinderRadialCompartment$clone()`

Method `new()`**:** Instantiates a new cylinder radial compartment.

Usage:

```
CylinderRadialCompartment$new(radius, diffusivity)
```

Arguments:

`radius` A numeric value specifying the radius of the cylinder in meters.

`diffusivity` A numeric value specifying the diffusivity within the cylinder in $m^2.s^{-1}$.

Returns: An instance of the `CylinderRadialCompartment` class.

Method `get_signal()`**:** Computes the signal attenuation predicted by the model.

Usage:

```
CylinderRadialCompartment$get_signal(
  small_delta,
  big_delta,
  G,
  echo_time = NULL,
  n_max = 20L,
  m_max = 50L
)
```

Arguments:

`small_delta` A numeric value specifying the duration of the gradient pulse in seconds.

`big_delta` A numeric value specifying the duration between the gradient pulses in seconds.

`G` A numeric value specifying the strength of the gradient in $T.m^{-1}$.

`echo_time` A numeric value specifying the echo time in seconds.

`n_max` An integer value specifying the maximum order of the Bessel function. Defaults to `20L`.

`m_max` An integer value specifying the maximum number of extrema for the Bessel function. Defaults to `50L`.

Returns: A numeric value storing the predicted signal attenuation.

Examples:

```
sodermanComp <- SodermanCompartment$new(
  radius = 1e-6,
  diffusivity = 2.0e-9
)
sodermanComp$get_signal(0.03, 0.03, 0.040)

staniszComp <- StaniszCompartment$new(
  radius = 1e-6,
  diffusivity = 2.0e-9
)
staniszComp$get_signal(0.03, 0.03, 0.040)

neumanComp <- NeumanCompartment$new(
  radius = 1e-6,
  diffusivity = 2.0e-9
)
neumanComp$get_signal(0.03, 0.03, 0.040, echo_time = 0.040)

callaghanComp <- CallaghanCompartment$new(
  radius = 1e-6,
  diffusivity = 2.0e-9
)
callaghanComp$get_signal(0.03, 0.03, 0.040)

vanGelderenComp <- VanGelderenCompartment$new(
  radius = 1e-6,
```

```
  diffusivity = 2.0e-9
)
vanGelderenComp$get_signal(0.03, 0.03, 0.040)
```

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
CylinderRadialCompartment$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
## ------------------------------------------------
## Method `CylinderRadialCompartment$get_signal`
## ------------------------------------------------

sodermanComp <- SodermanCompartment$new(
  radius = 1e-6,
  diffusivity = 2.0e-9
)
sodermanComp$get_signal(0.03, 0.03, 0.040)

staniszComp <- StaniszCompartment$new(
  radius = 1e-6,
  diffusivity = 2.0e-9
)
staniszComp$get_signal(0.03, 0.03, 0.040)

neumanComp <- NeumanCompartment$new(
  radius = 1e-6,
  diffusivity = 2.0e-9
)
neumanComp$get_signal(0.03, 0.03, 0.040, echo_time = 0.040)

callaghanComp <- CallaghanCompartment$new(
  radius = 1e-6,
  diffusivity = 2.0e-9
)
callaghanComp$get_signal(0.03, 0.03, 0.040)

vanGelderenComp <- VanGelderenCompartment$new(
  radius = 1e-6,
  diffusivity = 2.0e-9
)
vanGelderenComp$get_signal(0.03, 0.03, 0.040)
```

NeumanCompartment *Neuman's model for restricted diffusion in a cylinder*

Description

A class to model restricted diffusion in a cylinder using the Neuman's model.

Super class

`midi::CylinderRadialCompartment` -> `NeumanCompartment`

Methods

Public methods:

- `NeumanCompartment$clone()`

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
NeumanCompartment$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

References

Neuman, C. H. (1974). Spin echo of spins diffusing in a bounded medium. The Journal of Chemical Physics, 60(11), 4508-4511.

plot.bundle *Plots a cross section of a cylinder bundle*

Description

Plots a cross section of a cylinder bundle

Usage

```
## S3 method for class 'bundle'
plot(x, grid_size = 100L, ...)
```

Arguments

`x`	An object of class `bundle` as generated by `simulate_bundle()`.
`grid_size`	An integer value specifying the number of points on which the unit circle should be discretized to draw the spheres. Defaults to `100L`.
`...`	Additional arguments to be passed to the `ggplot2::autoplot()` method.

Value

Nothing.

Examples

```
density <- 0.5
voxel_size <- 5 # micrometers
withr::with_seed(1234, {
  out <- simulate_bundle(density, voxel_size)
})
plot(out)
```

`plot3d` *Plots a 3D representation of a cylinder bundle using* **plotly**

Description

Plots a 3D representation of a cylinder bundle from an object of class `bundle` as generated by `simulate_bundle()` using **plotly**.

Usage

```
plot3d(b, show_linear_mesh = FALSE)
```

Arguments

`b`	An object of class `bundle` as generated by `simulate_bundle()`.
`show_linear_mesh`	A logical value indicating whether the linear mesh of each cylinder should be displayed. Defaults to `FALSE` for computational efficiency.

Value

An HTML widget of class `plotly::plotly` storing the 3D visualization of the cylinder bundle.

Examples

```
density <- 0.5
voxel_size <- 5 # micrometers
withr::with_seed(1234, {
  out <- simulate_bundle(density, voxel_size)
})
plot3d(out)
```

run_app *Runs the MIDI Shiny web application*

Description

This is a helper function to run the MIDI Shiny web application in the default web browser.

Usage

```
run_app()
```

Value

Nothing, but launches the Shiny app in the default web browser.

Examples

```
run_app()
```

simulate_bundle *Generates a cross section of a cylinder bundle*

Description

Generates a cross section of a cylinder bundle with a given density and voxel size. The cross section is generated by simulating a random distribution of cylinders and computing the intersection of the cylinders with a plane. The radii of the cylinders are drawn from a Gamma distribution fitted from data retrieved by histology in the genu of the corpus callosum (Aboitiz et al., 1992). The number of cylinders is determined by the density and the voxel size.

Usage

```
simulate_bundle(
  density = 0.5,
  voxel_size = 10,
  rel_tol = 0.001,
  verbose = FALSE
)
```

Arguments

density	A numeric value between 0 and 1 specifying the density of the cylinders in the voxel. Defaults to 0.5.
voxel_size	A numeric value specifying the size of the voxel in micro- meters. Defaults to 10.
rel_tol	A numeric value specifying the relative tolerance to reach the target volume defined as density * voxel_size^2. Defaults to 1e-3.
verbose	A logical value specifying whether to print messages. Defaults to FALSE.

Value

A list with the following components:

- sections: A numeric matrix with 3 columns:
  - x: The x-coordinates of the centers of the cylinders;
  - y: The y-coordinates of the centers of the cylinders;
  - r: The radii of the cylinders in micrometers.
- voxel_size: The size of the voxel in micrometers

References

Aboitiz, F., Scheibel, A. B., Fisher, R. S., & Zaidel, E. (1992). Fiber composition of the human corpus callosum. Brain research, 598(1-2), 143-153.

Examples

```
density <- 0.5
voxel_size <- 5 # micrometers
withr::with_seed(1234, {
  out <- simulate_bundle(density, voxel_size)
})

# Actual density in the simulated substrate
sum(out$sections[, "r"]^2 * pi) / voxel_size^2
```

SodermanCompartment *Soderman's model for restricted diffusion in a cylinder*

Description

A class to model restricted diffusion in a cylinder using the Soderman's model.

Super class

midi::CylinderRadialCompartment -> SodermanCompartment

Methods

Public methods:

- SodermanCompartment$clone()

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
SodermanCompartment$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

References

Söderman, O., & Jönsson, B. (1995). Restricted diffusion in cylindrical geometry. Journal of Magnetic Resonance, Series A, 117(1), 94-97.

`StaniszCompartment` *Stanisz's model for restricted diffusion in a cylinder*

Description

A class to model restricted diffusion in a cylinder using the Stanisz's model.

Super class

`midi::CylinderRadialCompartment` -> `StaniszCompartment`

Methods

Public methods:

- StaniszCompartment$clone()

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
StaniszCompartment$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

References

Stanisz, G. J., Wright, G. A., Henkelman, R. M., & Szafer, A. (1997). An analytical model of restricted diffusion in bovine optic nerve. Magnetic Resonance in Medicine, 37(1), 103-111.

VanGelderenCompartment *Van Gelderen's model for restricted diffusion in a cylinder*

Description

A class to model restricted diffusion in a cylinder using the Van Gelderen's model.

Super class

`midi::CylinderRadialCompartment` -> `VanGelderenCompartment`

Methods

Public methods:

- `VanGelderenCompartment$clone()`

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
VanGelderenCompartment$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

References

Vangelderen, P., DesPres, D., Vanzijl, P. C. M., & Moonen, C. T. W. (1994). Evaluation of restricted diffusion in cylinders. Phosphocreatine in rabbit leg muscle. Journal of Magnetic Resonance, Series B, 103(3), 255-260.

Index