# Package: dti (via r-universe)

October 29, 2026

**Version** 1.5.4.3

**Date** 2024-09-26

**Title** Analysis of Diffusion Weighted Imaging (DWI) Data

**Author** Karsten Tabelow [aut, cre], Joerg Polzehl [aut], Felix Anker [ctb]

**Maintainer** Karsten Tabelow <karsten.tabelow@wias-berlin.de>

**Depends** R (>= 3.5.0), awsMethods (>= 1.1-1)

**SystemRequirements** gsl

**Imports** methods, parallel, adimpro (>= 0.9), aws (>= 2.4.1), rgl, oro.nifti (>= 0.3.9), oro.dicom, gsl, quadprog

**LazyData** TRUE

**Description** Diffusion Weighted Imaging (DWI) is a Magnetic Resonance Imaging modality, that measures diffusion of water in tissues like the human brain. The package contains R-functions to process diffusion-weighted data. The functionality includes diffusion tensor imaging (DTI), diffusion kurtosis imaging (DKI), modeling for high angular resolution diffusion weighted imaging (HARDI) using Q-ball-reconstruction and tensor mixture models, several methods for structural adaptive smoothing including POAS and msPOAS, and a streamline fiber tracking for tensor and tensor mixture models. The package provides functionality to manipulate and visualize results in 2D and 3D.

**License** GPL (>= 2)

**Copyright** This package is Copyright (C) 2005-2020 Weierstrass Institute for Applied Analysis and Stochastics.

**URL** https://www.wias-berlin.de/research/ats/imaging/

**Suggests** covr

**RoxygenNote** 6.1.0

**NeedsCompilation** yes

**Config/pak/sysreqs** libgsl0-dev

**Repository** https://cran.r-universe.dev

**Date/Publication** 2024-09-26 12:10:03 UTC

**RemoteUrl** https://github.com/cran/dti

**RemoteRef** HEAD

**RemoteSha** 6461869b28b1b554cee5846f14916bd7bd5492c1

# Contents

dti-package . . . 3
AdjacencyMatrix . . . 5
awssigmc . . . 6
colqFA . . . 8
combineDWIdata . . . 9
dkiTensor-methods . . . 10
dti.options . . . 11
dti.smooth-methods . . . 12
dtiIndices-methods . . . 14
dtiTensor-methods . . . 15
dwi-class . . . 17
dwi.smooth-methods . . . 22
dwiMD-methods . . . 24
dwiMixtensor-methods . . . 25
dwiQball-methods . . . 27
dwiRiceBias-methods . . . 28
dwiSqrtODF-methods . . . 29
extract-methods . . . 30
getmask-methods . . . 32
getsdofsb-methods . . . 33
medinria . . . 34
optgrad . . . 35
optgradients . . . 35
plot-methods . . . 35
polyeder . . . 37
print-methods . . . 38
readDWIdata . . . 39
sdpar-methods . . . 41
setmask-methods . . . 42
show-methods . . . 43
show3d-methods . . . 43
showFAColorScale . . . 47
subsetg . . . 47
summary-methods . . . 48
tracking-methods . . . 49

**Index** **52**

dti-package *Analysis of Diffusion Weighted Imaging (DWI) Data*

## Description

Diffusion Weighted Imaging (DWI) is a Magnetic Resonance Imaging modality, that measures diffusion of water in tissues like the human brain. The package contains R-functions to process diffusion-weighted data. The functionality includes diffusion tensor imaging (DTI), diffusion kurtosis imaging (DKI), modeling for high angular resolution diffusion weighted imaging (HARDI) using Q-ball-reconstruction and tensor mixture models, several methods for structural adaptive smoothing including POAS and msPOAS, and a streamline fiber tracking for tensor and tensor mixture models. The package provides functionality to manipulate and visualize results in 2D and 3D.

## Details

The DESCRIPTION file:

| | |
|---|---|
| Package: | dti |
| Version: | 1.5.4.3 |
| Date: | 2024-09-26 |
| Title: | Analysis of Diffusion Weighted Imaging (DWI) Data |
| Authors@R: | c(person("Karsten", "Tabelow", role = c("aut", "cre"), email = "karsten.tabelow@wias-berlin.de"), pe |
| Author: | Karsten Tabelow [aut, cre], Joerg Polzehl [aut], Felix Anker [ctb] |
| Maintainer: | Karsten Tabelow <karsten.tabelow@wias-berlin.de> |
| Depends: | R (>= 3.5.0), awsMethods (>= 1.1-1) |
| SystemRequirements: | gsl |
| Imports: | methods, parallel, adimpro (>= 0.9), aws (>= 2.4.1), rgl, oro.nifti (>= 0.3.9), oro.dicom, gsl, quadprog |
| LazyData: | TRUE |
| Description: | Diffusion Weighted Imaging (DWI) is a Magnetic Resonance Imaging modality, that measures diffusi |
| License: | GPL (>= 2) |
| Copyright: | This package is Copyright (C) 2005-2020 Weierstrass Institute for Applied Analysis and Stochastics. |
| URL: | https://www.wias-berlin.de/research/ats/imaging/ |
| Suggests: | covr |
| RoxygenNote: | 6.1.0 |

Index of help topics:


| | |
|---|---|
| AdjacencyMatrix | Create an adjacency matrix from fiber tracking results |
| awssigmc | Estimate noise variance for multicoil MR systems |
| colqFA | FA map color scheme |
| combineDWIdata | Combine two objects of class "dtiData") |
| dkiTensor-methods | Diffusion Kurtosis Imaging (DKI) |
| dti-package | Analysis of Diffusion Weighted Imaging (DWI) |

| | |
|---|---|
| | Data |
| dti.options | Set and manipulate image orientations for plots. |
| dti.smooth-methods | Methods for Function 'dti.smooth' in Package 'dti' |
| dtiIndices-methods | Methods for Function 'dtiIndices' in Package 'dti' |
| dtiTensor-methods | Methods for Function 'dtiTensor' in Package 'dti' |
| dwi-class | Class "dwi" |
| dwi.smooth-methods | Smooth DWI data |
| dwiMD | Methods for Mean Diffusivity in Package 'dti' |
| dwiMixtensor-methods | Methods for Function 'dwiMixtensor' in Package 'dti' |
| dwiQball-methods | Methods for Function 'dwiQball' in Package 'dti' |
| dwiRiceBias-methods | Correction for Rician Bias |
| dwiSqrtODF-methods | Methods for positive definite EAP and ODF estimation in Package 'dti' |
| extract-methods | Methods for Function 'extract' and '[' in Package 'dti' |
| getmask-methods | Methods for Function 'getmask' in Package 'dti' |
| getsdofsb-methods | Estimate the noise standard deviation |
| medinria | Read/Write Diffusion Tensor Data from/to NIFTI File |
| optgrad | Optimal gradient directions |
| optgradients | Optimal gradient directions for number of gradients between 6 and 162 |
| plot-methods | Methods for Function 'plot' in Package 'dti' |
| polyeder | Polyeders derived from the Icosahedron (icosa0) by sequential triangulation of surface triangles |
| print-methods | Methods for Function 'print' in Package 'dti' |
| readDWIdata | Read Diffusion Weighted Data |
| sdpar-methods | Methods for Function 'sdpar' in Package 'dti' |
| setmask-methods | Methods for Function 'setmask' in Package 'dti' |
| show-methods | Methods for Function 'show' in Package 'dti' |
| show3d-methods | Methods for Function 'show3d' in Package 'dti' |
| showFAColorScale | Writes an image with the colqFA colorscale to disk. |
| subsetg | Create an objects of class "dtiData" containing only a subset of gradient directions. |
| summary-methods | Methods for Function 'summary' in Package 'dti' |
| tracking-methods | Methods for Function 'tracking' in Package 'dti' |

## Author(s)

Karsten Tabelow [aut, cre], Joerg Polzehl [aut], Felix Anker [ctb]

Maintainer: Karsten Tabelow <karsten.tabelow@wias-berlin.de>

## References

J. Polzehl, K. Tabelow (2019). Magnetic Resonance Brain Imaging: Modeling and Data Analysis Using R. Springer, Use R! series. Doi:10.1007/978-3-030-29184-6.

S. Mohammadi, K. Tabelow, L. Ruthotto, Th. Feiweier, J. Polzehl, and N. Weiskopf, *High-resolution diffusion kurtosis imaging at 3T enabled by advanced post-processing*, 8 (2015), 427.

S. Becker, K. Tabelow, S. Mohammadi, N. Weiskopf, and J. Polzehl, *Adaptive smoothing of multi-shell diffusion weighted magnetic resonance data by msPOAS*, NeuroImage 95 (2014), pp. 90-105.

S. Becker, K. Tabelow, H.U. Voss, A. Anwander, R.M. Heidemann and J. Polzehl, *Position-orientation adaptive smoothing of diffusion weighted magnetic resonance data (POAS)*, Medical Image Analysis, 16 (2012), pp. 1142-1155.

J. Polzehl and K. Tabelow, *Beyond the diffusion tensor model: The package* **dti**, Journal of Statistical Software, 44 no. 12 (2011) pp. 1-26.

K. Tabelow, H.U. Voss and J. Polzehl, *Modeling the orientation distribution function by mixtures of angular central Gaussian distributions*, Journal of Neuroscience Methods, 203 (2012), pp. 200-211.

J. Polzehl and K. Tabelow, *Structural adaptive smoothing in diffusion tensor imaging: The R package dti*, Journal of Statistical Software, 31 (2009) pp. 1–24.

K. Tabelow, J. Polzehl, V. Spokoiny and H.U. Voss. *Diffusion Tensor Imaging: Structural adaptive smoothing*, NeuroImage 39(4), 1763-1773 (2008).

## See Also

fmri aws oro.nifti

## Examples

```
## Not run: demo(dti_art)
## Not run: demo(mixtens_art)
```

---

`AdjacencyMatrix` *Create an adjacency matrix from fiber tracking results*

---

## Description

The function takes two objects, `fiberobj` with class 'dwiFiber' containing fiber tracking results and an array or nifti-object containing atlas information. For each combination of regions defined in the atlas the number of fibers connecting these regions is calculated, resulting in a matrix of fiber counts. As default this matrix is standardized and the diagonal elements are set to zero.

## Usage

```
AdjacencyMatrix(fiberobj, atlas, labels = NULL,
  method = c("standardize", "counts"), diagelements = FALSE,
  symmetric=TRUE, verbose = FALSE)
```

### Arguments

| | |
|---|---|
| fiberobj | an object of class 'dwiFiber' |
| atlas | an object of class 'array' or 'nifti' containing region indices as intensities. The atlas needs to be registered to DWI (subject) space, with array dimension corresponding to fiberobj@ddim |
| labels | optional labels for the regions. Will be used as dimnames of the resulting matrix. |
| method | either "standardize" or "counts", determines if fiber counts or a standardized (default) matix is returned. |
| diagelements | logical, if FALSE the diagonal elements of the standardized matrix are set to zero (default). |
| symmetric | logical, with ni the number of fibers originating if FALSE standardized values counts(i,j)/ni, if TRUE we get counts(i,j)/sqrt(nj*nj). |
| verbose | logical, if TRUE report pairwise fiber counts. |

### Value

A matrix with dimensions equal to the number of regions defined in the atlas and dimnames given by labels or by the region number. The matrix contains fiber counts or values standardized with the number of fibers ni, nj originating/ending from the pair of regions. Depending on symmetric standardization is with 1/sqrt(ni*nj) or with 1/ni.

### Author(s)

Joerg Polzehl <polzehl@wias-berlin.de>

### See Also

dwiFiber

---

## awssigmc *Estimate noise variance for multicoil MR systems*

---

### Description

The distribution of image intensity values $S_i$ divided by the noise standard deviation in $K$-space $\sigma$ in dMRI experiments is assumed to follow a non-central chi-distribution with $2L$ degrees of freedom and noncentrality parameter $\eta$, where $L$ refers to the number of receiver coils in the system and $\sigma\eta$ is the signal of interest. This is an idealization in the sense that each coil is assumed to have the same contribution at each location. For realistic modeling $L$ should be a locally smooth function in voxel space that reflects the varying local influence of the receiver coils in the the reconstruction algorithm used.

The functions assume $L$ to be known and estimate either a local (function awslsigmc) or global ( function awssigmc) $\sigma$ employing an assumption of local homogeneity for the noncentrality parameter $\eta$.

Function afsigmc implements estimates from Aja-Fernandez (2009). Function aflsigmc implements the estimate from Aja-Fernandez (2013).

## Usage

```
awssigmc(y, steps, mask = NULL, ncoils = 1, vext = c(1, 1), lambda = 20,
         h0 = 2, verbose = FALSE, sequence = FALSE, hadj = 1, q = 0.25,
         qni = .8, method=c("VAR","MAD"))
awslsigmc(y, steps, mask = NULL, ncoils = 1, vext = c(1, 1), lambda = 5, minni = 2,
         hsig = 5, sigma = NULL, family = c("NCchi"), verbose = FALSE,
         trace=FALSE, u=NULL)
afsigmc(y, level = NULL, mask = NULL,  ncoils = 1,   vext = c( 1, 1),
        h = 2, verbose = FALSE, hadj = 1,
        method = c("modevn","modem1chi","bkm2chi","bkm1chi"))
aflsigmc(y, ncoils, level = NULL, mask = NULL, h=2, hadj=1,  vext = c( 1, 1))
```

## Arguments

| | |
|---|---|
| y | 3D array, usually obtained from an object of class dwi as obj@si[,,,i] for some i, i.e. one 3D image from an dMRI experiment. |
| steps | number of steps in adapive weights smoothing, used to reveal the unerlying mean structure. |
| mask | restrict computations to voxel in mask, if is.null(mask) all voxel are used. In function afsigmc mask should refer to background for method %in% c("modem1chi","bkm2chi","bkm1c and to voxel within the head for method=="modevn". |
| ncoils | number of coils, or equivalently number of effective degrees of freedom of non-central chi distribution divided by 2. |
| vext | voxel extentions |
| lambda | scale parameter in adaptive weights smoothing |
| h0 | initial bandwidth |
| verbose | if verbose==TRUE density plots and quantiles of local estimates of sigma are provided. |
| trace | if trace==TRUE intermediate results for each step are returned in component tergs for all voxel in mask. |
| sequence | if sequence=TRUE a vector of estimates for the noise standard deviation sigma for the individual steps is returned instead of the final value only. |
| hadj | adjustment factor for bandwidth (chosen by bw.nrd) in mode estimation |
| q | quantile to be used for interquantile-differences. |
| qni | quantile of distribution of actual sum of weights $N_i = \sum_j w_{ij}$ in adaptive smoothing. Only voxel i with $N_i > q_{qni}(N_.)$ are used for variance estimation. Should be larger than 0.5. |
| method | in case of function awssigmc the method for variance estimation, either "VAR" (variance) or "MAD" (mean absolute deviation). In function afsigmc see last column in Table 2 in Aja-Fernandez (2009). |
| level | threshold for background separation. Used if !is.null(level) to redefine mask |
| h | bandwidth for local avaeraging |

| | |
|---|---|
| minni | Minimum sum of weights for updating values of sigma. |
| hsig | Bandwidth of the median filter. |
| sigma | Initial estimate for sigma |
| family | One of "Gauss" or "NCchi" (default) defining the probability distribution to use. |
| u | if verbose==TRUE an array of noncentrality paramters for comparisons. Internal use for tests only |

## Value

a list with components

| | |
|---|---|
| sigma | either a scalar or a vector of estimated noise standard deviations. |
| theta | the estimated mean structure |

## Author(s)

J\"org Polzehl <polzehl@wias-berlin.de>

## References

K. Tabelow, H.U. Voss, J. Polzehl, Local estimation of the noise level in MRI using structural adaptation, Medical Image Analysis, 20 (2015), pp. 76–86.

---

# colqFA *FA map color scheme*

---

## Description

Color map implementing the FA color scheme develop at Uniklinikum Muenster (M. Deppe)

## Usage

```
colqFA
```

## Format

A vector with 256 RGB color values.

`combineDWIdata` *Combine two objects of class "dtiData")*

## Description

This function creates a dtiData-object from two compatible dtiData-objects. Compatible means that the spatial dimensions coincide, but gradients and b-values may be different.

## Usage

```
combineDWIdata(x1, x2, s0strategy = "first")
```

## Arguments

| | |
|---|---|
| `x1` | Object of class `"dtiData"` |
| `x2` | Object of class `"dtiData"` |
| `s0strategy` | Character, determines how the unweighted S0 images are handled. Six strategies are implemented. `s0strategy="first"` copies the S0 images from object `x1`, `s0strategy="second"` copies the S0 images from object `x2`, `s0strategy="both"` used the S0 images from both objects. `s0strategy="rfirst"` creates one average S0 image from object `x1`, `s0strategy="rsecond"` creates one average S0 image from object `x2`, `s0strategy="rboth"` creates one average S0 image from the S0 images in both objects. |

## Details

The function can be used to merge two objects of class "dtiData" under the condition that the information in slot `ddim` in both objects is identical. Also slots `voxelext`, `orientation` and `rotation` should be indentical.

## Value

An object of class "dtiData".

## Author(s)

Karsten Tabelow <tabelow@wias-berlin.de>
J\"org Polzehl <polzehl@wias-berlin.de>

## See Also

`dtiData`, `readDWIdata`, `dtiData`, `subsetg`

dkiTensor-methods *Diffusion Kurtosis Imaging (DKI)*

## Description

These methods estimate, in each voxel, the diffusion kurtosis tensor (and the diffusion tensor) and some scalar indices.

## Usage

```
  ## S4 method for signature 'dtiData'
dkiTensor(object, method=c("CLLS-QP", "CLLS-H", "ULLS", "QL", "NLR"),
                  sigma=NULL, L=1, mask=NULL,
                  mc.cores=setCores(, reprt=FALSE), verbose=FALSE)
  ## S4 method for signature 'dkiTensor'
dkiIndices(object, mc.cores=setCores(, reprt=FALSE),
                  verbose=FALSE)
```

## Arguments

| | |
|---|---|
| object | Object of class "dtiData" |
| method | Method for tensor estimation. May be "CLLS-QP" for a qudratic programm solution for the constrained optimization (requires package quadprog), "CLLS-H" for a heuristic approximation described in Tabesh et al. (2011), or "ULLS" for an unconstrained linear least squares estimation. "QL" and "NLR" correspond to the use of unconstrained quasi-likelihood and nonlinear regression, respectively. |
| sigma | Scale parameter of intensity distribution (unprocessed). Used with method="QL" in the calculation of the expected intensity values. |
| L | Effective number of coils, 2*L are the degrees of freedom of the intensity distribution (unprocessed). The default corresponds, e.g., to the case of a SENSE reconstruction. Used with method="QL" in the calculation of the expected intensity values. |
| mask | argument to specify a precomputed brain mask |
| mc.cores | Number of cores to use. Defaults to number of threads specified for openMP, see documentation of package **awsMethods**. Not yet fully implemented for these methods. |
| verbose | Verbose mode. |

## Value

An object of class "dkiTensor" or "dkiIndices".

### Methods

`signature(object = "ANY")` Returns a warning

`signature(object = "dtiData")` The method `"dkiTensor"` estimates the diffusion kurtosis model, i.e., the kurtosis tensor and the diffusion tensor.

`signature(object = "dkiTensor")` The method `"dkiIndices"` estimates some scalar indices from the kurtosis tensor. The method is still experimental, some quantities may be removed in future versions, other might be included.

### Author(s)

Karsten Tabelow <tabelow@wias-berlin.de>

### References

A. Tabesh, J.H. Jensen, B.A. Ardekani, and J.A. Helpern, *Estimation of tensors and tensor-derived measures in diffusional kurtosis imaging*, Magnetic Resonance in Medicine, 65, 823-836 (2011).

E.S. Hui, M.M. Cheung, L. Qi, and E.X. Wu, *Towards better MR characterization of neural tissues using directional diffusion kurtosis analysis*, Neuroimage, 42, 122-134 (2008).

J. Polzehl, K. Tabelow (2019). Magnetic Resonance Brain Imaging: Modeling and Data Analysis Using R. Springer, Use R! series. Doi:10.1007/978-3-030-29184-6.

https://www.wias-berlin.de/projects/matheon_a3/

### See Also

`dtiData`, `readDWIdata`, `dtiData`, `dkiTensor` `dkiIndices`

---

## dti.options *Set and manipulate image orientations for plots.*

---

### Description

The function can be used to adjust to radiological conventions in image displays.

### Usage

```
dti.options(...)
```

### Arguments

| | |
|---|---|
| `...` | The following parameters can be used to determine the behaviour of the plot method for 3D image data in subsequent calls:<br>• swapx - swap image x axis for display<br>• swapy - swap image y axis for display<br>• swapz - swap image z axis for display<br>all default to FALSE. |

## Value

returns specified display orientations.

## Author(s)

Joerg Polzehl <polzehl@wias-berlin.de>

---

`dti.smooth-methods` *Methods for Function 'dti.smooth' in Package 'dti'*

---

## Description

The function provides structural adaptive smoothing for diffusion weighted image data within the context of an diffusion tensor (DTI) model. It implements smoothing of DWI data using a structural assumption of a local (anisotropic) homogeneous diffusion tensor model (in case a `"dtiData"`-object is provided). It also implements structural adaptive smoothing of a diffusion tensor using a Riemannian metric (in case a `"dtiTensor"`-object is given), although we strictly recommend to use the first variant due to methodological reasons.

## Usage

```
## S4 method for signature 'dtiData'
dti.smooth(object, hmax=5, hinit=NULL, lambda=20, tau=10, rho=1,
         graph=FALSE,slice=NULL, quant=.8, minfa=NULL, hsig=2.5,
         lseq=NULL, method="nonlinear", rician=TRUE,
         niter=5,result="Tensor")
```

## Arguments

| | |
|---|---|
| `object` | Either an object of class `"dtiData"` or an object of class `"dtiTensor"` |
| `hmax` | Maximal bandwidth |
| `hinit` | Initial bandwidth (default 1) |
| `lambda` | Critical parameter (default 20) |
| `tau` | Critical parameter for orientation scores (default 10) |
| `rho` | Regularization parameter for anisotropic vicinities (default 1) |
| `graph` | "logical": Visualize intermediate results (default FALSE) |
| `slice` | slice number, determines the slice used in visualization |
| `quant` | determines `minfa` as corresponding quantile of FA if `is.null(minfa)` |
| `minfa` | minimal anisotropy index (FA) to use in visualization |
| `hsig` | bandwidth for presmoothing of variance estimates |
| `lseq` | sequence of correction factors for `lambda` |
| `method` | Method for tensor estimation. May be `"linear"`, `"nonlinear"` |

| | |
|---|---|
| `rician` | "logical": apply a correction for Rician bias. This is still experimental and depends on spatial independence of errors. |
| `niter` | Maximum number of iterations for tensor estimates using the nonlinear model. |
| `result` | Determines the created object. Alternatives are `"Tensor"` for create a dtiTensor-object and `"dtiData"` for a dtiData-object containing a smoothed data cube. |

## Value

An object of class `dtiTensor`.

## Methods

**object = "ANY"** Returns a warning.

**object = "dtiData"** We highly recommend to use the method `dti.smooth` on DWI data directly, i.e. on an object of class `"dtiData"`, due to methodological reasons, see Tabelow et al. (2008). It is usually not necessary to use any other argument than `hmax`, which defines the maximum bandwidth of the iteration.

If `model=="linear"` estimates are obtained using a linearization of the tensor model. This was the estimate used in Tabelow et.al. (2008). `model=="nonlinear"` uses a nonlinear regression model with reparametrization that ensures the tensor to be positive semidefinite, see Koay et.al. (2006). If `varmethod=="replicates"` the error variance is estimated from replicated gradient directions if possible, otherwise (default) an estimate is obtained from the residual sum of squares. If `volseq==TRUE` the sum of location weights is fixed to $1.25^k$ within iteration $k$ (does not depend on the actual tensor). Otherwise the ellipsoid of positive location weights is determined by a bandwidth $h_k = 1.25^(k/3)$.

## Author(s)

Karsten Tabelow <tabelow@wias-berlin.de>
J\"org Polzehl <polzehl@wias-berlin.de>

## References

J. Polzehl and K. Tabelow, *Beyond the diffusion tensor model: The package* **dti**, Journal of Statistical Software, to appear.

K. Tabelow, H.U. Voss and J. Polzehl, *Modeling the orientation distribution function by mixtures of angular central Gaussian distributions*, Journal of Neuroscience Methods, to appear.

J. Polzehl and K. Tabelow, *Structural adaptive smoothing in diffusion tensor imaging: The R package dti*, Journal of Statistical Software, 31 (2009) pp. 1–24.

K. Tabelow, J. Polzehl, V. Spokoiny and H.U. Voss. *Diffusion Tensor Imaging: Structural adaptive smoothing*, NeuroImage 39(4), 1763-1773 (2008).

https://www.wias-berlin.de/projects/matheon_a3/

## See Also

`dtiData`, `readDWIdata`, `dtiTensor-methods`, `dtiIndices-methods`, `medinria` , `dtiData`, `dtiTensor`, `dtiIndices`

`dtiIndices-methods` *Methods for Function 'dtiIndices' in Package 'dti'*

## Description

The method creates estimates of the fractional anisotropy (FA) and relative anisotropy (RA) indices, the main directions of anisotropy and several statistics used for visualization.

## Usage

```
  ## S4 method for signature 'dtiTensor'
dtiIndices(object, mc.cores = setCores(,reprt=FALSE))
```

## Arguments

| | |
|---|---|
| `object` | Object of class `"dtiTensor"` |
| `mc.cores` | Number of cores to use. Defaults to number of threads specified for openMP, see documentation of package **awsMethods**. Our experience suggests to use 4-6 cores if available. |

## Value

An object of class `"dtiIndices"`.

## Methods

**obj = "ANY"** Returns a warning.

**obj = "dtiTensor"** Estimate tensor indices like trace, fractional and geodesic anisotropy, main diffusion direction and shape parameters.

## Author(s)

Karsten Tabelow <tabelow@wias-berlin.de>
J\"org Polzehl <polzehl@wias-berlin.de>

## References

J. Polzehl, K. Tabelow (2019). Magnetic Resonance Brain Imaging: Modeling and Data Analysis Using R. Springer, Use R! series. Doi:10.1007/978-3-030-29184-6.

J. Polzehl and K. Tabelow, *Beyond the diffusion tensor model: The package* **dti**, Journal of Statistical Software, to appear.

K. Tabelow, H.U. Voss and J. Polzehl, *Modeling the orientation distribution function by mixtures of angular central Gaussian distributions*, Journal of Neuroscience Methods, to appear.

J. Polzehl and K. Tabelow, *Structural adaptive smoothing in diffusion tensor imaging: The R package dti*, Journal of Statistical Software, 31 (2009) pp. 1–24.

K. Tabelow, J. Polzehl, V. Spokoiny and H.U. Voss. *Diffusion Tensor Imaging: Structural adaptive smoothing*, NeuroImage 39(4), 1763-1773 (2008).

https://www.wias-berlin.de/projects/matheon_a3/

## See Also

medinria, dtiTensor-methods, dtiTensor, dtiIndices

## Examples

```
## Not run: demo(dti_art)
```

---

# dtiTensor-methods *Methods for Function 'dtiTensor' in Package 'dti'*

---

## Description

The method estimates, in each voxel, the diffusion tensor from the DWI data contained in an object of class "dtiData".

## Usage

```
  ## S4 method for signature 'dtiData'
dtiTensor(object, method=c( "nonlinear", "linear", "quasi-likelihood"),
          sigma = NULL, L = 1, mask=NULL, mc.cores = setCores( , reprt = FALSE))
```

## Arguments

| | |
|---|---|
| object | Object of class "dtiData" |
| method | Method for tensor estimation. May be "linear", or "nonlinear". method=="quasi-likelihood" solves the nonlinear regression problem with the expected value of the signal as regression function and weighting according to the signal variance. |
| sigma | (local) scale parameter of the signal's distribution. |
| L | (local) effective degrees of freedom. |
| mask | argument to specify a precomputed brain mask |
| mc.cores | Number of cores to use. Defaults to number of threads specified for openMP, see documentation of package **awsMethods**. Our experience suggests to use 4-6 cores if available. |

## Value

An object of class "dtiTensor".

## Methods

**obj = "ANY"** Returns a warning.

**obj = "dtiData"** Estimate diffusion tensor from data in each voxel with the different options for the regression type and model for variance estimation. If `method=="linear"` estimates are obtained using a linearization of the tensor model. This was the estimate used in Tabelow et.al. (2008). `method=="nonlinear"` uses a nonlinear regression model with reparametrization that ensures the tensor to be positive semidefinite, see Koay et.al. (2006). The imlementation is based on R's internal C code for the BFGS optimization. `method=="quasi-likelihood"` solves the nonlinear regression problem with the expected value of the signal as regression function and weighting according to the signal variance. Tis requires additional parameters `sigma` and `L` characterizing the distribution of the signal. If `varmethod=="replicates"` the error variance is estimated from replicated gradient directions if possible, otherwise an estimate is obtained from the residual sum of squares. If `varmodel=="global"` a homogeneous variance is assumed and estimated as the median of the local variance estimates. `sigma` and `2*L` are the scale parameter and degrees of freedom of the (local) signal distribution. `L` characterizes the effective number of coils. Both parameters are either scalars or arrays of the size of the images.

## Author(s)

Karsten Tabelow <tabelow@wias-berlin.de>
J\"org Polzehl <polzehl@wias-berlin.de>

## References

J. Polzehl and K. Tabelow, *Beyond the diffusion tensor model: The package* **dti**, Journal of Statistical Software, 44(12), 1-26 (2011).

K. Tabelow, H.U. Voss and J. Polzehl, *Modeling the orientation distribution function by mixtures of angular central Gaussian distributions*, Journal of Neuroscience Methods, 203(1), 200-211 (2012).

J. Polzehl and K. Tabelow, *Structural adaptive smoothing in diffusion tensor imaging: The R package dti*, Journal of Statistical Software, 31(9) 1-24 (2009).

K. Tabelow, J. Polzehl, V. Spokoiny and H.U. Voss. *Diffusion Tensor Imaging: Structural adaptive smoothing*, NeuroImage 39(4), 1763-1773 (2008).

C.G. Koay, J.D. Carew, A.L. Alexander, P.J. Basser and M.E. Meyerand. *Investigation of Anomalous Estimates of Tensor-Derived Quantities in Diffusion Tensor Imaging*, Magnetic Resonance in Medicine, 2006, 55, 930-936.

J. Polzehl, K. Tabelow (2019). Magnetic Resonance Brain Imaging: Modeling and Data Analysis Using R. Springer, Use R! series. Doi:10.1007/978-3-030-29184-6.

https://www.wias-berlin.de/projects/matheon_a3/

## See Also

`dtiData`, `readDWIdata`, `dtiIndices-methods`, `medinria`, `dtiData`, `dtiTensor` `dwiMixtensor`

## Examples

```
## Not run: demo(dti_art)
```

dwi-class *Class "dwi"*

## Description

The family of "dwi" classes is used for Diffusion Weighted Imaging (DWI) data and, within the Diffusion Tensor Model (DTI), diffusion tenors and its indices.

## Objects from the Class

"dwi" is only a superclass, no instances should be created. However, objects can be created by calls of the form new("dwi", ...). "dtiData", "dtiTensor", and "dtiIndices" can be created from their correspondingly named functions and methods.

## Slots

.Data: Object of class "list", usually empty.

gradient: Object of class "matrix", matrix of dimension c(3,ngrad) containing gradient directions.

btb: Object of class "matrix", matrix of dimension c(6,ngrad) obtained from gradient directions.

bvalue: Object of class "numeric", of length ngrad containing b-values if available.

ngrad: Object of class "integer", number of gradients (including zero gradients).

s0ind: Object of class "integer", index of zero gradients within the sequence 1:ngrad.

replind: Object of class "integer", index (identifier) of unique gradient directions. Used to characterize replications in the gradient design by identical indices. length is ngrad.

ddim: Object of class "integer", dimension of subcube defined by xind, yind and zind.

ddim0: Object of class "integer", dimension of original image cubes. Vector of length 3.

xind, yind, zind: Objects of class "integer", index for subcube definition in x-, y- and z-direction.

voxelext: Object of class "numeric", voxel extensions in x-, y- and z-direction. Vector of length 3.

orientation: Object of class "integer", orientation of data according to AFNI convention. Vector of length 3.

rotation: Object of class "matrix", optional rotation matrix for gradient directions.

level: Object of class "numeric", minimal valid S0-level. No evaluation will be performed for voxels with S0-values less than level.

source: Object of class "character", name of the source imgage file or source directory.

call: Object of class "call", call that created the object.

For class "dtiData":

si: Object of class "array", Diffusion Weighted Data.

sdcoef: Object of class "numeric", Parameters of the model for error standard deviation as a function of the mean. First two entries refer to intercept and slope of a linear function, third and fourth value are the endpoints of the interval of linearity. Contains rep(0,4) if not set. If the function

For class "dtiTensor":

D: Object of class "array", estimated tensors, dimension c(6,ddim). Tensors are stored as upper diagonal matrices.

th0: Object of class "array", estimated intensities in S0 images, dimension ddim

sigma: Object of class "array", estimated error variances if method=="linear", zero otherwise.

scorr: Object of class "numeric", estimated spatial correlations in coordinate directions

bw: Object of class "numeric", bandwidth for a Gaussian kernel that approximately creates the estimated spatial correlations. Needed for adjustments of critical values in the adaptive smoothing algorithm used in function dti.smooth

mask: Object of class "array", logical indicating the voxel where the tensor was estimated.

hmax: Object of class "numeric", maximal bandwidth in case of adaptive smoothing, 1 otherwise.

outlier: Object of class "numeric", index of voxel where physical constraints are not met, i.e. where the observed values in gradient images Si were larger than the corresponding S0 values. These are probably motion effects or registration errors. Values are replaced by the corresponding (mean) S0 values.

scale: Numerical value corresponding to the 95% quantile of the maximal eigenvalues of estimated tensors within the mask. Used for scaling in function show3d.dtiTensor

method: Object of class "character", either "linear" or "nonlinear" or "unknown". Indicates the regression model used for estimating the tensors.

For class "dtiIndices":

fa: Object of class "array", Fractional anisotropy values (FA)

ga: Object of class "array", Geodetic anisotropy values (GA)

md: Object of class "array", Mean diffusivity values (MD)

andir: Object of class "array", Main directions of anisotropy

bary: Object of class "array", Shape parameters

method: Object of class "character" either "linear" or "nonlinear" or "unknown". Indicates the regression model used for estimating the tensors.

For class "dkiTensor":

D: Object of class "array", estimated tensors, dimension c(6,ddim). Tensors are stored as upper diagonal matrices.

W: Object of class "array", estimated kurtosis tensors, dimension c(15,ddim).

th0: Object of class "array", estimated intensities in S0 images, dimension ddim

sigma: Object of class "array", estimated error variances if method=="linear", zero otherwise.

scorr: Object of class "numeric", estimated spatial correlations in coordinate directions

`bw`: Object of class `"numeric"`, bandwidth for a Gaussian kernel that approximately creates the estimated spatial correlations. Needed for adjustments of critical values in the adaptive smoothing algorithm used in function `dti.smooth`

`mask`: Object of class `"array"`, logical indicating the voxel where the tensor was estimated.

`hmax`: Object of class `"numeric"`, maximal bandwidth in case of adaptive smoothing, 1 otherwise.

`outlier`: Object of class `"numeric"`, index of voxel where physical constraints are not met, i.e. where the observed values in gradient images Si were larger than the corresponding S0 values. These are probably motion effects or registration errors. Values are replaced by the corresponding (mean) S0 values.

`scale`: Numerical value corresponding to the 95% quantile of the maximal eigenvalues of estimated tensors within the mask. Used for scaling in function `show3d.dtiTensor`

`method`: Object of class `"character"`, either `"linear"` or `"nonlinear"` or `"unknown"`. Indicates the regression model used for estimating the tensors.

For class `"dkiIndices"`:

`fa`: Object of class `"array"`, Fractional anisotropy values (FA)

`ga`: Object of class `"array"`, Geodetic anisotropy values (GA)

`md`: Object of class `"array"`, Mean diffusivity values (MD)

`andir`: Object of class `"array"`, Main directions of anisotropy

`bary`: Object of class `"array"`, Shape parameters

`k1`: Object of class `"array"`, Kurtosis along DT (Hui et al. 2008)

`k2`: Object of class `"array"`, Kurtosis along DT (Hui et al. 2008)

`k3`: Object of class `"array"`, Kurtosis along DT (Hui et al. 2008)

`mk`: Object of class `"array"`, Mean kurtosis (Hui et al. 2008)

`mk2`: Object of class `"array"`, Mean Kurtosis (Tabesh et al. (2011))

`kaxial`: Object of class `"array"`, Axial kurtosis (Hui et al. 2008)

`kradial`: Object of class `"array"`, Radial kurtosis (Hui et al. 2008)

`fak`: Object of class `"array"`, Kurtosis anisotropy (Hui et al. 2008)

`method`: Object of class `"character"` either `"linear"` or `"nonlinear"` or `"unknown"`. Indicates the regression model used for estimating the tensors.

For class `"dwiQball"`:

`order`: Object of class `"integer"`, maximal order of Spherical Harmonics to use, needs to be even.

`forder`: Object of class `"integer"`, maximal order Gaussian-Laguerre functions in SPF basis (for EAP estimation)

`zeta`: Object of class `"numeric"`, Scale parameter used in Gaussian-Laguerre functions (for EAP estimation)

`lambda`: Object of class `"numeric"`, nonnegative regularization parameter.

`sphcoef`: Object of class `"array"`, estimated coefficients for spherical harmonics, dimension `c((order+1)*(order+2)/2,ddim)`.

`sigma`: Object of class `"array"`, estimated error variances if `method=="linear"`, zero otherwise.

`scorr`: Object of class `"numeric"`, estimated spatial correlations in coordinate directions

`bw`: Object of class `"numeric"`, bandwidth for a Gaussian kernel that approximately creates the estimated spatial correlations. Needed for adjustments of critical values in the adaptive smoothing algorithm used in function `dti.smooth`

`mask`: Object of class `"array"`, logical indicating the voxel where the tensor was estimated.

`hmax`: Object of class `"numeric"`, maximal bandwidth in case of adaptive smoothing, 1 otherwise.

`outlier`: Object of class `"numeric"`, index of voxel where physical constraints are not met, i.e. where the observed values in gradient images Si were larger than the corresponding S0 values. These are probably motion effects or registration errors. Values are replaced by the corresponding (mean) S0 values.

`scale`: Numerical value corresponding to the 95% quantile of the maximal eigenvalues of estimated tensors within the mask. Used for scaling in function `show3d.dwiQball`

`what`: Object of class `"character"`, `"ODF"`, `"wODF"`, `"aODF"` or `"ADC"`. Indicates if the object contains coefficients of the orientation density function (ODF (Descoteaux 2007), wODF (Sapiro(2009) or aODF) or the apparent diffusion coefficient (ADC). Coefficients are computed with respect to spherical harmonics of the specified order.

For class `"dwiFiber"`:

`fibers`: Object of class `"matrix"`, Matrix of fibers. The first three columns contain the coordinates of the track points, the last three columns the direction vectors for each of these points.

`startind`: Object of class `"integer"`, indices for the first dimension of `fibers` where coordinates for a new fiber start.

`roix`: Object of class `"integer"`, coordinate range of region of interest in x-direction

`roiy`: Object of class `"integer"`, coordinate range of region of interest in x-direction

`roiz`: Object of class `"integer"`, coordinate range of region of interest in x-direction

`method`: Object of class `"character"`, fiber tracking method.

`minfa`: Object of class `"numeric"`, minimal fractional anisotropy index

`maxangle`: Object of class `"numeric"`, maximal angle between fibres.

For class `"dwiMixtensor"`:

`model`: Object of class `"character"`, characterizes the type of the mixed tensor model. Currently the only implemented model is `model="homogeneous_prolate"`.

`ev`: Object of class `"array"`, estimated eigenvalues, dimension `c(2,ddim)`

`mix`: Object of class `"array"`, estimated mixture coefficients, dimension `c(nmix,ddim)`. `nmix` is the number of mixture components specified.

`orient`: Object of class `"array"`, estimated tensor orientations, dimension `c(2,nmix,ddim)`

`th0`: Object of class `"array"`, estimated intensities in S0 images, dimension `ddim`

`sigma`: Object of class `"array"`, estimated error variances if `method=="linear"`, zero otherwise.

`scorr`: Object of class `"numeric"`, estimated spatial correlations in coordinate directions

bw: Object of class "numeric", bandwidth for a Gaussian kernel that approximately creates the estimated spatial correlations. Needed for adjustments of critical values in the adaptive smoothing algorithm used in function dti.smooth

mask: Object of class "array", logical indicating the voxel where the tensor was estimated.

hmax: Object of class "numeric", maximal bandwidth in case of adaptive smoothing, 1 otherwise.

outlier: Object of class "numeric", index of voxel where physical constraints are not met, i.e. where the observed values in gradient images Si were larger than the corresponding S0 values. These are probably motion effects or registration errors. Values are replaced by the corresponding (mean) S0 values.

scale: Numerical value corresponding to the 95% quantile of the maximal eigenvalues of estimated tensors within the mask. Used for scaling in function show3d.dtiTensor

method: Object of class "character", either "mixtensor" or "Jian". Indicates the regression model used for estimating the tensors.

## Methods

Methods only operate on subclasses "dtiData", "dtiTensor", "dtiIndices", "dwiQball" and "dwiFiber".

**dti.smooth** Create estimates of diffusion tensors in each voxel using structural adaptive spatial smoothing.

**dtiTensor** signature(object = "dtiData"): Create estimates of diffusion tensors in each voxel.

**dtiIndices** signature(object = "dtiTensor"): Create estimates of diffusion tensors indices in each voxel.

**tracking** signature(object = "dtiTensor") or signature(object = "dtiIndices"): Fiber tracking.

**dtiQball** signature(object = "dtiData"): Create estimates of ADC-parameters with respect to a sherical harmonics ortho-normal system.

**show3d** Method for Function 'show3d' in Package 'dti'.

**plot** Method for Function 'plot' in Package 'dti'.

**print** Method for Function 'print' in Package 'dti'.

**summary** Method for Function 'summary' in Package 'dti'.

## Author(s)

Karsten Tabelow <tabelow@wias-berlin.de>
J\"org Polzehl <polzehl@wias-berlin.de>

## References

J. Polzehl and K. Tabelow, *Beyond the diffusion tensor model: The package* **dti**, Journal of Statistical Software, to appear.

K. Tabelow, H.U. Voss and J. Polzehl, *Modeling the orientation distribution function by mixtures of angular central Gaussian distributions*, Journal of Neuroscience Methods, to appear.

J. Polzehl and K. Tabelow, *Structural adaptive smoothing in diffusion tensor imaging: The R package dti*, Journal of Statistical Software, 31 (2009) pp. 1–24.

K. Tabelow, J. Polzehl, V. Spokoiny and H.U. Voss. *Diffusion Tensor Imaging: Structural adaptive smoothing*, NeuroImage 39(4), 1763-1773 (2008).

## See Also

dtiData, readDWIdata, sdpar-methods, getsdofsb-methods, dwiRiceBias-methods, dtiTensor-methods, dwiMixtensor-methods, dti.smooth-methods, dwi.smooth-methods, dtiIndices-methods, dwiQball-methods, tracking-methods, show3d-methods, plot-methods, print-methods, summary-methods, extract-methods

---

# dwi.smooth-methods *Smooth DWI data*

---

## Description

Adaptive smoothing of DWI data. Smoothing is performed both in space and on the sphere (e.g. between images obtained for different gradient directions) employing a natural geometrical distance ( in SE(3)). Structural adaptation is used in space only. Method `dwi.smooth` refers to the original POAS approach for single shell data. Method `dwi.smooth.ms` implements an improved method that is applicable for both single and multi-shell data.

## Usage

```
  ## S4 method for signature 'dtiData'
dwi.smooth(object, kstar, lambda=20, kappa0=NULL, mask=NULL, ncoils=1,
                          sigma=NULL, level=NULL, vred=4, verbose=FALSE, dist=1,
                                 model=c("Gapprox","Gapprox2","Chi","Chi2"))
  ## S4 method for signature 'dtiData'
dwi.smooth.ms(object, kstar, lambda=12, kappa0=.5, ncoils=1,
                           sigma=NULL, ws0=1, level=NULL, mask = NULL, xind=NULL,
                                    yind=NULL, zind=NULL, verbose=FALSE,
                                    usemaxni=TRUE, memrelease = TRUE)
```

## Arguments

| | |
|---|---|
| `object` | Object of class `"dtiData"` |
| `kstar` | Number of steps in structural adaptation |
| `lambda` | Scale parameter in adaptation |
| `kappa0` | determines amount of smoothing on the sphere. Larger values correspond to stronger smoothing on the sphere. If `kappa0=NULL` a value is that corresponds to a variace reduction with factor `vred` on the sphere. |
| `ncoils` | Number of coils in MR system |
| `sigma` | Error standard deviation. Assumed to be known and homogeneous in the current implementation. A reasonable estimate may be defined as the modal value of standard deviations obtained using method `getsdofsb`. |

| | |
|---|---|
| level | Threshold for image intensities when setting mask. |
| mask | Binary 3D image defining a mask |
| vred | Used if kappa0=NULL to specify the variance reduction on the sphere when suggesting a value of kappa0 |
| xind | index for x-coordinate |
| yind | index for y-coordinate |
| zind | index for z-coordinate |
| verbose | If verbose=TRUE additional reports are given. |
| dist | Distance in SE3. Reasonable values are 1 (default, see Becker et.al. 2012), 2 ( a slight modification of 1: with k6^2 instead of abs(k6)) and 3 (using a 'naive' distance on the sphere) |
| model | Determines which quantities are smoothed. Possible values are "Chi" for observed values (assumed to be distributed as noncentral Chi with 2*ncoils degrees of freedom), "Chi2" for squares of observed values (assumed to be distributed as noncentral Chi-squared with 2*ncoils degrees of freedom). "Gapprox" and "Gapprox2" use a Gaussian approximation for the noncentral Chi distribution to smooth ovserved and squared values, respectively. |
| ws0 | Factor to downweight information from S0 images, defaults to 1/numer of s0 images. |
| usemaxni | If "usemaxni==TRUE" a strikter penalization is used. |
| memrelease | If "memrelease==TRUE" try to release allocated memory whenever possible. |

## Value

An object of class "dtiData" with smoothed diffusion weighted images.

## Methods

signature(object) = "ANY" Returns a warning.

signature(object) = "dtiData" Smoothing of DWI data

## Author(s)

Karsten Tabelow <tabelow@wias-berlin.de>
J\"org Polzehl <polzehl@wias-berlin.de>

## See Also

dtiData, dtiData,

dwiMD-methods *Methods for Mean Diffusivity in Package 'dti'*

## Description

Compute mean diffusivity (MD) from dtiData or dtiTensor objects

## Usage

```
  ## S4 method for signature 'dtiData'
dwiMD(object, eps=.05)
  ## S4 method for signature 'dtiTensor'
dwiMD(object)
```

## Arguments

| | |
|---|---|
| object | Object of class "dtiData" or "dtiTensor" |
| eps | tolerance in search for good gradient combinations. |

## Value

Array of mean diffusivities.

## Methods

signature(object = "ANY") Returns a warning

signature(object = "dtiData") searches for three gradients that enable best MD evaluation. Returns MD.

signature(object = "dtiTensor") calculates MD values from estimated difusion tensors

## Author(s)

Karsten Tabelow <tabelow@wias-berlin.de>

## See Also

dtiData, dtiTensor,

dwiMixtensor-methods *Methods for Function 'dwiMixtensor' in Package 'dti'*

## Description

The method estimates, in each voxel, a mixture of radial symmetric tensors from the DWI data contained in an object of class "dtiData".

## Usage

```
## S4 method for signature 'dtiData'
dwiMixtensor(object, maxcomp=3,
          model=c("MT","MTiso","MTisoFA","MTisoEV"), fa=NULL,
          lambda=NULL, mask=NULL, reltol=1e-10, maxit=5000, ngc=1000,
          nguess=100*maxcomp^2, msc=c("BIC","AIC","AICC","none"),
          mc.cores = setCores(,reprt=FALSE))
## S4 method for signature 'dwiMixtensor,dwiMixtensor'
dwiMtCombine(mtobj1, mtobj2, msc="BIC", where=NULL)
```

## Arguments

| | |
|---|---|
| object | Object of class "dtiData" |
| maxcomp | Maximal number of mixture components. |
| model | Specifies the mixture model used. "MT" corresponds to a mixture of prolate tensors, "MTiso" includes an isotropic compartment, "MTisoFA" additionally fixes FA to the value given in argument fa and "MTisoEV" uses eigenvalues specified by fa and lambda. |
| fa | Value for FA in case of model="MTisoFA" or model="MTisoEV" |
| lambda | Value for first eigenvalue in case of model="MTisoEV" |
| mask | Brain mask |
| reltol | Relative tolerance for R's optim() function. |
| maxit | Maximal number of iterations in R's optim() function. |
| ngc | provide information on number of voxel processed, elapsed time and estimated remaining time after ngc voxel. |
| nguess | number of guesses in search for initial estimates |
| msc | Criterion used to select the order of the mixture model, either BIC (Bayes Information Criterion) AIC (Akaike Information Criterion) or AICC ((Bias-)Corrected Akaike Information Criterion). None may be specified to only correct for underestimation of variances. |
| mtobj1 | For method "dwiMtCombine" an "dwiMixtensor"-object. |
| where | Mask of voxel for which "dwiMtImprove" or "dwiMtCombine" should be performed. |

| | |
|---|---|
| mtobj2 | For method "dwiMtCombine" an "dwiMixtensor"-object obtained from the same "dwiData" object. The maximum number of components in mtobj2 should preferably be less or equal to the maximum number of components in mtobj1. |
| mc.cores | Number of cores to use. Defaults to number of threads specified for openMP, see documentation of package **awsMethods**. Our experience suggests to use 4-6 cores if available. |

## Details

For model="MT" the function estimates, in each voxel, a mixture of radial symmetric (prolate) tensors from the DWI data contained in an object of class "dtiData". The number of mixture components is selected depending on the data, with a maximum number of components specified by maxcomp. Optimization is performed usin R's internal BFGS code with mixture weights (volumes of compartments corresponding to a tensor component) computed using the Lawson-Hannson NNLS code. model="MT" is only available for single shell data. In case of model="MTiso" the model additionally contains an isotropic compartment. Optimization uses the internal L-BFGS-B code. model="MTisoFA" and model="MTisoEV" fix FA and eigenvalues of the prolate tensors, respectively, in the tensor mixture model with isotropic compartment.

The method "dwiMtCombine" enables to combine results obtained for the same dwi data set with different specifications, e.g. for maximum number of components mcomp and settings that influence initial estimates. The combined result contains in each voxel the best result from both reconstructions with respect to the specified model selection criterion msc.

## Value

An object of class "dwiMixtensor".

## Author(s)

Karsten Tabelow <tabelow@wias-berlin.de>
J\"org Polzehl <polzehl@wias-berlin.de>

## References

Jian et al. (2007), A novel tensor distribution model for the diffusion-weighted MR signal, *NeuroImage* **37**, 164–176.

J. Polzehl, K. Tabelow (2019). Magnetic Resonance Brain Imaging: Modeling and Data Analysis Using R. Springer, Use R! series. Doi:10.1007/978-3-030-29184-6.

## See Also

dtiData, readDWIdata, medinria, dtiData, dwiMixtensor

## Examples

```
## Not run: demo(mixtens_art)
```

dwiQball-methods *Methods for Function 'dwiQball' in Package 'dti'*

## Description

The method estimates, in each voxel, the coefficients of an expansion of the apparent diffusion cefficient (ADC) with respect to a apherical harmonics orthonormal system from the DWI data contained in an object of class "dtiData".

## Usage

```
  ## S4 method for signature 'dtiData'
dwiQball(object, what="wODF", order=4, lambda=0, mask=NULL)
```

## Arguments

| | |
|---|---|
| object | Object of class "dtiData" |
| what | Determines quantity to estimate, coefficients ot the orientation density function (ODF) (what="ODF", what="wODF", what="aODF") or the apparent diffusion coefficient (ADC) (what="ADC") with respect to spherical harmonics of the up to the specified order. |
| order | even integer: maximum order of the sperical harmonics expansion |
| lambda | nonnegative regularization parameter. |
| mask | optional brain mask |

## Value

An object of class "dwiQball".

## Methods

**obj = "ANY"** Returns a warning.

**obj = "dtiData"** Estimate, in each voxel, the coefficients of an expansion of the orientation density function (ODF) or the apparent diffusion coefficient (ADC) with respect to a apherical harmonics orthonormal system. Note that the maxima of the ADC have no direct interpretation as fibre orientations.

## Author(s)

Karsten Tabelow <tabelow@wias-berlin.de>
J\"org Polzehl <polzehl@wias-berlin.de>

## References

M. Descoteaux, E. Angelino, S. Fitzgibbons and R. Deriche, *Regularized, Fast and Robust Analytical Q-Ball Imaging*, Magnetic Resonance Methods, 2007, 58, 497-512.

J. Polzehl, K. Tabelow (2019). Magnetic Resonance Brain Imaging: Modeling and Data Analysis Using R. Springer, Use R! series. Doi:10.1007/978-3-030-29184-6.

## See Also

dtiData, readDWIdata, dtiIndices-methods, medinria, dtiData, dtiTensor

## Examples

```
## Not run: demo(dti_art)
```

---

dwiRiceBias-methods *Correction for Rician Bias*

---

## Description

Correction for Rician Bias assuming known variance parameter

## Usage

```
## S4 method for signature 'dtiData'
dwiRiceBias(object, sigma=NULL, ncoils=1)
```

## Arguments

| | |
|---|---|
| object | Object of class "dtiData" |
| sigma | Scale parameter that relates the distribution of the signal to a $\chi_{2L}$ distribution |
| ncoils | number of effective coils in parallel imaging, the related $\chi$ distribution has 2*ncoils degrees of freedom. |

## Value

An object of class "dtiData".

## Methods

**object = "ANY"** Returns a warning.

**object = "dtiData"** Returns a dtiData object with bias-corrected image intensities.

## Author(s)

Karsten Tabelow <tabelow@wias-berlin.de>
J\"org Polzehl <polzehl@wias-berlin.de>

## See Also

dtiData, dtiTensor-methods, dwiMixtensor-methods, dtiData, dtiTensor, dwiMixtensor,

dwiSqrtODF-methods *Methods for positive definite EAP and ODF estimation in Package* **dti**

## Description

Compute a positive definite estimate of the Ensemble Average Propagator (EAP) and Orientation Density Function (ODF) using the approach of Cjeng et. al (2012).

## Usage

```
  ## S4 method for signature 'dtiData'
dwiSqrtODF(object,what="sqrtODF",order=4,forder=1,lambda=0,D0=1.4e-3)
```

## Arguments

| | |
|---|---|
| object | Object of class "dtiData" |
| what | Character, currently only "sqrtODF" is possible |
| order | Even integer, Order of spherical harmonics approximation. |
| forder | Integer, Order of radial approximation. |
| lambda | Non-negative, Regularization parameter. |
| D0 | Numeric vector, grid of diffusivity parameters, typically about 1e-3. |

## Methods

`signature(object = "ANY")` Returns a warning.

`signature(object = "dtiData")` Compute a positive definite estimate of the Ensemble Average Propagator (EAP) and Orientation Density Function (ODF) using the approach of Cjeng et. al (2012).

## Author(s)

Karsten Tabelow <tabelow@wias-berlin.de>
J\"org Polzehl <polzehl@wias-berlin.de>

## References

J. Cheng, T. Jiang and R. Deriche. *Nonnegative Definite EAP and ODF Estimation via a Unified Multi-Shell HARDI Reconstruction*, MICCAI 2012.

## See Also

dtiData, readDWIdata, dtiData, dwiQball

extract-methods *Methods for Function 'extract' and '[' in Package 'dti'*

## Description

The methods extract and/or compute specified statistics from object of class "dtiData", "dtiTensor", and "dtiIndices". This can be restricted to a subset of voxel.

## Usage

```
  ## S4 method for signature 'dtiData'
extract(x,
    what=c("data","gradient","btb","s0","sb","siq"),
    xind=TRUE, yind=TRUE, zind=TRUE)
  ## S4 method for signature 'dtiTensor'
extract(x, what=c("tensor", "fa", "ga", "md", "evalues",
      "andir", "s0", "mask", "bic", "aic", "outlier"),
    xind=TRUE, yind=TRUE, zind=TRUE, mc.cores = setCores(, reprt = FALSE))
  ## S4 method for signature 'dwiMixtensor'
extract(x, what=c("w0","andir", "order", "ev", "mix", "s0",
      "mask", "fa", "eorder", "bic", "aic"), xind=TRUE, yind=TRUE, zind=TRUE)
  ## S4 method for signature 'dtiIndices'
extract(x, what=c("fa", "andir", "ga", "md", "bary"),
    xind=TRUE, yind=TRUE, zind=TRUE)
  ## S4 method for signature 'dwiQball'
extract(x, what=c("sphcoef", "s0", "mask", "bic", "aic",
      "outlier"), xind=TRUE, yind=TRUE, zind=TRUE)
  ## S4 method for signature 'dtiData'
x[i, j, k, drop=FALSE]
  ## S4 method for signature 'dtiTensor'
x[i, j, k, drop=FALSE]
  ## S4 method for signature 'dtiIndices'
x[i, j, k, drop=FALSE]
  ## S4 method for signature 'dkiTensor'
x[i, j, k, drop=FALSE]
  ## S4 method for signature 'dkiIndices'
x[i, j, k, drop=FALSE]
  ## S4 method for signature 'dwiQball'
x[i, j, k, drop=FALSE]
```

## Arguments

| | |
|---|---|
| x | Object of class dti |
| i | vector of x-coordinates, defaults to whole range. |
| j | vector of y-coordinates, defaults to whole range. |
| k | vector of z-coordinates, defaults to whole range. |

| | |
|---|---|
| xind | vector of x-coordinates, defaults to whole range. |
| yind | vector of y-coordinates, defaults to whole range. |
| zind | vector of z-coordinates, defaults to whole range. |
| what | Statistic to extract. See Methods Section for details. |
| drop | unused. |
| mc.cores | Number of cores to use. Defaults to number of threads specified for openMP, see documentation of package **awsMethods**. Our experience suggests to use 4-6 cores if available. |

## Value

For function `extract` a list with components carrying the names of the options specified in argument `what`. For `"["` the cutted object.

## Methods

The generic extract function `"["` does what it is expected to do: it extracts parts of the object specified by `i`, `j`, and `k`.

Returns a warning for `extract`. Generic funtion for `"["` returns an object of same class with data clipped to the indices specified in arguments `i`, `j` and `k`.

**x = "ANY", "dtiData"** Extraction of squared gradient matrix `"btb"` or of S0 `"s0"`, Sb `"sb"`, Si/mean(SO) `"siq"` or all images `"data"` restricted to the cube defined by arguments `i`, `j` and `k`.

**x = "dtiIndices"** Returns an array containing the specified statistics, i.e. fractional anisotropy `"fa"`, geodesic anisotropy `"ga"`, mean diffusivity `"md"`, main direction of anisotropy `"andir"` and/or shape parameters `"bary"`, as specified in argument `what`. Information is extracted for voxel within the cube defined by `xind`, `yind`, and `zind`.

**x = "dtiTensor"** Returns a list with component names corresponding to `what` containing the specified statistics, i.e. fractional anisotropy `"fa"`, geodesic anisotropy `"ga"`, mean diffusivity `"md"`, eigenvalues `"evalues"`, main direction of anisotropy `"andir"`, the tensor `"tensor"` the estimated S0 image `"s0"`, the values of the model selection criteia BIC `"bic"` or AIC `"aic"` and/or the mask used to restrict computations `"mask"`, as specified in argument `what`. Information is extracted for voxel within the cube defined by arguments `xind`, `yind` and `zind`.

**x = "dwiMixtensor"** Returns a list with component names corresponding to `what` containing the specified statistics. Possible values for `what` are `"w0"` (size of isotropic department), `"order"` (estimated number of mixture components), `"eorder"` effective order), `"ev"` (eigenvalues), `"mix"` (mixture weights), `"andir"` (main directions of diffusion), `"fa"` (FA index), `"s0"` (the estimated S0 image), the values of the model selection criteia BIC `"bic"` or AIC `"aic"` and `mask` (the mask used to restrict computations). Information is extracted for voxel within the cube defined by arguments `xind`, `yind` and `zind`.

**x = "dwiQball"** Returns an array containing the specified statistics, the estimated coefficients with respect to the selected spherical harmonics basis `"sphcoef"`, the estimated S0 image `"s0"`, the values of the model selection criteia BIC `"bic"` or AIC `"aic"` and/or the mask used to restrict computations `"mask"`, as specified in argument `what`. Information is extracted for voxel within the cube defined by arguments `xind`, `yind` and `zind`.

### Author(s)

Karsten Tabelow <tabelow@wias-berlin.de>
J\"org Polzehl <polzehl@wias-berlin.de>

### See Also

dtiData, dtiTensor, dtiIndices dwiMixtensor, dwiQball

---

## getmask-methods *Methods for Function 'getmask' in Package 'dti'*

---

### Description

Create a mask containing voxel inside the head

### Usage

```
## S4 method for signature 'dtiData'
getmask(object, level = NULL, prop = 0.4, size = 3)
```

### Arguments

| | |
|---|---|
| object | an object of class "dtiData" |
| level | S0 intensity value to be used to discriminate between voxel inside and outside the brain. A good value of level may be determined using method sdpar in advance. |
| prop | proportion of voxel in test area with s0 value larger than level needed to decide for a voxel inside the brain |
| size | size of a cube defining a test area |

### Value

The function returns an object of class dtiData.

### Methods

**obj = "ANY"** Returns a warning

**obj = "dtiData"** Create a mask containing voxel inside the head

**obj = "array"** Create a mask containing voxel inside the head

### Author(s)

Karsten Tabelow <tabelow@wias-berlin.de>
J\"org Polzehl <polzehl@wias-berlin.de>

### See Also

dtiData, readDWIdata, dti.smooth, sdpar

getsdofsb-methods *Estimate the noise standard deviation*

## Description

Estimate the noise standard deviation. Uses an assumption that the standard deviation is a linear function of the expected mean for image intensities. qA0 and qA1 define quantiles of observed image intensities that define the range of values where this assumption is made.

## Usage

```
  ## S4 method for signature 'dtiData'
getsdofsb(object,qA0=.1,qA1=.98,nsb=NULL,level=NULL)
```

## Arguments

| | |
|---|---|
| object | Object of class "dtiData" |
| qA0 | level for lower quantile of image intensities |
| qA1 | level for upper quantile of image intensities |
| nsb | number of diffusion weighted image to use |
| level | level for mask |

## Value

An object of class "dtiData" with results in slot sdcoef in components 5: intercept parameter, 6: slope parameter for linear model, 7: lower bound (depending on qA0) and 8: upper bound (depending on qA1).

## Methods

signature(object) = "ANY" Returns a warning.

signature(object) = "dtiData" Returns a dtiData object with estimated standard deviation parameters in slot sdcoef.

## Author(s)

Karsten Tabelow <tabelow@wias-berlin.de>
J\"org Polzehl <polzehl@wias-berlin.de>

## See Also

dtiData, dwi.smooth-methods, dtiData,

`medinria` *Read/Write Diffusion Tensor Data from/to NIFTI File*

## Description

Read/Write diffusion tensor data from/to NIfTI file. Interface functions to MedINRIA.

## Usage

```
medinria2tensor(filename)
tensor2medinria(obj, filename, xind = NULL, yind = NULL, zind = NULL)
```

## Arguments

| | |
|---|---|
| `filename` | file name for the tensor data. |
| `obj` | object of class `"dtiTensor"` |
| `xind` | index to define a subcube in x-direction. If `is.null(xind)` all voxel indices are used. |
| `yind` | index to define a subcube in y-direction. If `is.null(yind)` all voxel indices are used. |
| `zind` | index to define a subcube in z-direction. If `is.null(zind)` all voxel indices are used. |

## Value

For function `medinria2tensor`: object of class `"dtiTensor"`.

## Author(s)

Karsten Tabelow `<tabelow@wias-berlin.de>`
J\"org Polzehl `<polzehl@wias-berlin.de>`

## References

P. Fillard, J. Souplet and N. Toussaint *Medical Image Navigation and Research Tool by INRIA (MedINRIA)*, INRIA Sophia Antipolis - Research Project ASCLEPIOS 2007

https://www-sop.inria.fr/asclepios/software/MedINRIA/

## See Also

`dtiTensor`, `dtiTensor-methods` `dtiIndices-methods`

## Examples

```
## Not run: demo(dti_art)
```

optgrad *Optimal gradient directions*

## Description

List containing gradient directions minimizing Coulomb forces on the sphere following a proposal by D. Jones (1999) for number of gradients between 6 and 162.

optgradients *Optimal gradient directions for number of gradients between 6 and 162*

## Description

Optimal gradient directions minimizing symmetrized Coulomb forces on the sphere following a proposal by Jones et al. (1999). These directions define an optimal design in DWI for given number of gradients.

## Usage

```
optgrad
```

## Format

a list with name optgrad and component ngrad-5 containing a matrix with ngrad gradients as columns.

plot-methods *Methods for Function 'plot' in Package 'dti'*

## Description

Visualization of objects of class "dtiData", "dtiIndices", "dtiTensor"and class "dwiMixtensor"

## Usage

```
## S4 method for signature 'dtiData'
plot(x, y, slice=1, gradient=NULL, view= "axial", show=TRUE,
  density=FALSE, xind=NULL, yind=NULL, zind=NULL, mar=par("mar"), mgp=par("mgp"), ...)
## S4 method for signature 'dtiTensor'
plot(x, y, slice=1, view="axial", quant=0, minfa=NULL, contrast.enh=1,
    what="fa", qrange=c(.01,.99), xind=NULL, yind=NULL, zind=NULL,
    mar=par("mar"), mgp=par("mgp"), ...)
## S4 method for signature 'dwiMixtensor'
```

```
plot(x, y, slice=1, view="axial", what="fa", minfa=NULL,
   identify=FALSE, xind=NULL, yind=NULL, zind=NULL, mar=par("mar"), mgp=par("mgp"), ...)
## S4 method for signature 'dtiIndices'
plot(x, y, slice=1, view= "axial", method=1, quant=0, minfa=NULL,
   show=TRUE, identify=FALSE, density=FALSE, contrast.enh=1, what="fa",
   xind=NULL, yind=NULL, zind=NULL, mar=par("mar"), mgp=par("mgp"), ...)
## S4 method for signature 'dwiFiber'
plot(x, y, ...)
## S4 method for signature 'dkiIndices'
plot(x, y, slice=1, what=c("md", "fa", "mk", "mk2",
     "k1", "k2", "k3", "kaxial", "kradial", "fak"), xind=NULL, yind=NULL,
     mar=par("mar"), mgp=par("mgp"), ...)
```

## Arguments

| | |
|---|---|
| x | Object of class "dtiIndices", "dtiData" or "dtiTensor" |
| y | Not used |
| slice | Slice number |
| view | Choose "sagittal", "coronal", or "axial" view here |
| gradient | Index of data cube to plot. Defaults to the first S0 image. |
| method | Method for color coding tensor indices. |
| quant | If is.null(minfa) specify minfa as corresponding quantile of the fractal anisotropy (FA) index. |
| minfa | Display only information for voxel with (G)FA>minfa |
| show | Visualize information in a graphics device (for classes "dtiData" and "dtiIndices" only). |
| identify | Enable identification of coordinates by mouse actions, logical with default FALSE. Uses function identify. (for classes "dtiIndices" and "dwiMixtensor" only) |
| density | Show density of S0(Sb)-values (for class "dtiData") or densities of fractal anisotropy (FA) or geodesic anisotropy (GA) ( for class "dtiIndices"). |
| contrast.enh | Enhance image contrast using min(1,x$anindex/contrast.enh instead of the anisotropy index itself. Effective values are within the interval (0,1). |
| what | In case of class "dtiIndices" what="ga" uses geodesic anisotropy (GA) in contrast to what="fa" for fractional anisotropy (FA). For class "dwiMixtensor" what="fa" for FA and what="order" for the number of mixture components may be chosen. |
| mar | Graphical parameter for par. |
| mgp | Graphical parameter for par. |
| qrange | Cut image intensity to these quantiles to avoid that outliers determine the dynamic range of the image. |
| xind | If provided restrict display to indices specified in xind for x-direction. |
| yind | If provided restrict display to indices specified in yind for y-direction. |
| zind | If provided restrict display to indices specified in zind for z-direction. |
| ... | currently not used |

### Methods

**x = "ANY"** Generic function: see `plot`.

**x = "dwi"** Returns a warning.

**x = "dtiData"** `gradient` can be used to specify a specific data cube associated with the index of a gradient direction. For objects of class `"dtiData"` images are produces that are scaled by the maximal observed image value. This guarantees that subsequently produced images are on a comparable grey scale. The resulting image of class `"adimpro"` from package **adimpro** is returned.

**x = "dtiIndices"** Color coded anisotropy maps are produced depending on the specification in `method`. `method==1`, `method==2`, `method==4` and `method==5` specify three different color schemes for directional FA-maps. `method==6` uses colored FA maps based on scheme developed at Uni Muenster (M. Deppe, Germany). `method==3` specifies visualization of dtiIndices using color coded shape parameters. If `identify==FALSE` the resulting image of class `"adimpro"` from package **adimpro**, otherwise a matrix with coordinates of identified voxel is returned.

**x = "dtiTensor"** The tensor itself, fractional anisotropy (FA), mean diffusivity (MD) and a color coded anisotropy map are provided. `NULL` is returned.

**x = "dwiMixtensor"** Depending of `what` images of FA (`what="fa"`), number of mixture components (`what="order"`), effective order (`what="eorder"`) or maximum eigenvalues (`what="ev"`). is returned.

**x = "dwiFiber"** Creates a density plot of fiber lengths. `NULL` is returned.

**x = "dkiIndices"** Prelimanary function to plot a slice of diffusion kurtosis indices: Mean Kurtosis `what="mk"` or `what="mk2"`, mean diffusivity `what="md"`, fractional anisotropy `what="fa"`.

### Author(s)

Karsten Tabelow <tabelow@wias-berlin.de>
J\"org Polzehl <polzehl@wias-berlin.de>

### See Also

`dtiIndices`, `dtiData`, `dtiTensor` `dwiMixtensor`

### Examples

```
## Not run: demo(dti_art)
```

---

## polyeder — *Polyeders derived from the Icosahedron (icosa0) by sequential triangulation of surface triangles*

---

### Description

icosa0 - icosa4 provide a description of regular polyeders derived from the Icosahedron (icosa0) by sequential triangulation of surface triangles

### Usage

```
icosa0
```

### Format

a list with components

1. vertices - array of dimension `c(3,nv)`. containing cartesian coordinate of the `nv` vertices.
2. indices - Indices of vertices that define surface triangles of the polyeder.
3. edges - Indices of vertices that define edges of the polyeder.
4. nv - number of vertices
5. ne - number of edges
6. ni - number of triangles

---

## `print-methods` *Methods for Function 'print' in Package 'dti'*

---

### Description

The function provides information on data dimensions, data source and existing slot-names for objects of class `"dti"`, `"dtiData"`, `"dtiTensor"`, `"dtiIndices"`, `"dkiIndices"`, `"dkiTensor"`, `"dwiMixtensor"`, `"dwiQball"` and `"dwiFiber"`.

### Usage

```
  ## S4 method for signature 'dwi'
print(x)
```

### Arguments

| | |
|---|---|
| `x` | Object of class `"dtiIndices"`, `"dtiData"`, `"dtiTensor"`, `"dkiIndices"`, `"dkiTensor"`, `"dwiFiber"`, `"dwiMixtensor"` or `"dwiQball"` |

### Methods

**x = "ANY"** Generic function: see `print`.

**x = "dwi"** The function provides information on data dimensions, data source and existing slot-names for objects of class `"dwi"`.

### Author(s)

Karsten Tabelow <tabelow@wias-berlin.de>
J\"org Polzehl <polzehl@wias-berlin.de>

### See Also

`dtiIndices`, `dtiData`, `dtiTensor` `dwiMixtensor` `dwiQball` `dwiFiber`

readDWIdata *Read Diffusion Weighted Data*

## Description

The functions create a "dtiData" object from Diffusion Weighted Data from medicial imaging files in a list of directories or from an imagefile, where the diffusion weighted data is given as 2-byte integer.

## Usage

```
dtiData(gradient, imagefile, ddim, bvalue = NULL, xind = NULL, yind = NULL, zind = NULL,
        level = 0, mins0value = 1, maxvalue = 32000, voxelext = c(1, 1, 1),
        orientation = c(0L, 2L, 5L), rotation = diag(3))
readDWIdata(gradient, dirlist, format = c("DICOM", "NIFTI", "ANALYZE", "AFNI"),
        nslice = NULL, order = NULL, bvalue = NULL,
        xind = NULL, yind = NULL, zind = NULL, level = 0, mins0value = 1,
        maxvalue = 32000, voxelext = NULL, orientation = c(0L, 2L, 5L),
        rotation = NULL, pattern = NULL, SPM2=TRUE, verbose = FALSE)
```

## Arguments

| | |
|---|---|
| gradient | matrix of diffusion gradients (including zero gradients for S0 images) |
| imagefile | name of data image file (binary 2Byte integers) |
| ddim | dimension of image cube (3D) |
| dirlist | list of directories containing the data files or name of a single data file (e.g. 4D NIFTI) |
| format | string specifying the medical imaging format, one of "DICOM", "NIFTI", "ANALYZE", or "AFNI" |
| nslice | number of slices (usually z-direction) |
| order | vector, specifying a different order of the data files, i.e. other than alphabetic order in the directories given by dirlist. If not given, 1:n is used for n data files (no order change). |
| bvalue | vector of b-values (default 0 for S0 and 1 for Si) |
| xind | subindex for x-direction |
| yind | subindex for y-direction |
| zind | subindex for z-direction |
| level | determine mins0value as quantile of positive S0-values |
| mins0value | set voxel in S0-images with values less than level "inactive" |
| maxvalue | set voxel with values larger than maxvalue inactive |
| voxelext | voxel extensions in coordinate directions |
| orientation | orientations of data as coded in AFNI |

| | |
|---|---|
| rotation | optional rotation matrix for the coordinate system. |
| pattern | pattern for file matching in the directories dirlist. |
| SPM2 | Enable some non-standard NIfTI files produced by SPM to be readable. |
| verbose | some progress reports if TRUE |

## Details

The function dtiData creates an object of class "dtiData" from an image file, where the diffusion weighted data is given as 2-byte integer. This image file has to be prepared by the user. Use writeBin to write out first all S0 images and than all Si images. The gradient should be created according to this order. Run the demo in order to have an example, how to do this!

The function readDWIdata reads the data files given in the directories in dirlist in alphabetic order. The order can be changed using the order argument: If filelist is the vector of files in alphabetic order, they are read in the order filelist[order]. If order is not given order <- 1:n is used (no change!). The medical imaging format is given by format and can be one of "DICOM", "NIFTI", "ANALYZE", or "AFNI". The number of slices of the three dimensional data cube is given by nslice. The diffusion gradients are provided as matrix gradient.

xind, yind, and zind define a region of interest as indices. If not given 1:dim[i] is used. level determine mins0value as quantile of positive S0-values. mins0value sets voxel in S0-images with values less than level "inactive". maxvalue sets voxel with values larger than maxvalue inactive.

voxelext defines the voxel extension, overwrites the values found in the imaging files. orientation codes the data orientation in AFNI notation.

## Value

An object of class "dtiData".

## Author(s)

Karsten Tabelow <tabelow@wias-berlin.de>
J\"org Polzehl <polzehl@wias-berlin.de>

## References

J. Polzehl, K. Tabelow (2019). Magnetic Resonance Brain Imaging: Modeling and Data Analysis Using R. Springer, Use R! series. Doi:10.1007/978-3-030-29184-6.

https://afni.nimh.nih.gov/pub/dist/src/README.attributes

## See Also

dti.smooth, dtiTensor-methods, dtiData

## Examples

```
## Not run: demo(dti_art)
```

sdpar-methods *Methods for Function 'sdpar' in Package 'dti'*

## Description

This function estimates the parameters of a piecewise linear model for the dependence between error standard deviation and mean.

## Usage

```
  ## S4 method for signature 'dtiData'
sdpar(object,level=NULL,sdmethod="none",interactive=TRUE,threshfactor=1)
```

## Arguments

| | |
|---|---|
| object | An object of class dtiData |
| level | Suggested value for slot level. As a default the value in object@level is used. The value determines the lower endpoint of the linear section in the model for error standard deviation as a function of the mean. |
| sdmethod | Method for estimating voxelwise standard deviations if replicates of zero weighted images are available, can be set to "sd" or "mad". "none" specifies that no variance model is fitted |
| interactive | If TRUE a density of values in zero weighted images is plotted together with the specification of the lower endpoint of the interval of linearity. A good choice of this point should correspond, if present, to the minimum between the first two modes of the density estimate. The value can be changed or accepted. If changed a new value for slot lambda is set. |
| threshfactor | Factor for threshold-value selected if function is run in interactive mode. May be used to correct results if automatic threshold selection fails. |

## Value

The function returns an object of class dtiData.

## Methods

**obj = "ANY"** Returns a warning

**obj = "dtiData"** Estimate parameters of a model for the dependence between error standard deviation and mean.

## Author(s)

Karsten Tabelow <tabelow@wias-berlin.de>
J\"org Polzehl <polzehl@wias-berlin.de>

## See Also

dtiData, readDWIdata, dti.smooth, dtiTensor,

## Examples

```
## Not run: demo(dti_art)
```

---

setmask-methods *Methods for Function 'setmask' in Package 'dti'*

---

## Description

Read mask definition from NIfTI file and include it in dtiData object

## Usage

```
## S4 method for signature 'dtiData'
setmask(object, maskfile)
```

## Arguments

| | |
|---|---|
| object | an object of class "dtiData" |
| maskfile | NIfTI file containing mask definition. Dimension need to be compatible, i.e. either equal object@ddim0 or object@ddim |

## Value

The function returns an object of class dtiData.

## Methods

**obj = "ANY"** Returns a warning

**obj = "dtiData"** Set mask definition in dtiObject using information provided as NIfTI file as e.g. provided by fsl_bet.

## Author(s)

Karsten Tabelow <tabelow@wias-berlin.de>
J\"org Polzehl <polzehl@wias-berlin.de>

## See Also

dtiData, readDWIdata,

show-methods *Methods for Function 'show' in Package 'dti'*

## Description

The function provides information on data dimensions, data source and existing slot-names for objects of class "dti", "dtiData", "dtiTensor", "dwiMixtensor", "dtiIndices", "dwiQball" or "dwiFiber"

## Usage

```
  ## S4 method for signature 'dti'
show(object)
```

## Arguments

| | |
|---|---|
| object | Object of class dtiIndices, dtiData, dtiTensor, dkiTensor, dkiIndices, dwiMixtensor, dwiQball or dwiFiber |

## Methods

**x = "ANY"** Generic function.

**x = "dti"** The function provides information on data dimensions, data source and existing slot-names for objects of class "dti" and classes that extent "dti".

## Author(s)

Karsten Tabelow <tabelow@wias-berlin.de>
J\"org Polzehl <polzehl@wias-berlin.de>

## See Also

dtiIndices, dtiData, dtiTensor dwiMixtensor dwiQball dwiFiber

show3d-methods *Methods for Function 'show3d' in Package 'dti'*

## Description

The function provides 3D visualization of "dtiData", "dtiTensor", "dwiQball" and "dtiIndices" objects using the "rgl"-package. Functionality of the rgl-package allows to rotate and zoom the displayed object.

## Usage

```
## S4 method for signature 'dtiData'
show3d(obj, xind=NULL, yind=NULL, zind=NULL, quant=.8,
         scale=.4,
         bgcolor="black", add=FALSE, maxobjects=729, what=c("adc","data"),
         minalpha=1, nn=1, normalize=FALSE, box=FALSE, title=FALSE, ...)
## S4 method for signature 'dtiTensor'
show3d(obj, xind=NULL, yind=NULL, zind=NULL, method=1,
         minfa=.3, mask=NULL, fibers=FALSE,
         maxangle = 30,level=0, quant=.8, scale=.4, bgcolor="black", add=FALSE,
        subdivide=2, maxobjects=729, what=c("tensor","adc","odf"), odfscale = 1,
         minalpha=.25, normalize=NULL, box=FALSE, title=FALSE,...)
## S4 method for signature 'dkiTensor'
show3d(obj, xind=NULL, yind=NULL, zind=NULL, method=1,
         minfa=.3, mask=NULL, level=0, quant=.8, scale=.4, bgcolor="black",
         add=FALSE, subdivide=2, maxobjects=729, what=c("KT", "DT"),
         minalpha=.25, normalize=NULL, box=FALSE, title=FALSE,...)
## S4 method for signature 'dtiIndices'
show3d(obj, index=c("fa","ga"), xind=NULL, yind=NULL,
         zind=NULL, method=1,
         minfa=0, bgcolor="black", add=FALSE, lwd=1, box=FALSE,
         title=FALSE, ...)
## S4 method for signature 'dwiMixtensor'
show3d(obj, xind=NULL, yind=NULL, zind=NULL, minfa=.3,
         minorder = 1, mineo=1, fibers=FALSE, maxangle=30, level=0,
         quant=.8, scale=.4, bgcolor="black", add=FALSE,
         subdivide=3, maxobjects=729, what=c("odf","axis","both"), odfscale=1,
         minalpha=1, lwd=3, box=FALSE, title=FALSE, ...)
## S4 method for signature 'dwiQball'
show3d(obj, xind=NULL, yind=NULL, zind=NULL, level=0,
         quant=.8, scale=0.4, odfscale=1, bgcolor="black", add=FALSE,
         subdivide=3, maxobjects=729, minalpha=1, box=FALSE,
         title=FALSE, ...)
## S4 method for signature 'dwiFiber'
show3d(obj, add=FALSE, bgcolor="black", box=FALSE,
         title=FALSE, lwd=1, delta=0, ...)
```

## Arguments

| | |
|---|---|
| obj | An object of class dtiData, dtiTensor, dtiIndices,dwiMixTensor or dwiQball |
| xind | vector of x-coordinates, defaults to whole range. |
| yind | vector of y-coordinates, defaults to whole range. |
| zind | vector of z-coordinates, defaults to whole range. |
| quant | Quantile of maximal radii of objects used for scaling. |
| scale | Scale factor for the size of objects |
| bgcolor | Backgroundcolor for rgl-display |

| | |
|---|---|
| add | If true information is added to the current device, otherwise a new device is opened. |
| maxobjects | Maximal size of data cube (in voxel) to display |
| minalpha | Minimum value for transparency. |
| nn | Number of nearest neighbors used for interpolation onto a regular polyeder. |
| normalize | If TRUE normalize values (project to interval (0,1) within each voxel). For tensor objects normalize=NULL specifies a default depending on the content of argument what (normalize <- switch(what,"tensor"=FALSE,"adc"=TRUE)). |
| box | Logical, add a bounding box. |
| title | Either a character string specifying a title or a logical. If title==TRUE a default title characterizing the type of plot is generated. |
| method | method==1 and method==2 specify two different color schemes for directional FA-maps. |
| minfa | Minimal FA value for dtiTensor objects and for dwiMixtensor objects. |
| mask | additional mask for dtiTensor objects. |
| minorder | Minimal order for dwiMixtensor objects. |
| mineo | Minimal effective order for dwiMixtensor objects. |
| fibers | If TRUE show fibers starting in voxel with fa>=minfa, order>=minorder and eorder>=mineo, the last two effective for dwiMixtensor objects only. |
| maxangle | argument for fibertracking |
| level | Radius of sphere used as support for ODF visualisation |
| subdivide | Level of subdivisions for meshing, level 0:4 correspond to use of c(12,42,162,642,2562) vertices per tensor, respectively. |
| what | For dtiTensor-objects either "tensor" for visualization using ellipsoids, "adc" for Apparent Diffusion Coefficients or "odf" for the Orientation Density Function. For dwiMixtensor-objects possible specifications are "odf", "axis" and "both", with the latter superposing the estimated main directions on the estimated ODF. For "axis"(and "both") the length of the axis corresponds to the mixture weights. For dtiData-objects choices are either "data" or "adc". |
| odfscale | Determines visualisation of the Orientation density function (ODF). For odfscale=3 the ODF values are rescaled such that the volume of the displayes objects is constant. odfscale=1 uses the values of the ODF as radii in the corrsponding vertice direction of the specified polyhedron. This can lead to extremely large volumes in case of one mixture component with high excentricity. values of odfscale inbetween 1 and 3 are possible and allow to balance between volume based visualization and emphazising highly structured ODF's. |
| lwd | Linewidth for visualization of dtiIndices objects. |
| index | Eiter "FA" for fractional anisotropy index or "GA" for geodesic anisotropy index. |
| delta | if delta>0 Join line segments in fiber objects as long as acos of directions is smaller than delta. Reduces the size of 3D object at the cost of resolution. |
| ... | Additional parameters passed to function rgl.par from the rgl-package. |

## Value

The function returns the number of the current rgl-device.

## Methods

**obj = "ANY"** Returns a warning

**obj = "dtiData"** Empirical ADC's are visualized at the voxel centers. Color is determined by gradient directions, ADC values are reflected by both radial extend and transparancy. The value of `maxobjects` limits the size of datacube and may be increased on hardware with suitable graphics capabilities.

**obj = "dtiIndices"** Objects are visualized as a collection of line segments with location given by the voxel center, orientation and color determined by the main direction of inisotropy and length corresponding to either fractional or geodesic anisotropy as specified in `index`.

Displayed objects are restricted to voxel with an fractional (geodesic) anisotropy larger than `level`.

**obj = "dtiTensor"** Ellipsoids/ADC's are visualized at the voxel centers. Orientation and size correspond to the tensor values, color is determined by the main direction of anisotropy using the colorsceme specified with `method`. The fractional anisotropy value is coded as transparency. The value of `maxobjects` limits the size of datacube and may be increased on hardware with suitable graphics capabilities.

**obj = "dkiTensor"** Prelimenary show3d method for diffusion kurtosis tensors.

**obj = "dwiQball"** Estimated ODF/ADC's are visualized at the voxel centers. Color is determined by directions, ODF/ADC values are reflected by both radial extend and transparancy. The value of `maxobjects` limits the size of datacube and may be increased on hardware with suitable graphics capabilities.

**obj = "dwiFiber"** Display and combine fibres generated by function tracking.

Displays can be closed using function `rgl.close`

## Author(s)

Karsten Tabelow <tabelow@wias-berlin.de>
J\"org Polzehl <polzehl@wias-berlin.de>

## See Also

`dtiIndices-methods`, `dti.smooth`, `dtiTensor`, `dtiIndices`

## Examples

```
## Not run: demo(dti_art)
```

showFAColorScale *Writes an image with the colqFA colorscale to disk.*

## Description

Writes an image (type PNG) with the colqFA colorscale to disk.

## Usage

```
showFAColorScale(filename = "FAcolorscale.png")
```

## Arguments

| | |
|---|---|
| filename | Name of file to write. |

## See Also

See Also colqFA

subsetg *Create an objects of class "dtiData" containing only a subset of gradient directions.*

## Description

This function creates an object of class "dtiData" that containes only a subset, defined by an index vector, of the S0 and diffusion weighted images. This function may e.g. be used to separate information measured on different shells.

## Usage

```
subsetg(x, ind)
```

## Arguments

| | |
|---|---|
| x | Object of class "dtiData" |
| ind | Indexvector containing values between 1 and x@ngrad. |

## Value

An object of class "dtiData".

## Author(s)

Karsten Tabelow <tabelow@wias-berlin.de>
J\"org Polzehl <polzehl@wias-berlin.de>

## See Also

dtiData, readDWIdata, dtiData, combineDWIdata

---

summary-methods *Methods for Function 'summary' in Package 'dti'*

---

## Description

The method provides summary information for objects of class "dti".

## Usage

```
  ## S4 method for signature 'dwi'
summary(object, ...)
```

## Arguments

| | |
|---|---|
| object | Object of class "dti", "dtiData", "dtiTensor", "dwiMixtensor", "dtiIndices", "dkiIndices", "dkiTensor", "dwiQball" or "dwiFiber". |
| ... | Additional arguments in ... are passed to function quantile, e.g. argument probs may be specified here. |

## Methods

**object = "ANY"** Generic function: see summary.

**object = "dwi"** The function provides summary information for objects of class "dwi", "dtiData", "dtiTensor", "dwiMixtensor", "dtiIndices", "dkiIndices", "dkiTensor", "dwiQball" and , "dwiFiber"

## Author(s)

Karsten Tabelow <tabelow@wias-berlin.de>
J\"org Polzehl <polzehl@wias-berlin.de>

## See Also

dtiIndices, ] dtiData, dtiTensor dwiMixtensor dwiQball dwiFiber

tracking-methods *Methods for Function 'tracking' in Package 'dti'*

## Description

The function provides fiber tracking of "dtiTensor", "dtiIndices", and "dwiMixtensor" objects and methods for fiber manipulations.

## Usage

```
## S4 method for signature 'dtiTensor'
tracking(obj, roix=NULL, roiy=NULL, roiz=NULL, mask=NULL,
                    method="LINEPROP", minfa=0.3, maxangle=30, subsample = 1)
## S4 method for signature 'dtiIndices'
tracking(obj, roix=NULL, roiy=NULL, roiz=NULL, mask=NULL,
                    method="LINEPROP", minfa=0.3, maxangle=30, subsample = 1)
## S4 method for signature 'dwiMixtensor'
tracking(obj, roix=NULL, roiy=NULL, roiz=NULL, mask=NULL,
                    method="LINEPROP", minfa=0.3, maxangle=30, subsample = 1,
                    mincompartsize = 0)
## S4 method for signature 'dwiFiber'
selectFibers(obj, roix=NULL, roiy=NULL, roiz=NULL, mask=NULL,
                    minlength=1)
## S4 method for signature 'dwiFiber'
reduceFibers(obj, maxdist=1, ends=TRUE)

## S4 method for signature 'dwiFiber,dwiFiber'
combineFibers(obj, obj2)

## S4 method for signature 'dwiFiber,dwiFiber'
touchingFibers(obj, obj2, maxdist=1, combine=FALSE)
```

## Arguments

| | |
|---|---|
| obj | An object of class "dtiTensor", "dtiIndices", or "dwiMixtensor" for tracking() and "dwiFiber" for selectFiber(), conbineFibers(). |
| obj2 | An object of class "dwiFiber" for combineFibers(). |
| roix | Indices defining the ROI in x direction. Currently min/max is used to define ROIx |
| roiy | Indices defining the ROI in y direction. Currently min/max is used to define ROIy |
| roiz | Indices defining the ROI in z direction. Currently min/max is used to define ROIz |
| mask | Mask defining seed points for tracking |
| method | Method for fibre tracking. "LINEPROP" is simple line propagation algorithm which is the default. |

| | |
|---|---|
| minfa | Minimal FA to follow the tracks. default 0.3 |
| maxangle | Maximal angle between fiber in adjacent voxels. default 30 degree. |
| subsample | Subsampling order of the data to get more dense fibre tracks. Note, that objects become very(!) large. |
| minlength | Minimal length of fibers to be selected. |
| maxdist | Maximal supremum distance between fibers in mm |
| ends | Logical: Use only endpoints of shorter fibers for distance (TRUE) or compute distances using full fiber-length (FALSE). Default (TRUE) removes more fibers and is significantly faster. |
| mincompartsize | Minimal size of a compartment in dwiMixtensor that will be used in fiber tracking. |
| combine | If combine=TRUE fibers selected from obj are combined with the fibers from obj2. |

## Value

The function returns an object of class dwiFiber.

## Methods

**obj = "dtiTensor"** Fiber tracking is performed on the estimated vector field of principal diffusion direction using the method method. Currently only line propagation is implemented. The resulting tracks can be visualized using function show3d.

**obj = "dtiIndices"** Fiber tracking is performed on the estimated vector field of principal diffusion direction using the method method. Currently only line propagation is implemented. The resulting tracks can be visualized using function show3d.

**obj = "dwiMixtensor"** Fiber tracking is performed on the estimated vector fields of diffusion direction in the mixed tensor model using the method method. Currently only line propagation is implemented. The resulting tracks can be visualized using function show3d.

**obj = "dwiFiber"** selectFibers produces a dwiFiber-object containing all fibers that cross the region of interest and exceed a minimum length. reduceFibers eliminates all fibers that are within a maximum supremum distance of maxdist mm of a longer fiber. reduceFibers allows to reduce the size of a dwiFiber-object considerably but is slow !

signature(obj1 = "dwiFiber", obj2 = "dwiFiber") combineFibers produces a dwiFiber-object containing all fibers that are in one of the supplied objects. touchingFibers takes all fibers from obj that have a minimum distance to a fiber in obj2 of less than maxdist. If combine=TRUE these fibers are combined with the fibers from obj2.

## Author(s)

Karsten Tabelow <tabelow@wias-berlin.de>, Joerg Polzehl <polzehl@wias-berlin.de>

## References

J. Polzehl, K. Tabelow (2019). Magnetic Resonance Brain Imaging: Modeling and Data Analysis Using R. Springer, Use R! series. Doi:10.1007/978-3-030-29184-6.

## See Also

dtiTensor,dtiIndices,dwiFiber, show3d, summary, print

# Index


∗ **IO**
  medinria, 34
  readDWIdata, 39
∗ **array**
  AdjacencyMatrix, 5
∗ **classes**
  dwi-class, 17
∗ **cluster**
  AdjacencyMatrix, 5
∗ **datasets**
  colqFA, 8
  optgrad, 35
  optgradients, 35
  polyeder, 37
∗ **hplot**
  plot-methods, 35
  show3d-methods, 43
∗ **iplot**
  show3d-methods, 43
∗ **manip**
  combineDWIdata, 9
  dtiIndices-methods, 14
  dtiTensor-methods, 15
  dwiMixtensor-methods, 25
  dwiQball-methods, 27
  dwiRiceBias-methods, 28
  extract-methods, 30
  getsdofsb-methods, 33
  showFAColorScale, 47
  subsetg, 47
∗ **methods**
  dkiTensor-methods, 10
  dti.smooth-methods, 12
  dtiTensor-methods, 15
  dwi.smooth-methods, 22
  dwiMD-methods, 24
  dwiMixtensor-methods, 25
  dwiQball-methods, 27
  dwiRiceBias-methods, 28
  dwiSqrtODF-methods, 29
  extract-methods, 30
  getmask-methods, 32
  getsdofsb-methods, 33
  plot-methods, 35
  print-methods, 38
  sdpar-methods, 41
  setmask-methods, 42
  show-methods, 43
  show3d-methods, 43
  summary-methods, 48
  tracking-methods, 49
∗ **misc**
  dti.options, 11
∗ **models**
  dtiIndices-methods, 14
  dtiTensor-methods, 15
  dwiMixtensor-methods, 25
  dwiQball-methods, 27
  dwiSqrtODF-methods, 29
∗ **model**
  dkiTensor-methods, 10
  dwiMD-methods, 24
∗ **package**
  dti-package, 3
∗ **smooth**
  awssigmc, 6
  dti.smooth-methods, 12
  dwi.smooth-methods, 22
∗ **utilities**
  print-methods, 38
  show-methods, 43
  summary-methods, 48
[,ANY-method *(*extract-methods*)*, 30
[,dkiIndices-method *(*extract-methods*)*, 30
[,dkiTensor-method *(*extract-methods*)*, 30
[,dtiData-method *(*extract-methods*)*, 30
[,dtiIndices-method *(*extract-methods*)*,

30
[,dtiTensor-method (extract-methods), 30
[,dwiMixtensor-method (extract-methods), 30
[,dwiQball-method (extract-methods), 30
[-methods (extract-methods), 30

AdjacencyMatrix, 5
aflsigmc (awssigmc), 6
afsigmc (awssigmc), 6
awslsigmc (awssigmc), 6
awssigmc, 6

colqFA, 8, *47*
combineDWIdata, 9, *48*
combineFibers (tracking-methods), 49
combineFibers,dwiFiber,dwiFiber-method (tracking-methods), 49
combineFibers-methods (tracking-methods), 49

dkiIndices, *11*
dkiIndices (dkiTensor-methods), 10
dkiIndices,ANY-method (dkiTensor-methods), 10
dkiIndices,dkiTensor-method (dkiTensor-methods), 10
dkiIndices-class (dwi-class), 17
dkiIndices-methods (dkiTensor-methods), 10
dkiTensor, *11*
dkiTensor (dkiTensor-methods), 10
dkiTensor,ANY-method (dkiTensor-methods), 10
dkiTensor,dtiData-method (dkiTensor-methods), 10
dkiTensor-class (dwi-class), 17
dkiTensor-methods, 10
dti (dti-package), 3
dti-package, 3
dti.options, 11
dti.smooth, *32*, *40*, *42*, *46*
dti.smooth (dti.smooth-methods), 12
dti.smooth,ANY-method (dti.smooth-methods), 12
dti.smooth,dtiData-method (dti.smooth-methods), 12
dti.smooth,dtiTensor-method (dti.smooth-methods), 12
dti.smooth-methods, 12
dtiData, *9*, *11*, *13*, *16*, *22–24*, *26*, *28*, *29*, *32*, *33*, *37*, *38*, *40*, *42*, *43*, *48*
dtiData (readDWIdata), 39
dtiData-class (dwi-class), 17
dtiIndices, *13*, *15*, *32*, *37*, *38*, *43*, *46*, *48*, *51*
dtiIndices (dtiIndices-methods), 14
dtiIndices,ANY-method (dtiIndices-methods), 14
dtiIndices,dtiTensor-method (dtiIndices-methods), 14
dtiIndices-class (dwi-class), 17
dtiIndices-methods, 14
dtiTensor, *13*, *15*, *16*, *24*, *28*, *32*, *34*, *37*, *38*, *42*, *43*, *46*, *48*, *51*
dtiTensor (dtiTensor-methods), 15
dtiTensor,ANY-method (dtiTensor-methods), 15
dtiTensor,dtiData-method (dtiTensor-methods), 15
dtiTensor-class (dwi-class), 17
dtiTensor-methods, 15
dwi-class, 17
dwi.smooth (dwi.smooth-methods), 22
dwi.smooth,ANY-method (dwi.smooth-methods), 22
dwi.smooth,dtiData-method (dwi.smooth-methods), 22
dwi.smooth-methods, 22
dwi.smooth.ms (dwi.smooth-methods), 22
dwi.smooth.ms,ANY-method (dwi.smooth-methods), 22
dwi.smooth.ms,dtiData-method (dwi.smooth-methods), 22
dwiFiber, *6*, *38*, *43*, *48*, *50*, *51*
dwiFiber-class (dwi-class), 17
dwiMD (dwiMD-methods), 24
dwiMD,ANY-method (dwiMD-methods), 24
dwiMD,dtiData-method (dwiMD-methods), 24
dwiMD,dtiTensor-method (dwiMD-methods), 24
dwiMD-methods, 24
dwiMixtensor, *16*, *26*, *28*, *32*, *37*, *38*, *43*, *48*
dwiMixtensor (dwiMixtensor-methods), 25
dwiMixtensor,ANY-method (dwiMixtensor-methods), 25
dwiMixtensor,dtiData-method (dwiMixtensor-methods), 25

dwiMixtensor-class (dwi-class), 17
dwiMixtensor-methods, 25
dwiMtCombine (dwiMixtensor-methods), 25
dwiMtCombine,ANY-method (dwiMixtensor-methods), 25
dwiMtCombine,dwiMixtensor,dwiMixtensor-method (dwiMixtensor-methods), 25
dwiMtCombine-methods (dwiMixtensor-methods), 25
dwiQball, *29*, *32*, *38*, *43*, *48*
dwiQball (dwiQball-methods), 27
dwiQball,ANY-method (dwiQball-methods), 27
dwiQball,dtiData-method (dwiQball-methods), 27
dwiQball-class (dwi-class), 17
dwiQball-methods, 27
dwiRiceBias (dwiRiceBias-methods), 28
dwiRiceBias,ANY-method (dwiRiceBias-methods), 28
dwiRiceBias,dtiData-method (dwiRiceBias-methods), 28
dwiRiceBias-methods, 28
dwiSqrtODF (dwiSqrtODF-methods), 29
dwiSqrtODF,ANY-method (dwiSqrtODF-methods), 29
dwiSqrtODF,dtiData-method (dwiSqrtODF-methods), 29
dwiSqrtODF-methods, 29

extract (extract-methods), 30
extract,ANY-method (extract-methods), 30
extract,dtiData-method (extract-methods), 30
extract,dtiIndices-method (extract-methods), 30
extract,dtiTensor-method (extract-methods), 30
extract,dwiMixtensor-method (extract-methods), 30
extract,dwiQball-method (extract-methods), 30
extract-methods, 30

getmask (getmask-methods), 32
getmask,ANY-method (getmask-methods), 32
getmask,array-method (getmask-methods), 32
getmask,dtiData-method (getmask-methods), 32
getmask-methods, 32
getsdofsb (getsdofsb-methods), 33
getsdofsb,ANY-method (getsdofsb-methods), 33
getsdofsb,dtiData-method (getsdofsb-methods), 33
getsdofsb-methods, 33

icosa0 (polyeder), 37
icosa1 (polyeder), 37
icosa2 (polyeder), 37
icosa3 (polyeder), 37
icosa4 (polyeder), 37

MedINRIA (medinria), 34
medinria, *13*, *15*, *16*, *26*, *28*, 34
medinria2tensor (medinria), 34

optgrad, 35
optgradients, 35

plot, *37*
plot,ANY-method (plot-methods), 35
plot,dkiIndices-method (plot-methods), 35
plot,dtiData-method (plot-methods), 35
plot,dtiIndices-method (plot-methods), 35
plot,dtiTensor-method (plot-methods), 35
plot,dwi-method (plot-methods), 35
plot,dwiFiber-method (plot-methods), 35
plot,dwiMixtensor-method (plot-methods), 35
plot-methods, 35
polyeder, 37
print, *38*, *51*
print,ANY-method (print-methods), 38
print,dkiIndices-method (print-methods), 38
print,dkiTensor-method (print-methods), 38
print,dtiData-method (print-methods), 38
print,dtiIndices-method (print-methods), 38
print,dtiTensor-method (print-methods), 38
print,dwi-method (print-methods), 38

print,dwiFiber-method (print-methods), 38
print,dwiMixtensor-method (print-methods), 38
print,dwiQball-method (print-methods), 38
print-methods, 38

readDWIdata, *9*, *11*, *13*, *16*, *22*, *26*, *28*, *29*, *32*, 39, *42*, *48*
reduceFibers (tracking-methods), 49
reduceFibers,dwiFiber-method (tracking-methods), 49

sdpar, *32*
sdpar (sdpar-methods), 41
sdpar,ANY-method (sdpar-methods), 41
sdpar,dtiData-method (sdpar-methods), 41
sdpar-methods, 41
selectFibers (tracking-methods), 49
selectFibers,dwiFiber-method (tracking-methods), 49
selectFibers-methods (tracking-methods), 49
setmask (setmask-methods), 42
setmask,ANY-method (setmask-methods), 42
setmask,dtiData-method (setmask-methods), 42
setmask-methods, 42
show,ANY-method (show-methods), 43
show,dkiIndices-method (show-methods), 43
show,dkiTensor-method (show-methods), 43
show,dti-method (show-methods), 43
show,dtiData-method (show-methods), 43
show,dtiIndices-method (show-methods), 43
show,dtiTensor-method (show-methods), 43
show,dwiFiber-method (show-methods), 43
show,dwiMixtensor-method (show-methods), 43
show-methods, 43
show3d, *50*, *51*
show3d (show3d-methods), 43
show3d,ANY-method (show3d-methods), 43
show3d,dkiTensor-method (show3d-methods), 43
show3d,dtiData-method (show3d-methods), 43
show3d,dtiIndices-method (show3d-methods), 43
show3d,dtiTensor-method (show3d-methods), 43
show3d,dwiFiber-method (show3d-methods), 43
show3d,dwiMixtensor-method (show3d-methods), 43
show3d,dwiQball-method (show3d-methods), 43
show3d-methods, 43
showFAColorScale, 47
subsetg, *9*, 47
summary, *48*, *51*
summary,ANY-method (summary-methods), 48
summary,dkiIndices-method (summary-methods), 48
summary,dkiTensor-method (summary-methods), 48
summary,dtiData-method (summary-methods), 48
summary,dtiIndices-method (summary-methods), 48
summary,dtiTensor-method (summary-methods), 48
summary,dwi-method (summary-methods), 48
summary,dwiFiber-method (summary-methods), 48
summary,dwiMixtensor-method (summary-methods), 48
summary,dwiQball-method (summary-methods), 48
summary-methods, 48

tensor2medinria (medinria), 34
touchingFibers (tracking-methods), 49
touchingFibers,dwiFiber,dwiFiber-method (tracking-methods), 49
touchingFibers-methods (tracking-methods), 49
tracking (tracking-methods), 49
tracking,ANY-method (tracking-methods), 49
tracking,dtiIndices-method (tracking-methods), 49
tracking,dtiTensor-method (tracking-methods), 49
tracking,dwiMixtensor-method (tracking-methods), 49

tracking-methods, 49

writeBin, *40*