Package ‘dti’

March 25, 2013

Version 1.1-4

Date 2013-03-25

Title Analysis of diffusion weighted imaging (DWI) data

Author Karsten Tabelow <karsten.tabelow@wias-berlin.de> Joerg Polzehl <joerg.polzehl@wias-berlin.de>

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

Depends R (>= 2.14.0), methods, awsMethods (>= 1.0-1), adimpro, oro.nifti (>= 0.3.9), oro.dicom, rgl, gsl, parallel

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), structural adaptive smoothing in in case of (DTI) (K. Tabelow, J. Polzehl, V. Spokoiny, and H.U. Voss, Diffusion Tensor Imaging: Structural Adaptive Smoothing, Neuroimage 39(4), 1763-1773 (2008)), modeling for high angular resolution diffusion weighted imaging (HARDI) using Q-ball-reconstruction and tensor mixture models 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) 2005-2013 Weierstrass Institute for Applied Analysis and Stochastics.

URL http://www.wias-berlin.de/projects/matheon_a3

NeedsCompilation yes

Repository CRAN

Date/Publication 2013-03-25 15:41:10
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), structural adaptive smoothing in case of (DTI) (K. Tabelow, J. Polzehl, V. Spokoiny, and H.U. Voss, Diffusion Tensor Imaging: Structural Adaptive Smoothing, Neuroimage 39(4), 1763-1773 (2008)), modeling for high angular resolution diffusion weighted imaging (HARDI) using Q-ball-reconstruction and
tensor mixture models 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

Package: dti
Version: 0.9-3
Date: 2010-10-21
Depends: R (>= 2.5.0), adimpro, fmri, rgl
License: GPL (>=2)
URL: http://www.wias-berlin.de/projects/matheon_a3

The package is based on S4 classes and methods. For help on a specific topic use class ?class-name> for classes, methods ?method-name> for methods and ?function-name> for all other functions.

Index:

dti-class Classes "dti", "dtiData", "dtiTensor", "dtiIndices", "dwiQball", "dwiMixtensor" and "dwiFiber"

dtiData Read Diffusion Weighted Data from Image File

readDWIData Read Diffusion Weighted Data from Directory

dti.smooth Structural Adaptive Smoothing
dtiTensor-methods Methods for Function 'dtiTensor'
dtiIndices-methods Methods for Function 'dtiIndices'
dwiQball-methods Methods for Function 'dwiQball'
dwiMixtensor-methods Methods for Function 'dwiMixtensor'

extract-methods Methods for Functions 'extract' and "[
plot-methods Methods for Function 'plot'
show-methods Methods for Function 'show'
show3d-methods Methods for Function 'show3d'
print-methods Methods for Function 'print'
summary-methods Methods for Function 'summary'

tracking-methods Methods for Functions 'tracking', 'selectFibers' and 'reduceFibers'
tensor2nifti Write Tensor Data as NIFTI File
medinria2tensor Read Tensor Data from NIFTI File
Author(s)

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

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

References


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

Examples

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

---

**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 function assumes \( L \) to be known and estimates a global \( \sigma \) employing an assumption of local homogeneity for the noncentrality parameter \( \eta \).

Usage

```r
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 = 0.8, method=c("VAR","MAD"))
```
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 adaptive weights smoothing, used to reveal the underlying mean structure.

- **mask**
  restrict computations to voxel in mask, if `is.null(mask)` all voxel are used.

- **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.

- **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 $N_i > q_{qni}(N_i)$ are used for variance estimation. Should be larger than 0.5.

- **method**
  method for variance estimation, either "VAR" (variance) or "MAD" (mean absolute deviation).

Value

- a list with components

  - **sigma**
    either a scalar or a vector of estimated noise standard deviations.

  - **theta**
    the estimated mean structure

Author(s)

Jörg Polzehl <polzehl@wias-berlin.de>

References

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 identical.
**Value**

An object of class "dtiData".

**Author(s)**

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

**See Also**

dtiData, readDWIdata, dtiData, subset

---

**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**

```r
## 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)
dti.smooth-methods

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


http://www.wias-berlin.de/projects/matheon_a3/
dtiIndices-methods

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

Methods

Methods for Function ‘dtiIndices’ in Package ‘dti’

dtiIndices-methods

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>
Jörg Polzehl <polzehl@wias-berlin.de>

References


\url{http://www.wias-berlin.de/projects/matheon_a3/}

\section*{See Also}

\texttt{medinria, dtiTensor-methods, dtiTensor, dtiIndices}

\section*{Examples}

\begin{verbatim}
## Not run: demo(dti_art)
\end{verbatim}

\section*{dtiTensor-methods \textit{Methods for Function 'dtiTensor' in Package 'dti'}}

\subsection*{Description}

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

\subsection*{Usage}

\begin{verbatim}
## S4 method for signature 'dtiData'
dtiTensor(object, method=c("nonlinear", "linear"),
          mc.cores = setCores(, reprt = FALSE))
\end{verbatim}

\subsection*{Arguments}

\begin{itemize}
  \item \texttt{object} Object of class "dtiData"
  \item \texttt{method} Method for tensor estimation. May be "linear", or "nonlinear".
  \item \texttt{mc.cores} Number of cores to use. Defaults to number of threads specified for openMP, see documentation of package \texttt{awsMethods}. Our experience suggests to use 4-6 cores if available.
\end{itemize}

\subsection*{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). 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.

Author(s)

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

References


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

See Also
dtiData, readDWIData, dtiIndices-methods, medinria, dtiData, dtiTensor dwiMixtensor

Examples

## Not run: demo(dti_art)
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.
- `ddimo`: 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 image 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.
sdcoeff: 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 "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 y-direction

roiz: Object of class "integer", coordinate range of region of interest in z-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 fibers.

For class "dwiMixtensor":

model: Object of class "character", characterizes the type of the mixed tensor model. Currently 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
dwi-class

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 $S_i$ were larger than the corresponding $S_0$ values. These are probably motion effects or registration errors. Values are replaced by the corresponding (mean) $S_0$ 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 spherical 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>
Jörg Polzehl <polzehl@wias-berlin.de>
**References**


**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.

**Usage**

```r
## S4 method for signature 'dtiData'
dwi.smooth(object, kstar, lambda=2/zero.noslash, kappa/zero.noslash=NULL, ncoils=1, sigma=NULL, level=NULL, vred=4, xind=NULL, yind=NULL, verbose=FALSE, dist=1, model=c("Gapprox", "Gapprox2", "Chi", "Chi2"))
```

**Arguments**

- `object`: Object of class "dtiData"
- `kstar`: Number of steps in structural adaptation
- `lambda`: Scale parameter in adaptation
- `kappa`: determines amount of smoothing on the sphere. Larger values correspond to stronger smoothing on the sphere. If `kappa=NULL` a value is that corresponds to a variance 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.
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 observed and squared values, respectively.

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>
Jörg Polzehl <polzehl@wias-berlin.de>

See Also

dtiData, dtiData.

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, reltol=1e-6, 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"
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 under-estimation 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 "dtiData" 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 using R’s internal BFGS code with mixture weights (volumes of compartments corresponding to a tensor component) computed using the Lawson-Hannson NNLS code. 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>
Jörg Polzehl <polzehl@wias-berlin.de>

References


See Also

dtiData, readDWIData, medinria, dtiData, dwiMixtensor

Examples

```r
## 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 coefficient (ADC) with respect to a spherical harmonics orthonormal system from the DWI data contained in an object of class "dtiData".

**Usage**

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

object Object of class "dtiData"
what Determines quantity to estimate, coefficients of 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 spherical harmonics expansion
lambda nonnegative regularization parameter.

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 spherical 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>
Jörg Polzehl <polzehl@wias-berlin.de>

References


See Also
dtiData, readDWIData, dtiIndices-methods, medinria, dtiData, dtiTensor

Examples

## Not run: demo(dti_art)
Correction for Rician Bias assuming known variance parameter

Usage

```r
## 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 $2\times ncoils$ 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>
Jörg Polzehl <polzehl@wias-berlin.de>

See Also

dtiData, dtiTensor-methods, dwiMixtensor-methods, dtiData, dtiTensor, dwiMixtensor,
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, D=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.
- `D`: 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>
Jörg Polzehl <polzehl@wias-berlin.de>

References


See Also

dtiData, readDWIdata, dtiData, dwiQball
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

```r
## 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","evals","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("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)
```

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 code `"[" 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 function 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 "values", main direction of anisotropy "andir", the tensor "tensor" the estimated S0 image "s0", the values of the model selection criteria 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 "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 criteria BIC "bic" or AIC "aic" and/or 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 criteria 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
Methods for Function 'getmask' in Package 'dti'

Description

Create a mask containing voxel inside the head

Usage

```r
## 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
Estimate the noise standard deviation. Uses an assumption that the standard deviation is a linear function of the expected mean for image intensities. \( qA_0 \) and \( qA_1 \) define quantiles of observed image intensities that define the range of values where this assumption is made.

### Usage

```r
## S4 method for signature 'dtiData'
getsdofsb(object, qA/zero.noslash = .1, qA1 = .98, nsb = NULL, level = NULL)
```

### Arguments

- `object`: Object of class "dtiData"
- `qA/zero.noslash`: 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 \( qA_0 \)) and 8: upper bound (depending on \( qA_1 \)).

### 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 &lt;tabelow@wias-berlin.de&gt;
Jörg Polzehl &lt;polzehl@wias-berlin.de&gt;

### See Also

dtiData, dwi.smooth-methods, dtiData,
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>
Jörg Polzehl <polzehl@wias-berlin.de>

References

http://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**

```r
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**

```r
## 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=c(3,3,3,.3),
mgp=c(2,1,0), ...)

## S4 method for signature 'dtiTensor'
plot(x, y, slice=1, view="axial", quant=/zero.noslash, minfa=NULL, contrast.enh=1,
what="fa", qrange=c(.01,.99), xind=NULL, yind=NULL, zind=NULL,
mar=c(2,2,2,.2), mgp=c(2,1,0), ...)
```
## 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=c(2,2,2,.2),mgp=c(2,1,0), ...)

## 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=c(3,3,3,.3), mgp=c(2,1,0), ...)

## S4 method for signature 'dwiFiber'
plot(x, y, ...)

### 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 dti-Indices 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.

Author(s)

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

See Also
dtiIndices, dtiData, dtiTensor dwiMixtensor

Examples

## Not run: demo(dti_art)

### pmatrix

Parallelize columnwise computations on a matrix using forking (generalizes function 'pvec' from package 'parallel')

Description

parallelizes the execution of a function on matrix elements by splitting the matrix by columns and submitting each part to one core. The function must take a matrix argument, with computations depending on row entries. It creates a matrix with same number of columns as on input, but possibly different number of rows.
### Usage

```r
pmatrix(x, FUN, ..., mc.cores = setCores(,reprt=FALSE))
plmatrix(x, FUN, ..., mc.cores = setCores(,reprt=FALSE))
```

### Arguments

- **x**: matrix to operate on.
- **FUN**: function depending on a matrix argument, and providing a matrix valued result.
- **...**: see documentation for function `pvec`
- **mc.cores**: number of threads to use. Defaults to number of threads specified for openMP, see documentation of package `awsMethods`.

### Details

This functions are wrappers to functions `parCapply` and `parLapply` from `parallel`.

### Value

Returns a matrix containing the results for the individual columns of `x` as columns.

### Note

These functions are wrappers to functions `parCapply` and `parLapply` from `parallel`.

### Author(s)

Jörg Polzehl <polzehl@wias-berlin.de>

---

**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

`icosax`
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

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", "dwiMixtensor", "dwiQball" and "dwiFiber".

Usage

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

Arguments

x Object of class "dtiIndices", "dtiData", "dtiTensor", "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>
Jörg 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 medical 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,
xind = NULL, yind = NULL, zind = NULL, level = 0, mins0value = 1,
maxvalue = 32000, voxelext = NULL, orientation = c(0L, 2L, 5L),
rotation = NULL, pattern = NULL, 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
- 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.
- 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` determines `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>
Jörg Polzehl <polzehl@wias-berlin.de>

References

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

See Also

dti.smooth, dtiTensor-methods, dtiData

Examples

```r
### 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.
sdpar-methods

Usage

```r
## S4 method for signature 'dtiData'
sdpar(object, level=NULL, sdmethod="sd", 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, either "sd" or "mad".
- **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

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

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, 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>
Jörg Polzehl <polzehl@wias-berlin.de>

See Also

dtiIndices, dtiData, dtiTensor dwiMixtensor dwiQball dwiFiber
show3d-methods

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 '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, ...)

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=FALSE specifies a default depending on the content of argument what
(normalize <- switch(what,"tensor"=FALSE,"adc"=TRUE)).
show3d-methods

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>box</td>
<td>Logical, add a bounding box.</td>
</tr>
<tr>
<td>title</td>
<td>Either a character string specifying a title or a logical. If title==TRUE a default title characterizing the type of plot is generated.</td>
</tr>
<tr>
<td>method</td>
<td>method==1 and method==2 specify two different color schemes for directional FA-maps.</td>
</tr>
<tr>
<td>minfa</td>
<td>Minimal FA value for dtiTensor objects and for dwiMixtensor objects.</td>
</tr>
<tr>
<td>mask</td>
<td>Additional mask for dtiTensor objects.</td>
</tr>
<tr>
<td>minorder</td>
<td>Minimal order for dwiMixtensor objects.</td>
</tr>
<tr>
<td>mineo</td>
<td>Minimal effective order for dwiMixtensor objects.</td>
</tr>
<tr>
<td>fibers</td>
<td>If TRUE show fibers starting in voxel with (fa&gt;=\text{minfa}, \text{order}&gt;=\text{minorder}) and (eorder&gt;=\text{mineo}), the last two effective for dwiMixtensor objects only.</td>
</tr>
<tr>
<td>maxangle</td>
<td>Argument for fibertracking</td>
</tr>
<tr>
<td>level</td>
<td>Radius of sphere used as support for ODF visualisation</td>
</tr>
<tr>
<td>subdivide</td>
<td>Level of subdivisions for meshing, level 0:4 correspond to use of (c(12,42,162,642,2562)) vertices per tensor, respectively.</td>
</tr>
<tr>
<td>what</td>
<td>For dtiTensor-objects either &quot;tensor&quot; for visualization using ellipsoids, &quot;adc&quot; for Apparent Diffusion Coefficients or &quot;odf&quot; for the Orientation Density Function. For dwiMixtensor-objects possible specifications are &quot;odf&quot;, &quot;axis&quot; and &quot;both&quot;, with the latter superposing the estimated main directions on the estimated ODF. For &quot;axis&quot;(and &quot;both&quot;) the length of the axis corresponds to the mixture weights. For dtiData-objects choices are either &quot;data&quot; or &quot;adc&quot;.</td>
</tr>
<tr>
<td>odfsclale</td>
<td>Determines visualisation of the Orientation density function (ODF). For odfsclale=3 the ODF values are rescaled such that the volume of the displays objects is constant. odfsclale=1 uses the values of the ODF as radii in the corresponding vertice direction of the specified polyhedron. This can lead to extremely large volumes in case of one mixture component with high eccentricity. values of odfsclale inbetween 1 and 3 are possible and allow to balance between volume based visualization and emphasising highly structured ODF’s.</td>
</tr>
<tr>
<td>lwd</td>
<td>Linewidth for visualization of dtiIndices objects.</td>
</tr>
<tr>
<td>index</td>
<td>Either &quot;FA&quot; for fractional anisotropy index or &quot;GA&quot; for geodesic anisotropy index.</td>
</tr>
<tr>
<td>...</td>
<td>Additional parameters passed to function rgl.par from the rgl-package.</td>
</tr>
</tbody>
</table>

**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 transparency. The value of maxobjects limits the size of datacube and may be increased on hardware with suitable graphics capabilities.
### showFAColorScale

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 anisotropy 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 = "dtiIndices"** 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 colorscheme 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 = "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 transparency. 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>`
Jörg Polzehl `<polzehl@wias-berlin.de>`

### See Also

dtiIndices-methods, dti.smooth, dtiTensor, dtiIndices

### Examples

```r
## 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

```r
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 contains 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>
Jürg 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

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

**object**
Object of class "dti", "dtiData", "dtiTensor", "dwiMixtensor", "dtiIndices", "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", "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

---

### Description

The function provides fiber tracking of "dtiTENSOR", "dtiIndices", and "dwiMixtensor" objects.

### Usage

```r
## 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)
## 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)
```
Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>obj</td>
<td>An object of class &quot;dtiTensor&quot;, &quot;dtiIndices&quot;, or &quot;dwiMixtensor&quot; for tracking() and &quot;dwiFiber&quot; for selectFiber().</td>
</tr>
<tr>
<td>roix</td>
<td>Indices defining the ROI in x direction. Currently min/max is used to define ROIx</td>
</tr>
<tr>
<td>roiy</td>
<td>Indices defining the ROI in y direction. Currently min/max is used to define ROIy</td>
</tr>
<tr>
<td>roiz</td>
<td>Indices defining the ROI in z direction. Currently min/max is used to define ROIz</td>
</tr>
<tr>
<td>mask</td>
<td>Mask defining seed points for tracking</td>
</tr>
<tr>
<td>method</td>
<td>Method for fibre tracking. &quot;LINEPROP&quot; is simple line propagation algorithm which is the default.</td>
</tr>
<tr>
<td>minfa</td>
<td>Minimal FA to follow the tracks. default 0.3</td>
</tr>
<tr>
<td>maxangle</td>
<td>Maximal angle between fiber in adjacent voxels. default 30 degree.</td>
</tr>
<tr>
<td>subsample</td>
<td>Subsampling order of the data to get more dense fibre tracks. Note, that objects become very(!) large.</td>
</tr>
<tr>
<td>minlength</td>
<td>Minimal length of fibers to be selected.</td>
</tr>
<tr>
<td>maxdist</td>
<td>Maximal supremum distance between fibers in mm</td>
</tr>
<tr>
<td>ends</td>
<td>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.</td>
</tr>
</tbody>
</table>

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 !

Author(s)

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

dtiTensor, dtiIndices, dwiFiber, show3d, summary, print
Index

*Topic IO
  medinria, 27
  readDWIdata, 33

*Topic classes
  dwi-class, 12

*Topic datasets
  colqFA, 6
  optgrad, 28
  optgradients, 28
  polyeder, 31

*Topic hplot
  plot-methods, 28
  show3d-methods, 36

*Topic iplot
  show3d-methods, 36

*Topic manip
  combineDWIdata, 6
  dtiIndices-methods, 9
  dtiTensor-methods, 10
  dwiMixtensor-methods, 17
  dwiQball-methods, 19
  dwiRiceBias-methods, 21
  extract-methods, 23
  getsdofsb-methods, 26
  showFAColorScale, 39
  subsetg, 40

*Topic methods
  dti.smooth-methods, 7
  dtiTensor-methods, 10
  dwi.smooth-methods, 16
  dwiMixtensor-methods, 17
  dwiQball-methods, 19
  dwiRiceBias-methods, 21
  dwiSqrtODF-methods, 22
  extract-methods, 23
  getmask-methods, 25
  getsdofsb-methods, 26
  plot-methods, 28
  print-methods, 32
  sdpar-methods, 34
  show-methods, 36
  show3d-methods, 36
  summary-methods, 40
  tracking-methods, 41

*Topic models
  dtiIndices-methods, 9
  dtiTensor-methods, 10
  dwiMixtensor-methods, 17
  dwiQball-methods, 19
  dwiSqrtODF-methods, 22

*Topic package
dti-package, 2

*Topic smooth
  awssigmc, 4
  dti.smooth-methods, 7
  dwi.smooth-methods, 16

*Topic utilities
  print-methods, 32
  show-methods, 36
  summary-methods, 40
  [,ANY-method (extract-methods), 23
  [,dtiData-method (extract-methods), 23
  [,dtiIndices-method (extract-methods), 23
  [,dtiTensor-method (extract-methods), 23
  [,dwiMixtensor-method (extract-methods), 23
  [,dwiQball-method (extract-methods), 23
  [,dwiSqrtODF-method (extract-methods), 23
  awssigmc, 4
  colqFA, 6, 39
  combineDWIdata, 6, 40
  dti (dti-package), 2
  dti-package, 2
  dti.smooth, 25, 34, 35, 39
  dti.smooth (dti.smooth-methods), 7
summary, dtiTensor-method
  (summary-methods), 40
summary, dwi-method (summary-methods), 40
summary, dwiFiber-method
  (summary-methods), 40
summary, dwiMixtensor-method
  (summary-methods), 40
summary, dwiQball-method
  (summary-methods), 40
summary-methods, 40
tensor2medinria (medinria), 27
tracking (tracking-methods), 41
tracking, ANY-method (tracking-methods), 41
tracking, dtiIndices-method
  (tracking-methods), 41
tracking, dtiTensor-method
  (tracking-methods), 41
tracking, dwiMixtensor-method
  (tracking-methods), 41
tracking-methods, 41
writeBin, 34