Package 'dti'

October 12, 2009

Version 0.8-2

Date 2009-10-12

Title DTI/DWI Analysis

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

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

Depends R (>= 2.5.0), methods, adimpro, fmri, rgl

Suggests gsl

Description Diffusion Weighted Imaging 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 in the context of the diffusion tensor model (DTI). This includes the calculation of anisotropy measures and, most important, the implementation of our structural adaptive smoothing algorithm as described in K. Tabelow, J. Polzehl, V. Spokoiny, and H.U. Voss, Diffusion Tensor Imaging: Structural Adaptive Smoothing, Neuroimage 39(4), 1763-1773 (2008).

License GPL (>= 2)

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

Repository CRAN

Date/Publication 2009-10-12 10:42:20

R topics documented:

`dti-package` *DTI Analysis*

Description

Diffusion Weighted Imaging 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 in the context of the diffusion tensor model (DTI). This includes the calculation of anisotropy measures and, most important, the implementation of our structural adaptive smoothing algorithm as described in Tabelow et al. (2008).

Details

Package:	dti
Version:	0.5-7
Date:	2008-07-24
Depends:	R (>= 2.5.0), adimpro, fmri, rgl
License:	GPL (>=2)
Copyright:	2008 Weierstrass Institute for Applied Analysis and Stochastics.
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"`
`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'
extract-methods	Methods for Function 'extract'
plot-methods	Methods for Function 'plot'
show3d-methods	Methods for Function 'show3d'
print-methods	Methods for Function 'print'
summary-methods	Methods for Function 'summary'
tensor2medinria	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

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

Polzehl, J. and Tabelow, K. (2008) Structural Adaptive Smoothing in Diffusion Tensor Imaging: the R package dti, WIAS-Preprint 1382.

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

Examples

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

---

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, sli
```

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 minanindex as corresponding quantile of FA if is.null(minanindex)
minanindex	minimal anisotropy index 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"
varmethod	Specifies the method for estimating the error variance. May be varmethod=="replicates", or "residuals".
varmodel	Specifies the model for the variance. May be "global", or "local".
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⟩
Jörg Polzehl ⟨polzehl@wias-berlin.de⟩

References

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

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

http://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, which)
```

Arguments

`object`	Object of class `"dtiTensor"`
`which`	Indices should be created, currently not implemented.

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⟩

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="nonlinear", varmethod="replicates", varmodel="local")
```

Arguments

`object`	Object of class `"dtiData"`
`method`	Method for tensor estimation. May be `"linear"`, or `"nonlinear"`.
`varmethod`	Specifies the method for estimating the error variance. May be `"replicates"`.
`varmodel`	Specifies the model for the variance. May be `"global"`, or `"local"`.

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⟩
Jörg Polzehl ⟨polzehl@wias-berlin.de⟩

References

K. Tabelow, J. Polzehl, H.U. Voss, and V. Spokoiny. *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.

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

See Also

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

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.

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.

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 "dwiQball":

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

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", either "Qball" or "ADC". Indicates if the object contains coefficients of the orientation density function (Qball) or the apparent diffusion coefficient (ADC). Coefficients are computed with respect to spherical harmonics of the specified order.

Methods

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

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.

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

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⟩

See Also

dtiData, readDWIdata, sdpar-methods, dtiTensor-methods, dti.smooth-methods, dtiIndices-methods, dwiQball-methods, plot-methods, print-methods, summary-methods, extract-methods

---

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="Qball", order=4, lambda=0)
```

Arguments

`object`	Object of class `"dtiData"`
`what`	Determines quantity to estimate, either coefficients ot the orientation density function (ODF) (`what="Qball"` 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.

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⟩
Jörg 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.

See Also

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

Examples

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

`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="data", xind=TRUE, yind=TRUE, zind=TRUE)
## S4 method for signature 'dtiTensor':
extract(x, what="tensor", xind=TRUE, yind=TRUE, zind=TRUE)
## S4 method for signature 'dtiIndices':
extract(x, what=c("fa","andir"), xind=TRUE, yind=TRUE, zind=TRUE)
## S4 method for signature 'dwiQball':
extract(x, what="sphcoef", 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 '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.

Value

For function `extract` a list with components carrying the names of the options specified in argument `what`. For indexing function, 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) or all images (Si) 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 an array 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) 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 = "dwiQball" Returns an array containing the specified statistics, the estimated coefficients with respect to the selected spherical harmonics basis, the estimated S0 image (s0) 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⟩
Jörg Polzehl ⟨polzehl@wias-berlin.de⟩

See Also

dtiData, dtiTensor, dtiIndices dwiQball

---

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⟩
Jörg 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

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

See Also

dtiTensor, dtiTensor-methods dtiIndices-methods

Examples

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

---

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

Description

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

Usage

```
## S4 method for signature 'dtiData':
plot(x, y, slice=1, gradient=NULL, view= "axial", show=TRUE, density=FALSE, xind=
## S4 method for signature 'dtiTensor':
plot(x, y, slice=1, view="axial", quant=0, minanindex=NULL, contrast.enh=1, what=
## S4 method for signature 'dtiIndices':
plot(x, y, slice=1, view= "axial", method=1, quant=0, minanindex=NULL, show=TRUE,
```

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(minanindex) specify minanindex as corresponding quantile of the fractal anisotropy (FA) index.
minanindex	Display only information for voxel with FA>minanindex
show	Visualize information in a graphics device (for classes "dtiData" and "dtiIndices" 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	If what="GA" use geodesic anisotropy (GA) instead of fractional anisotropy (FA).
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==3 specifies visualization of dtiIndices using color coded shape parameters. The resulting image of class "adimpro" from package **adimpro** is returned.

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

Author(s)

Karsten Tabelow ⟨tabelow@wias-berlin.de⟩
Jörg Polzehl ⟨polzehl@wias-berlin.de⟩

See Also

dtiIndices, dtiData, dtiTensor

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

```
icosax
```

Format

a list with components

vertices	array of dimension `c(3,nv)` containing cartesian coordinate of the `nv` vertices.
indices	Indices of vertices that define surface triangles of the polyeder.
edges	Indices of vertices that define edges of the polyeder.
nv	number of vertices
ne	number of edges
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" and "dtiIndices".

Usage

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

Arguments

x	Object of class dtiIndices, dtiData, dtiTensor 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 dwiQball

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, xind = NULL, yind = NULL, zind = NULL, level =
readDWIdata(gradient, dirlist, format, nslice, order = NULL, xind = NULL, yind =
```

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).
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

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⟩
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

```
## 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="sd",interactive=TRUE)
```

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.

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⟩
Jörg Polzehl ⟨polzehl@wias-berlin.de⟩

See Also

dtiData, readDWIdata, dti.smooth, dtiTensor,

Examples

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

---

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" and "dtiIndices".

Usage

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

Arguments

object	Object of class dtiIndices, dtiData or dtiTensor

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".

Author(s)

Karsten Tabelow ⟨tabelow@wias-berlin.de⟩
Jörg Polzehl ⟨polzehl@wias-berlin.de⟩

See Also

dtiIndices, dtiData, dtiTensor

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,nx=NULL,ny=NULL,nz=NULL,center=NULL,scale=.5,bgcolor="black",add=FALSE
## S4 method for signature 'dtiTensor':
show3d(obj,nx=NULL,ny=NULL,nz=NULL,center=NULL,method=1,level=0,scale=.5,bgcolor=
## S4 method for signature 'dtiIndices':
show3d(obj, index="FA", nx=NULL, ny=NULL, nz=NULL, center=NULL, method=1, level=0
## S4 method for signature 'dwiQball':
show3d(obj,nx=NULL,ny=NULL,nz=NULL,center=NULL,scale=0.5,bgcolor="black",add=FALS
```

Arguments

obj	An object of class dtiTensor or dtiIndices
nx	Number of voxel in x-direction
ny	Number of voxel in y-direction
nz	Number of slices
center	Vector of length 3 specifying the center of the data cube (class dtiData or dtiTensor) or center of display (class dtiIndices)
scale	Scale factor for the size of polyeders dtiData and ellipsoids (dtiTensor)
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.
power	Exponent in visualization, defaults to 1. heighter values may be used for increased contrast.
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(tolower(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.
`level`	Minimal FA value of tensors.
`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 or `"ADC"` for Apparent Diffusion Coefficients. For dtiData-objects choices are either `"data"` or `"ADC"`.
`lwd`	Linewidth for visualization of dtiIndices objects.
`index`	Eiter `"FA"` for fractional anisotropy index or `"GA"` for geodesic anisotropy index.
`...`	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 = "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.

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

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

---

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", "dtiIndices" or "dwiQball".
...	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" and "dtiIndices".

Author(s)

Karsten Tabelow ⟨tabelow@wias-berlin.de⟩
Jörg Polzehl ⟨polzehl@wias-berlin.de⟩

See Also

dtiIndices,] dtiData, dtiTensor dwiQball

Index