Weierstrass Institute for
Applied Analysis and Stochastics

DFG-Forschungszentrum MATHEON
Mathematik für Schlüsseltechnologien

Diffusion weighted imaging: the dti package

Jörg Polzehl and Karsten Tabelow

UseR! Tutorial: Medical image analysis for structural and functional MRI

Content

1 DWI physics and data acquisition

- The geometry of the data

MRI and DWI physics

- Strong magnetic field (usually $1.5 - 3$ Tesla(T), up to 10.5 T)
- Radio frequency pulse at Lamour-frequency
- Measuring relaxation times (T_1 (z-direction), T_2 (phase coherence in x-y), and $T_2^\star$) of magnetic spin excitation in receiver coil(s)
- Image generation by 2D-FFT

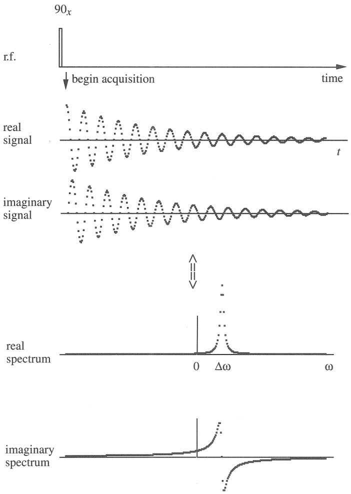
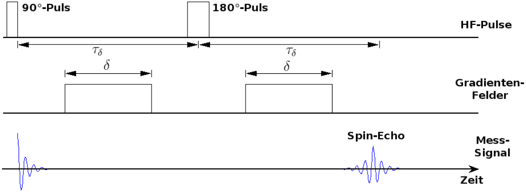


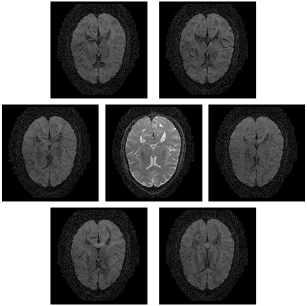
Fig. 2.7 Free Induction Decay (F I D) following a single 90° r.f. pulse. The real and imaginary parts of the signal correspond to the in-phase and quadrature receiver outputs. The signal is depicted with receiver phase $\phi = 0$ and, on complex Fourier transformation, gives real absorption and imaginary dispersion spectra at the offset frequency, $\Delta\omega = \omega_0 - \omega$.

- Different usage of gradient magnetic fields
- Measuring diffusion of water ...
- ... for a direction $\vec{g}$ specified by gradient magnetic fields
- Restricted water diffusion within neuronal fiber bundles
- Focus on brain white matter anatomy instead of grey matter functionality (fMRI)



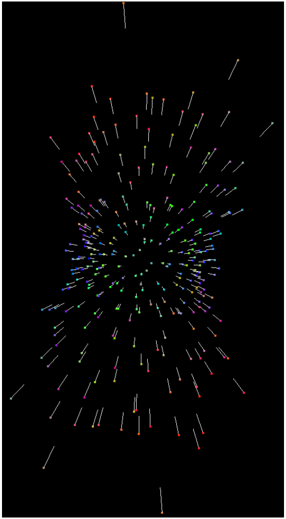
DWI - Data Properties

- Noise (Rician distribution)
- Motion artifacts, magnetic field inhomogeneity, multicoil measurement, correlated data, ghosts
- Partial volume effects: $f(V) = \int_V f(\vec{x}) d^3x$
- Sensitive at μm scale vs. measurement at mm scale
- $S(\vec{g}) = S_0 \exp(-bD(\vec{g}))$ (apparent diffusion coefficient ADC $D(\vec{g})$)



Geometry of DWI Data

- 3D + S^2 data
- Measurements of integral values on a regular grid of voxel (size $\approx 1mm^3$)
- Structures of interest have a diameter of $10-30\mu m$ and length of up to $10cm$
- $1-30$ measurements without gradient field (S_0)
- $12-180$ measurements with additional gradient ($S(\vec{g})$)
- gradient directions uniformly sampled from the sphere S^2
- Observations live in an 3D orientation score $R^3 \rtimes S^2$.



ADC $-log(S_b/S_0)$, 140 gradients in one voxel

2 The dti package

- Data input
- Class DWI

Package: dti
Version: 0.9-2.1
Date: 2010-07-15
Title: Analysis of diffusion weighted imaging (DWI) data
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: ... see package ...
License: GPL (>= 2)
Copyright: This package is Copyright (C) 2005-2010
Weierstrass Institute for Applied Analysis and Stochastics (WIAS)
URL: http://www.wias-berlin.de/projects/matheon_a3

- Data usually as DICOM, NIFTI, ANALYZE or AFNI-Files
- Functions

```
readDWIdata(gradient, dirlist, format, nslice = NULL, order = NULL, xind=NULL,
        yind=NULL, zind=NULL, level=0, mins0value=0, maxvalue=10000,
        voxelext=NULL, orientation=c(0,2,5), rotation=diag(3))
dtiData(gradient,imagefile,ddim,xind=NULL,yind=NULL,zind=NULL,level=0,
        mins0value=0,maxvalue=10000,voxelext=c(1,1,1),
        orientation=c(0,2,5),rotation=diag(3))
```

- Main arguments:
  - gradient - gradient directions (including zero gradients for S0 images)
  - dirlist - list of directories containing the data files
  - format - one of ”DICOM”, ”NIFTI”, ”ANALYZE”, or ”AFNI”
  - imagefile - name of data image file (binary 2Byte integers)
  - ddim - array dimensions
  - xind, yind, zind - indices for subimages
  - mins0value - threshold for a mask defined by S0 intensity

Examples

- Reading imaging data (DICOM) with restriction to a subregion

```
R> grad <- read.table("gradient.txt")
R> dwiobj <- readDWIdata(grad, c("datadir/s0011/","datadir/s0012/"),
                   "DICOM",72,xind=129:196,yind=129:196,zind=25:30)
R> dwiobj <- sdpar(dwiobj) # interactive choice of mins0value and variance est.
R> save(dwiobj,file="s00.rsc")
```

- Creating a binary file inbetween (using functions from package fmri):

```
R> con <- file("S-all","wb")
R> for (gg in 1:16) {
R>    data <- read.ANALYZE(filename[gg])
R>    writeBin(as.integer(extract.data(data)),con,2)
R> }
R> close(con)
R> dwiobj <- dtiData(grad,"S-all",ddim)
```

- both create an object of class dwiData

Class definition:

```
setClass("dwi",
    representation(.Data = "list",
          call = "list", # object history
          gradient = "matrix", # gradient matrix (3xngrad)
          btb    = "matrix", # matrix (6xngrad)
          ngrad  = "integer", # number of gradients
          s0ind  = "integer", # indices of s0 images
          replind = "integer", # replications in gradient design
          ddim   = "integer", # actual image dimension
          ddim0  = "integer", # initial image dimension
          xind   = "integer", # x-index of actual cube
          yind   = "integer", # y-index of actual cube
          zind   = "integer", # z-index of actual cube
          voxelext = "numeric", # voxelsize
          level  = "numeric", # threshold for mask
          orientation = "integer", # orientation for data cube
          rotation = "matrix", # rotation matrix for coordinate system (not yes used)
          source = "character") # image source
    )
```

3 Diffusion Tensor Imaging (DTI)

- Diffusion based diagnostics
- Classes dtiTensor and dtiIndices
- Manipulating Objects
- Visualization and Output
- Smoothing in DTI

- Assumes homogeneity within a voxel
- Diffusion characterized by a symmetric positive semi-definite 3×3 matrix $\mathscr{D}$
- Nonlinear Model

$$\begin{aligned} S_i(\vec{g}) &\sim Rice(\zeta_i(\vec{g}), \sigma_i^2) \\ \zeta_i(\vec{g}) &= \theta_i \exp(-b\vec{g}^\top \mathscr{D}_i \vec{g}) \end{aligned}$$

- Nonlinear regression with positivity constraints

$$\begin{aligned} \mathbf{R}(\zeta, \theta, \mathscr{D}) &= \sum_j \frac{(\zeta(\vec{g}_j) - \theta \exp(-b\vec{g}_j^\top \mathscr{D}_i \vec{g}_j))^2}{\sigma_{j,i}^2} \\ \begin{pmatrix} \hat{\theta}_i \\ \hat{\mathscr{D}}_i \end{pmatrix} &= \arg\min_{\theta, \mathscr{D}} \mathbf{R}(\hat{\zeta}_i, \theta, \mathscr{D}) \end{aligned}$$

- Call

  dtiTensor(dwiDataobj, ...)

- Mean diffusivity (MD) $Tr(\mathscr{D}) = \mu_1 + \mu_2 + \mu_3$
- Fractional anisotropy (FA)

$$FA = \sqrt{\frac{3}{2}}\sqrt{\frac{(\mu_1 - \langle\mu\rangle)^2 + (\mu_2 - \langle\mu\rangle)^2 + (\mu_3 - \langle\mu\rangle)^2}{\mu_1^2 + \mu_2^2 + \mu_3^2}}, \qquad \langle\mu\rangle = \frac{1}{3}\sum_i \mu_i$$

- Geodesic anisotropy (GA) (Fletcher (2004), Corouge (2006))

$$GA = (\sum_{i=1}^{3}(log(\mu_i) - \overline{log(\mu)})^2)^{1/2}, \quad \overline{log(\mu)} = \frac{1}{3}\sum_{i=1}^{3} log(\mu_i)$$

- Bary-coordinates (characterizing spherical, planar and linear shape)

$$C_s = \frac{\mu_3}{\langle\mu\rangle} \quad C_p = \frac{2(\mu_2 - \mu_3)}{3\langle\mu\rangle} \quad C_l = \frac{(\mu_1 - \mu_2)}{3\langle\mu\rangle}$$

- Call

  dtiIndices(dtiTensorobj, ...)

- Demo

  R> demo(dti_art)

  uses 3 different data sets generated from artificial tensor configurations

- Example data set

```
R> library(dti)
R> load("s00.rsc")
R> dtobj <- dtiTensor(dwiobj)
start nonlinear regression Wed Jul 14 21:56:44 2010
successfully completed nonlinear regression  Wed Jul 14 21:57:24 2010
estimated spatial correlations Wed Jul 14 21:57:31 2010
first order  correlation in x-direction 0.544
first order  correlation in y-direction 0.448
first order  correlation in z-direction 0.345
estimated corresponding bandwidths Wed Jul 14 21:57:31 2010
estimated scale information Wed Jul 14 21:57:32 2010
R> dtind <- dtiIndices(dtobj)
R> save(dtobj,dtind,file="s00tens.rsc")
```

Classes dtiTensor and dtiIndices

Class definitions:

```
setClass("dtiTensor",
        representation(method = "character",# either "nonlinear" or "linear"
                       D      = "array",# estimated tensors dimension c(6,ddim)
                       th0    = "array",# estimated base intensity
                       sigma  = "array",#
                       scorr  = "array",# estimated spatial correlations
                       bw     = "numeric",# bandwidth characterizing spatial corr.
                       mask   = "array",# mask of aktiv voxel
                       hmax   = "numeric",# maximal bandwidth for smoothing
                       outlier = "numeric",# voxel with inappropriate S0
                       scale  = "numeric"),# scale info for visualization
        contains=c("list","dwi"))

setClass("dtiIndices",
        representation(method = "character",#either "nonlinear" or "linear"
                       fa     = "array",# Fractional anisotropy (FA)
                       ga     = "array",# Geodesic anisotropy (GA)
                       md     = "array",# Mean diffusivity
                       andir  = "array",# Main anisotropy directions
                       bary   = "array"),# bary coordinates
        contains=c("list","dwi"))
```

Summarizing and manipulating objects

- Data preprocessing: Function sdpar

  dwiDataobj <- sdpar(dwiDataobj)

  estimates error variances, correlations, interactive thresholds
- Methods print, show and summary for all objects of dwi based classes
- Index operations: Methods “[” exist for objects of all dwi based classes.

  reducedobj <- obj[xind,yind,zind]

  reducedobj has same class as obj.
- Extraction of information by

  statlist <- extract(obj,what, xind=TRUE, yind=TRUE, zind=TRUE)

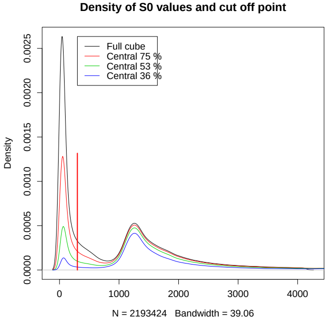
  Arguments:
  - obj - object of class dwiData, dtiTensor, dtiIndices
  - what - character vector specifying which information to extract
    - dwiData: any of “data”, “s0”, “sb”, “gradient”, “btb”, “siq”
    - dtiTensor: any of “fa”, “ga”,“md”,“andir”, “evalues”,“s0”,“mask”,“outlier”
    - dtiIndices: any of “fa”, “ga”,“md”,“andir”, “bary”
  - xind, yind, zind - index vectors defining a subcube

  statlist will be a list with components containing the requested information

Examples

```
R> library(dti)
R> load("nydwdatareduced.rsc")
R> dwobj <- sdpar(dwobj,level=300)
R> summary(dwobj)
 Object of class dtiData
 Generated by calls    :
[[1]]
readDWIdata(bvec, paste(filepre, "/s0004", sep = ""), format = "DICOM",
   xind = 48:204, yind = 19:234, nslice = 66, voxelext = voxelext)

 Source-Filename      : ~tabelow/DATA/dti_ny/data/e006353/s0004
 Dimension            : 157x216x66
 Number of Gradients  : 150
 Voxel extensions     : 0.9x0.9x1.8
 Index of S0-Images   : 1x2x3x4x5x6x7x8x9x10
 Quantiles of S0-values:
  0%   25%   50%   75%  100%
   1    67   693  1540 10000
 Mean S0-value        : 998
 Threshold for mask   : 300
R> nytens <- dtiTensor(dwobj)
R> save(nytens,file="nytens.rsc")
```



Selecting the cut off level by sdpar

Examples

```
R> library(dti)
R> load("nytens.rsc")
R> summary(nytens)
 Object of class dtiTensor
 Generated by calls    :
[[1]]
readDWIdata(bvec, paste(filepre, "/s0004", sep = ""), format = "DICOM",
   xind = 48:204, yind = 19:234, nslice = 66, voxelext = voxelext)
[[2]]
dtiTensor(dwobj)
 Source-Filename       : ~tabelow/DATA/dti_ny/data/e006353/s0004
 Dimension             : 157x216x66
 Number of Gradients   : 150
 Voxel extensions      : 0.9x0.9x1.8
 Quantiles of S0-values:
     0%     25%     50%     75%    100%
    1.0    68.1   678.0  1530.0 10000.0
 Mean S0-value         : 991
 Voxel in mask         : 1298464
 Spatial smoothness    : 2.87x2.36x0.503
 mean variance         : NA
 hmax                  : 1
 Number of outliers    : 1025043
```

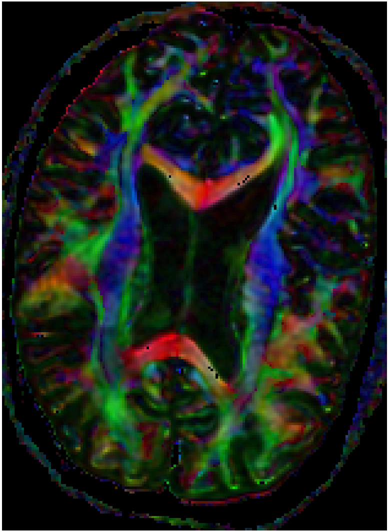
Examples

```
> nytind <- dtiIndices(nytens[31:120,31:180,11:60])
> summary(nytind)
  Object of class dtiIndices
  Generated by calls    :
[[1]]
readDWIdata(bvec, paste(filepre, "/s0004", sep = ""), format = "DICOM",
    xind = 48:204, yind = 19:234, nslice = 66, voxelext = voxelext)
[[2]]
dtiTensor(dwobj)
[[3]]
nytens[31:120, 31:180, 11:60]
[[4]]
dtiIndices(nytens[31:120, 31:180, 11:60])
  Source-Filename       : ~tabelow/DATA/dti_ny/data/e006353/s0004
  Dimension             : 90x150x50
  Number of Gradients   : 150
  Voxel extensions      : 0.9x0.9x1.8
  Percentage of zero values      : 6.49 %
  Quantiles of positive FA-values:
     0%     25%     50%     75%    100%
0.00644 0.12700 0.23900 0.42300 0.99700
  Quantiles of positive GA-values:
     0%     25%     50%     75%    100%
0.00912 0.18000 0.34700 0.65800 7.09000
  Quantiles of positive MD-values:
    0%    25%    50%    75%   100%
0.0001 0.5900 0.6870 1.0200 4.3400
```

- 2D visualization based on package adimpro
- Generic plot functions for all classes
- Example

```
R> library(dti)
R> load("nytens.rsc")
R> nytind <- dtiIndices(nytens)
R> img <- plot(nytind,slice=35,
                  view="axial")
R> write.image(img,"nyccfa35.png")
```

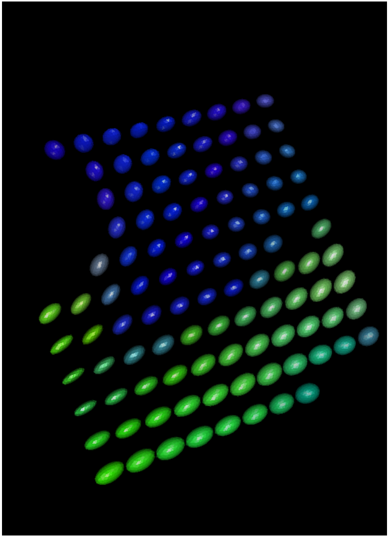
  provides a color coded directional FA map
- output as adimpro image except for dtiTensor objects



- 3D visualization based on package rgl
- show3d method for all classes
- Example

```
R> library(dti)
R> load("nytens.rsc")
R> nytind <- dtiIndices(nytens)
R> show3d(nytens, center=
     c(50,160,35),nx=11,ny=11,nz=1)
R> show3d(nytind,center =
     c(79, 108, 35), nz = 3)
```

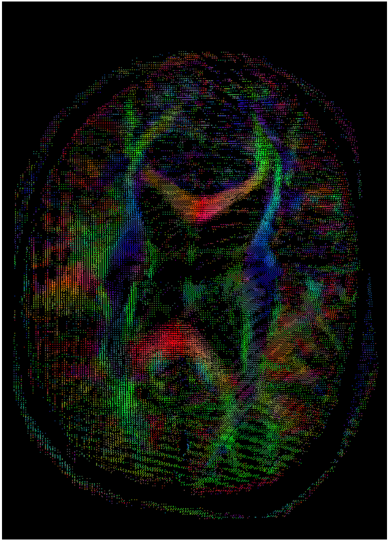
provides a 3D visualization of estimated tensors and



- 3D visualization based on package rgl
- show3d method for all classes
- Example

```
R> library(dti)
R> load("nytens.rsc")
R> nytind <- dtiIndices(nytens)
R> show3d(nytens, center=
     c(50,160,35),nx=11,ny=11,nz=1)
R> show3d(nytind,center =
     c(79, 108, 35), nz = 3)
```

provides a 3D visualization of estimated tensors and a visualization of main diffusion directions (color coded) and FA (length of lines) for slices 34 - 36



- Interface to MedINRIA

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

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

  Functions to read and write diffusion tensors in NIFTI Format

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

- Use function write.image from package adimpro to save 2D illustrations and
- rgl.snapshot from package rgl for 3D snapshots

Why smooth at all ?

- reduce noise, increase SNR
- enables to reduce number of measured gradients, i.e. scan time ...

Established approaches:

- Tensor space has a Riemannian metric
- Smoothing using log-Euclidian metric (Fillard et.al. IEEE TMI 2007)
- Smoothing using Riemannian metric (Pennec et.al. International Journal of Computer Vision, 2006, Fletcher, 2004)
- Anisotropic Geodesic Diffusion (Zhang & Hancock, 2006)

Remarks:

- Nonadaptive smoothing leads to deterioration of structure (blurring)
- Smoothing within tensor space does not allow for Rice bias correction
- Structural adaptation provides a better alternative

Structural adaptive smoothing algorithm:

- **Initialization:** $k=1$, $h^{(1)}=c_h$. Set $\hat{\zeta}^{(0)}_{b,i}=S_{b,i}$, $\hat{\mathscr{D}}^{(0)}_i$, $\hat{\theta}^{(0)}_{0,i}$ $N^{(0)}_i=1$.
- **Adaptation:** For every pair i,j compute

$$
\begin{aligned}
s^{(k)}_{ij} &= \frac{N^{(k-1)}_i}{\lambda}\left(\mathbf{R}(\hat{\zeta}^{(k-1)}_{.,i},\hat{\theta}^{(k-1)}_{0,j},\hat{\mathscr{D}}^{(k-1)}_j)-\mathbf{R}(\hat{\zeta}^{(k-1)}_{.,i},\hat{\theta}^{(k-1)}_{0,i},\hat{\mathscr{D}}^{(k-1)}_i)\right)\\
w^{(k)}_{ij} &= K_{\text{loc}}\left(\Delta(i,j,\tilde{\mathscr{D}}^{(k-1)}_i)/h^{(k)}\right)K_{\text{st}}\left(s^{(k)}_{ij}\right),
\end{aligned}
$$

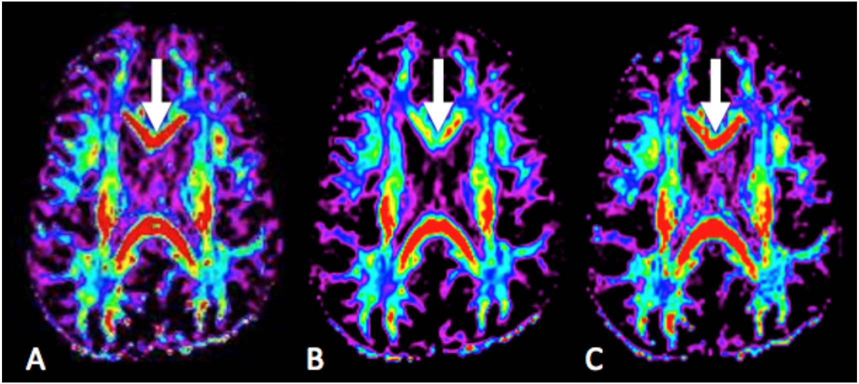
- **Rice bias correction:** Compute $\hat{\zeta}^{(k)}_{.,i}=(\hat{\zeta}^{(k)}_{b_1,i},\ldots\hat{\zeta}^{(k)}_{b_{N_{grad}},i})$

$$
\hat{\zeta}^{(k)}_{.,i}=\arg_{\zeta}\max_{\zeta,\sigma} l(S_{1,1}\ldots,S_{n,N_{grad}};\zeta,\sigma,W^{(k)}) \qquad W^{(k)}_i=(w^{(k)}_{i1},\ldots,w^{(k)}_{in})
$$

- **Estimation of diffusion weighted images:** $\begin{pmatrix}\hat{\theta}^{(k)}_{0,i}\\ \hat{\mathscr{D}}^{(k)}_i\end{pmatrix}=\arg\min_{\theta,\mathscr{D}}\mathbf{R}(\hat{\zeta}^{(k)}_{.,i},\theta,\mathscr{D})$, Set $N^{(k)}_i=\sum_{j=1}^n w^{(k)}_{ij}$.
- **Stopping:** Stop if $k=k^*$, otherwise $h^{(k+1)}=c_h h^{(k)}$, $k:=k+1$

Comparison

- Adaptive smoothing provides more stable estimates without loss of structure
- enables to reduce recording time



A: unsmoothed B: non-adaptive C: adaptive

```
demo(dti_art)
dti.smooth(dwiDataobj,hmax=5) # generates a smoothed tensor object
dti.smooth(dwiDataobj,hmax=5,result="dtiData") # generates a smoothed data object
```

4 High Angular Resolution Diffusion Weighted Imaging (HARDI)

- Statistical modeling II: The orientation distribution function
- Q-ball imaging
- Tensor mixture models

- Diffusion Tensor Imaging (DTI):

$$E(\vec{g}) = \frac{\mathrm{E}\,S(\vec{g})}{\mathrm{E}\,S_0} = \exp(-b\vec{g}^\top \mathscr{D}\vec{g})$$

- Assumption: homogeneous fiber structure within a voxel
- Reality: high precentage of voxel with fiber crossings or bifurcations

A more accurate description:

- $P(\vec{r},\vec{r}\,',\tau)$ probability for a particle to diffuse from position $\vec{r}\,'$ to $\vec{r}$ in time τ
- Mean diffusion function (over a voxel V:

$$P(\vec{R},\tau) = \int_{\vec{r}\,'\in V,\ \vec{R}=\vec{r}-\vec{r}\,'} P(\vec{r},\vec{r}\,',\tau)p(\vec{r}\,')d\vec{r}\,'$$

- Orientation density function (ODF) (weighted radial projection of P, Aganji 2009)

$$\psi_{(w)}(\vec{u},\tau) = \int_0^\infty r^2 P(r\vec{u},\tau)dr$$

Relation between $E(\vec{g})$ and $\psi_{(w)}$

- Represent $\vec{g} = q\vec{u}$ by (q, θ, ϕ)
- The Fourier transform of $r^2 P(r\vec{g})$ is

$$\begin{aligned} -\bigtriangledown^2 E(\vec{g}) &= -\frac{1}{\vec{g}} \frac{\delta^2}{\delta \vec{g}^2}(qE) + \bigtriangledown_b^2 E \\ \bigtriangledown_b^2 E &= \frac{1}{q^2}\left[\frac{1}{\sin(\phi)} \frac{\delta}{\delta\theta}(\sin\theta \frac{\delta E}{\delta\theta}) + \frac{1}{\sin^2\theta} \frac{\delta^2 E}{\delta\phi^2}\right] \end{aligned}$$

- Funk-Radon transform (line integral over unit equator): for $f : R^3 \to R$ symmetric and $F(\vec{g})$ it's 3D Fourier transform

$$\int_0^\infty f(r\vec{g})dr = \frac{1}{8\pi^2} \iint_{\vec{u}^\perp} F(\vec{g})d^2\vec{g}$$

- Q-Ball imaging (weighted version - see Aganj et al. (2009)) $\theta \equiv \pi/2$

$$\psi_{(w)}(\vec{u}) = \frac{1}{4\pi} - \frac{1}{8\pi^2} \int_0^{2\pi} \int_0^\infty \frac{1}{q} \bigtriangledown_b^2 E dq d\phi = \frac{1}{4\pi} - \frac{1}{8\pi^2} \int_0^{2\pi} \bigtriangledown_b^2 ln(-lnE))d\phi$$

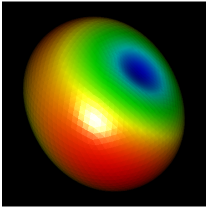
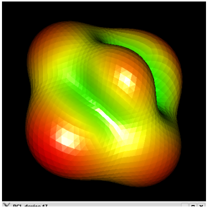
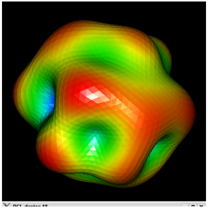
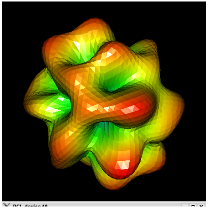
Q-Ball Imaging: Strengths and problems

- Integration not feasible, therefore expansion into spherical harmonics (Descoteaux et al., (2007), Aganj et al. (2009))

$$\ln(-\ln E(\vec{g}_i)) = \sum_{j=1}^{J} c_j Y_j(\vec{g}_i) \qquad \text{for} \quad i = 1, \dots, N$$

$$\psi_w(\vec{u}) = \frac{1}{2\sqrt{\pi}} Y_1(\vec{u}) - \frac{1}{16\pi^2} \sum_{j=2}^{J} 2\pi P_{k_j}(0) k_j (k_j + 1) c_j Y_j(\vec{u})$$

- Fast (linear), high-frequency artifacts (needs regularization), ODF via Funk-Radon transform is non-linear in E $(ln(-lnE))$.

Class definition:

```
setClass("dwiQball",
    representation(what = "character", # type of estimate (default "wODF")
              order  = "integer", # specified order
              lambda = "numeric", # regularization parameter
              sphcoef = "array",  # estimated SH coefficients
              varsphcoef = "array", # variance estimates
              th0 = "array", # mean S0 values
              sigma  = "array", # estimated error variances
              scorr  = "array", # spatial correlations
              bw     = "numeric", # same
              mask   = "array", # mask of active voxel
              hmax   = "numeric", # not yet used
              outlier = "numeric", # unreasonable voxel
              scale  = "numeric"), # used for visualization
    contains=c("list","dwi")
```

Example:

```
R> load("nydwdatareduced.rsc")
R> nyqball <- dwiQball(dwobj, order=8, lambda=1e-2)
```

- anisotropic Gaussian diffusion (single fiber bundle)

$$P(r\vec{u},\tau) = \frac{1}{\sqrt{|\mathscr{D}|(4\pi\tau)^3}} \exp\left(-r^2 \frac{\vec{u}^T \mathscr{D}^{-1}\vec{u}}{4\tau}\right)$$

- ODF

$$\psi(\vec{u},\tau) = (4\pi)^{-1}|\mathscr{D}|^{-1/2}(\vec{u}^\top \mathscr{D}^{-1}\vec{u})^{-3/2}$$

  Angular central Gaussian distribution

- Assumes a mixture of fiber bundles in each voxel
- Each fiber bundle can be described by a tensor model
- Model:

$$\frac{S(\vec{g})}{S_0} = \sum_i w_i \exp(-b\vec{g}^\top \mathscr{D}_i^{-1}\vec{g}) \quad \sum_i w_i = 1, \quad w_i \geq 0$$

- parameter identifiability ? to flexible ...
- corresponding ODF:

$$\psi(\vec{u},\tau) = (4\pi)^{-1}\sum_i w_i|\mathscr{D}_i|^{-1/2}(\vec{u}^\top \mathscr{D}_i^{-1}\vec{u})^{-3/2}$$

- Assumption: Homogeneous geometry of fibres $\mapsto$ rotational symmetric (prolate) tensors of same eccentricities
- Model:

$$\frac{S(\vec{g})}{S_0} = \sum_i w_i \exp(-b\vec{g}^\top(\lambda_2 I_3 + (\lambda_1 - \lambda_2) d_i d_i^\top)\vec{g}) \quad \sum_i w_i = 1, \quad w_i \geq 0$$
$$= \sum_i \tilde{w}_i \exp(-\theta(\vec{g}^\top d_i)^2) \quad \tilde{w}_i \geq 0$$

with $w_i = \tilde{w}_i / \sum(\tilde{w}_i)$, $b\lambda_2 = \log(\sum(\tilde{w}_i))$ and $\theta = b(\lambda_1 - \lambda_2)$.

- Separable nonlinear least squares problem with constraints on linear parameters.

$$min_{(\theta,d_1,\ldots d_k)} min_{\tilde{w}_i \geq 0} \sum_j^N \left(\frac{S(\vec{g}_j)}{S_0} - \sum_i \tilde{w}_i \exp(-\theta(\vec{g}^\top d_i)^2)\right)^2$$

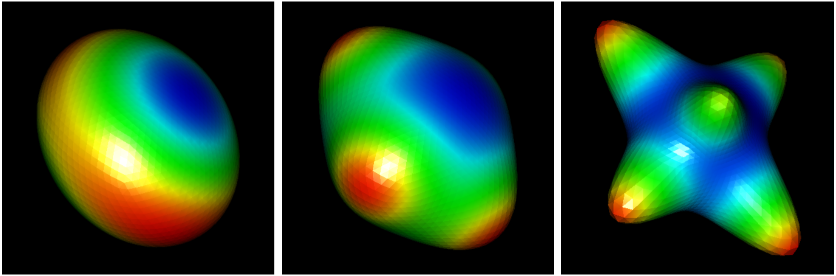
- Problem: difficult to solve for low SNR, initial estimates ...

How many components ?

- Model selection problem
- Nested models for orders $k = K, \ldots, 1$
- Order selected by Bayesian Information Criterion (BIC) with automatic reduction in case of zero weights

ODF-representation: Mixture of angular central Gaussian distributions (density $f(x,D) = |D|^{-1/2}(x^\top Dx)^{-3/2}$).

Tensor Order 2 Order 4



Class dwiMixtensor

Class definition

```
setClass("dwiMixtensor",
        representation(method = "character", #
                       model  = "character", #
                       ev     = "array", # ev[1,,,]+ev[2,,,], ev[2,,,] are the eigenvalues
                       mix    = "array", # mixture coefficients
                       orient = "array", # orientations
                       order  = "array", # estimated mixture order
                       p      = "numeric", # p in "method"=="Jian"
                       th0    = "array", # mean S0
                       sigma  = "array", # estimated error variances
                       scorr  = "array", # spatial correlations
                       bw     = "numeric", # same
                       mask   = "array", # mask of active voxel
                       hmax   = "numeric", #
                       outlier = "numeric", # inreasonable voxel
                       scale  = "numeric"), # used for visualization
        contains=c("list","dwi"))
```

Demo: artificial tensor models of order 3

```
demo(mixtens_art)
```

- Generalized fractional anisotropy (gfa)

$$gfa = \frac{\mu_1 - \mu_2}{\sqrt{\mu_1^2 + 2\mu_2^2}}$$

- Effective order (between 0 and m)

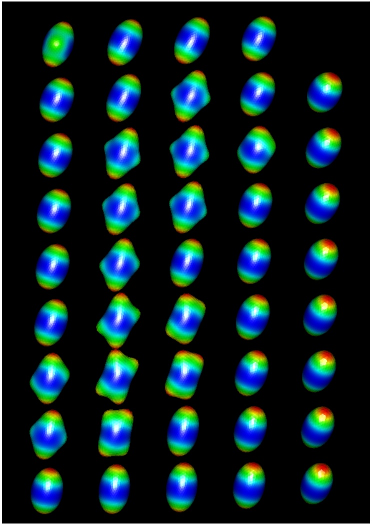
$$eorder = \sum_{k=1}^{m} (2k-1) w_k$$

- Extract function

```
R> library(dti)
R> load("nydwdatareduced.rsc")
R> nymix4 <- dwiMixtensor(dwobj, maxcomp=4) # expensive
R> save(nymix4,file="nymix4.rsc")
R> load("nymix4.rsc")
R> nymix4char <- extract(nymix4,c("gfa","eorder","ev","order","mix",
                 "andir","s0","mask"))
R> summary(nymix4)
```

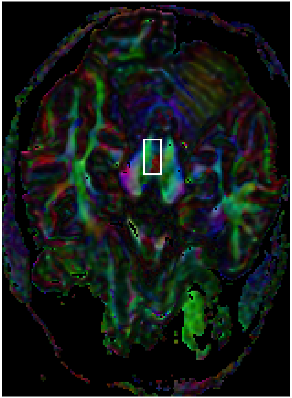
Example

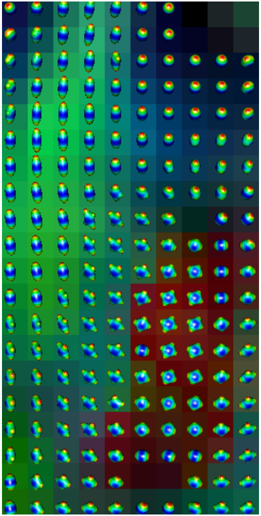
```
R> library(dti)
R> load("nymix4.rsc")
R> summary(nymix4)
 Object of class dwiMixtensor
 Generated by calls    :
[[1]]
readDWIdata(bvec, paste(filepre, "/s0004", sep = ""),
   format = "DICOM",xind = 48:204, yind = 19:234,
   nslice = 66, voxelext = voxelext)
[[2]]
dwiMixtensor(dwobj, maxcomp = 4)
 Source-Filename       : ....
 Dimension             : 157x216x66
 Number of Gradients   : 150
 Voxel extensions      : 0.9x0.9x1.8
 Quantiles of S0-values:
    0%     25%     50%     75%    100%
   1.0    68.1   691.0  1540.0 10000.0
 Mean S0-value         : 998
 Voxel in mask         : 1301852
 Spatial smoothness    : 0x0x0
 mean variance         : 0.0107
 hmax                  : 1
 Number of outliers    : 1025043
 Numbers od mixture components: 708616 350351 239711 3096 78
R> show3d(nymix4 , center=c(52,151,35),nx=5,ny=9,nz=1)
R> rgl.snapshot("nymix435.png")
```

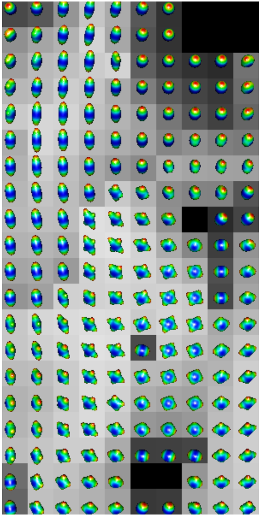


Comparison of tensor and tensor mixtures results

Color coded directional FA (tensor model)







Center: dwiMixtensor results (order 5) overlayed on color coded FA. Right: dwiMixtensor results (order 5) overlayed on GFA

5 Fiber Tracking

- Class dwiFiber

Class definition:

```
setClass("dwiFiber",
         representation(call = "list",
                        fibers = "matrix",
                        startind = "integer",
                        roimask = "raw",
                        method = "character",
                        minanindex = "numeric",
                        maxangle   = "numeric"),
        contains=c("list","dwi"))
```

tracking - Method for classes dtiTensor and dwiMixtensor:

```
R> library(dti)
R> load("nymix4.rsc")
R> nytracks35 <- tracking(nymix4,roiz=35)
```

Methods for class dwiFiber: summary, print, show, plot, show3d, selectFibers

```
R> nytracksz35x79 <- selectFibers(nytracks35,roix=79)
```

```
R> summary(nytracks35)
 Object of class dwiFiber
 Generated by calls    :
[[1]] ...
[[3]]
tracking(nymix4, roiz = 35)
 Source-Filename       : ~tabelow/...
 Dimension            : 157x216x66
 Number of Gradients   : 150
 Voxel extensions      : 0.9x0.9x1.8
 Minimum FA  : 0.3
 Maximum angle : 30
 Number of fibers : 10503
 Quantiles of fiber lengths:
 0%  25%  50%  75% 100%
  5    9   16   31  184
 Total number of line segments : 254433
R> show3d(nytracks35)
R> rgl.snapshot("nytracks35.png")
}
```

Fibers crossing slice 35, dwiMixtensor model max. order 4

6 Outlook and further reading

Things to come:

- general approach to smoothing of DWI data
- stabilization of tensor mixture models
- additional HARDI models
- better integration with other packages
- connectivity maps (see e.g. Hagmann et.al. PLOSone (2007)), Pittsburgh Brain competition

Collaborations

Joint Work with:

- Henning Voss, Weill Medical College, Cornell University

Cooperation:

- Citigroup Biomedical Imaging Center, Weill Medical College, Cornell University
- University of Münster
- BNIC, Charitè, Berlin
- Max-Plank Institute for Human Cognitive and Brain Sciences, Leipzig

R-Community:

- CRAN Task View: Medical Image Analysis
  Jonathan Clayden, Pierre Lafaye de Micheaux, Volker Schmid, Brandon Whitcher

Acknowledgments:

- We thank the Weill Medical College, Cornell University, the Max Planck Institute for Human Cognitive and Brain Sciences and University of Münster and the NIH/NCRR Center for Integrative Biomedical Computing (P41-RR12553) for providing functional and diffusion-weighted MR datasets.

Susumu Mori (2007).
Introduction to Diffusion Tensor Imaging
Elsevier

Tabelow, K., Polzehl, J., Spokoiny, V., Voss, H. (2008).
Diffusion tensor imaging - spatial adaptive smoothing
Neuroimage, 39:1763–1773.

Polzehl, J., Tabelow, K. (2009).
Structure adaptive smoothing Diffusion Tensor Imaging data: the R Package dti
Journal of Statistical Software, 31, 1–24.

Tabelow, K., Polzehl, J. (2010).
Expansion of the orientation distribution function in terms of the angular central Gaussian distribution.
Manuscript