Package 'dmri.tracking'

October 13, 2022

Type Package

Title DiST - Diffusion Direction Smoothing and Tracking

Version 0.1.0

Depends R (>= 3.5.0)

Maintainer Seungyong Hwang <syhwang@ucdavis.edu>

Description It provides functions to apply the deterministic tracking algorithm - DiST (Wong et al 2016) <doi:10.1214/15-AOAS880> and to plot tractography results.

License MIT + file LICENSE

URL https://github.com/vic-dragon/dmri.tracking

BugReports https://github.com/vic-dragon/dmri.tracking/issues

Encoding UTF-8

Imports rgl

RoxygenNote 7.1.1

Suggests rmarkdown, knitr, testthat (>= 3.0.0), covr

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation yes

Author Seungyong Hwang [aut, cre],
Raymond Wong [aut],
Seungyong Hwang [aut]

Repository CRAN

Date/Publication 2021-06-09 11:30:02 UTC

R topics documented:

tractography *Tractography*

Description

Visualize the result from v.track. r package rgl is required.

Usage

```
tractography(loc, vec)
```

Arguments

loc	Voxel coordinates that the reconstructed fiber go through
vec	Diffusion direction that used to reconstruct the fiber

Value

No return value. The reconstructed fiber is visualized on the opened rgl window.

Author(s)

Raymond Wong, Seungyong Hwang (Maintainer: <syhwang@ucdavis.edu>)

References

R. K. W. Wong, T. C. M. Lee, D. Paul, J. Peng and for the Alzheimer's Disease Neuroimaging Initiative. (2016) "Fiber Direction Estimation, Smoothing and Tracking in Diffusion MRI". The Annals of Applied Statistics, 10(3), 1137-1156.

See Also

The tracking result from v.track can be used for tractography in dmri.tracking package.

Examples

```
#Load an example output from the peak detection algorithm
load(system.file("extdata", "peakresult.rda", package = "dmri.tracking"))

str(peak.result)  #Output from the peak detection algorithm

#Apply Tracking algorithm
result = v.track(v.obj = peak.result, max.line=500)


#Plot tracking result.

library(rgl)
open3d()
```

```
for (iind in (result$sorted.iinds[result$sorted.update.ind])){
  cat(iind,"\n")
  tractography(result$tracks1[[iind]]$inloc, result$tracks1[[iind]]$dir)
  tractography(result$tracks2[[iind]]$inloc, result$tracks2[[iind]]$dir)
}

#An example to prepare v.obj is available in https://github.com/vic-dragon/dmri.tracking
```

`v.track` *Deterministic tracking algorithm – DiST*

Description

v.track is used to apply the deterministic tracking algorithm – DiST (Wong et al 2017) It can be used to carry out the neuronal fiber reconstruction based on the peak detection results with local fiber estimation. Peak detection algorithm can be found in example_HCP_analysis.py from github-repository

Usage

```
v.track(
  v.obj,
  max.line = 100,
  nproj = 1,
  elim = T,
  elim.thres = 1,
  thres.ang = 0.5235988
)
```

Arguments

`v.obj` An list type object which contains the following components:

- vec: A matrix containing the estimated peak directions.
- loc: A matrix containing the 'braingrid' coordinates of the corresponding estimated peak direction.
- map: A vector containing the voxel indicator of corresponding estimated peak direction.
- rmap: A vector specifying the location in 'map' of each voxel.
- n.fiber: A vector specifying the number of peaks at each voxel.
- n.fiber2: A vector specifying the number of peaks corresponding to 'map'.
- braingrid: A array specifying the normalized voxel coordinates.
- xgrid.sp,ygrid.sp,zgrid.sp: A numeric value specifying the voxel size (mm) in x, y, z-axis, respectively. (e.g. Voxel size in HCP dMRI: 1.25mm * 1.25mm * 1.25mm)
- Example can be found in main page of github-repository

max.line	A integer value specifying the maximum number of voxels that the reconstructed fibers can go through. The value can depend on the size of ROI.
nproj	A integer value specifying the number of neighborhood voxels if the algorithm cannot find any viable direction nearby.
elim	logical. If TRUE, 'sorted.update.ind' returns whether the reconstructed fiber is greater than 'elim.thres'
elim.thres	A numeric value specifying the lower limit length of reconstructed fibers.
thres.ang	A numeric value specifying the threshold to determine whether the destination voxel have a viable direction (default value: pi/6). i.e., the algorithm will be proceeded, if the angular difference of the diffusion direction between the previous voxel and the destination voxel is smaller than 'thres.ang'.

Value

Result of deterministic tracking algorithm

- v.obj: Input of v.track
- track1, track2: A list containing the reconstructed fiber information
  - inloc: The voxel coordinates that the reconstructed fiber went through
  - dir: The diffusion direction that used to reconstruct fiber
  - iinds: The indicator of voxel that the reconstructed fiber went through
  - change: logical, If TRUE, that voxel pass the angular difference threshold (thres.ang)
  - pvox: The passed voxel
  - pdir: The passed direction
  - pdis: The length of fiber between the passed information and the previous one.
  - ppdis: The distance of the voxel between the passed one and the previous one.
- n.iinds: Number of voxels the reconstructed fiber went through
- n.use.iind: Number of times that each estimated direction is used.
- lens: The reconstructed fiber lengths.
- sorted.iinds: The ordered reconstructed fibers
- sorted.update.ind: Whether the reconstructed fiber is greater than elim.thres

Author(s)

Raymond Wong, Seungyong Hwang (Maintainer: <syhwang@ucdavis.edu>)

References

R. K. W. Wong, T. C. M. Lee, D. Paul, J. Peng and for the Alzheimer's Disease Neuroimaging Initiative. (2016) "Fiber Direction Estimation, Smoothing and Tracking in Diffusion MRI". The Annals of Applied Statistics, 10(3), 1137-1156.

See Also

tractography for plotting tractography based on the tracking result from v.track in dmri.tracking package.

Examples

```
#Load an example output from the peak detection algorithm
load(system.file("extdata", "peakresult.rda", package = "dmri.tracking"))

str(peak.result)  #Output from the peak detection algorithm

#Apply Tracking algorithm
result = v.track(v.obj = peak.result, max.line=500)


#Plot tracking result.

library(rgl)
open3d()
for (iind in (result$sorted.iinds[result$sorted.update.ind])){
  cat(iind,"\n")
  tractography(result$tracks1[[iind]]$inloc, result$tracks1[[iind]]$dir)
  tractography(result$tracks2[[iind]]$inloc, result$tracks2[[iind]]$dir)
}

#An example to prepare v.obj is available in https://github.com/vic-dragon/dmri.tracking
```

Index