Supplementary Materials of "The Mean Shape under the Relative Curvature Condition".

Mohsen Taheri , Stephen M. Pizer and Jörn Schulz

Corresponding author email: MohsenTaheriShalmani@gmail.com

October 27, 2024

This document provides R code documentation related to the Supplementary Materials of the manuscript titled "The Mean Shape under the Relative Curvature Condition".

For additional information, please visit the corresponding author's GitHub repository: Https://Github.com/MohsenTaheriShalmani/Elliptical_Tubes.

Contents

1 Description

The primary script file is `main.R`, which is supported by two additional scripts in the `Functions` folder: `ETRep_Functions.R`, containing functions specific to ETRep analysis, and `ETRep_MathFunctions.R`, which includes general mathematical functions. An ETRep encapsulates the characteristics of an *elliptical tube* (e-tube), including the size and orientation of its elliptical cross-sections, positioned according to the material frames along the spine, as illustrated in Figure 1.

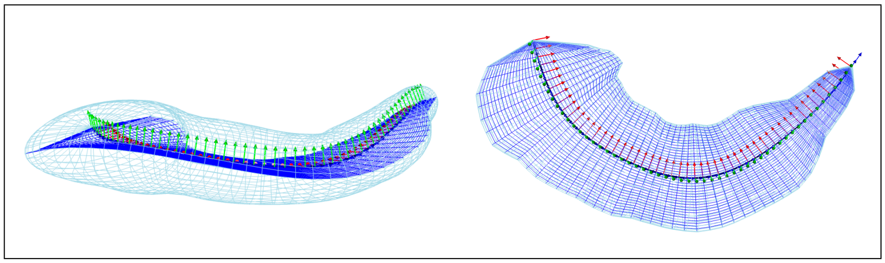

Figure 1: ETRep representation of the left hippocampus with 53 cross-sections. Arrows highlight material frames aligned along the spine.

The `main.R` script is divided into two main sections: 1. Transformation and 2. Simulation. The Transformation section provides examples for calculating both intrinsic and non-intrinsic means using intrinsic and non-intrinsic transformations between two e-tubes, represented by their corresponding *elliptical tube representations* (ETReps), as discussed in the main manuscript. The simulation section provides an example of ETRep simulation, as discussed in the article's Supplementary Materials.

The required libraries for running the `main.R` script are: `shapes`, `rgl`, `Morpho`, `matlib`, `RiemBase`, `doBy`, `plotrix`, `Directional`, `RSpincalc`, `rotations`, `SphericalCubature`, `Rvcg`, `fields`, `Matrix`, `pracma`, `truncnorm`, `ggplot2`, `reshape2`, and `dplyr`.

2 Functions

The functions essential for ETRep analysis are outlined below, covering transformation calculations, mean computation, ETRep size estimation, simulation, and plotting.

2.1 Create a Discrete Elliptical Tube

The function `create_Elliptical_Tube` constructs a discrete e-tube and represents it as an ETRep.

Input Parameters

- `numberOfFrames`: Integer, specifies the number of consecutive material frames.
- `method`: String, either "basedOnEulerAngles" or "basedOnMaterialFrames", determining whether the material frames are defined by Euler angles or given material frames.

- `EulerAngles_Matrix`: `numberOfFrames` $\times$ 3 matrix, with each row containing three Euler angles to define a material frame.
- `materialFramesBasedOnParents`: Array of dimensions $3 \times 3 \times$ `numberOfFrames`, defining the predetermined material frames.
- `ellipseResolution`: Integer, specifies the boundary resolution of the elliptical cross-sections (default is 10).
- `ellipseRadii_a`: Real vector with `numberOfFrames` elements, defining the primary radii of the cross-sections.
- `ellipseRadii_b`: Real vector with `numberOfFrames` elements, defining the secondary radii of the cross-sections.
- `connectionsLengths`: Real vector with `numberOfFrames` elements, defining the lengths of the spinal connection vectors.
- `plotting`: Logical, enables plotting of the ETRep (default is TRUE).

Output

- `ETRep`: A list containing the generated tube's details, including:
  - Orientation of material frames based on `materialFramesBasedOnParents`,
  - Twisting angles `theta_angles`,
  - Curvature angles `phi_angles_bend`,
  - Principal radii of the cross-sections `ellipseRadii_a` and `ellipseRadii_b`,
  - Projection lengths, `r_project_lengths`, according to Equation (1) in the main article,
  - Lengths of the spinal connection vectors `connectionsLengths`,
  - Boundary points of the cross-sections, stored as `boundaryPoints` for the ETRep-PDM.

Example

```
numberOfFrames<-15
ellipseResolution<-10
EulerAngles_alpha<-c(rep(0,numberOfFrames))
EulerAngles_beta<-c(rep(-pi/20,numberOfFrames))
EulerAngles_gamma<-c(rep(0,numberOfFrames))
EulerAngles_Matrix =cbind(EulerAngles_alpha,EulerAngles_beta,EulerAngles_gamma)
ellipseRadii_a<-rep(3,length.out=numberOfFrames)
ellipseRadii_b<-rep(2,length.out=numberOfFrames)
connectionsLengths<-rep(4,numberOfFrames)
tube<-create_Elliptical_Tube(numberOfFrames=numberOfFrames,
                             EulerAngles_Matrix =EulerAngles_Matrix,
                             method="basedOnEulerAngles",
                             ellipseResolution=ellipseResolution,
                             ellipseRadii_a=ellipseRadii_a,
                             ellipseRadii_b=ellipseRadii_b,
                             connectionsLengths=connectionsLengths,
                             plotting=TRUE)
```

2.2 Plotting

The plot_Elliptical_Tube function plots an ETRep.

Input Parameters

- tube: A list containing the information of an ETRep.
- plot_boundary: Logical, enables plotting of the boundary (default is TRUE).
- plot_frames: Logical, enables plotting the material frames (default is TRUE).
- frameScaling: Real value for scaling the frames.
- plot_r_project: Logical, enables plotting of the projection of the cross-sections along the normals based on Equation (1) of the main article (default is TRUE).
- plot_r_max: Logical, enables plotting of the maximum possible size of the projection of the cross-sections along the normals, i.e., $\frac{x_i}{\boldsymbol{v}_i}$ based on Equation (1) of the main article (default is FALSE).

Output

- Just a graphical output.

Example

```
plot_Elliptical_Tube(e_tube = tube_A)
```

2.3 Check the validity of a tube

The check_Tube_Legality function checks the RCC and the validity of the principal radii $\forall i a_i > b_i$.

Input Parameters

- tube: A list containing the information of an ETRep.

Output

- Logical (TRUE as valid and False as invalid)

Example

```
check_Tube_Legality(tube = tube)
```

2.4 Transformations

The nonIntrinsic_Transformation_Elliptical_Tubes and intrinsic_Transformation_Elliptical_Tubes functions perform non-intrinsic and intrinsic transformations from one ETRep to another. Both ETReps must have the same number of spinal points.

Input Parameters

- tube1: A list containing the details of the first ETRep.
- tube2: A list containing the details of the second ETRep.
- numberOfSteps: Integer, specifying the number of transformation steps.
- plotting: Logical, enables visualization during transformation (default is TRUE).

Output

- `tubes`: A list containing `numberOfSteps` intermediate ETReps.

Example

```
# Non-intrinsic transformation example
nonIntrinsic_Transformation_Elliptical_Tubes(tube1 = tube1,
                                             tube2 = tube2,
                                             numberOfSteps = 7,
                                             plotting = TRUE)
# Intrinsic transformation example
intrinsic_Transformation_Elliptical_Tubes(tube1 = tube1,
                                          tube2 = tube2,
                                          numberOfSteps = 7,
                                          plotting = TRUE)
```

2.5 Mean calculation

The `nonIntrinsic_mean_tube` and `intrinsic_mean_tube` functions calculate the non-intrinsic and intrinsic mean of a set of ETReps, respectively. All ETReps must have the same number of spinal points.

Input Parameters

- `tubes`: A list of ETReps.
- `type`: String, either "ShapeAnalysis" and "sizeAndShapeAnalysis" for calculating the mean with or without scaling (default is "sizeAndShapeAnalysis").
- `plotting`: Logical, enables visualization of the mean (default is TRUE).

Output

- `tube`: An ETRep as a list representing the mean ETRep.

Example

```
# Non-intrinsic mean
nonIntrinsic_mean_tube(tubes = tubes, plotting = TRUE)
nonIntrinsicMeanTube

# Intrinsic mean
intrinsic_mean_tube(tubes = tubes, plotting = TRUE)
intrinsicMeanTube
```

2.6 Simulation

The `simulate_etube` function generates a set of ETReps based on a reference tube by adding variation to the tube elements, as discussed in the article's Supplementary Materials.

Input Parameters

- `referenceTube`: A list consists of the information of an ETRep as the reference e-tube.
- `numberOfSimulation`: Number of random samples.
- `sd_v`: Standard deviation for the $\boldsymbol{v}_i$ vectors.
- `sd_psi`: Standard deviation for the roll angles ψ_i.
- `sd_x`: Standard deviation for the length of the spinal connection vectors x_i.
- `sd_a`: Standard deviation for the first principal radii a_i.
- `sd_b`: Standard deviation for the second principal radii b_i.
- `rangeSdScale`: A range for adding random scaling to the generated samples.
- `plotting`: Logical, enables visualization of the sample (default is FALSE).

Output

- `tubes`: A list containing `numberOfSimulation` of random ETReps.

Example

```
#create a reference tube
numberOfFrames<-15
ellipseResolution<-10
EulerAngles_alpha<-c(rep(0,numberOfFrames))
EulerAngles_beta<-c(rep(-pi/20,numberOfFrames))
EulerAngles_gamma<-c(rep(0,numberOfFrames))
EulerAngles_Matrix<-cbind(EulerAngles_alpha,EulerAngles_beta,EulerAngles_gamma)
ellipseRadii_a<-rep(3,length.out=numberOfFrames)
ellipseRadii_b<-rep(2,length.out=numberOfFrames)
connectionsLengths<-rep(3,numberOfFrames)
referenceTube<-
  create_Elliptical_Tube(numberOfFrames=numberOfFrames,
                         EulerAngles_Matrix =EulerAngles_Matrix,
                         method="basedOnEulerAngles",
                         ellipseResolution=10,
                         ellipseRadii_a=ellipseRadii_a,
                         ellipseRadii_b=ellipseRadii_b,
                         connectionsLengths=connectionsLengths,
                         plotting=TRUE)

# Generate random samples
simulate_etube(referenceTube = referenceTube,
               numberOfSimulation=5,
               sd_v=0.1,
               sd_psi=0.001,
               sd_x=0.01,
               sd_a=0.01,
               sd_b=0.01,
               rangeSdScale=c(1,2),
               plotting=TRUE)
```