Package 'RvtkStatismo'

June 23, 2014

Type Package

Title Integrates statismo and R using the vtkStandardMeshRepresenter

Version 0.2.140623

Date 2014-06-23

Author Stefan Schlager, the authors of Statismo

Maintainer Stefan Schlager <zarquon42@gmail.com>

Description Integrates statismo and R using the vtkStandardMeshRepresenter.
Statismo shape models will be stored as objects of class ``pPCA". (this is work in progress).

License GPL >=2

Imports Rcpp (>= 0.11.1),Morpho,Rvcg

LinkingTo Rcpp,RcppEigen

SystemRequirement VTK5.8, statismo (>= 0.9 best ist freshly from github)

URL http://github.com/zarquon42b/RvtkStatismo, URL:
http://github.com/statismo/statismo

R topics documented:

`Rvtk-package` *Integrates statismo and R using the vtkStandardMeshRepresente*

Description

Integrates statismo and R using the vtkStandardMeshRepresenter. Statismo shape models will be stored as objects of class "pPCA". (this is work in progress).

Details

Package:	Rvtk
Type:	Package
Version:	0.2.140623
Date:	2014-06-23
License:	GPL
LazyLoad:	yes

Author(s)

Stefan Schlager

Maintainer: Stefan Schlager <zarquon42@gmail.com>

References

To be announced

`getCoordVar` *get per coordinate variance from a statistical model*

Description

get per coordinate variance from a statistical model

Usage

```
getCoordVar(model)
```

Arguments

model	object of class pPCA

Note

calculates the per-coordinate variance as described in Luethi(2009)

References

Lüthi M, Albrecht T, Vetter T. 2009. Probabilistic modeling and visualization of the flexibility in morphable models. In: Mathematics of Surfaces XIII. Springer. p 251-264

---

getDataLikelihood *calculate probability/coefficients for a matrix/mesh given a statistical model*

Description

calculate probability for a matrix/mesh given a statistical model

Usage

```
getDataLikelihood(x, model, align = FALSE, use.lm)

## S3 method for class 'matrix'
getDataLikelihood(x, model, align = FALSE, use.lm = NULL)

## S3 method for class 'mesh3d'
getDataLikelihood(x, model, align = FALSE, use.lm = NULL)

getCoefficients(x, model, align = TRUE, use.lm = NULL)
```

Arguments

x	matrix or mesh3d
model	a model of class pPCA
align	logical: if TRUE the data will be aligned to the model's mean
use.lm	integer vector specifying row indices of the coordinates to use for rigid registration on the model's meanshape.

Value

`getProb` returns a probability, while `getCoefficients` returns the (scaled) scores in the pPCA space.

`mesh2vtp` *exports a triangular mesh of class mesh3d to a vtp file*

Description

exports a triangular mesh of class mesh3d to a vtp file

Usage

```
mesh2vtp(mesh, filename = dataname)
```

Arguments

`mesh`	mesh of class mesh3d
`filename`	character

`meshalign` *align meshes stored in a list by their vertices*

Description

align meshes stored in a list by their vertices

Usage

```
meshalign(meshlist, scale = TRUE, array = FALSE)
```

Arguments

`meshlist`	list containing triangular meshes of class "mesh3d"
`scale`	logical: request scaling during alignment
`array`	logical: if TRUE the superimposed vertices will be returned as 3D array.

Value

returns a list of aligned meshes or an array of dimensions k x 3 x n, where k=number of vertices and n=sample size.

meshlist2array *convert meshes to array consisting of vertex coordinates*

Description

convert meshes to array consisting of vertex coordinates

Usage

```
meshlist2array(meshlist)
```

Arguments

meshlist	list containing triangular meshes of class "mesh3d"

Value

returns an array with k x 3 x n dimensions where k=number of vertices, and n=sample size.

pPCA/pPCAconstr *calculate or modify a probablistic PCA based on 3D-coordinates*

Description

calculate or modify a probablistic PCA based on 3D-coordinates

Usage

```
pPCA(array, align = TRUE, sigma = NULL, exVar = 1, scale = TRUE,
  representer = NULL)

pPCAconstr(array, align = TRUE, missingIndex, deselect = FALSE,
  sigma = NULL, exVar = 1, representer = NULL, scale = TRUE,
  fullfit = FALSE)

setMod(procMod, sigma, exVar)

## S3 method for class 'pPCA'
setMod(procMod, sigma = NULL, exVar = 1)

## S3 method for class 'pPCAconstr'
setMod(procMod, sigma = NULL, exVar = 1)
```

Arguments

array	array of dimensions k x 3 x n, where k=number of coordinates and n=sample size.
align	logical: if TRUE, the data will be aligned first
missingIndex	integer vector: specifies which points are missing in the constrained model
deselect	logical: if TRUE, missingIndex references the existing coordinates instead of the missing ones.
procMod	object of class "pPCA" or "pPCAconstr"
sigma	estimate of error variance (sensible is a value estimating coordinate error in terms of observer error)
exVar	numeric value with `0 < exVar <= 1` specifying the PCs to be included by their cumulative explained Variance
representer	a triangular mesh, where the vertices correspond to the coordinates in `array`
scale	logical: allow scaling in Procrustes fitting
fullfit	logical: if FALSE only the non-missing points will be used for registration.

Value

`pPCA` and `pPCAconstr` return a probabilistic PCA model of class "pPCA" or "pPCAconstr" respectively. `predictPCA` and `predictPCAcond` select the most probable shape within a given model (within defined boundaries), `setMod` is used to modify existing models by changing sigma and exVar.

References

Lüthi M, Albrecht T, Vetter T. 2009. Probabilistic modeling and visualization of the flexibility in morphable models. In: Mathematics of Surfaces XIII. Springer. p 251-264

Examples

```
require(Morpho)
data(boneData)
model <- pPCAconstr(boneLM[,,-1],missingIndex=3:4)
## change parameters without recomputing Procrustes fit
model1 <- setMod(model, sigma=1, exVar=0.8)
```

---

predictpPCA/predictpPCAconstr

predict or restrict a mesh or matrix based on a statistical model

---

Description

predict or restrict a mesh or matrix based on a statistical model

Usage

```
predictpPCAconstr(x, model, representer, origSpace = TRUE, pPCA = FALSE,
  ...)

## S3 method for class 'matrix'
predictpPCAconstr(x, model, representer = TRUE,
  origSpace = TRUE, pPCA = FALSE, ...)

## S3 method for class 'mesh3d'
predictpPCAconstr(x, model, representer = TRUE, sdmax,
  origSpace = TRUE, pPCA = FALSE, ...)

predictpPCA(x, model, representer = TRUE, ...)

## S3 method for class 'matrix'
predictpPCA(x, model, representer = TRUE, origSpace = TRUE,
  use.lm = NULL, sdmax, mahaprob = c("none", "chisq", "dist"),
  align = TRUE, ...)

## S3 method for class 'mesh3d'
predictpPCA(x, model, representer = TRUE, origSpace = TRUE,
  use.lm = NULL, sdmax, mahaprob = c("none", "chisq", "dist"),
  align = TRUE, ...)

## S3 method for class 'numeric'
predictpPCA(x, model, representer = TRUE, ...)
```

Arguments

x	a matrix, a mesh3d or a vector (for pPCA models) containing standardized variables within the PC-space
model	model of class pPCA or pPCAconstr
representer	if TRUE and the model contains a representer mesh, a surface mesh will be returned, coordinate matrix otherwise.
origSpace	logical: rotate the estimation back into the original coordinate system.
pPCA	logical: if TRUE, a constrained pPCA model is returned. "chisq" uses the Chi-Square distribution of the squared Mahalanobisdistance, while "dist" restricts the values to be within a multi-dimensional sphere of radius sdmax. If FALSE the probability will be determined per PC separately.
use.lm	optional: integer vector specifying row indices of the coordinates to use for rigid registration on the model's meanshape.
sdmax	maximum allowed standard deviation (per Principal axis) within the model space. Defines the probabilistic boundaries.
mahaprob	character: if != "none", use mahalanobis-distance to determine overall probability (of the shape projected into the model space.

Value

predictpPCA returns a matrix/mesh3d restricted to the boundaries given by the modelspace.

predictpPCAconstr returns a list with

estim	matrix/mesh3d representing the mean of the restricted space
pPCA	if pPCA = TRUE a pPCA model representing the gaussian subspace given the constraints is returned
rot	the transformation of x into the modelspace that can be reverted by calling rotreverse from the package Morpho

---

read.vtk *imports vtk and vtp files*

Description

imports vtk and vtp files

Usage

```
read.vtk(filename)
```

Arguments

filename	character string

Value

list of class mesh3d

---

rigidAlign *Fast Procrustes align of coordinates*

Description

Fast Procrustes align of coordinates

Usage

```
rigidAlign(array, scale = TRUE, missingIndex, deselect = FALSE)
```

Arguments

array	array of coordinates
scale	logical: request scaling during alignment
missingIndex	integer vector: specifies which points are missing (for building constrained model)
deselect	logical: if TRUE, missingIndex references the existing coordinates instead of the missing ones.

Value

a list containing

rotated	array containing registered coordinates
mshape	matrix containing meanshape

---

statismoBuildModel *generate a statistical model using an array of superimposed landmarks or a list of meshes*

Description

generate a statistical model using an array of superimposed landmarks

Usage

```
statismoBuildModel(x, representer, sigma = 0, scale = TRUE)
```

Arguments

x	array of aligned 3D-coordinates or a list of aligned registered meshes.
representer	matrix or triangular mesh of class "mesh3d" with vertices corresponding to rows in the array.
sigma	noise in the data
scale	logical: set to TRUE, if scaling was involved in the registration.

Examples

```
require(Morpho)
data(boneData)
align <- rigidAlign(boneLM)$rotated
mymod <- statismoBuildModel(align,representer=align[,,1],sigma=2,scale=TRUE)
## save it
statismoSaveModel(mymod,"mymod.h5")
```

`statismoGPmodel` *expands a models variability by adding a Gaussian kernel function*

Description

expands a models variability by adding a Gaussian kernel function to the empiric covariance matrix and builds a low-rank approximation of the resulting PCA

Usage

```
statismoGPmodel(model, useEmpiric = TRUE, kernel = list(c(100, 70)),
  ncomp = 10, nystroem = 500)
```

Arguments

`model`	shape model of class "pPCA"
`useEmpiric`	logical: if TRUE, the empiric covariance kernel will be added to the Gaussian ones.
`kernel`	a list containing two valued vectors containing with the first entry specifiying the bandwidth and the second the scaling of the Gaussian kernels (currently only the first list entry is used)
`ncomp`	integer: number of PCs to approximate
`nystroem`	number of samples to compute Nystroem approximation of eigenvectors

Value

returns a shape model of class "pPCA"

Examples

```
### this is a silly example with only 10 landmarks
require(Morpho)
data(boneData)
align <- ProcGPA(boneLM,CSinit=FALSE, scale=TRUE,silent = TRUE)$rotated
mod <- statismoBuildModel(align)
GPmod <- statismoGPmodel(mod,kernel=list(c(10,1),c(1,1)))##extend flexibility using two Gaussian kernels
GPmodNoEmp <- statismoGPmodel(mod,kernel=list(c(10,1),c(1,1)),useEmpiric = FALSE)##extend flexibility using two
PC1orig <- predictpPCA(2,mod)# get shape in 2sd of first PC of originial model
PC1 <- predictpPCA(2,GPmod)# get shape in 2sd of first PC of the extended model
PC1NoEmp <- predictpPCA(2,GPmodNoEmp)# get shape in 2sd of first PC
##visualize the differences from the mean (green spheres)
deformGrid3d(PC1,GPmod$mshape,ngrid=0)##
deformGrid3d(PC1NoEmp,GPmod$mshape,ngrid=0,col1=4,add=TRUE)##only deviates in 5 landmarks from the mean (dark bl
deformGrid3d(PC1orig,GPmod$mshape,ngrid=0,col1=5,add=TRUE)
```

statismoLoadModel/statismoSaveModel

save and load a statistical model of class pPCA to statismo hdf5 format

Description

save and load a statistical model of class pPCA to statismo hdf5 format

Usage

```
statismoSaveModel(model, modelname = dataname)

statismoLoadModel(modelname, scale = TRUE)
```

Arguments

model	object of class pPCA
modelname	filename to read/save

Value

statismoLoadModel returns an object of class "pPCA" while statismoSaveModel saves an object of class pPCA to disk in the statismo file format.

StatismoModelMembers *Implementation/Emulation of the statsimo StatisticalModel class.*

Description

Implementation/Emulation of the statsimo StatisticalModel class.

Usage

```
GetPCABasisMatrix(model)

GetOrthonormalPCABasisMatrix(model)

GetNoiseVariance(model)

GetMeanVector(model)

GetPCAVarianceVector(model)

ComputeLogProbabilityOfDataset(model, dataset)
```

```
ComputeProbabilityOfDataset(model, dataset)

DrawMean(model)

ComputeCoefficientsForDataset(model, dataset)
```

Arguments

model	object of class "pPCA"
dataset	an (already aligned) mesh or k x 3 matrix containing the datasets coordinates.

Details

see http://statismo.github.io/statismo/classdoc/html/classstatismo_1_1StatisticalModel.html for details.

Value

functions return matrices, (log)-probabilties or coefficients for specific dataset

Index