Assessing the agreement between 3D meshes using `MeshAgreement` for R

Daniel Wollschlaeger
wollschlaeger@uni-mainz.de

Heiko Karle
karle@uni-mainz.de

University Medical Center Mainz, Germany

November 14, 2022

Contents

1 Introduction

`MeshAgreement` is an add-on package for the free statistical environment R[1] (R Development Core Team, 2022). It provides functionality to read 3D mesh files, to calculate distance-based as well as volume-overlap-based agreement measures for 3D structures, and to plot the meshes.

The application motivating development of `MeshAgreement` is to compare delineated structures for radiotherapy treatment planning. In order to export 3D mesh files in PLY format from Varian Eclipse, you can use an ESAPI script included in the package. The path to the script can be found like this - re-run in current R session to find the correct path on a given system:

```
esapi_location <- system.file("extdata", package="MeshAgreement")
list.files(esapi_location, full.names=TRUE)

## [1] "C:/Users/Daniel/AppData/Local/Temp/RtmpCqQKBR/Rinst2fd45a29554c/MeshAgreement/extdata/
## [2] "C:/Users/Daniel/AppData/Local/Temp/RtmpCqQKBR/Rinst2fd45a29554c/MeshAgreement/extdata/
## [3] "C:/Users/Daniel/AppData/Local/Temp/RtmpCqQKBR/Rinst2fd45a29554c/MeshAgreement/extdata/
```

[1] A free short introduction to R can be found at https://www.statmethods.net/.

```
## [4] "C:/Users/Daniel/AppData/Local/Temp/RtmpCqQKBR/Rinst2fd45a29554c/MeshAgreement/extdata/
## [5] "C:/Users/Daniel/AppData/Local/Temp/RtmpCqQKBR/Rinst2fd45a29554c/MeshAgreement/extdata/
## [6] "C:/Users/Daniel/AppData/Local/Temp/RtmpCqQKBR/Rinst2fd45a29554c/MeshAgreement/extdata/
## [7] "C:/Users/Daniel/AppData/Local/Temp/RtmpCqQKBR/Rinst2fd45a29554c/MeshAgreement/extdata/
```

Computational geometry is carried out mainly using the `CGAL` library (CGAL Project, 2022) via package `RcppCGAL` (Dunipace & the CGAL Project, 2022) used in package `cgalMeshes` (Laurent, 2022b). Distance maps are calculated using the `VCG` library (Visual Computing Lab of the Italian National Research Council Institute ISTI, 2022) via package `Rvcg` (Schlager, 2017).

To install `MeshAgreement`, you need a current version of R and be online. Preferably, a free development environment like RStudio (Posit Software, PBC, 2022) should be used.

2 Interfaces

`MeshAgreement` provides two interfaces geared towards users with different levels of familiarity with R: The regular command line functions and a built-in web application.

2.1 R command line interface

Users familiar with R can use the `MeshAgreement` package functions from the R command line. This facilitates statistical post-processing of results with the full capabilities of R. After installing `MeshAgreement`, you should be able to run (function `get_mesh_agree()` is explained in section 4):

```
## load MeshAgreement package - required for all following tasks
library(MeshAgreement, verbose=FALSE)

## get agreement measures for all pairs from list of meshes
## data_heart_obsL: list of sample meshes included in MeshAgreement
heartL <- mesh3dL_to_cgalMeshL(data_heart_obsL)

## omit JSC/DSC to reduce run-time
agreeW <- get_mesh_agree(heartL, do_ui=FALSE, silent=TRUE)
agreeW

##          mesh1       mesh2     group  DCOM HD_max HD_avg    ASD   RMSD
## 1 Obs01_HEART Obs02_HEART strct_001 2.612 14.055 13.928 1.4366 2.2942
## 2 Obs01_HEART Obs03_HEART strct_001 4.778 14.126 14.112 2.3048 3.7402
## 3 Obs02_HEART Obs03_HEART strct_001 2.698 14.135 13.656 2.2127 3.2330
## 4  Obs01_AOKL  Obs02_AOKL strct_002 1.294  4.164  3.697 0.7241 0.9642
## 5  Obs01_AOKL  Obs03_AOKL strct_002 1.874  4.305  4.200 1.0613 1.3626
## 6  Obs02_AOKL  Obs03_AOKL strct_002 3.017  5.454  5.067 1.5629 1.9340
##   JSC DSC
## 1  NA  NA
## 2  NA  NA
## 3  NA  NA
## 4  NA  NA
## 5   NA   NA
## 6   NA   NA
```

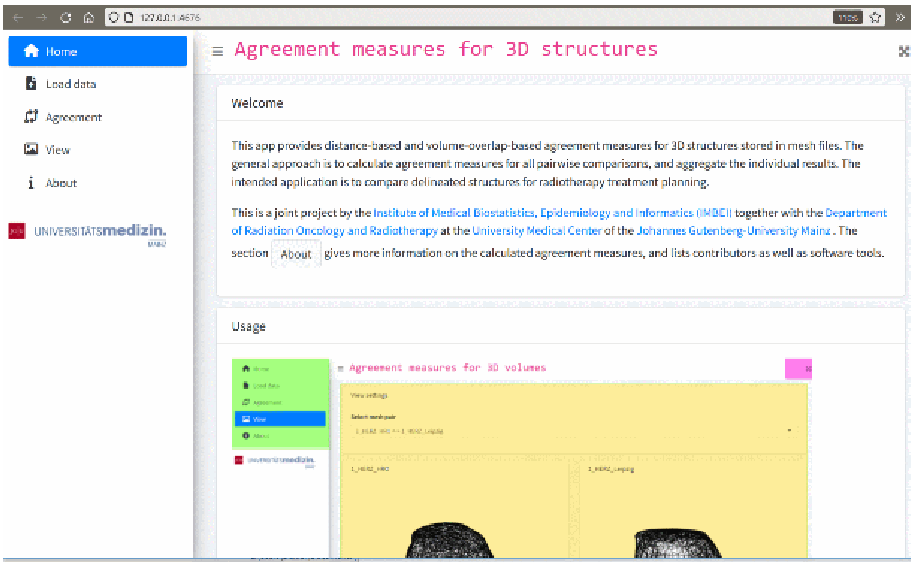
2.2 Web-based graphical user interface

For users who are unfamiliar with R, `MeshAgreement` includes a `Shiny`-based web application (Chang et al., 2022) running locally that eliminates the need to use R syntax.[2] Note that packages `shiny` (Chang et al., 2022), `bs4Dash` (Granjon, 2022), `DT` (Xie, Cheng, & Tan, 2022), `sortable` (de Vries, Schloerke, & Russell, 2022), and `rgl` (Murdoch & Adler, 2022) need to be installed to run the GUI. The different analysis steps are displayed in figures 1, 2, 3, 4, 5, and 6.

```
## install required packages
# install.packages(c("shiny", "bs4Dash", "DT", "sortable", "rgl"))

## load MeshAgreement package
# library(MeshAgreement, verbose=FALSE)

## start Shiny app
# run_gui()
```



Agreement measures for 3D structures

Welcome

This app provides distance-based and volume-overlap-based agreement measures for 3D structures stored in mesh files. The general approach is to calculate agreement measures for all pairwise comparisons, and aggregate the individual results. The intended application is to compare delineated structures for radiotherapy treatment planning.

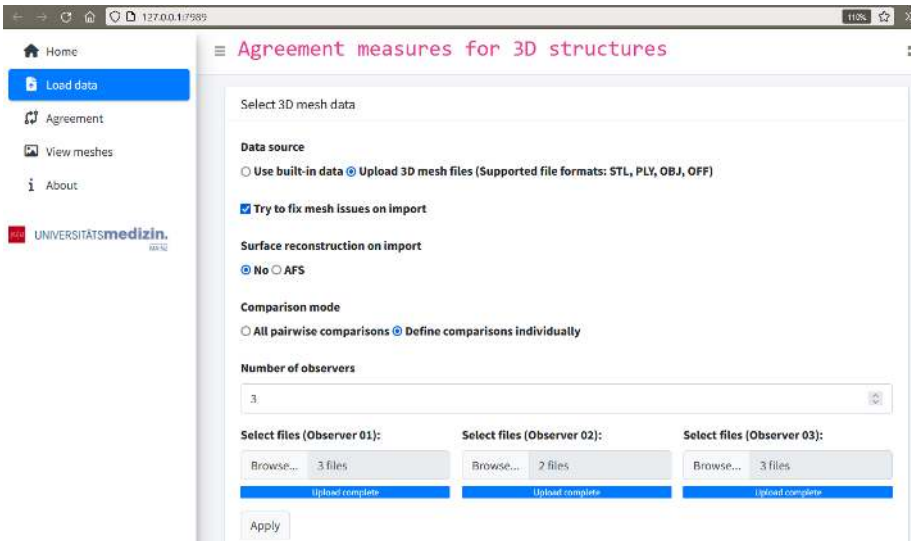
This is a joint project by the Institute of Medical Biostatistics, Epidemiology and Informatics (IMBEI) together with the Department of Radiation Oncology and Radiotherapy at the University Medical Center of the Johannes Gutenberg-University Mainz . The section About gives more information on the calculated agreement measures, and lists contributors as well as software tools.

Figure 1: Welcome page in the `MeshAgreement` web application

3 Read mesh files

Supported file formats are STL, PLY, OBJ, and OFF. If the same structures are contoured by three different observers, and the resulting mesh files are stored in three corresponding directories, reading in the observer/mesh list can look like this:

[2] A live demo is available at: http://shiny.imbei.uni-mainz.de:3838/MeshAgreement/



≡ Agreement measures for 3D structures

Select 3D mesh data

Data source

○ Use built-in data ● Upload 3D mesh files (Supported file formats: STL, PLY, OBJ, OFF)

☑ Try to fix mesh issues on import

Surface reconstruction on import

● No ○ AFS

Comparison mode

○ All pairwise comparisons ● Define comparisons individually

Number of observers

3

Select files (Observer 01): Browse... 3 files Upload complete

Select files (Observer 02): Browse... 2 files Upload complete

Select files (Observer 03): Browse... 3 files Upload complete

Apply

Figure 2: Importing files with options in the MeshAgreement web application

```
# ff1 <- list.files("c:/meshes/obs1", pattern="PLY$", full.names=TRUE)
# ff2 <- list.files("c:/meshes/obs2", pattern="PLY$", full.names=TRUE)
# ff3 <- list.files("c:/meshes/obs3", pattern="PLY$", full.names=TRUE)
# obsL <- read_mesh(list(Obs01=ff1, Obs02=ff2, Obs03=ff3),
#                   reconstruct="AFS")
```

If a single structure is contoured by different observers, and all files are stored in the same directory, reading in requires two steps: First, the mesh files are imported into a mesh list, and second, the mesh list is transformed by assigning each mesh to a different observer.

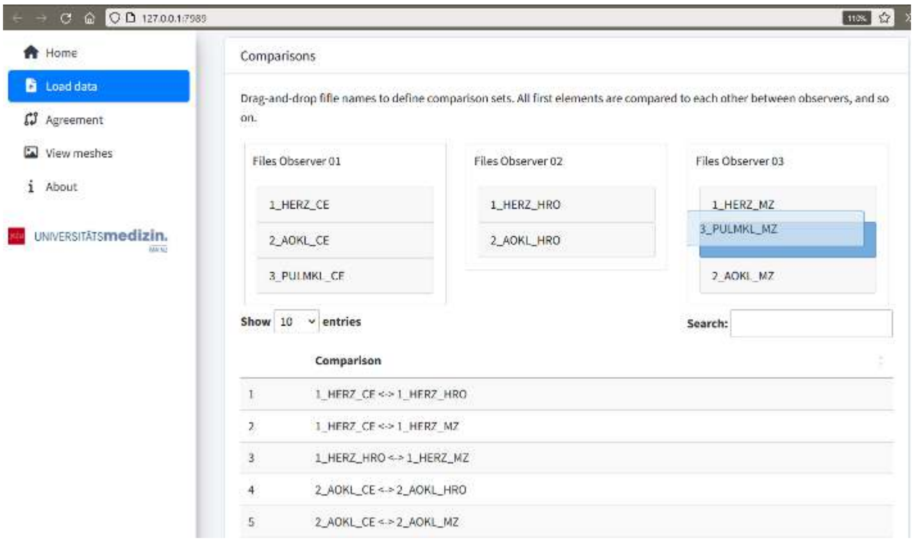
```
# ff    <- list.files("c:/meshes/", pattern="PLY$", full.names=TRUE)
# meshL <- read_mesh_obs(ff)

## assign each mesh to a different observer to enable all
## pairwise comparisons
# obsL <- meshL_to_observerL(meshL)
```

Information on the imported meshes can be printed.

```
## data_heart_obsL: list of sample meshes included in MeshAgreement
heartL <- mesh3dL_to_cgalMeshL(data_heart_obsL)
print_mesh(heartL)


## Mesh: Obs01_HEART
## Volume: 652172.69
## Centroid: [18.71, -45.00, -1379.33]
##
## Mesh: Obs01_AOKL
## Volume: 11640.76
## Centroid: [-2.16, -47.56, -1349.71]
```



Comparisons

Drag-and-drop fiile names to define comparison sets. All first elements are compared to each other between observers, and so on.

	Comparison
1	1_HERZ_CE <-> 1_HERZ_HRO
2	1_HERZ_CE <-> 1_HERZ_MZ
3	1_HERZ_HRO <-> 1_HERZ_MZ
4	2_AOKL_CE <-> 2_AOKL_HRO
5	2_AOKL_CE <-> 2_AOKL_MZ

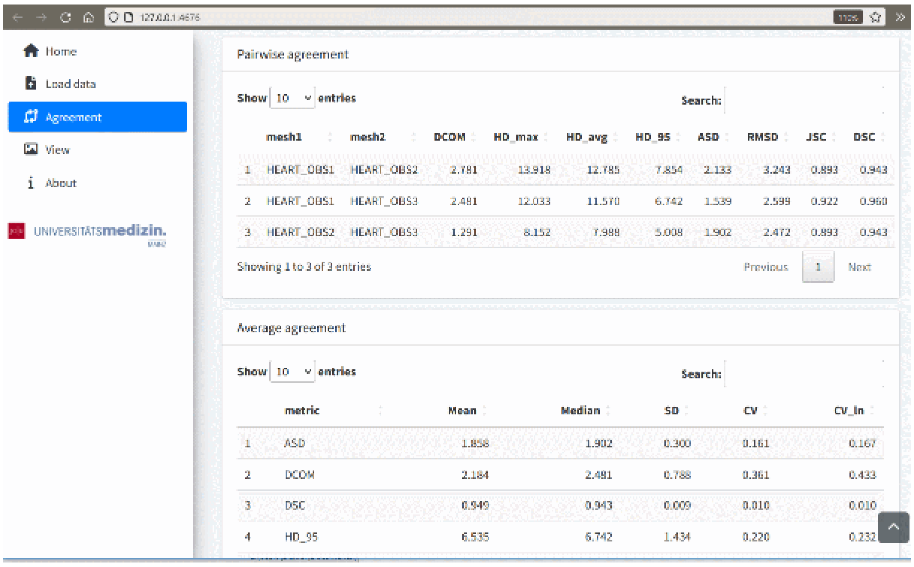
Figure 3: Defining comparisons for agreement measures by drag-and-drop of file lists in the `MeshAgreement` web application

```
##
## Mesh: Obs02_HEART
## Volume: 659868.94
## Centroid: [17.69, -44.01, -1377.14]
##
## Mesh: Obs02_AOKL
## Volume: 11461.88
## Centroid: [-3.31, -48.15, -1349.73]
##
## Mesh: Obs03_HEART
## Volume: 580062.62
## Centroid: [18.50, -44.38, -1374.60]
##
## Mesh: Obs03_AOKL
## Volume: 10454.66
## Centroid: [-0.83, -46.89, -1348.58]
```

4 Mesh agreement measures

You can calculate distance-based as well as volume-overlap-based agreement measures for all pairwise comparisons between meshes. The following measures are included (Sherer et al., 2021; Heimann & et al., 2009; Fotina, Lütgendorf-Caucig, Stock, Pötter, & Georg, 2012; Babalola et al., 2009; Hanna, Hounsell, & O'Sullivan, 2010; Jaccard, 1912; Dice, 1945):

- Distance-based measures
  - DCOM: Euclidean distance between the respective center of mass of both meshes
  - HDmax: Hausdorff distance - worst case, maximum of both directed Hausdorff distances



Pairwise agreement

Show 10 entries   Search:

	mesh1	mesh2	DCOM	HD_max	HD_avg	HD_95	ASD	RMSD	JSC	DSC
1	HEART_OBS1	HEART_OBS2	2.781	13.918	12.785	7.854	2.133	3.243	0.893	0.943
2	HEART_OBS1	HEART_OBS3	2.481	12.033	11.570	6.742	1.539	2.599	0.922	0.960
3	HEART_OBS2	HEART_OBS3	1.291	8.152	7.988	5.008	1.902	2.472	0.893	0.943

Showing 1 to 3 of 3 entries   Previous 1 Next

Average agreement

Show 10 entries   Search:

	metric	Mean	Median	SD	CV	CV_ln
1	ASD	1.858	1.902	0.300	0.161	0.167
2	DCOM	2.184	2.481	0.788	0.361	0.433
3	DSC	0.949	0.943	0.009	0.010	0.010
4	HD_95	6.535	6.742	1.434	0.220	0.232

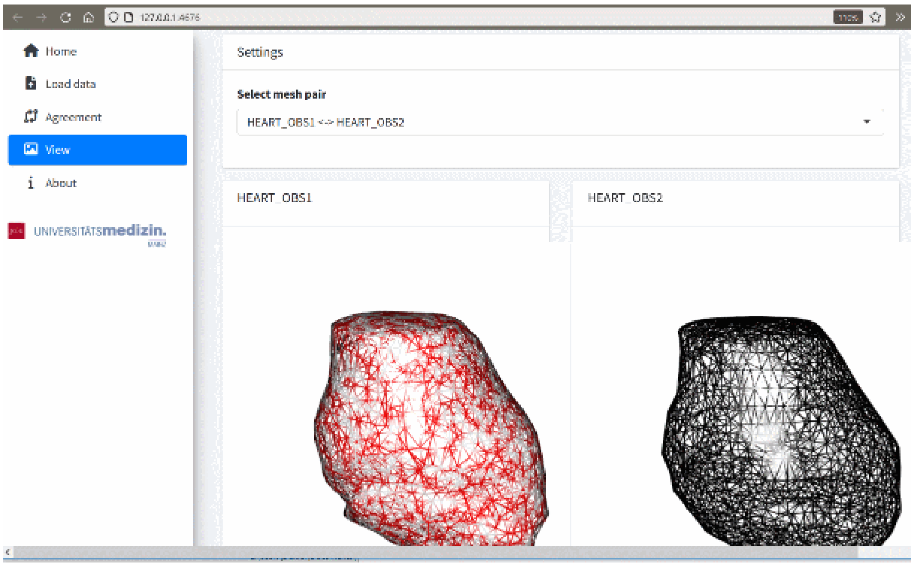
Figure 4: Display distance-based and volume-overlap-based agreement measures for pairwise comparisons as well as as aggregated agreement over all pairs in the `MeshAgreement` web application

  - HDavg: Hausdorff distance - average, arithmetic mean of both directed Hausdorff distances
  - ASD: Average symmetric surface distance
  - RMSD: Root mean squared symmetric surface distance
- Volume-overlap-based measures
  - JSC: Jaccard similarity coefficient
  - DSC: Dice similarity coefficient
  - Note that using package `Boov` (Laurent, 2022a) may have better performance for some meshes than the default `cgalMeshes`. Using `Boov` requires installing package `Boov` as well as setting option `boov=TRUE` when calling agreement functions.

The functions that calculate agreement measures all have two versions.

- The main version of each function operates on an observer/mesh list as generated by `read_mesh()`. These functions are `get_mesh_metro()` as an interface to the `Rvcg::vcgMetro()` distance map function, `get_mesh_ui()` to calculate the structures' union/intersection with corresponding volumes, and `get_mesh_agree()`, which does both of these tasks and summarizes results in a data frame.
- A second version of each function operates on a single pair of meshes as generated by `get_mesh_pairs()`. These functions are `get_mesh_metro_pair()`, `get_mesh_ui_pair()`, and `get_mesh_agree_pair()`.

```
## already called above
# heartL <- mesh3dL_to_cgalMeshL(data_heart_obsL)
# agreeW <- get_mesh_agree(heartL, silent=TRUE)
agreeW
```



Select mesh pair

HEART_OBS1 <-> HEART_OBS2

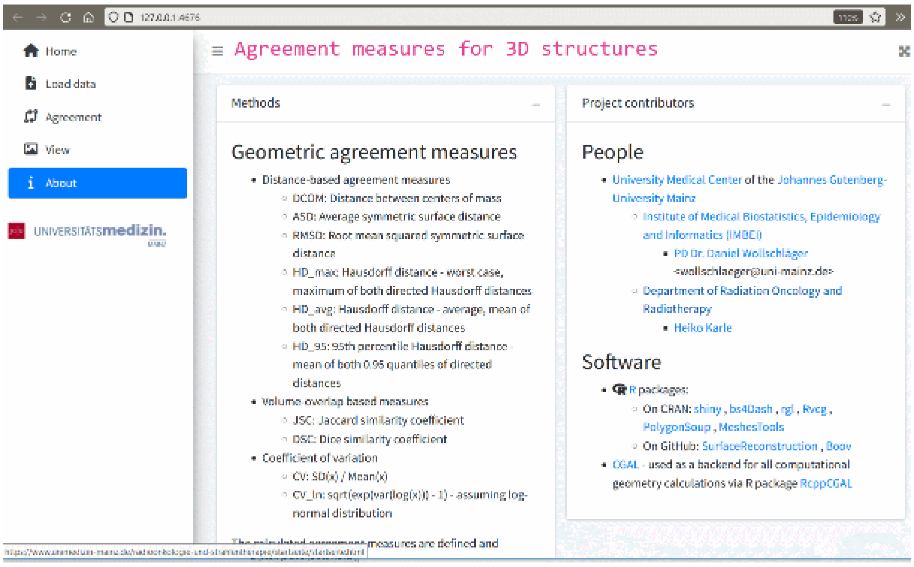
Figure 5: View pairs of imported meshes together with a color-coded distance map in the `MeshAgreement` web application

```
##          mesh1       mesh2     group  DCOM HD_max HD_avg    ASD   RMSD
## 1 Obs01_HEART Obs02_HEART strct_001 2.612 14.055 13.928 1.4366 2.2942
## 2 Obs01_HEART Obs03_HEART strct_001 4.778 14.126 14.112 2.3048 3.7402
## 3 Obs02_HEART Obs03_HEART strct_001 2.698 14.135 13.656 2.2127 3.2330
## 4  Obs01_AOKL  Obs02_AOKL strct_002 1.294  4.164  3.697 0.7241 0.9642
## 5  Obs01_AOKL  Obs03_AOKL strct_002 1.874  4.305  4.200 1.0613 1.3626
## 6  Obs02_AOKL  Obs03_AOKL strct_002 3.017  5.454  5.067 1.5629 1.9340
##   JSC DSC
## 1  NA  NA
## 2  NA  NA
## 3  NA  NA
## 4  NA  NA
## 5  NA  NA
## 6  NA  NA
```

A utility function transforms the data frame returned by `get_mesh_agree()` to long format which may be more convenient to post-process.

```
agreeL <- get_mesh_agree_long(agreeW)
agreeL
```

```
##          mesh1       mesh2     group metric observed
## 1  Obs01_HEART Obs02_HEART strct_001   DCOM   2.6123
## 2  Obs01_HEART Obs03_HEART strct_001   DCOM   4.7784
## 3  Obs02_HEART Obs03_HEART strct_001   DCOM   2.6983
## 4   Obs01_AOKL  Obs02_AOKL strct_002   DCOM   1.2938
## 5   Obs01_AOKL  Obs03_AOKL strct_002   DCOM   1.8738
## 6   Obs02_AOKL  Obs03_AOKL strct_002   DCOM   3.0174
## 7  Obs01_HEART Obs02_HEART strct_001 HD_max  14.0552
```


127.0.0.1:4676
110%
Home
Load data
Agreement
View
About
UNIVERSITÄTSmedizin. MAINZ
Methods
Project contributors
https://www.unimedizin-mainz.de/radioonkologie-und-strahlentherapie/startseite/startseite.html

Agreement measures for 3D structures

Methods

Geometric agreement measures

- Distance-based agreement measures
  - DCOM: Distance between centers of mass
  - ASD: Average symmetric surface distance
  - RMSD: Root mean squared symmetric surface distance
  - HD_max: Hausdorff distance - worst case, maximum of both directed Hausdorff distances
  - HD_avg: Hausdorff distance - average, mean of both directed Hausdorff distances
  - HD_95: 95th percentile Hausdorff distance - mean of both 0.95 quantiles of directed distances
- Volume overlap based measures
  - JSC: Jaccard similarity coefficient
  - DSC: Dice similarity coefficient
- Coefficient of variation
  - CV: SD(x) / Mean(x)
  - CV_ln: sqrt(exp[var(log(x))] - 1) - assuming log-normal distribution

The calculated agreement measures are defined and

Project contributors

People

- University Medical Center of the Johannes Gutenberg-University Mainz
  - Institute of Medical Biostatistics, Epidemiology and Informatics (IMBEI)
    - PD Dr. Daniel Wollschläger <wollschlaeger@uni-mainz.de>
  - Department of Radiation Oncology and Radiotherapy
    - Heiko Karle

Software

- R packages:
  - On CRAN: shiny , bs4Dash , rgl , Rvcg , PolygonSoup , MeshesTools
  - On GitHub: SurfaceReconstruction , Boov
- CGAL - used as a backend for all computational geometry calculations via R package RcppCGAL

Figure 6: Background information on the MeshAgreement web application

```
## 8  Obs01_HEART Obs03_HEART strct_001 HD_max  14.1261
## 9  Obs02_HEART Obs03_HEART strct_001 HD_max  14.1345
## 10  Obs01_AOKL  Obs02_AOKL strct_002 HD_max   4.1635
## 11  Obs01_AOKL  Obs03_AOKL strct_002 HD_max   4.3051
## 12  Obs02_AOKL  Obs03_AOKL strct_002 HD_max   5.4539
## 13 Obs01_HEART Obs02_HEART strct_001 HD_avg  13.9283
## 14 Obs01_HEART Obs03_HEART strct_001 HD_avg  14.1124
## 15 Obs02_HEART Obs03_HEART strct_001 HD_avg  13.6563
## 16  Obs01_AOKL  Obs02_AOKL strct_002 HD_avg   3.6970
## 17  Obs01_AOKL  Obs03_AOKL strct_002 HD_avg   4.1998
## 18  Obs02_AOKL  Obs03_AOKL strct_002 HD_avg   5.0675
## 19 Obs01_HEART Obs02_HEART strct_001    ASD   1.4366
## 20 Obs01_HEART Obs03_HEART strct_001    ASD   2.3048
## 21 Obs02_HEART Obs03_HEART strct_001    ASD   2.2127
## 22  Obs01_AOKL  Obs02_AOKL strct_002    ASD   0.7241
## 23  Obs01_AOKL  Obs03_AOKL strct_002    ASD   1.0613
## 24  Obs02_AOKL  Obs03_AOKL strct_002    ASD   1.5629
## 25 Obs01_HEART Obs02_HEART strct_001   RMSD   2.2942
## 26 Obs01_HEART Obs03_HEART strct_001   RMSD   3.7402
## 27 Obs02_HEART Obs03_HEART strct_001   RMSD   3.2330
## 28  Obs01_AOKL  Obs02_AOKL strct_002   RMSD   0.9642
## 29  Obs01_AOKL  Obs03_AOKL strct_002   RMSD   1.3626
## 30  Obs02_AOKL  Obs03_AOKL strct_002   RMSD   1.9340
## 31 Obs01_HEART Obs02_HEART strct_001    JSC       NA
## 32 Obs01_HEART Obs03_HEART strct_001    JSC       NA
## 33 Obs02_HEART Obs03_HEART strct_001    JSC       NA
## 34  Obs01_AOKL  Obs02_AOKL strct_002    JSC       NA
## 35  Obs01_AOKL  Obs03_AOKL strct_002    JSC       NA
## 36  Obs02_AOKL  Obs03_AOKL strct_002    JSC       NA
## 37 Obs01_HEART Obs02_HEART strct_001    DSC       NA
## 38 Obs01_HEART Obs03_HEART strct_001    DSC        NA
## 39 Obs02_HEART Obs03_HEART strct_001    DSC        NA
## 40  Obs01_AOKL  Obs02_AOKL strct_002    DSC        NA
## 41  Obs01_AOKL  Obs03_AOKL strct_002    DSC        NA
## 42  Obs02_AOKL  Obs03_AOKL strct_002    DSC        NA
```

Agreement measures for all pairwise comparisons for a structure between observers may be aggregated to assess overall agreement.

```
agree_aggrW <- get_mesh_agree_aggr(agreeW)
agree_aggrW

##        group metric   Mean Median      SD       CV    CV_ln
## 1  strct_001    ASD  1.985  2.213 0.47691 0.240296 0.266515
## 2  strct_001   DCOM  3.363  2.698 1.22652 0.364709 0.349718
## 3  strct_001 HD_avg 13.899 13.928 0.22941 0.016506 0.016534
## 4  strct_001 HD_max 14.105 14.126 0.04353 0.003086 0.003089
## 5  strct_001   RMSD  3.089  3.233 0.73365 0.237497 0.254928
## 6  strct_002    ASD  1.116  1.061 0.42209 0.378192 0.399385
## 7  strct_002   DCOM  2.062  1.874 0.87700 0.425380 0.444355
## 8  strct_002 HD_avg  4.321  4.200 0.69329 0.160430 0.159621
## 9  strct_002 HD_max  4.641  4.305 0.70765 0.152484 0.147962
## 10 strct_002   RMSD  1.420  1.363 0.48745 0.343209 0.358828
```

A utility function transforms the returned data frame to long format which may be more convenient to post-process.

```
agree_aggrL <- get_mesh_agree_aggr_long(agree_aggrW)
agree_aggrL

##        group metric statistic  observed
## 1  strct_001    ASD      Mean  1.984673
## 2  strct_001   DCOM      Mean  3.363023
## 3  strct_001 HD_avg      Mean 13.898998
## 4  strct_001 HD_max      Mean 14.105270
## 5  strct_001   RMSD      Mean  3.089118
## 6  strct_002    ASD      Mean  1.116085
## 7  strct_002   DCOM      Mean  2.061682
## 8  strct_002 HD_avg      Mean  4.321430
## 9  strct_002 HD_max      Mean  4.640835
## 10 strct_002   RMSD      Mean  1.420278
## 11 strct_001    ASD    Median  2.212678
## 12 strct_001   DCOM    Median  2.698318
## 13 strct_001 HD_avg    Median 13.928299
## 14 strct_001 HD_max    Median 14.126059
## 15 strct_001   RMSD    Median  3.232959
## 16 strct_002    ASD    Median  1.061255
## 17 strct_002   DCOM    Median  1.873818
## 18 strct_002 HD_avg    Median  4.199766
## 19 strct_002 HD_max    Median  4.305101
## 20 strct_002   RMSD    Median  1.362647
## 21 strct_001    ASD        SD  0.476910
## 22 strct_001   DCOM        SD  1.226524
## 23 strct_001 HD_avg        SD  0.229412
## 24 strct_001 HD_max        SD  0.043532
## 25 strct_001   RMSD        SD  0.733655
## 26 strct_002    ASD        SD  0.422095
## 27 strct_002   DCOM        SD  0.876998
## 28 strct_002 HD_avg        SD  0.693289
## 29 strct_002 HD_max        SD  0.707651
## 30 strct_002   RMSD        SD  0.487452
## 31 strct_001    ASD        CV  0.240296
## 32 strct_001   DCOM        CV  0.364709
## 33 strct_001 HD_avg        CV  0.016506
## 34 strct_001 HD_max        CV  0.003086
## 35 strct_001   RMSD        CV  0.237497
## 36 strct_002    ASD        CV  0.378192
## 37 strct_002   DCOM        CV  0.425380
## 38 strct_002 HD_avg        CV  0.160430
## 39 strct_002 HD_max        CV  0.152484
## 40 strct_002   RMSD        CV  0.343209
## 41 strct_001    ASD     CV_ln  0.266515
## 42 strct_001   DCOM     CV_ln  0.349718
## 43 strct_001 HD_avg     CV_ln  0.016534
## 44 strct_001 HD_max     CV_ln  0.003089
## 45 strct_001   RMSD     CV_ln  0.254928
## 46 strct_002    ASD     CV_ln  0.399385
## 47 strct_002   DCOM     CV_ln  0.444355
## 48 strct_002 HD_avg     CV_ln  0.159621
## 49 strct_002 HD_max     CV_ln  0.147962
## 50 strct_002   RMSD     CV_ln  0.358828
```

Acknowledgements

The authors thank Stéphane Laurent, Eric Dunipace and Stefan Schlager for creating the necessary R packages for handling 3D meshes with methods from the CGAL and VCG libraries.

References

Babalola, K., Patenaude, B., Aljabar, P., Schnabel, J., Kennedy, D., Crum, W., … Rueckert, D. (2009). An evaluation of four automatic methods of segmenting the subcortical structures in the brain. *NeuroImage*, *47*, 1435–1447.

CGAL Project. (2022). CGAL: The computational geometry algorithms library [Computer software]. URL https://www.cgal.org/ (Version 5.5.1)

Chang, W., Cheng, J., Allaire, J. J., Sievert, C., Schloerke, B., Xie, Y., ... Borges, B. (2022). shiny: Web application framework for R [Computer software]. URL https://CRAN.R-project.org/package=shiny (R package version 1.7.2)

de Vries, A., Schloerke, B., & Russell, K. (2022). sortable: Drag-and-drop in 'shiny' apps with 'SortableJS' [Computer software]. URL https://CRAN.R-project.org/package=sortable (R package version 0.4.6)

Dice, L. (1945). Measures of the amount of ecologic association between species. *Ecology*, *26*, 297–302.

Dunipace, E., & the CGAL Project. (2022). RcppCGAL: Rcpp integration for cgal [Computer software]. URL https://CRAN.R-project.org/package=RcppCGAL (R package version 5.4.1)

Fotina, I., Lütgendorf-Caucig, C., Stock, M., Pötter, R., & Georg, D. (2012). Critical discussion of evaluation parameters for inter-observer variability in target definition for radiation therapy. *Strahlenther Onkol*, *188*, 160–167.

Granjon, D. (2022). bs4Dash: A 'Bootstrap 4' version of 'shinydashboard' [Computer software]. URL https://CRAN.R-project.org/package=bs4Dash (R package version 2.1.0)

Hanna, G., Hounsell, A., & O'Sullivan, J. (2010). Geometrical Analysis of Radiotherapy Target Volume Delineation: a Systematic Review of Reported Comparison Methods. *Clin Oncol*, *22*, 515–525.

Heimann, T., & et al. (2009). Comparison and evaluation of methods for liver segmentation from CT datasets. *IEEE Trans Med Imaging*, *18*, 1251–1265.

Jaccard, P. (1912). The distribution of flora in the alpine zone. *New Phytologist*, *11*, 37–50.

Laurent, S. (2022a). Boov: Boolean operations on volumes [Computer software]. URL https://CRAN.R-project.org/package=Boov (R package version 1.0.0)

Laurent, S. (2022b). cgalMeshes: R6 based utilities for 3D meshes [Computer software]. URL https://CRAN.R-project.org/package=cgalMeshes (R package version 1.0.0)

Murdoch, D., & Adler, D. (2022). rgl: 3D visualization using OpenGL [Computer software]. URL https://CRAN.R-project.org/package=rgl (R package version 0.110.2)

Posit Software, PBC. (2022). RStudio: Integrated development environment for R [Computer software]. Boston, MA. URL https://posit.co/products/open-source/rstudio/ (Version 2022.07.2)

R Development Core Team. (2022). R: A Language and Environment for Statistical Computing [Computer software manual]. Vienna, Austria. URL https://www.r-project.org/

Schlager, S. (2017). Morpho and Rvcg – shape analysis in R. In G. Zheng, S. Li, & G. Szekely (Eds.), *Statistical shape and deformation analysis* (pp. 217–256). Academic Press. URL https://CRAN.R-project.org/package=Rvcg

Sherer, T., Lin, D., Elguindi, S., Duke, S., Tan, L., Cacicedo, J., ... Gillespie, E. (2021). Metrics to evaluate the performance of auto-segmentation for radiation treatment planning: A critical review. *Radiother Oncol*, *160*, 185–191.

Visual Computing Lab of the Italian National Research Council Institute ISTI. (2022). The visualization and computer graphics library (VCG) [Computer software]. URL http://vcg.isti.cnr.it/ (Version 2022.02)

Xie, Y., Cheng, J., & Tan, X. (2022). DT: A wrapper of the JavaScript library 'DataTables' [Computer software]. URL https://CRAN.R-project.org/package=DT (R package version 0.26)