

Using the 'align_shapes' function in 'auto3dgm'

Chris Glynn

December 12, 2013

1 Introduction

The 'auto3dgm' package provides functions in R for users to align shapes using an automated 3-dimensional geometric morphometric analysis. The main function that users interact with is `align_shapes`:

```
align_shapes(Bm_path, Levels, Ids, Names, Dataset, Run, Initialize, Return)
```

This vignette explains the steps required to get this function to work with your data. I will walk through each of the function inputs and explain where to find the outputs.

2 Before using 'auto3dgm'

IMPORTANT NOTE: Please update your installation of R to 3.0.2 or greater.

Make sure that you have the directory structure set up properly. Set a parent directory. This will be the function argument **Bm_path**. Inside the parent directory, `.../Bm_path`, there will be two subdirectories: one for the datasets called 'meshes' and one called 'output' where alignment files will be written.

1. Mesh folder. This folder contains the different data sets that you will need to reference.

- `...Bm_path/meshes`

Each data set warrants its own sub-directory. In developing this package, I worked with two different datasets: teeth shapes and astragali shapes. Within the meshes folder, there should be two separate folders for astragali meshes and teeth meshes.

- `...Bm_path/meshes/teeth_dataset`
- `...Bm_path/meshes/astragali_dataset`

Within each dataset, there should be an ".off" file for each shape that you wish to align. In addition, there should be two further sub-directories: one for low-resolution files and one for subsampled versions of the original files. The low-resolution files are used for visualization in MeshLab or other software platforms. The subsampled version is the file from which shape alignments are computed. If you do not have subsampled versions of the ".off" files, the `align_shapes` function itself will subsample the original shapes and create them and write them to this directory, so it must exist. The function will not generate low-resolution ".off" files.

These need to be user supplied and are essential for visual representation. As an example, let's consider the file "001.off" in the teeth dataset. There should be the following files. Note that the files should have the same name but are located in different places.

- Original shape file:
.../Bm_path/meshes/teeth_dataset/001.off
 - Low-resolution shape file:
.../Bm_path/meshes/teeth_dataset/lowres/001.off
 - Subsampled shape file:
.../Bm_path/meshes/teeth_dataset/subsampled/001.off
2. Output folder. This is where you will find all the files generated by the align_shapes function. It will contain aligned output files. The main task of the user here is to create a sub-directory for each dataset above. Again using the ankle bones and the teeth bones, you need to create the two following directories.
- **.../Bm_path/output/teeth_run**
 - **.../Bm_path/output/astragali_run**

The output files for each data set will be contained in their respective folders.

3 Function Arguments

1. Bm_path. Set the Bm_path as described above.
2. Levels. This is a vector of 2 elements which are the number of vertices to subsample from the original data files. The first element should be lower than the second element. There is a tradeoff between alignment accuracy and computational speed. Higher subsampled points leads to a more accurate alignment but comes at a higher compute time. I use Levels = c(64,128) most of the time.
3. Ids. This is a vector of filenames from the meshes. For the files "001.off" and "002.off", the shape ids are "001" and "002". The ids must correspond in this way. Ids = c("001", "002").
4. Names. This is a vector of names for the shapes which may not be characterized by the filename. It could be a bone or specimen name. The names for the teeth shapes that I worked with are "a19", "a13", etc. It is left to the user to define these names, though the vector of names and vector of ids must be the same length. Names = c("a19", "a13")
5. Dataset. This character variable is name of the dataset and must match with the name of the folder in the mesh directory. As an example, the directory **.../Bm_path/meshes/teeth_dataset/001.off** yields a dataset name of "teeth_dataset". There must be this correspondence for the function to find the appropriate files.
6. Run. This character variable is the name of the sub-folder in the output folder where this batch of output files will be written. When you run the analysis for different subsets of shapes and different subsampling levels, it is helpful to be able to store them all in different places. The Run variable allows you to specify a folder inside the output folder where the output of a particular run will be stored.

7. Initialize. This is a logical variable. If Initialize=TRUE, then the align_shapes function will simply load in the shape files into an R object called ds and return that object. It will NOT perform any alignment. If Initialize=FALSE, the function will perform the alignment algorithm.
8. Return. This is a character vector of data objects to return. To return the global alignment data structure, Return = "ga". To return the pairwise alignment data structure, Return="pw_rotations". To return the dataset of shapes, Return="ds". To return all three, Return=c("ds", "ga", "pw_rotations"). To return no data objects, Return=NA.

4 Finding your Output Files

Assume that you have run the align_shapes function with the teeth dataset. There are several output files of interest. Some users may simply be after the aligned shapes themselves. Each input shape will be aligned and saved. Further, for each level of subsampling (in this example 64 points and 128 points), there will be an alignment file with all shapes aligned in one file. Lastly, there is a file where the shapes are connected in 3D space through the minimum spanning tree. Each type of file has a bullet point below and describes what is in the file and where it can be found.

- Aligned Files.
.../Bm_path/output/teeth_run/Aligned_Shapes
 Each input shape will have a corresponding output file where the shape is aligned. Since there could be lots of these files, there is a subdirectory in the output folder.
- MST.
.../Bm_path/output/teeth_run/MST.jpg
 This is where you will find a jpeg of the Minimum Spanning Tree.
- The alignment file for the 64 point subsampling.
.../Bm_path/output/teeth_run/alignment.off
 This file contains rows of the aligned teeth with 10 shapes per row. There is a marker to denote where the first shape in the first row begins. From there the shapes are simply ordered according to the sequence of shape ids supplied by the user.
- The alignment file for the 128 point subsampling.
.../Bm_path/output/teeth_run/alignment.2.off
 Much like the file above. The minimum spanning tree is computed for 64 points and then re-used to compute the shape alignments with the higher number of points. In this case 128. This file can be found at
- A 3D representation of the aligned teeth in the minimum spanning tree.
.../Bm_path/output/teeth_run/map.off
- For each level of subsampling, there is a corresponding Morphologika file.
.../Bm_path/output/teeth_run/morphologika.off
.../Bm_path/output/teeth_run/morphologika.2.off
 These shapes are normalized so that they are all on the same size scale.
- Lastly there are unscaled shapes in a Morphologika files.
.../Bm_path/output/teeth_run/morphologika_unscaled.off
.../Bm_path/output/teeth_run/morphologika.2_unscaled.off,