Using the 'align_shapes' function in 'auto3dgm'

Chris Glynn

February 26, 2014

1 Introduction

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

```
align_shapes(Input_dir,Output_dir, Levels, Ids, Names)
```

This function returns a list which contains the dataset of shapes, the global alignment, and pairwise alignments.

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.

The package requires a bit of overhead in setting up an input directory structure. The package computes the alignment between shapes by subsampling landmarks from the original shape files. For visual representation, a low-resolution shape file is also necessary. The input directory will contain the original shape files. It will also contain subdirectories for low-resolution files and subsampled files.

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

   - **.../Input_dir**

   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: **.../Input_dir/001.off**
- Low-resolution shape file: **.../Input_dir/lowres/001.off**
- Subsampled shape file: **.../Input_dir/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 and can be any existing directory. Alternatively, the program will create the output folder if it does not already exist.

   - **.../Output_dir**

3 Function Arguments

1. Input_dir. Set up the input directory as described above.

2. Output_dir. Specify an output directory where all output files can be written.

3. 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.

4. 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”).

5. 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”)

4 Finding your Output Files

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.
  .../Output_dir/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.
  .../Output_dir/MST.jpg
  This is where you will find a jpeg of the Minimum Spanning Tree.

- The alignment file for the 64 point subsampling.
  .../Output_dir/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.
  .../Output_dir/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.
  .../Output_dir/map.off

- For each level of subsampling, there is a corresponding Morphologika file.
  .../Output_dir/morphologika.off
  .../Output_dir/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.
  .../Output_dir/morphologika_unscaled.off
  .../Output_dir/morphologika_2_unscaled.off,